Detecting Social Bots by Jointly Modeling Deep Behavior and Content Information

Chiyu Cai[*†] Linjing Li[*] Daniel Zeng[*‡]
[*] The State Key Laboratory of Management and Control for Complex Systems,
Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[†] School of Computer and Control Engineering, University of Chinese Academy of Sciences, China
[‡] Department of Management Information Systems, The University of Arizona, Tucson, AZ 85721, USA
{caichiyu2014,linjing.li,dajun.zeng}@ia.ac.cn

ABSTRACT

Bots are regarded as the most common kind of malwares in the era of Web 2.0. In recent years, Internet has been populated by hundreds of millions of bots, especially on social media. Thus, the demand on effective and efficient bot detection algorithms is more urgent than ever. Existing works have partly satisfied this requirement by way of laborious feature engineering. In this paper, we propose a deep bot detection model aiming to learn an effective representation of social user and then detect social bots by jointly modeling social behavior and content information. The proposed model learns the representation of social behavior by encoding both endogenous and exogenous factors which affect user behavior. As to the representation of content, we regard the user content as temporal text data instead of just plain text as be treated in other existing works to extract semantic information and latent temporal patterns. To the best of our knowledge, this is the first trial that applies deep learning in modeling social users and accomplishing social bot detection. Experiments on real world dataset collected from Twitter demonstrate the effectiveness of the proposed model.

KEYWORDS

bot detection; deep learning; behavior factors; temporal content

1 INTRODUCTION

As the mainstream platform for netizens to share and express information, social media play an increasingly significant role as information source. With more and more netizens turning to social media, malwares like social bots attempt to sway public opinion or individual via leveraging the influence of social network. Social bot is a bot in social media, which is any automated account that automatically produces content and interacts with humans, trying to mimic and alter their behavior [5]. While some bots are benign, there are still many harmful bots designed with the goals of spreading spam, persuading, or deceiving. Social bots have been known to inhabit social media platforms for a few years [1]. According to a recent Twitter SEC filing, approximately 8.5% of all Twitter users are bots[1]. Therefore, social bots bring negative effects to the user security and social environment.

"Bot or Not ?" is a challenging task since bots have risen to prominence on social media. Bot detection is an area of active research in recent years. The goal of bot detection is to discover or recognize bots among a number of social accounts. Distinguishing human and bots can help users get effective information, focus on valuable social accounts, avoid network traps, and ensure their own security. Therefore, bot detection is a highly demanded and valuable research problem.

Early methods for bot detection relied on ad hoc strategies, the famous one was the honeypot trap [9]. Linguistic clues and network structure were intuitive features which often used by machine learning algorithms to distinguish bot from human accounts. Content features were extracted using natural language processing (NLP) algorithms, such as word frequency and part-of-speech (PoS) [3, 7, 10]. Network structure features of social media platform include clustering coefficient [2], centrality measures, distribution of followers and followees, community detection [14] and so on. To achieve better performances, recent works had paid more attention to incorporate more relevant information which can be extracted from social platform, such as topics [11], sentiment [4], behavior traits [6, 8]. However, complex features were manually designed in most of the existing methods, this feature engineering was labor intensive and depended on external tools and resources. This paper strives to shed some light on this problem.

In this paper, we propose a novel deep bot detection model (DBDM) which focus on learning social user representation automatically and identifying bots depend on user representation. To model social behavior, DBDM takes endogenous and exogenous factors which affect user behavior into accounts. Beyond traditional linguistic features, DBDM regards user history tweets as temporal text data instead of plain text and explores semantic information and latent temporal patterns using a CNN-LSTM network. This paper is a first step towards utilizing deep learning in modeling social media user (call "social user" for short) and performing bot detection, which avoid cumbersome feature engineering.

The rest of this paper is organized as follows. Section 2 presents the bot detection model, experiments and findings are presented in Section 3, and Section 4 is the conclusion.

CIKM'17 , November 6–10, 2017, Singapore, Singapore
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-4918-5/17/11…$15.00
https://doi.org/10.1145/3132847.3133050

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

2 DEEP BOT DETECTION MODEL

In this section, we propose the novel Deep Bot Detection Model (DBDM) aims to capture the latent features of social behavior and content information about user, then it is employed to learn user representations which used to detect bots. The DBDM mainly consists of three layers: *input*, *representation*, and *fusing*, as shown in Figure 1. The input layer receives tweets and timestamps, and converts each tweet to a tweet matrix using word embedding. The representation layer includes two components, namely the social behavior component and the temporal content component. The details of the two components are described in the following subsection. Then a fusing layer jointly generates the representation of user through incorporation of the information from both behavior and content. These three components can be jointly trained together. On top of the fusing layer, we add a fully connected hidden layer and a softmax layer to obtain the classification label (bot or human).

2.1 Social Behavior Representation

In this paper, we take two behavior types into account, posting and retweeting. We analyze user behavior in social platform from endogenous factors and exogenous factors. Endogenous factors play a major role in individuals' activities in daily life. The famous factor is the circadian rhythm. A circadian rhythm is an individual difference in personality [13], it is believed to be the cause of why some individuals prefer to work and exercise in the morning hours while others prefer evening hours. Exogenous factors are mainly cultural or environment influences. On weekends and holidays, people may spend more time on social media, thus the probability of posting new tweets will be higher. Moreover, some hot social or emergent events also have a profound impact on social users' posting behaviors.

To extract endogenous factors in both posting and retweeting, we design the input of our model as vectors that each vector contains behavior information from one day. We can explore the latent endogenous factors like circadian rhythm through feeding these input vectors into deep neural network.

We regard one day, say d, as a sequence of T timestamps $\mathcal{T} = [t_1, t_2, \cdots, t_T]$, each interval between two consecutive timestamps is one minute, thus $T = 60 \times 24 = 1440$. The posting behavior of user u in day d is serialized based on the timestamps. We calculate the number of tweets posted by user u at each timestamp t and treat it as the weight at this timestamp. Therefore, we can then map the posting behavior of user u in day d as a feature vector $p_{ud} \in \mathbb{R}^{1\times T}$. In a period of D days, posting behavior of user u can thus be coined into a feature vector sequence $\mathbf{P}_u$:

$$\mathbf{P}_u = [p_{u1}, p_{u2}, \cdots, p_{uD}]. \tag{1}$$

We deal with the retweeting behavior in a similar way and represent retweeting behavior of user u in day d by a feature vector $r_{ud} \in \mathbb{R}^{1\times T}$. Then, we can build a vector sequence for retweeting behavior indicate as $\mathbf{R}_u$:

$$\mathbf{R}_u = [r_{u1}, r_{u2}, \cdots, r_{uD}]. \tag{2}$$

To capture exogenous factors, we first need to store behavior information during a period of time and then explore behavior information in different days. Therefore, we employ LSTM to store historical information in our model. The memory cells of LSTM allow our model possess the ability of exploring the latent behavior patterns over several days for social users. We feed posting vector sequence $\mathbf{P}_u$ and retweeting vector sequence $\mathbf{R}_u$ into a 2-layer LSTM. We regard the output of the hidden state at the last time step of LSTM as the posting behavior representation $\mathbf{Pr}_u$ and retweeting behavior representation $\mathbf{Rr}_u$.

2.2 Temporal Content Representation

In this component, we regard users' history tweets as temporal text data instead of as plain text in other existing works. We build a CNN-LSTM network that learns the user content representation UC_u which not only captures semantic information but also learns temporal patterns.

The content used to describe user, say u, can be treated as a sequence of tweets $C_u = \left[S_{u1}, S_{u2}, \cdots, S_{u|C_u|}\right]$, where $|C_u|$ is the number of tweets posted by user u. In the content component, a CNN is employed to extract high-level representation of each tweet which is regarded as the input for the LSTM layer. We convert S_{uj} into a matrix $\mathbf{S} \in \mathbb{R}^{e\times w}$, where e is the dimension of word embedding and w is the length of tweet. And we feed the matrix $\mathbf{S}$ into a convolutional layer. The convolutional layer is composed of s filters $\{\mathbf{F}_\ell \in \mathbb{R}^{e\times m} | \ell = 1, 2, 3, \cdots, s\}$, where m is the width of each filter. 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 $\mathbf{F}_\ell$, the k-th element $c_{\ell k}$ of $\mathbf{c}_l$ is calculated as:

$$c_{\ell k} = (\mathbf{S} * \mathbf{F}_\ell)_k = \sum_{mw} \left(\mathbf{S}_{[:,k-m+1:k]} \odot \mathbf{F}_\ell\right)_{mw}, \tag{3}$$

where $*$ denotes convolution and $\mathbf{S}_{[:,k-m+1:k]}$ is a matrix slice of dimension m along the column-wise. We choose ReLU as the nonlinear function. The output of the convolutional layer is then passed to a max pooling layer. The final pooled representation is transposed as a row vector $twt_{uj} \in \mathbb{R}^{1\times s}, j = 1, 2, 3, \cdots, |C_u|$.

With those $|C_u|$ number of tweet representation, we then concatenate them as a single sequence SC_u, i.e.,

$$SC_u = [twt_{u1}, twt_{u2}, \cdots, twt_{u|C_u|}]. \tag{4}$$

We feed the sequence SC_u as the input sequence into the LSTM network, and obtain the user content representation $\mathbf{UC}_u$. LSTM possesses memory cells to store information of history tweets and provides the ability to extract temporal patterns of user u hidden in the time-series data.

2.3 Fusing Layer

Based on the above analysis, the user representation is highly related to three key factors: posting behavior, retweeting behavior and content information. Therefore, the fusing layer jointly models the information from the above components and the user representation can be calculated as follows:

$$\mathbf{U}_u = \mathbf{B} + (\mathbf{Pr}_u + \mathbf{V} \cdot \mathbf{Rr}_u) + \mathbf{W} \cdot \mathbf{UC}_u, \tag{5}$$

where "+" denotes element-wise addition. $\mathbf{W}$ denotes the weight for content feature. $\mathbf{Pr}_u + \mathbf{V} \cdot \mathbf{Rr}_u$ denotes the behavior information, and parameter $\mathbf{V}$ is designed to balance the influence of posting and retweeting. $\mathbf{B}$ is the bias for overall function. $\mathbf{W}$, $\mathbf{V}$ and $\mathbf{B}$ are the weight matrices that need to be learned.

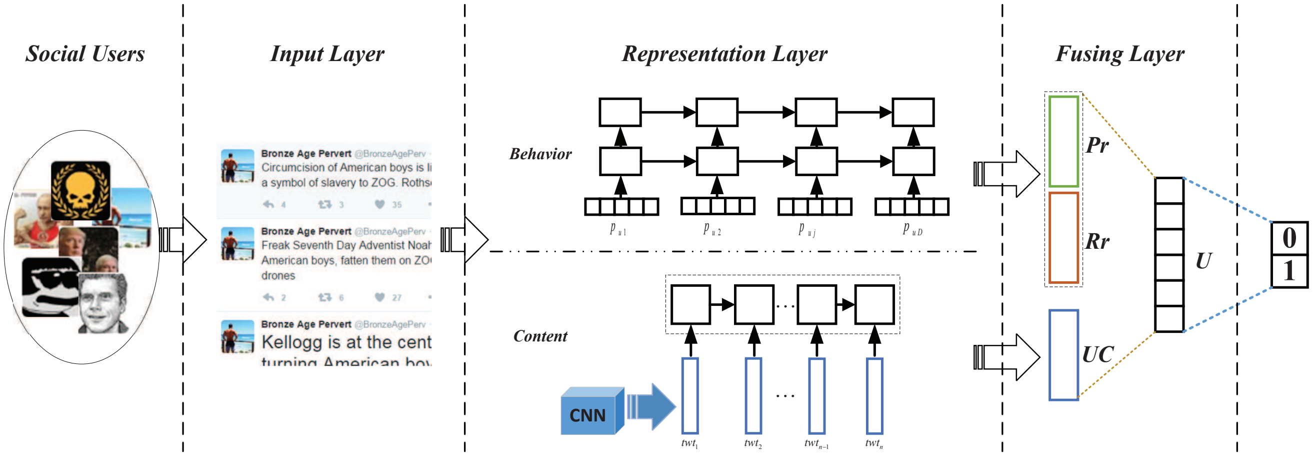
Social Users
Input Layer
Representation Layer
Fusing Layer
Bronze Age Pervert @BronzeAgePerv
Circumcision of American boys is li
a symbol of slavery to ZOG. Roths
Bronze Age Pervert @BronzeAgePerv
Freak Seventh Day Adventist Noah
American boys, fatten them on ZO
drones
Bronze Age Pervert @BronzeAgePerv
Kellogg is at the cent
turning American boy
Behavior
p_{u1}
p_{u2}
p_{uj}
p_{uD}
Content
CNN
twt_1
twt_2
twt_{n-1}
twt_n
Pr
Rr
UC
U
0
1

Figure 1: The overview of Deep Bot Detection Model

Letting $\mathbf{U}_u$ denotes the representation of user u, we formulate our bot detection model as follow:

$$\begin{aligned} P(y = l|\theta) &\propto \mathbf{O}_l \cdot g(\mathbf{U}_u) + \mathbf{b}_l, \qquad (6) \\ g(\mathbf{x}) &= ReLU(\mathbf{H} \cdot \mathbf{x} + \mathbf{h}), \end{aligned}$$

where $g(\cdot)$ denote the activations of a hidden layer, and θ is the set of parameters to be estimated during training. **H** and **h** are the weights and bias of the hidden layer. **O** and **b** are the weights and bias of the output layer.

3 EXPERIMENT

3.1 Dataset

To evaluate the performance of proposed DBDM, we use the honeypot dataset published in [12]. This dataset is a public dataset for bot detection which collected from Twitter, it provides a large number of accounts and indicates the label (bot or human) 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. Accounts with less than 200 tweets were discarded, resulting in 2742 bots and 2916 human accounts. The details of dataset are summarized in Table 1.

Table 1: Summary of datasets

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

3.2 Training Details

As the convolutional layer in our model requires fixed-length input, all tweets are padded into the maximum length which we defined. The word embeddings are initiated with publicly available word2vec tool and the dimension of word embedding is set as 200.

Table 2: Performance of different features

Methods	**Precision**	**Recall**	**F1**
Only Behavior	76.51	84.64	80.37
Only Content	88.37	82.60	85.39
RSC	79.52	77.79	78.65

The word embeddings are fine-tuned along with other model parameters during the training phase. The entire model is trained to minimize the cross-entropy error through Adadelta [15] 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.

3.3 Baselines

To assess the performance of our proposed model, we compared our model against the following four baselines:

- RSC[6][2] is a generative model aims to capture temporal activities of users in social media. It has the ability of matching four discovered patterns in the distribution of postings inter-arrival times.
- Boosting[10] method includes four type features: user demographics, friendship networks, content, and history by boosting of random forest classifier.
- BoostOR[12] 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[7][3] utilizes user data and tweet content, including username, user ID, keyword, hashtag, topic, and etc. It develops a bot analysis framework consisted of bot analyser and content analyser.

[2] http://github.com/alceufc/rsc_model

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

3.4 Evaluation Measures

To measure the performance of the proposed method and baselines, we adopted the standard metrics **precision**, **recall**, and **F1 score**. We conducted evaluation through 10-fold cross-validation. For each split part, we trained our model with 80% data, tune model with 10% data, and the remaining 10% data is used for testing. These data splits are kept fixed in all the experiments.

3.5 Results and Discussions

We first conducted a series of test using only behavior component or content component, and observed the effect of behavior information and content information respectively. We deleted the fusing layer and added the hidden layer and the classifier on top of the representation layer directly. We selected hyper-parameters of deep neural network in DBDM by cross validation. Finally, the memory dimension of all LSTM networks was set as 256. For the CNN in the content components, the number of filters and the filter width were set as 256 and 3. The number of hidden units in the hidden layer was set as 256. The experimental results are shown in Table 2.

We can observe that only using content information perform better than behavior information. This result means that traditional content feature may contain more effective information for bot detection. Meanwhile, a reason cannot be ignored is that many bots become sophisticated and mimicked human behavior patterns in recent years.

For behavior modeling, we also tested RSC model in our dataset. RSC model only uses behavior information to detect bots. Compared to RSC model, “Only Behavior” achieves a better performance. This empirical result indicates that the proposed social behavior modeling method is an effective way to model user behavior in social media.

According to above experiments, we further conducted experiments to compare the performance of DBDM with all baselines and the experimental results are shown in Table 3.

Table 3: 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
DBDM	87.58	89.04	**88.30**

It can be seen from the results in Table 2 and Table 3 that DBDM works better than only using behavior or content information. This further implies jointly modeling behavior and content information could provide a positive effect on bot detection. Comparing with all baseline algorithms, our DBDM achieves the highest F1 score of 88.30%. This demonstrates that the proposed DBDM is an effective method in bot detection.

4 CONCLUSIONS

In this paper, we proposed a novel bot detection model (DBDM) by jointly modeling the social behaviors and content information. DBDM avoids the cumbersome feature engineering and learns the joint representations automatically. Inspired by endogenous and exogenous factors in user behavior, we develop a deep model to learn social behavior representation in DBDM. Different from considering history tweets just as plain text like previous works done, DBDM regards the user content as temporal text data and captures semantic information and latent temporal patters. To our best knowledge, this work is the first trial which applies deep neural network in modeling social users and detecting social bots. The proposed model is a general model to represent Internet users. In the future, we plan to employ this model into other study, such as alias matching, author identification.

ACKNOWLEDGMENTS

This work was supported in part by the National Key R&D Program of China under Grant No. 2016QY02D0205, the National Natural Science Foundation of China under Grant Nos. 71202169, 71602184, 71621002, 61671450, U1435221, 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] Yazan Boshmaf, Ildar Muslukhov, Konstantin Beznosov, and Matei Ripeanu. 2013. Design and analysis of a social botnet. *Computer Networks* 57, 2 (2013), 556–578.

[2] Qiang Cao, Michael Sirivianos, Xiaowei Yang, and Tiago Pregueiro. 2012. Aiding the detection of fake accounts in large scale social online services. In *Presented as part of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*. 197–210.

[3] Zi Chu, Steven Gianvecchio, Haining Wang, and Sushil Jajodia. 2010. Who is tweeting on Twitter: human, bot, or cyborg?. In *Proceedings of the 26th annual computer security applications conference*. ACM, 21–30.

[4] John P Dickerson, Vadim Kagan, and VS Subrahmanian. 2014. Using sentiment to detect bots on Twitter: Are humans more opinionated than bots?. In *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*. IEEE, 620–627.

[5] Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. 2014. The rise of social bots. *arXiv preprint arXiv:1407.5225* (2014).

[6] Alceu Ferraz Costa, Yuto Yamaguchi, Agma Juci Machado Traina, Caetano Traina Jr, and Christos Faloutsos. 2015. Rsc: Mining and modeling temporal activity in social media. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 269–278.

[7] Zafar Gilani, Liang Wang, Jon Crowcroft, Mario Almeida, and Reza Farahbakhsh. 2016. 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, 37–38.

[8] Yuede Ji, Qiang Li, Yukun He, and Dong Guo. 2015. BotCatch: leveraging signature and behavior for bot detection. *Security and Communication Networks* 8, 6 (2015), 952–969.

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

[10] Kyumin Lee, Brian David Eoff, and James Caverlee. 2011. Seven Months with the Devils: A Long-Term Study of Content Polluters on Twitter.. In *ICWSM*.

[11] Juan Martinez-Romo and Lourdes Araujo. 2013. Detecting malicious tweets in trending topics using a statistical analysis of language. *Expert Systems with Applications* 40, 8 (2013), 2992–3000.

[12] Fred Morstatter, Liang Wu, Tahora H Nazer, Kathleen M Carley, and Huan Liu. 2016. 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, 533–540.

[13] Iren Tankova, Ana Adan, and Gualberto Buela-Casal. 1994. Circadian typology and individual differences. A review. *Personality and individual differences* 16, 5 (1994), 671–684.

[14] Bimal Viswanath, Ansley Post, Krishna P Gummadi, and Alan Mislove. 2010. An analysis of social network-based sybil defenses. *ACM SIGCOMM Computer Communication Review* 40, 4 (2010), 363–374.

[15] Matthew D Zeiler. 2012. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* (2012).