Twitter Spam Detection based on Deep Learning

Tingmin Wu, Shigang Liu, Jun Zhang and Yang Xiang
School of Information Technology
Deakin University
221 Burwood Hwy,Burwood,
VIC 3125, Australia
{tingminw, shigang, jun.zhang, yang.xiang}@deakin.edu.au

ABSTRACT

Twitter spam has long been a critical but difficult problem to be addressed. So far, researchers have developed a series of machine learning-based methods and blacklisting techniques to detect spamming activities on Twitter. According to our investigation, current methods and techniques have achieved the accuracy of around 80%. However, due to the problems of spam drift and information fabrication, these machine-learning based methods cannot efficiently detect spam activities in real-life scenarios. Moreover, the blacklisting method cannot catch up with the variations of spamming activities as manually inspecting suspicious URLs is extremely time-consuming. In this paper, we proposed a novel technique based on deep learning techniques to address the above challenges. The syntax of each tweet will be learned through WordVector Training Mode. We then constructed a binary classifier based on the preceding representation dataset. In experiments, we collected and implemented a 10-day real Tweet datasets in order to evaluate our proposed method. We first studied the performance of different classifiers, and then compared our method to other existing text-based methods. We found that our method largely outperformed existing methods. We further compared our method to non-text-based detection techniques. According to the experiment results, our proposed method was more accurate.

CCS Concepts

•**Security and privacy** → **Phishing;** *Spoofing attacks;*

Keywords

Twitter spam detection; deep learning; social network security

1. INTRODUCTION

In the recent years, online social networks have become increasingly prevalent platforms where users can post their messages and share ideas around the world. Particularly, Twitter tends to attract users as it provides free microblogging service for customers to broadcast or discover messages within 140 characters, follow other users and so on through multiple devices [12]. For each month, there are even 42 million new accounts created in Twitter [3]. With the popularity of Twitter, criminal accounts can post plenty of spam, which may include suspicious URLs to redirect users to phishing or malicious websites [3]. Consequently, Twitter spam becomes a severe problem and has bad influence on individuals' networking experience. It was reported that 8% URLs were spam in a 2-million-URL dataset [11]. Moreover, it was even more baleful than email spam, with the click-through rate at 0.13%, against much lower result of email spam only at 0.0003%∼0.0006%.

There are many researchers focusing on tackling the problem for the purpose of maintaining the social network security in Twitter by filtering spam. For instance, a tweet content-based classifier was built according to message linguistic analysis [25]. But it could not generate a set of comparable results since only one algorithm was employed in its mechanism. Lately, most researches have put emphasis on establishing machine learning-based binary classifiers with the input of statical features [3, 27, 31, 13, 7, 20]. The features could be picked up from Twitter's Streaming APIs and calculated by a JSON object, and they include account-level attributes (like number of followings, amount of followers and age of account) and user-level attributes (such as quantity of URLs, digits, hashtags in the tweet respectively)[7]. However, there existed some issues in terms of feature extraction and unsatisfied accuracy. In the procedure of collecting data, it was observed that Twitter spam would drift [7, 6] and features were easily fabricated [13, 26, 33].Besides, by organizing the performance of existing researches, the average of accuracy can only achieve at 85% or so. Another technique covers blacklist services, but it was showed that above 90% users might click the malicious URLs before they were blacklisted [6]. At the same time, blacklisting techniques are extremely time-consuming due to individuals participation for unsolicited information recognition. Consequently, these challenges contribute to the motivation of our work.

To deal with those problems including single supporting algorithm, feature extraction issue, accuracy shortage and low speed, an effective classification method based on deep learning is proposed by this paper. Firstly, we apply Word2Vec to pre-process the tweets instead of feature extraction, where the technique adopted is an advanced lan-

ACSW '17, January 31-February 03, 2017, Geelong, Australia

DOI: http://dx.doi.org/10.1145/3014812.3014815

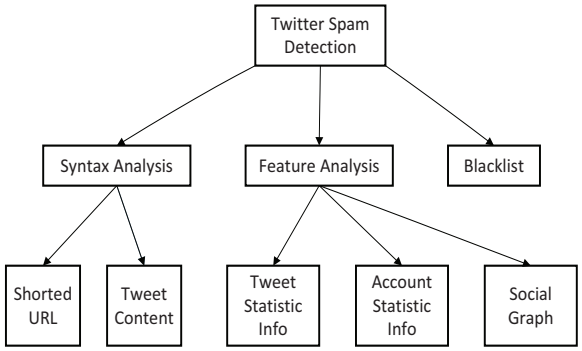

Figure 1: Twitter spam detection category

guage processing method in deep learning and it can convert word or document to representative vector [15]. Afterwards, a binary detection model is built on the basis of several machine learning algorithms to distinguish spam and non-spam. At the next stage, parameter setting is assigned for spam filtering. The experiments are set up with a real-world 10-day ground-truth dataset. The following step refers to compare our classification outcome to those also analyse the content of tweets. Finally, we make a further comparison between the new method and current detection techniques which do not rely on text analysis with respect to accuracy. As a result, our innovative methods are proved to outperform them.

The main contribution of our work is summarized as follows:

We put forward a new Twitter spam detection method based on deep learning which has addressed the challenges of existing classifiers (low speed, under-standard accuracy and characteristic extraction problem). The automatic system is operated fast without statistic feature input. What's more, it realizes higher accuracy of about 95% than current performance (87% averagely).

The rest of the paper is organized as follows. Section 2 shows some related researches on Twitter spam detection. Our innovative classification method based on deep learning is explained in detail at section 3. In Section 4, experimental setting and performance result are illustrated. And the comparisons between achieved result and existing accuracies (both at the same text-based and non-text-based classifiers) are also presented. In Section 5, we describe some factors which may influence the classifier performance. Finally, Section 5 concludes the paper.

2. RELATED WORK

Many efforts have been made to develop spam detection techniques on Twitter in the last decade. In this section, we explain the state of the art in three categories: syntax analysis, feature analysis, and blacklist techniques (See Figure 1).

2.1 Detection based on Syntax Analysis

Syntax-based detection methods analyse tweets at character or word level. There are some work focusing on inspecting shortened URLs inside tweets. Shortened URLs can be generated by shortener services by inputting a long URL, which are used by spammers to hide their malicious URLs. Lee and Kim [17] have proposed an innovative Twitter malicious URL detecting system according to the relevance of suspicious URLs, though their method could not be used for dynamic redirections. Wang, et al. [32] developed a dataset of click traffic feature to determine shortened URLs to be spam or not spam. Moreover, there are some work considering tweet content or text. For example, Tang, et al. [29] extracted tweet content for the input of classifier by employing a deep learning neural network model to learn syntactic contexts of embedded words and label information. For Tang, et al.ą́rs method, there was still room for improvement in terms of its performance (highest F1-measure 87.61%<90%). Rybina [25] also discussed the advantages of utilizing linguistic analysis to build text classification such as document-level modelling. However, the text classification technique did not include multiple machine learning methods, so it was not available to compare different performance results. Besides, the work [24] developed a text classifier on the basis of several different machine learning algorithms. These two work found that Naïve Bayes is the most efficient algorithm among all the methods in terms of accuracy and time cost (F1-measure >= 90%).

2.2 Detection based on Feature Analysis

In this field, there are also many methods that select account and/or tweet features as training data for the input of machine learning based classifiers, such as the work [3, 27, 31, 13, 7, 20]. Account features can be the age of user account, the number of followings, and the number of followers. Tweet features include the ratio of tweets which contain URLs, the average number of hashtags in a tweet and etc. To effectively select those features, some extraction techniques were used and developed. For example, Benevenuto, et al. [3] analysed ten most important attributes by χ^2 method, and Chen, et al. [7] collected features from Twitter's Streaming APIs and a JSON object. After the feature dataset is built, it is vital to determine the optimal classification algorithm to train them. Therefore, the focus of current Twitter spam detection methods is to examine different machine learning techniques. For example, Wang [31] chose Bayesian theory as the learning model to detect spammers. Benevenuto, et al. [3] inspected both spam and spammers based on the Supporting Vector Machine method. Stringhini, et al. [27] created a spam classifier by applying the Random Forest process. Lee, et al. [16] obtained features from spam profiles which were collected by honeypots, and then trained them in multiple machine learning algorithms, such as Decorate and LogitBoost [1]. There are two major problems of using feature-based machine learning classifiers. The first one is about Twitter spam drift which would influence the performance of a trained classifier [7, 6]. Liu, et al. [20] later solved this challenge by proposing a new technique which combined fuzzy-based redistribution and asymmetric sampling together. Another problem is that although Twitter data features can be easily extracted with the support of statistical methods, it is difficult to avoid feature fabrication in the data collection processes. To address the problem, researchers usually adopted social graph to expose robust features in order to prevent feature fabrication [13, 26, 33]. Jin, et al. [13] reported social network features which consisted of individual characteristics of user profiles and their behaviours. Song, et al. [26] generated Twitter social graphs and expose spam according to social distance

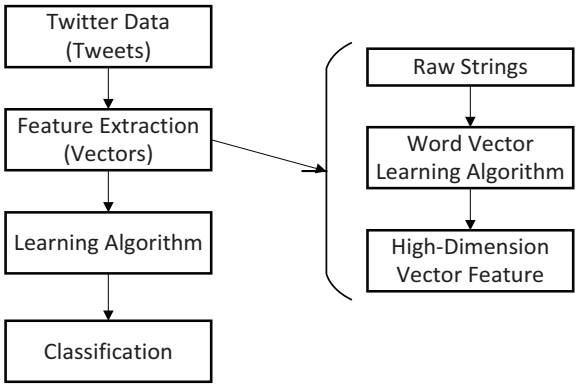

Figure 2: New Twitter classification workflow based on deep learning

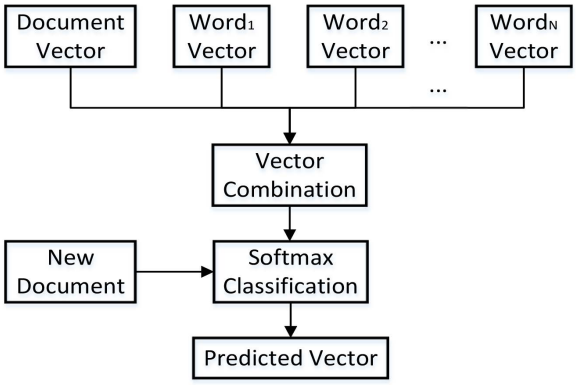

Figure 3: The procedure of learning document vector, where N represents the number of the words in a document.

??

and connectivity between followees and followers. Yang, et al. [33] constructed social graph according to the local clustering coefficient, betweenness centrality and bidirectional links ratio. Based on the social graphs, spamming account will be detected by analysing graph mathematical features. The features used in this method was proved to be more robust than existing algorithms. However, when considering time cost on data collection, this method becomes too complex to be used in the real world.

2.3 Blacklist Techniques

Blacklist techniques are commonly deployed in web filtering services such as Twitter spam detection, with the functionality of blocking malicious websites according to their information analysis like user feedback and website crawling. Ma, et al. [21] presented a lightweight blacklisting approach with lower cost than existing classifiers. Oliver, et al. [23] detected baleful URLs by using blacklisting technique which was integrated in a so-called Web Reputation Technology. However, this method has to rely on manual labelling which is too time-consuming.

In a nutshell, current spam detection methods on Twitter are still not sufficient to detect spamming activities quickly and accurately in terms of Recall, Precision and F1-measure. To achieve less time consuming and better performance turns into the motivation of our work.

3. DEEP LEARNING BASED CLASSIFIER

This section describes a new Twitter spam detection technique including vector-based characteristics training by WordVector techniques and binary classifier building using multiple machine learning algorithms. Figure 2 shows the workflow of distinguishing Twitter spam through our new method.

3.1 Deep Learning Primer

With the limitation of Natural Language Processing (NLP) ability for conventional machine learning algorithm using raw strings, deep learning was developed to be competent to understand and analyse text using a deep neural network with multiple layers [15]. Through the network, each output of the previous layer turns to be the input of the next level. In particular, deep leaning neural language techniques owe strong ability on language analysis, with distributed vectors trained under WordVector method [15]. Text-based vector representation for words are applied widely in systems of linguistic analysis [8, 28].

3.2 Detection Framework

Different from the conventional detection, we complete picking up attributes according to the content of tweets using Word2Vec instead of feature collection and generation.

First of all, we apply Word2Vec to map each word in the whole dataset into corresponding multidimensional vector. It employs a two-level neural network, where Huffman technique is used as hierarchical softmax to allocate codes to frequent words [22]. It improves the efficiency of training model, since high-frequency words can be processed fast [14]. Applying this technique, the word vector-based representation is trained through stochastic gradient descent and the gradient is achieved by backpropagation. What's more, optimal vectors are obtained for each word by CBOW or Skip-gram [22].

Furthermore, Doc2Vec training model is used to assign one vector representing every tweet using Paragraph Vector modelling [14]. Based on Word2Vec, a tweet-length document vector is trained by the combination of word vectors and unique document vector per record. By repeating the procedures, each optimal document-based vector can be learned (as shown in Figure **??**).

After document vectors with high-level dimension learned, they are treated as the input features of several machine learning techniques, such as the Random Forest or Neural Network, along with the label of spam/non-spam. The document representation $\vec{d}$ can be defined as

$$\vec{D} = \{d_1, d_2, \ldots, d_M\},$$

where M is the dimension amount of the document vector, d is the value for each level of it.

By adding the variable binary label, the tweet can be indicated as

$$\vec{t} = (\vec{D}, label),$$

where t represents the concatenate vector, and label is the tweet flag of spam or non-spam.

Thus, the training dataset T is expressed as

$$T = (\vec{t_1}, \vec{t_2}, \ldots, \vec{t_N}),$$

Table 1: The List of Methods for Comparison

Detection Method	Description
Text-based using Deep Learning (Internal)	Random Forest: classification applying the Random Forest algorithm [5] to process the word representation trained by WordVector Technique.
	Neural Network (MLP): detection method using the neural network MLP [10] with input extracted by WordVector.
	Decision Tree: employing a greedy splitting method to build a tree [9], along with WordVector pre-processing.
Traditional Text-based (Vertical Comparison)	Palladian: the text classifier working with n-grams which are a series of tokens for the length [30].
	Complementary Naive Bayes: Multinomial Naive Bayes model which can detect words distribution in documents [24].
	Complementary Naive Bayes (Frequencies): Complementary one with term frequency. [30]
Feature-based Supported by Machine Learning (Horizontal Comparisom)	Naive Bayes: a two-layer classification method, with one level representing the label of spam/non-spam,and another including a set of features [2].
	Random Forest: an anti-sensitive method, with an extra layer added [18].
	Decision Tree (C4.5): a traditional machine learning technique with multiple retrieving and ordering [19].

where N is the number of tweets in training dataset.

With a training dataset, a binary classification function C applying traditional machine learning methods is generated to predict attributes of testing data without labels, with a label vector $\vec{L}$ in the order of corresponding messages. It is demonstrated as

$$\vec{L} = (l_1, l_2, \ldots, l_n) = C(\vec{D}_1, \vec{D}_2, \ldots, \vec{D}_n),$$

where n is the tweets number of testing data.

4. PERFORMANCE EVALUATION

In the section, we first demonstrate our performance results using several different classification methods based on the ground-truth datasets. We then compare our experimental outcomes to two kinds of existing classifications which are text-based and non-text-based. These classifiers are shown in Table 3, where the first one part represents three classifiers in our proposed methods (internal) and the rest two parts are about similar (vertical) and different (horizontal) techniques.

4.1 Experiment Setting

4.1.1 Ground-Truth Twitter Dataset

To evaluate our proposed Twitter spam detection technique, we apply a real-life 10-day ground-truth dataset from Twitter, which contains 1,376,206 spam tweets and 673,836 non-spam messages excluding blank lines. We wrote a Java program to process the raw text including removing the blank lines and assigning labels classified as spam or non-spam. In order to determine the impact of dataset properties on performance, 4 sub-datasets are picked up continuously or randomly. The 4 samples are shown in Table 2. In theory, the ratio of spam to non-spam is 1:1, like Dataset 1 and Dataset 3. However, in the real world, there were approximately 95% tweets are non-spam [11]. Thus, we set the amount of non-spam to be 19 times as much as spam in Dataset 2 and Dataset 4.

4.1.2 Basic Parameters Setup

To achieve the whole workflow of our innovative detection technique, we build classification process on KNIME Analysis Platform [4]. Basically, we run the experiments on Windows 10 operation system at a server with Inter(R) Core(TM) i7 CPU of 12 GB. In the workflow, at the first layer, the WordVector Training Model is set as Doc2Vec, along with the learning rate of 2% and the layer size to be 200. Besides, another step consists of several different traditional machine learning models (one is utilized each time). To avoid error, we loop each experiment for 100 times and calculate the mean of each performance metric. For each run of experiments, the dataset is independently split into 60% training data and 40% testing data.

Table 2: Sample Datasets

Dataset No	Dataset Type	Spam : Non-spam
1	Continuous	5k : 5k
2	Continuous	5k : 95k
3	Random	5k : 5k
4	Random	5k : 95k

Table 3: Confusion Matrix

	Predicted	
	Spam	Non-spam
Spam	TP	FP
Non-spam	FN	TN

4.1.3 Performance Metrics

To evaluate the performance of our created classification and make it comparable to current approaches, we use Recall (Sensitivity), Precision, F-measure and Accuracy (ACC) to measure the capability of classifiers.

Traditionally, the result of spam classification contains the amount of projected spam and non-spam. Table 3 shows the variables TP (True Positive), FP (False Positive), TN (True Negative), and FN (False Negative). TP is the number of spam tweets which are correctly classified as spam, and FP represents the amount of non-spam which are wrongly labeled as spam. On the contrary, TN refers to the quantity of non-spam which are exactly considered as non-spam, while FN indicates the data for spam messages which are treated as spam by mistake.

Accuracy (ACC) means the ratio of tweets identified cor-

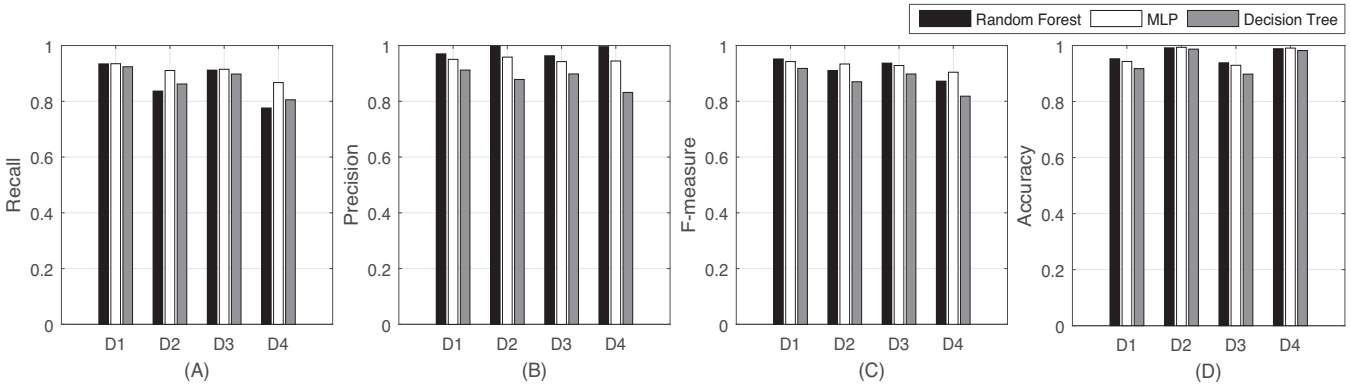

Figure 4: Performance Value of our detection method based on deep learning based on 4 sampled datasets. (A) Recall; (B) Precision; (C) F-measure; (D) Accuracy

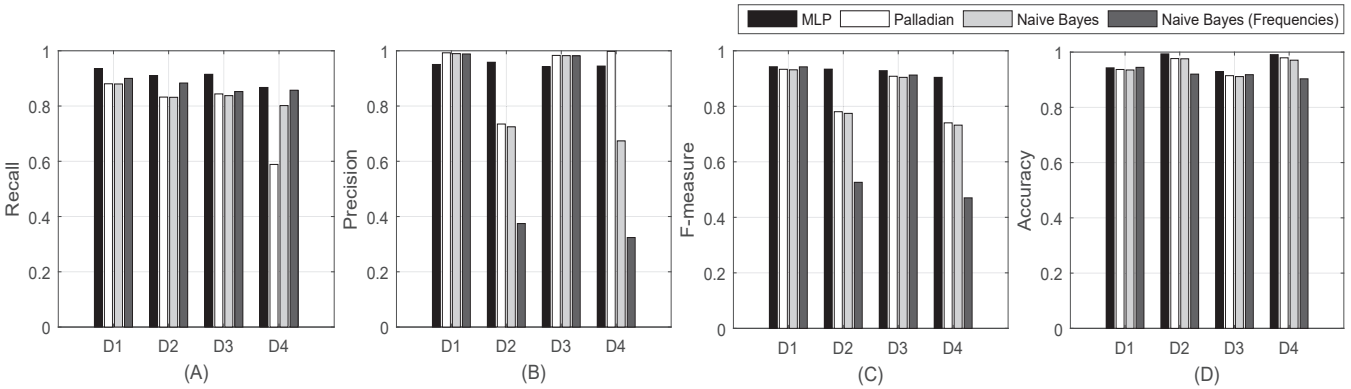

Figure 5: Vertical Comparison of performance values between our technique and traditional text-based detection approaches based on 4 sampled datasets. (A) Recall; (B) Precision; (C) F-measure; (D) Accuracy

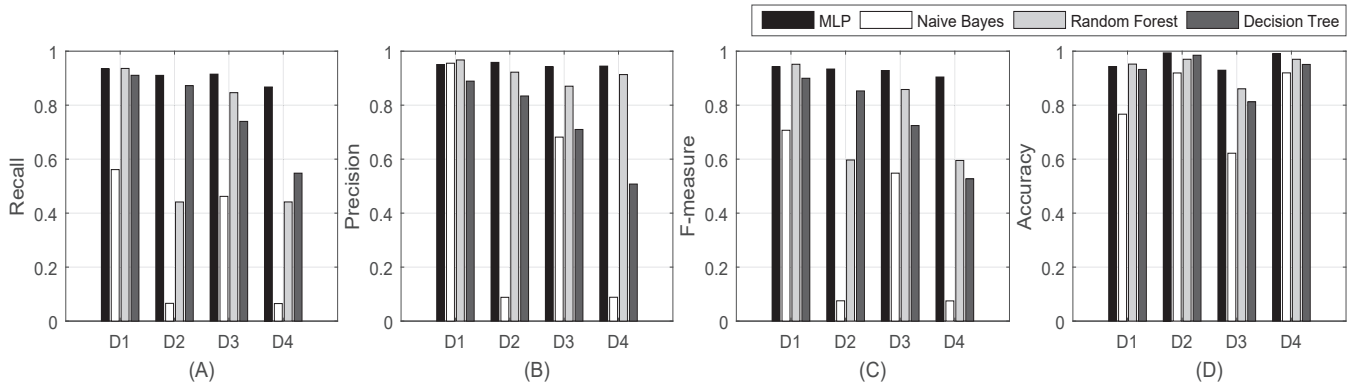

Figure 6: Horizontal Comparison of performance values between our technique and feature-based methods based on 4 sampled datasets. (A) Recall; (B) Precision; (C) F-measure; (D) Accuracy

Table 4: Impact of the Spam Ratio by Dataset 1 and 2 using MLP

Unit: %	Recall	Precision	F-measure	Accuracy
Dataset 1	93.48	95.04	94.25	94.30
Dataset 2	91.03	95.84	93.37	99.35

Table 5: Impact on Sample Dataset Discretisation of Dataset 1 and 3 using MLP

Unit: %	Recall	Precision	F-measure	Accuracy
Dataset 1	93.48	95.04	94.25	94.30
Dataset 3	91.48	94.23	92.83	92.94

rectly to all tweets. It is expressed as

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

Recall (Sensitivity) is defined as the ratio of correctly classified spam in total actual spam, as

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

Precision is defined as true projected spam to classified spam. It can be obtained by

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

F-measure is the harmonic mean of Precision and Recall, and it can be calculated as follow:

$$F{-}measure = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN}$$

4.2 Comparison of Classifiers

In this subsection, we evaluate the performance of our work through three different classifying algorithms with vectors input trained by WordVector technique in a deep learning style on four sampling datasets. The comparison results will suggest the optimal classifier that can be used in our method (i.e. internal comparison). The list of classifiers is shown in Table 1.

As is shown in Figure 4, all the three algorithms perform well. Almost all performance values are higher than 80%, and most of them are more than 90%. As represented in Figure 4, the technique of Random Forest outperforms the other two methods all the time at the aspects of Precision and Accuracy. Furthermore, MLP achieves the highest performance in the metrics of Recall, Precision and Accuracy over all the four datasets. For the F-measure, MLP achieves highest performance in Dataset 2 and 4, and the second best performance in Dataset 1 and 3. Since the ratio of spam on Dataset 2 and 4 is similar to the real world, it is reasonable to achieve the highest F-measure on them. Besides, there is no significant difference among the four performance metrics (all $\sim$ 95% averagely). In the following, we select MLP as our classification method along with WordVector technique to compare to other approaches as listed in Table 1.

4.3 Comparison (vs. Syntax-based Methods)

In this section, we compare our method to 3 existing text-based techniques vertically. Figure 5 describes the differences among different text-based methods. It indicates that our proposed method using MLP performs is better than all current work in terms of Recall, F-measure and Accuracy. For Precision, the performance of our method is about 25% higher than the second place on Dataset 2 but 5% less than the best for other datasets. It even achieves double F-measure of Naive Bayes (Frequencies) on Dataset 2 and 4. Overall, it outperforms all the others.

4.4 Comparison (vs. Feature-based Methods)

We further compare our method to other feature-based detection methods. The performances for all four metrics on for datasets are better than other all the time. As shown in Figure 6, the F-measure is much higher than others, with averagely 30% higher than Random Forest and almost nine times of Naive Bayes in Dataset 2 and 4. Even the Decision Tree method achieves almost the same as our method at Dataset 1, it only remains half when testing on Dataset 4.

5. DISCUSSION

According to our performance evaluation, there are two factors that affects the classifier function in terms of dataset: 1) the proportion of spam and non-spam and 2) the sample dataset discretisation.

5.1 Impact of Spam Ratio

We show the impact of the spam ratio in Table 4. It can be found that with the change of spam ratio, the performance of our proposed method remains stable. The best one achieves 2.45% on Racall. Therefore, it affects other text-based or non-text-based significantly. For example, in Figure 5, the F-measure of Naive Bayes (Frequencies) is only half in the ratio of 1:19 (spam:non-spam) dataset of it in 1:1 dataset. In addition, the F-measure of Naive Bayes is averagely 60% in 1:1 dataset, but it becomes one fifth in 1:19 dataset.

5.2 Impact of Sample Dataset Discretisation

We further study the impact of sample dataset discretisation. The results are shown in Table 5. It is found that with the change of the ratio of spam, the performance of our proposed method remains stable. The biggest difference is only 2.% on Recall. Accordingly, the performance on continuous dataset would be slightly better than randomly sampled dataset for all detection from Figure 4, 5 and 6.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we explored the issues on the current Twitter spam detection techniques, and proposed a new classification method based on deep learning algorithms to address them. For the purpose of judging its performance evaluation, we firstly collected a part of labeled data (376,206 spam and 73,836 non-spam tweets) from a 10-day ground-truth dataset with more than 600 million real-world tweets. Then we utilized WordVector technique for pre-processing them and converted them into high-dimension vectors.

Future work may include several aspects: 1) The evaluation of this paper is mainly on empirical studies. We will carry out theoretical studies on the outperformance of our methods in order to better understand the deep-learning based spam detection framework. This will in addition help us improve the performance. 2) We will compare more classifier and other methods in the future in order to demonstrate the pros and cons of our proposed method. 3) We

will finally collect more real data from social media, particularly the datasets from other social media such as Facebook and microblogs, and study the immigration of our spam detection framework. This part of work is very important to both industries and academia because social spam is also very critical in other social media platforms.

7. REFERENCES

[1] R. Aires, A. Manfrin, S. M. Aluísio, and D. Santos. *Which Classification Algorithm Works Best with Stylistic Features of Portuguese in Order to Classify Web Texts According to Users' needs?*. ICMC-USP, 2004.

[2] N. B. Amor, S. Benferhat, and Z. Elouedi. Naive bayes vs decision trees in intrusion detection systems. In *Proceedings of the 2004 ACM symposium on Applied computing*, pages 420–424. ACM, 2004.

[3] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting spammers on twitter. In *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)*, volume 6, page 12, 2010.

[4] M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, K. Thiel, and B. Wiswedel. Knime-the konstanz information miner: version 2.0 and beyond. *AcM SIGKDD explorations Newsletter*, 11(1):26–31, 2009.

[5] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

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

[7] C. Chen, J. Zhang, Y. Xie, Y. Xiang, W. Zhou, M. M. Hassan, A. AlElaiwi, and M. Alrubaian. A performance evaluation of machine learning-based streaming spam tweets detection. *IEEE Transactions on Computational Social Systems*, 2(3):65–76, 2015.

[8] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

[9] T. G. Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

[10] V. N. Ghate and S. V. Dudul. Optimal mlp neural network classifier for fault detection of three phase induction motor. *Expert Systems with Applications*, 37(4):3468–3481, 2010.

[11] 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*, pages 27–37. ACM, 2010.

[12] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 56–65. ACM, 2007.

[13] X. Jin, C. Lin, J. Luo, and J. Han. A data mining-based spam detection system for social media networks. *Proceedings of the VLDB Endowment*, 4(12):1458–1461, 2011.

[14] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196, 2014.

[15] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[16] K. Lee, J. Caverlee, and S. Webb. Uncovering social spammers: social honeypots+ machine learning. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 435–442. ACM, 2010.

[17] S. Lee and J. Kim. Warningbird: Detecting suspicious urls in twitter stream. In *NDSS*, volume 12, pages 1–13, 2012.

[18] A. Liaw and M. Wiener. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[19] S. Liu, J. Zhang, Y. Wang, and Y. Xiang. Fuzzy-based feature and instance recovery. In *Asian Conference on Intelligent Information and Database Systems*, pages 605–615. Springer, 2016.

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

[21] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Learning to detect malicious urls. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):30, 2011.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[23] J. Oliver, P. Pajares, C. Ke, C. Chen, and Y. Xiang. An in-depth analysis of abuse on twitter. *Trend Micro*, 225, 2014.

[24] J. D. Rennie, L. Shih, J. Teevan, D. R. Karger, et al. Tackling the poor assumptions of naive bayes text classifiers. In *ICML*, volume 3, pages 616–623. Washington DC), 2003.

[25] K. Rybina. *Sentiment analysis of contexts around query terms in documents*. PhD thesis, MasterâĂŹs thesis, 2012.

[26] J. Song, S. Lee, and J. Kim. Spam filtering in twitter using sender-receiver relationship. In *International Workshop on Recent Advances in Intrusion Detection*, pages 301–317. Springer, 2011.

[27] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 1–9. ACM, 2010.

[28] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[29] D. Tang, F. Wei, B. Qin, T. Liu, and M. Zhou. Coooolll: A deep learning system for twitter sentiment classification. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 208–212, 2014.

[30] D. Urbansky, K. Muthmann, P. Katz, and S. Reichert.

Tud palladian overview. *TU Dresden, Department of Systems Engineering, Chair Computer Networks, IIR Group*, 5, 2011.

[31] 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. IEEE, 2010.

[32] D. Wang, S. B. Navathe, L. Liu, D. Irani, A. Tamersoy, and C. Pu. Click traffic analysis of short url spam on twitter. In *Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 2013 9th International Conference Conference on*, pages 250–259. IEEE, 2013.

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