Spammer Detection via Combined Neural Network

Weiping Pei[1,2], Youye Xie[2], and Gongguo Tang[2(✉)]

[1] Biliang, Ltd., Zhuhai, China
[2] Colorado School of Mines, Golden, USA
{weipingpei,youyexie,gtang}@mines.edu

Abstract. Social networks, as an indispensable part of our daily lives, provide ideal platforms for entertainment and communication. However, the appearance of spammers who spread malicious information pollutes a network's reliability. Unlike email spammers detection, a social network account has several types of attributes and complicated behavior patterns, which require a more sophisticated detection mechanism. To address the above challenges, we propose several efficient profiles and behavioral features to describe a social network account and a combined neural network to detect the spammers. The combined neural network can process the features separately based on their mutual correlation and handle data with missing features. In experiments, the combined neural network outperforms several classical machine learning approaches and achieves 97.5% accuracy on real data. The proposed features and the combined neural network have already been applied commercially.

Keywords: Spammer detection · Social network · Deep learning
Data mining

1 Introduction

1.1 Background

The development of social network platforms, such as Twitter, Facebook, and Sina, have made communication around the world much more convenient. With the increasing impact of the social network, a significant number of spammers appears aiming to conduct malicious behaviors, such as spreading malicious URLs, posting abusive comments and hijacking the social hotspot. And that makes spammers detection one of the top priorities of social network companies. This paper focuses on spammers detection for the Sina microblogs, one of the most influential social network platforms in China, which allows users to post microblogs with less than 140 characters along with images or videos. The spammers are also called the paid posters or internet water army on Sina [3]. The analysis on feature effectiveness and the proposed combined neural network are

W. Pei and Y. Xie have contributed equally to this work.

P. Perner (Ed.): MLDM 2018, LNAI 10934, pp. 350–364, 2018.
https://doi.org/10.1007/978-3-319-96136-1_28

also applicable to other social networks such as Twitter and Facebook by simply changing the language analysis tool and modifying the number of input nodes accordingly.

1.2 Related Work

Spammers detection appears first in email applications, the goal of which is to filter out spam emails based on the email content [6,12], the abnormal behavior [10,17] and attachments. Social network companies such as Twitter and Sina face similar issues and many methods have been developed for spammers detection in social networks. O'Donovan et al. [11] analyzed the content features of social network accounts and suggested the usefulness of content features in spammers detection. As a complement, Liu et al. [9] studied the behavioral features and proposed a hybrid model to calculate the 'spamming value' for each account.

In addition, machine learning approaches have also produced promising results for spammers detection. Wang [15] proposed a Naïve Bayes method using three graph-based features (profile features) and three content features. Two support vector machine (SVM) based methods were developed by Cheng et al. [5] and Zheng et al. [18], whose approach detects spammers in two phases. The first phase utilizes content features and the second phase considers the topic of the microblogs with the help of a Latent Dirichlet Allocation (LDA) model [1]. Chakraborty et al. [2] designed a Social Profile Abuse Monitoring (SPAM) system which also adopts the multi-stages detection mechanism and achieves 89% accuracy.

However, most of the previous approaches assume that the account has all the features the detection model requires, which is not desirable in practice. Indeed, a new spammer account may not have any significant behavioral features at all. In addition, some proposed features, such as the content topic recognition and accounts similarity analysis, require more feature engineering effort. Our work contribute to the topic of spammers detection in the following ways:

- **Efficient profile and behavioral features.** We investigate the difference between spammers and non-spammers and propose the use of several profile and behavioral features that can be extracted easily from the social network accounts.
- **Combined neural network.** We propose a novel combined neural network that includes a linear regression model (LR) and two artificial neural networks (ANN) to incorporate different types of features. More importantly, the combined model is very flexible in practice since each sub-model within the combined neural network can perform the detection independently, by simply deactivating unwanted sub-models, in the case that some features are missing.
- **High detection accuracy.** We conduct experiments to study the effectiveness of the proposed features and the detection performance of the combined neural network and its sub-models. In our experiment, the combined neural network outperforms other classical machine learning models in literature by achieving 97.5% detection accuracy.

The paper is organized as follows. Section 2 introduces the profile and behavioral features that will be used for our detection model. In Sect. 3, we present our combined neural network and derive its training process. We study the effectiveness of the proposed network in spammers detection in Sect. 4. The paper is concluded in Sect. 5.

2 Features

A social network contains massive data that may change rapidly. So extracting distinctive features is a critical component for spammers detection. Based on the features categories, we consider four profile features and four behavioral features for our model and we will analyze those features' effectiveness in experiments. Some of the features in this paper have proven efficiency in the spammer detection problem [2,5,11,15].

2.1 Profile Features

A registered account in a social network typically has a public profile which describes the attributes of the user. The normal users are more likely to fill out the basic profile so that their friends could recognize them. However, it is cumbersome for spammers to create a lot of accounts with complete profile information. This section analyzes four profile features that are useful in spammers detection and summarizes them in Table 1.

- **Fans ratio.** The followers, also called fans, are users that follow one's account, while the followings are the users that one follows. Spammers would follow many normal users in the hope that they would follow back so that they can spread spam messages. And a few normal users would indeed follow a stranger. This results in a limited number of followers and lots of followings for spammers. We define *fans_ratio* to quantify this phenomenon

$$fans\_ratio = \frac{follower}{follower + following}$$

  where *follower* and *following* represent the number of followers and followings respectively. We collect 1000 spammers and non-spammers' *fans_ratio* in Fig. 1. It is obvious that spammers (red cross) are more likely to have lower *fans_ratio* than non-spammers (green cross).
- **Account level.** Each account has a level indicator that reflects the activeness of this account, determined by how long the account is established and how often the user posts a microblog. Normally, an active account would have a higher level than a silent one. We record 1000 spammers and non-spammers' *account levels* in Fig. 2. To compensate the level data's disproportion, all the levels are normalized within $[0, 1]$ for training and testing.
- **Verification.** The Sina microblog network enforces an identification policy. An account can earn a verification mark by verifying its owner's identity. Since detailed personal information are needed for verification, few spammers have such a verification mark.

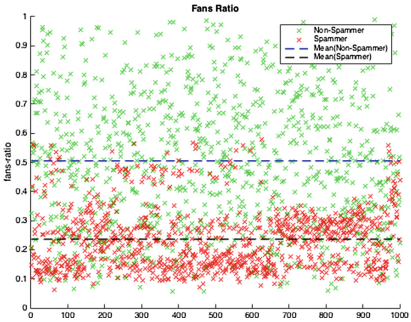

Fig. 1. The *fans_ratio* feature. (Color figure online)

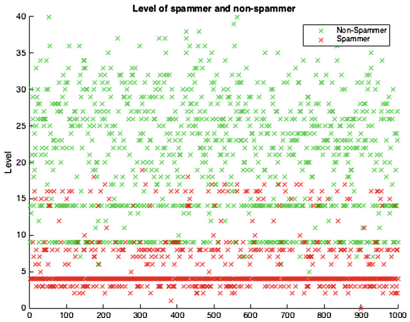

Fig. 2. The *account level* feature. (Color figure online)

Table 1. The proposed profile features.

Feature name	Definition
fans_ratio	The fans ratio
lv	The account level normalized within $[0, 1]$
is_common	The account is not verified
is_V	The account is verified as a personal account
is_expert	The account is verified as an expert's account
is_company	The account is verified as an organization's account
Brief information	The account has an introduction

– **Self introduction.** A brief self introduction can be added for each account that is displayed under the account name. Most spammers do not have a self introduction.

2.2 Behavioral Features

Besides the profile features, spammers' accounts also operate in a different manner due to their specific blogging purposes. Behavioral features provide a way to quantify that difference. We propose four behavioral features which can be easily extracted from the microblogs of an account.

– **User interaction.** For each microblog, other users can comment, repost it or leave a'like'. The microblog itself might also be a reposted one. The number of interaction activities reflects the attention a microblog has attracted. We binarize those features based on whether a particular microblog has the above interaction activities.
– **Special characters.** The microblog also provides other enhanced interaction using special characters. We summarize those special characters and their associated functions in Table 2.

Table 2. The special characters and proposed content features.

Special characters	Function
I	The subject of the message
@	Post a microblog and alert other users
#	Post a microblog with hashtags
//@	Repost others reposted microblogs
URLs	Attach the URLs
Video	Attach the video
Content features	Definition
Number of words	The number of words normalized within $[0, 1]$
Non-repeated words	Fraction of non-repeated words
Non-stop words	Fraction of non-stop words
Non-repeated and non-stop words	Fraction of non-repeated and non-stop words

– **Content features.** Instead of doing the content mining, this paper leverages simpler content features in terms of the text length and the valuable content. We segment the microblog's content and compute the corresponding content features as summarized in Table 2. Generally, stop words are a group of words that occur frequently but do not carry much information, like the word 'the' in English.
– **Publish time.** Controlling by the software, spammers keep posting even at deep night and the interval time between two microblogs is more regular compared to non-spammers. Therefore, we use an 8×1 vector, whose i-th entry is 1 when the microblog is posted during $[(i-1) \times 3 : 00 - (i \times 3) : 00]$, to indicate the publish time. A 7×1 vector is used to represent the day of the week of the posting.

3 The Combined Neural Network

In the past decade, neural networks have achieved great success in many machine learning tasks [8,13,16]. A properly designed neural network model has sufficient capacity to classify complicated data in high dimensional spaces. It is also found that hybrid models which utilize multiple models in a proper way can improve the classification performance [4]. However, they fail to consider the situation of missing data. Inspired by those ideas, we generalize and propose a combined neural network which hybridizes linear regression models and artificial neural networks. Specifically, for the spammer detection problem, our combined network is composed of one linear regression model (LR) and two neural networks (ANN) as shown in Fig. 3.

With the proposed combined network, highly correlated features can be processed together in a sub-networks to avoid the influence of other less related features. In practice, if some of the account's features are absent, the sub-models

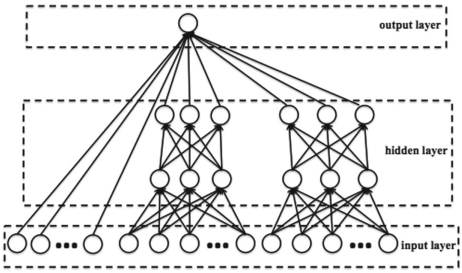

Fig. 3. Architecture of the proposed combined neural network.

Table 3. The combined neural network for spammers detection. The structures show the number of nodes from input to output layers for each sub-model.

Submodel	Feature name	Structures
LR	Profile features	[7 2]
ANN-1	Publish time	[150 150 52 2]
ANN-2	User interaction, characters and content features	[140 140 49 2]

can still perform the detection independently. In addition, the design reduces computational complexity compared to a fully connected neural network and makes it easier to handle inputs with different types of attributes.

Based on the features analysis in the last section, we category the features associated with an account into three classes, the profile features, the publish time and the behavioral features (except the publish time) whose dimensions are $7,150$ (15×10 for ten most recent microblogs) and 140 (14×10 for ten most recent microblogs) respectively. Therefore, given N undefined accounts, the network input is $\mathbf{X} = [\mathbf{X}^L, \mathbf{X}^{A_1}, \mathbf{X}^{A_2}]^T \in \mathbf{R}^{297 \times N}$ where $\mathbf{X}^L \in \mathbf{R}^{7 \times N}$, $\mathbf{X}^{A_1} \in \mathbf{R}^{150 \times N}$ and $\mathbf{X}^{A_2} \in \mathbf{R}^{140 \times N}$.

We model the profile features using linear regression (i.e. a neural network without hidden layers) since those features are almost linear separable based on the experiments. The rest two classes are modeled using two independent artificial neural networks due to their non-linear characteristics. Empirically, we observe that when the numbers of nodes in the first and second hidden layers equals the 100% and 35% of the number of nodes in the input layer, the combined model can achieve best performance as shown in Table 3.

3.1 Model Details

We first introduce parameters' notations for the combined neural network in Table 4. The corresponding matrices are written in boldface capital letters without subscripts. For example, $\mathbf{W}^{L,1}$ represents the weight matrix connecting the first and second layer in the linear model. In addition, since the input layer is the first layer, we have $x_i^L = a_i^{L,1}$ and $x_i^A = a_i^{A,1}$.

Table 4. The combined neural network's parameters.

Notation	Definition
y_i	The i-th node's output in the output layer
$w_{ij}^{L,l}, w_{ij}^{A,l}$	The weight parameters connecting the i-th node in the l layer to the j-th node in the $(l+1)$ layer in the linear model and ANN
$b_j^{L,l}, b_j^{A,l}$	The bias parameters connecting to the j-th node of the $(l+1)$ layer in the linear model and ANN
$z_i^{L,l}, z_i^{A,l}$	The i-th node's input of the l-th layer in the linear model and ANN
$a_i^{L,l}, a_i^{A,l}$	The i-th node's output of the l-th layer in the linear model and ANN

The linear model can be easily shown as a linear transformation $\mathbf{Z}^{L,2} = \mathbf{W}^{L,1}\mathbf{X}^L + \mathbf{B}^{L,1}$. Similarly, for the ANN, we have $\mathbf{Z}^{A,4} = \mathbf{W}^{A,3} f(\mathbf{W}^{A,2} f(\mathbf{W}^{A,1}\mathbf{X}^A + \mathbf{B}^{A,1}) + \mathbf{B}^{A,2}) + \mathbf{B}^{A,3}$ where $f(x) = \max(0, x)$ is the rectifier activation function. Based on the above equations, the output layer's input for the combined neural network is $\mathbf{Z} = \mathbf{Z}^{L,2} + \mathbf{Z}^{A_1,4} + \mathbf{Z}^{A_2,4}$.

A softmax function is applied in the output layer to calculate the posterior probabilities. For one particular data point, the network's output is

$$\mathbf{Y} = \left[\frac{\exp\left(z_1^{L,2} + z_1^{A_1,4} + z_1^{A_2,4}\right)}{\sum_{i=1}^{2} \exp\left(z_i^{L,2} + z_i^{A_1,4} + z_i^{A_2,4}\right)}, \frac{\exp\left(z_2^{L,2} + z_2^{A_1,4} + z_2^{A_2,4}\right)}{\sum_{i=1}^{2} \exp\left(z_i^{L,2} + z_i^{A_1,4} + z_i^{A_2,4}\right)} \right]^T$$

where $y_1 = P(\text{Spammer}|x)$ and $y_2 = P(\text{Normal}|x)$. If $y_1 \geq y_2$, the account will be classified as a spammer.

3.2 Model Training

We first derive the parameters derivatives for training. Assuming we have N data points, we evaluate the combined neural network's performance using the cross entropy

$$\mathbf{H} = -\frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{2} y_i^{\prime(n)} \log(y_i^{(n)})$$

where y_i' takes a value either 0 or 1 indicating the ground truth label and the minimum value of $\mathbf{H}$ is 0. Therefore, given N data points, the derivatives of the linear model parameters are:

$$\nabla \mathbf{H}(w_{ij}^{L,1}) = \frac{1}{N}\sum_{n=1}^{N} \nabla \mathbf{H}(w_{ij}^{L,1}, x^{(n)}, y^{(n)}) = \frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{2} \frac{\partial \mathbf{H}}{\partial z_k^{L,2,(n)}} \frac{\partial z_k^{L,2,(n)}}{\partial w_{ij}^{L,1}}$$
$$= \frac{1}{N}\sum_{n=1}^{N} (y_j^{(n)} - y_j^{'(n)}) a_i^{L,1,(n)}$$
$$\nabla \mathbf{H}(b_j^{L,1}) = \frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{2} \frac{\partial \mathbf{H}}{\partial z_k^{L,2,(n)}} \frac{\partial z_k^{L,2,(n)}}{\partial b_j^{L,1}} = \frac{1}{N}\sum_{n=1}^{N} (y_j^{(n)} - y_j^{'(n)})$$

in which

$$\frac{\partial \mathbf{H}}{\partial z_k^{L,2}} = \sum_{j=1}^{2} \frac{\partial \mathbf{H}}{\partial y_j} \frac{\partial y_j}{\partial z_k^{L,2}} = -y_k' + y_k \sum_{j=1}^{2} y_j' = y_k - y_k'$$
$$\frac{\partial z_k^{L,2}}{\partial w_{ij}^{L,1}} = \begin{cases} a_i^{L,1} & k = j \\ 0 & k \neq j \end{cases}, \quad \frac{\partial z_k^{L,2}}{\partial b_j^{L,1}} = \begin{cases} 1 & k = j \\ 0 & k \neq j. \end{cases}$$

The derivatives of the ANN can be calculated efficiently via backpropagation. We first compute the derivative with respect to each node's input, denoted as δ, and propagate it backward. For the output layer and the l-th ($l \leq 3$) layer

$$\delta_i^4 = \frac{\partial \mathbf{H}}{\partial z_i^{A,4}} = y_i - y_i', \quad \delta_i^l = \frac{\partial \mathbf{H}}{\partial z_i^{A,l}} = \sum_{j=1}^{n_{l+1}} \delta_j^{l+1} w_{ij}^{A,l} f'(z_i^{A,l})$$

where n_{l+1} is the total number of nodes in the $(l+1)$ layer (except the bias node) and $f'(z_i^{A,l})$ is the derivative of the rectifier function. $f'(z_i^{A,l}) = 1$ if $z_i^{A,l} \geq 0$ and $f'(z_i^{A,l}) = 0$ otherwise. Therefore, we get the derivatives with respect to the weights and biases for ANN with N input data points as follows.

$$\nabla \mathbf{H}(w_{ij}^{A,l}) = \frac{1}{N}\sum_{n=1}^{N} \nabla \mathbf{H}(w_{ij}^{A,l}, x^{(n)}, y^{(n)}) = \frac{1}{N}\sum_{n=1}^{N} \delta_j^{l+1,(n)} a_i^{A,l,(n)}$$
$$\nabla \mathbf{H}(b_j^{A,l}) = \frac{1}{N}\sum_{n=1}^{N} \delta_j^{l+1,(n)}$$

After calculating the derivatives, the Adam optimization algorithm [7] which updates the learning rate for each parameter adaptively with 0.05 initial learning rate is applied to train the combined network. Given N training samples, the batch size and number of epoches are $N/10$ and 1000 respectively. Dropout [14] is also performed in the ANN during training to prevent overfitting. Nodes in the first and second hidden layers are kept with probabilities 0.5 and 0.8.

4 Experiment and Analysis

We analyze the effectiveness of the selected features and compare the proposed combined neural network to other classical machine learning methods for spammers detection. Throughout the experiment, five-fold cross-validation is applied for a more accurate performance estimate.

4.1 Data and Evaluation Metrics

We collect about 5000 spammer accounts and 5000 non-spammer accounts through a media company and web crawling. For each account, we label it manually and extract its profile features and behavioral features accordingly. We evaluate the model performance based on four evaluation metrics, the accuracy, precision, recall and F1 score defined as follows (Table 5).

Table 5. The confusion matrix entries.

	Spammer	Non-spammer
Predicted spammer	True positive (TP)	False positive (FP)
Predicted non-spammer	False negative (FN)	True negative (TN)

- **Accuracy:** the ratio of the number of correctly classified accounts over the total number of accounts. $A = \frac{TP+TN}{TP+TN+FP+FN}$.
- **Precision:** the ratio of the number of correctly classified spammers to the number of accounts that are classified as spammers. $P = \frac{TP}{TP+FP}$.
- **Recall:** the ratio of the number of correctly classified spammers to the number of spammers. $R = \frac{TP}{TP+FN}$.
- **F1 score:** a measure to examine the test accuracy and is computed as $F1 = \frac{2\times P\times R}{P+R}$.

4.2 Features Effectiveness Analysis

Single Feature. In order to verify the proposed features' effectiveness, we apply four classical machine learning classifiers for spammers detection using different single features. Those classifiers, which are also popular in literature, are C4.5, classification and regression trees (CART), support vector machine (SVM) and the Naïve Bayes (NB) classifier. Considering that both the special characters and content features are extracted from texts, we also combine them as the text feature for evaluation. The detection accuracies are summarized in Table 6. We observe that the text features and the publish time is quite distinguishable between spammers and non-spammers and except the content feature, all features achieve at least 78.92% accuracy for four classifiers.

Combined Features. Instead of using only one type of features as the input, in this section, we try different combinations among features and record the detection accuracies in Table 7. We can observe that using combined features promotes the detection accuracy significantly compared to using the single feature. In addition, no matter which model we use, utilizing all features for detection achieves the highest accuracy. Notably, SVM using all features achieves 96.53% accuracy which proves the effectiveness of the proposed features.

Table 6. The detection accuracies using a single feature.

Feature name	C4.5	CART	SVM	NB
Profile features	85.40%	85.74%	85.52%	78.92%
User interaction	84.06%	83.96%	84.77%	85.65%
Special character	86.87%	86.41%	87.70%	88.96%
Content feature	74.55%	75.07%	73.91%	66.37%
Text features	85.87%	86.44%	89.19%	89.80%
Publish time	87.21%	86.93%	86.70%	86.61%

Table 7. The detection accuracies using combined features.

Features name	C4.5	CART	SVM	NB
Profile features and user interaction	88.95%	87.97%	89.21%	86.30%
Profile and text features	92.97%	93.39%	94.71%	90.36%
Profile features and publish time	92.49%	91.95%	90.89%	88.40%
User interaction and text features	91.70%	91.74%	95.81%	95.39%
User interaction and publish time	89.76%	90.20%	92.78%	91.63%
Text features and publish time	89.25%	89.35%	93.69%	91.18%
All features	94.55%	94.06%	96.53%	96.02%

Different Lengths of Behavioral Features. Furthermore, we study the influence of different lengths of behavioral features. For each account, we use its profile features as the initial state (horizontal axis = 1) and add its recent microblogs one by one from which we extract the account's behavioral features. The results are shown in Fig. 4. After adding five microblogs' behavioral features, the accuracies of SVM and NB tend to be steady while the accuracies of C4.5 and CART fluctuate within a narrow range. In order to obtain a higher accuracy while avoiding heavy features engineering effort, using behavioral features from the ten most recent posts are most favorable.

4.3 The Combined Neural Network

In this section, we conduct experiments to evaluate the detection performance of the proposed combined model and its sub-models.

Comparison to Classicial Models. We first compare the detection performance between the proposed combined neural network and other classical

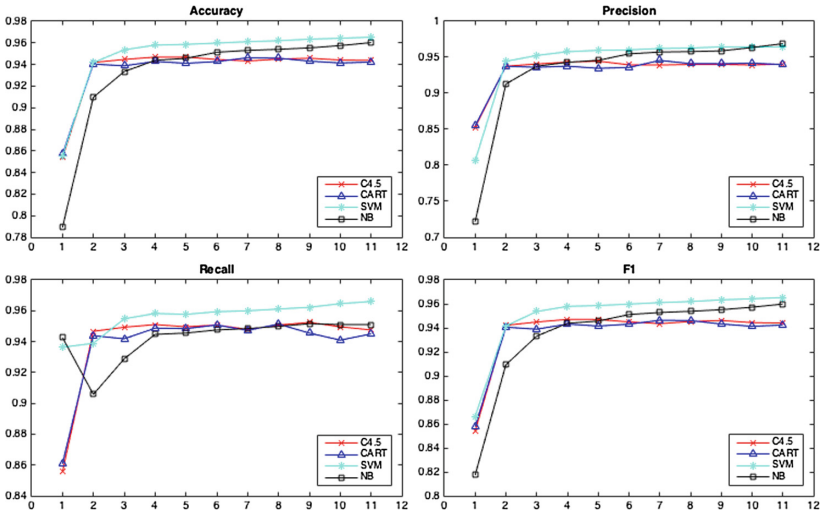

Fig. 4. The accuracy, precision, recall and F1 score with all profile features while different lengths of behavioral features.

Table 8. The detection performances for different models.

Model	Accuracy	Precision	Recall	F1 score
C4.5	94.55%	94.15%	95.01%	94.58%
CART	94.06%	93.82%	94.35%	94.08%
SVM	96.53%	96.46%	96.60%	96.53%
NB	96.02%	96.89%	95.10%	95.98%
Combined model	97.50%	98.24%	97.12%	97.68%

machine learning approaches. The comparison result is shown in Table 8 which shows that the proposed combined model achieves 97.5% accuracy and outperforms other classical machine learning classifiers in the spammer detection problem.

Sub-models Detection Performance. As we mentioned previously, each sub-model in the combined neural network can work independently in the case some features are absent. In this section, we train the combined network and extract each individual sub-model to examine its detection performance. The result is shown in Table 9.

The vote model is a multi-classifiers model that classifies an account based on the voting from three sub-models. We can find that all sub-models achieve promising detection accuracies between 84% and 97%. Therefore, it is not surprising that a vote model can achieve above 97% accuracy. However, a simple vote model ignores the relative magnitudes of each sub-model's output which

Table 9. The detection result using sub-models independently.

Model	Accuracy	Precision	Recall	F1 score
LR	84.50%	78.73%	94.58%	85.93%
ANN-1	95.68%	96.95%	75.68%	95.62%
ANN-2	96.36%	96.72%	95.97%	96.34%
Vote model	97.25%	97.80%	97.09%	97.44%
Combined model	97.50%	98.24%	97.12%	97.68%

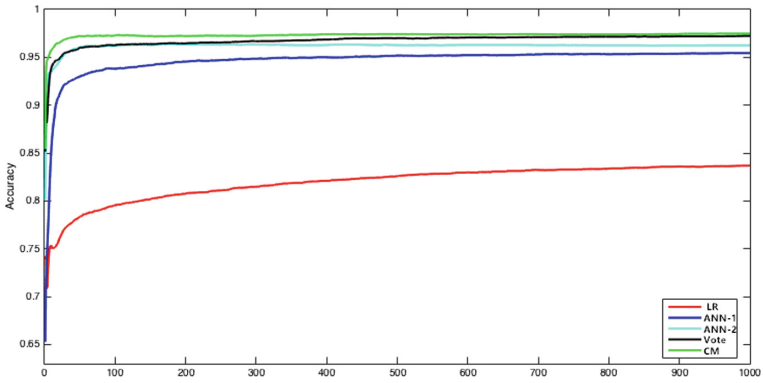

Fig. 5. Different models' accuracies versus the training iteration number.

limits its capability. We also record the sub-models training process in Fig. 5. We can observe that the combined model converges as fast as its sub-models but achieves higher accuracy. Noticeably, the linear model accuracy is far below the non-linear models due to the non-linear characteristics in the spammer detection task.

4.4 Performance Tracking

Although behavioral features are quite effective, it may update rapidly for active users. Spammers would also change their behavior pattern to escape from being detected by the platform. Therefore, it is important to examine how the fast changing environment affects the combined neural network's detection performance.

A Quick Test. We start the study from a quick test by first selecting 500 active accounts including 250 spammers and 250 non-spammers on December 25th, 2016 and implementing different classifiers on those accounts. After two months, we extracted those accounts' features again and their ten most recent microblogs were totally different from the previous. We then performed the classification again using the pre-trained models and recorded the detection accu-

racies in Table 10. All models' accuracies decrease after two months since those users may have already changed their microblog habits and so do the spammers. Fortunately, the combined model still achieve 96.21% accuracy and outperform other models.

Table 10. The accuracies after two months.

Date	C4.5	CART	SVM	NB	Combined model
12/25/16	95.99%	96.65%	98.88%	92.20%	99.11%
02/25/17	91.21%	87.04%	94.98%	91.63%	96.21%

Model Update. In this part, we take a closer look of the influence by the changing behavioral features. A cooperative company provides us with new accounts every day which allows us to update the combined model daily. Namely, we update the combined model using yesterday's data and test it on today's new coming data. The tracking of detection accuracies is shown in Table 11. Compared to the non-updated model, updated model performs much better and the accuracies are always beyond 97%. Therefore, it is necessary to update the model frequently.

Table 11. Accuracies tracking for the combined neural network.

Data collection date	Non-updated model	Updated model
02/19/17	93.21%	–
02/20/17	94.83%	97.24%
02/21/17	95.17%	97.24%
02/22/17	96.79%	98.57%
02/23/17	94.48%	97.59%
02/24/17	96.55%	97.24%
02/25/17	94.07%	98.15%

5 Conclusion

This paper proposes several efficient profile and behavioral features and a novel combined neural network for spammers detection in the social network. Among the massive information, our proposed features provide a decent norm to detect the spammers among legitimate users. The effectiveness of the proposed features is studied using several classical machine learning approaches. In addition, based on the correlation between different features, the combined neural network is proposed to handle the input with different types of attributes. The experiments on real world data demonstrate the efficiency and effectiveness of the proposed network which achieves 97.5% detection accuracy. Finally, we study how the combined neural network is affected by the rapidly changing internet environment.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**(Jan), 993–1022 (2003)
2. Chakraborty, A., Sundi, J., Satapathy, S., et al.: SPAM: a framework for social profile abuse monitoring. CSE508 report, Stony Brook University, Stony Brook (2012)
3. Chen, C., Wu, K., Srinivasan, V., Zhang, X.: Battling the internet water army: detection of hidden paid posters. In: 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 116–120. IEEE (2013)
4. Cheng, H.T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., et al.: Wide & deep learning for recommender systems. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, pp. 7–10. ACM (2016)
5. Cheng, Z., Kai, N., Zhiqiang, H.: Dynamic detection of spammers in Weibo. In: 2014 4th IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC), pp. 112–116. IEEE (2014)
6. Khan, A., Baharudin, B., Lee, L.H., Khan, K.: A review of machine learning algorithms for text-documents classification. J. Adv. Inf. Technol. **1**(1), 4–20 (2010)
7. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
9. Liu, Y., Wu, B., Wang, B., Li, G.: SDHM: a hybrid model for spammer detection in Weibo. In: 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 942–947. IEEE (2014)
10. Ming, L., Yunchun, L., Wei, L.: Spam filtering by stages. In: International Conference on Convergence Information Technology, pp. 2209–2213. IEEE (2007)
11. O'Donovan, J., Kang, B., Meyer, G., Hollerer, T., Adalii, S.: Credibility in context: an analysis of feature distributions in Twitter. In: 2012 International Conference on Privacy, Security, Risk and Trust (PASSAT) and 2012 International Conference on Social Computing (SocialCom), pp. 293–301. IEEE (2012)
12. Ruan, G., Tan, Y.: A three-layer back-propagation neural network for spam detection using artificial immune concentration. Soft. Comput. **14**(2), 139–150 (2010)
13. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al.: Mastering the game of go with deep neural networks and tree search. Nature **529**(7587), 484–489 (2016)
14. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
15. Wang, A.H.: Detecting spam bots in online social networking sites: a machine learning approach. In: Foresti, S., Jajodia, S. (eds.) DBSec 2010. LNCS, vol. 6166, pp. 335–342. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13739-6_25
16. Xie, Y., Tang, G., Hoff, W.: Chess piece recognition using oriented chamfer matching with a comparison to CNN. In: IEEE Winter Conference on Applications of Computer Vision (WACV) (2018)

17. Yeh, C.Y., Wu, C.H., Doong, S.H.: Effective spam classification based on meta-heuristics. In: 2005 IEEE International Conference on Systems, Man and Cybernetics, vol. 4, pp. 3872–3877. IEEE (2005)
18. Zheng, X., Wang, J., Jie, F., Li, L.: Two phase based spammer detection in Weibo. In: 2015 IEEE International Conference on Data Mining Workshop (ICDMW), pp. 932–939. IEEE (2015)