Behavior Enhanced Deep Bot Detection in Social Media

Chiyu Cai*†, Linjing Li*, *Member, IEEE*, and Daniel Zeng‡*†, *Fellow, IEEE*
* The State Key Laboratory of Management and Control for Complex Systems,
Institute of Automation, Chinese Academy of Sciences, Beijing 10090, China
† School of Computer and Control Engineering, University of Chinese Academy of Sciences, China
‡ Department of Management Information Systems, University of Arizona, Tucson, AZ 85721, USA
Email: {caichiyu2014, linjing.li, dajun.zeng}@ia.ac.cn

***Abstract*—Social bots are regarded as the most common kind of malwares in social platform. They can produce fake messages, spread rumours, and even manipulate public opinions. Recently, massive social bots are created and widely spread in social platform, they bring negative effects to public and netizen security. Bot detection aims to distinguish bots from human and it catches more and more attentions in recent years. In this paper, we propose a behavior enhanced deep model (BeDM) for bot detection. The proposed model regards user content as temporal text data instead of plain text to extract latent temporal patterns. Moreover, BeDM fuses content information and behavior information using deep learning method. To the best of our knowledge, this is the first trial that applies deep neural network in bot detection. Experiments on real world dataset collected from Twitter also demonstrate the effectiveness of our proposed model.**

I. Introduction

Social bot is a program inhabiting in social platform that automatically produces content or interacts with humans, trying to mimic and alter others' behavior [1]. Illegal users often use social bots to manipulate public opinions, spread rumours and produce fake rating or reviews. Therefore, these malicious social bots bring negative effects to public and individual security. Social bots have been known to inhabit social media platforms for a few years. According to a recent Twitter SEC filing, approximately 8.5% of all Twitter users are bots[1]. As the social media has been evolving gradually as the mainstream platform for Internet users, the issues caused by social bots are increasingly obvious.

The goal of bot detection is to recognize bots among a number of social accounts. Detecting bots can help maintain social stability, avoid network traps, and ensure safety of privacy. For all the reasons outlined above, researchers are engaged in the design of advanced methods to automatically detect social bots. Therefore, bot detection is a valuable and highly demanded research problem.

The challenge of bot detection has been pointed out by many teams in previous works. The classical method for bot detection exploits social honeypots [2]. Lee's team implemented a honeypot trap that managed to detect thousands of social bots. It shows the effectiveness of honeypot despite of its simple principle. Lee et al. [3] proposed a machine learning method through the boosting of random forests by considering four features: user demographics, friendship networks, content, and history. BoostOR [4] introduces a set of heuristics including fraction of retweets, average tweet length, fraction of URLs, and average time between tweets. Then a BoostOR algorithm is employed for bot detection based on these heuristics. Stweeler [5] took user data and tweet content as input to identify bots through its internal components, which using entropy, account properties, NLP classification, and ranking algorithms. However, complex features are manually designed in most of the existing methods, this feature engineering is labour intensive and depends on external tools and resources. This paper strives to shed some light on this problem.

In this paper, we first propose a behavior enhanced deep model (BeDM) on social user (or account of social media), then we apply it to detect bots under a deep learning framework consisted by CNN and LSTM blocks. The contribution of our work are as follows. BeDM fuses content information and behavior information which utilizes posting behavior and connecting behavior. Beyond traditional linguistic features, BeDM regards user's history tweets as temporal text data instead of plain text in existing methods which explores semantic information and latent temporal patterns using a CNN-LSTM network. It is a first step towards utilizing deep learning in bot detection, which avoid cumbersome feature engineering. We have also conducted a series of experiments on real world dataset from Twitter to validate the effectiveness of the proposed model.

II. The Proposed Method

In this section, we propose a behavior enhanced deep model (BeDM) for bot detection in social media. The overall structure of BeDM is shown in Figure 1. BeDM aims to capture the latent features by fusing content and behaivor information. The details of the proposed model are described in the following.

A. Content Features

In the following, we use u to denote a social user, all tweets posted by user u can be denoted as $\mathcal{C}_u = [S_{u1}, S_{u2}, \cdots, S_{u|\mathcal{C}_u|}]$, where $|\mathcal{C}_u|$ is the number of all history tweets. The order of a specific tweet is determined by its

[1] http://time.com/3103867/twitter-bots/

978-1-5090-6727-5/17/$31.00 ©2017 IEEE

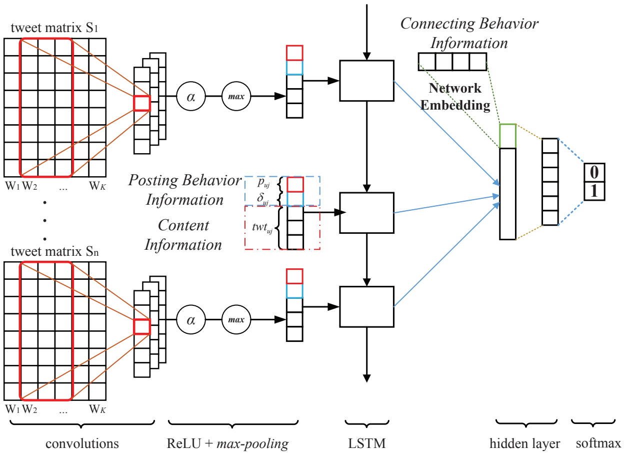

Fig. 1: The overview of Behavior enhanced Deep Model

posting time. In BeDM, we obtain the high-level representation of each tweet using the shared CNNs.

Given a pre-trained word embedding matrix $\mathbf{E} \in \mathbb{R}^{e \times |V|}$, where e is the dimension of word embedding and $|V|$ is the size of vocabulary V. Considering the j-th tweet S_{uj} formed by w words, we can convert S_{uj} as a $e \times w$ tweet matrix $\mathbf{S} \in \mathbb{R}^{e \times w}$ by replacing all words in S_{uj} with their corresponding word embeddings in matrix $\mathbf{E}$. Next, we feed the tweet matrix $\mathbf{S}$ into a convolutional layer. The convolution operation maps the input matrix $\mathbf{S}$ to a vector $\mathbf{c}_\ell \in \mathbb{R}^{w+m-1}$ by applying a specific convolutional filter. The convolutional layer is composed of s filters, these filters are convolved sequentially with matrix $\mathbf{S}$ to produce a feature map matrix $\mathbf{C} \in \mathbb{R}^{s \times (w+m-1)}$. We choose ReLU [6] as the non-linear function in BeDM which aims to speed up the training.

The output of the convolutional layer is then passed to the pooling layer. In BeDM, max-pooling is adopted to transform the resulting feature map as a single scalar. The final pooled representation is transposed as a row vector $twt_{uj} \in \mathbb{R}^{1 \times s}, j = 1, 2, 3, \cdots, |\mathcal{C}_u|$.

B. Behavior features

Beside text information, behavior information is also a significant part of each tweet. We incorporate behavior information into BeDM and use behavior information to promote the effectiveness of our proposed model.

We represent posting behavior using two features, timestamps and posting type. Timestamps are used to describe posting inter-arrival time pattern which is a significant temporal activity of users in social media. For tweet S_{uj}, we calculate the inter-arrival time δ_{uj}, corresponds to the time interval between two consecutive postings $t_j - t_{j-1}$ from the user u. Posting types are divided into original posting and retweeting, and we denote posting type of tweet S_{uj} as p_{uj} To merge content information and posting behavior, for tweet S_{uj}, we concatenates text vector, timestamp vector and posting type vector into a single vector:

$$T_{uj} = twt_{uj} \oplus \delta_{uj} \oplus p_{uj}, \tag{1}$$

T_{uj} represents the tweet S_{uj} from user u. To describe the full history tweets information for user u, we obtain a single sequence H_u, i.e.,

$$H_u = [T_{u1}, T_{u2}, \cdots, T_{u|\mathcal{C}_u|}], \tag{2}$$

where $|\mathcal{C}_u|$ denotes the number of all history tweets.

As shown in Figure 1, we feed the sequence H_u as the input sequence into a LSTM network. LSTM possesses memory cells to store history information and perform well on sequence modelling tasks. For our model, LSTM layer stores the context information in its memory cells and serves as the bridge among the tweets in full history for one social user, it provides the ability to extract high level latent features which hidden in time-series data. In this paper, we use h_u to denote the final vector computed for the tweet sequence H_u using LSTM.

We adopt network embedding to represent connecting behaviors for user u, it is an intuitive way to describe the current social network of u. In BeDM model, we employ DeepWalk [7] to convert the social network representation into a word embedding and utilize the Skip-gram model to generate network embedding. We denote the learnt network embedding as ct_u

For user u, we generate a final joint vector U_u by concatenating h_u and ct_u, i.e.,

$$U_u = h_u \oplus ct_u. \tag{3}$$

The joint vector U_u is then passed through a fully connected hidden layer adopted to capture the relations between behavior and content as follows,

$$g(x) = \alpha(w_h \cdot x + b), \tag{4}$$

where $g(\cdot)$ denotes the activation function of the hidden layer, w_h and b are the weight and bias.

Finally, output of the hidden layer is fed to the *softmax* layer which computes the probability distribution over the labels (bot or human).

C. Implementation Details

As the convolution layer in our model requires fixed-length input, all tweets are padded into maximum length len which we defined. For tweets have a longer length than len, we simply cut extra words at the end of these tweets to meet len. The word embeddings is initiated with publicly available word2vec tool and the dimension of word embedding is set as $e = 200$.

The entire model is trained by minimizing the cross-entropy error through Adadelta [8] which is an adaptive learning rate method. The number of mini-batches is set as 64 for optimization reason. The gradients are computed by back propagation algorithm and the parameters of the proposed model are trained through stochastic gradient descent algorithm.

III. Experiment

A. Dataset

We used a public dataset published in paper [4] which collected with honeypot method. This dataset provides a large number of accounts from Twitter and indicates the label (human or bot) of each account. We further collected the 1000 most recent tweets' information through Twitter API for each account recorded in the dataset. The information we crawled including content, behavior category (posting or retweeting) and timestamp. We discarded account with less than 200 tweets, and the details of dataset are summarized in Table I.

TABLE I: Summary of datasets

	#Accounts	#Tweets
Bot	2742	2,487,000
Human	2916	2,635,000

B. Baselines and Evaluation Metrics

We adopted evaluation criteria which are widely used in bot detection: **precision**, **recall**, and **F1 score**. We conducted evaluations through 10-fold cross-validation. For each split part, we trained our model with 80% data, tuned model with 10% data, and the remaining 10% data is used for testing. To validate the performance of the proposed model, we compared our model against three baselines: Boosting [3], BoostOR [4], and Stweeler [5][2]. All baselines have been introduced in previous section.

C. Performance of proposed method

In the experiments, we first conducted a series of test with different parameters of CNN, LSTM, and hidden layer. The CNN network was investigated by varying the number of filters and the filter width, set as 128, 256 and 2, 3 respectively. The memory dimension of LSTM was set as 256, and the number of hidden units in the hidden layer was set as 256. The experimental results are shown in Table II.

TABLE II: Performance of BeDM

CNN Filter		**LSTM**	**Precision**	**Recall**	**F1**
Size	**Width**	**Memory**			
128	2	256	87.02	85.21	86.11
256	2	256	87.17	86.26	86.71
128	3	256	87.69	85.56	86.61
256	3	256	88.41	86.26	**87.32**

Next, we conducted experiments with BeDM which only using content features. It can be seen from Table III that BeDM performs better than only using content information. Therefore, fusing content and behavior information could improve the performance in bot detection.

Finally, we conducted experiments to compare the performance of BeDM with all baselines and the results are shown in Table III. Comparing with the baselines, BeDM achieves the highest F1 score of 87.32%. This demonstrates that the proposed BeDM is an effective method for detecting bot.

TABLE III: Performance comparison

Methods	**Precision**	**Recall**	**F1**
Stweeler	83.38	88.23	85.74
Boosting	85.23	84.32	84.77
BoostOR	83.16	89.25	86.10
BeDM(only content)	83.49	84.73	84.11
BeDM	88.41	86.26	**87.32**

IV. Conclusion

In this paper, we proposed a behavior enhanced deep model (BeDM) for bot detection. BeDM fuses content and behavior information and learns the joint representation automatically. To our best knowledge, this work is the first trial which applies deep neural network in bot detection. We also conducted extensive experiments on a real world dataset collected from Twitter, the results showed the effectiveness and performance of our proposed method. In the future work, we plan to study the writing behavior modelling using deep neural network in social media which can also be used in user identification.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 71202169, 71272236, U1435221, 71602184, 71621002, 71672067, 61671450, the Key Research Program of the Chinese Academy of Sciences under Grant No. ZDRW-XH-2017-3, as well as the Early Career Development Award of SKLMCCS.

References

[1] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots," *arXiv preprint arXiv:1407.5225*, 2014.

[2] 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*. ACM, 2010, pp. 435–442.

[3] K. Lee, B. D. Eoff, and J. Caverlee, "Seven months with the devils: A long-term study of content polluters on twitter." in *ICWSM*, 2011.

[4] F. Morstatter, L. Wu, T. H. Nazer, K. M. Carley, and H. Liu, "A new approach to bot detection: striking the balance between precision and recall," in *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*. IEEE, 2016, pp. 533–540.

[5] Z. Gilani, L. Wang, J. Crowcroft, M. Almeida, and R. Farahbakhsh, "Stweeler: A framework for twitter bot analysis," in *Proceedings of the 25th International Conference Companion on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016, pp. 37–38.

[6] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.

[7] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 701–710.

[8] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[2] https://github.com/zafargilani/stcs