Social Bot Detection Based on Window Strategy

Boyu Qiao
Institute of Information Engineering,
Chinese Academy of Sciences
School of Cyber Security,
UCAS
Beijing, China
Email: qiaoboyu@iie.ac.cn

Kun Li*
Institute of Information Engineering,
Chinese Academy of Sciences
School of Cyber Security,
UCAS
Beijing, China
Email: likun@iie.ac.cn

Wei Zhou
Institute of Information Engineering
Chinese Academy of Sciences
School of Cyber Security
UCAS
Beijing, China
Email: zhouwei@iie.ac.cn

Zhou Yan*
State Key Laboratory of
Communication Content Cognition,
People's Daily Online
Beijing, China
Email: yanzhou@people.cn

Shilong Li
Institute of Information Engineering,
Chinese Academy of Sciences
School of Cyber Security,
UCAS
Beijing, China
Email: lishilong@iie.ac.cn

Songlin Hu
Institute of Information Engineering,
Chinese Academy of Sciences
School of Cyber Security,
UCAS
Beijing, China
Email: husonglin@iie.ac.cn

***Abstract*—With social bots evolving continually, the new bots post highly anthropomorphic posts to evade detection. For post content information, existing methods choose to extract features from single posts or use the rough characterization of the pre-training language model for bot detection. However, the evolution of bots has made them better at camouflage in previous posts processing techniques. According to our observation, the purpose of bot posting is to publicize different contents in different periods, and its posting often shows abnormal interest changes, while human posting is to express hobbies and daily life, and its interest changes are relatively stable. Therefore, we extract the interest changes between multiple posts published by users for bot detection. In this paper, we propose a social Bot detection model based on the Window Strategy(BotWS). Specifically, we first employ the window strategy to obtain the user posting windows, each containing multiple posts. Then, we extract the interest changes between the posting windows by multi-head attention mechanism. Finally, we embed the interest changes information into the user representation and construct a heterogeneous graph classification module to classify. We conduct experiments on three challenging datasets. Results indicate that our approach achieves state-of-the-art.**

***Index Terms*—Social bot detection, Window strategy, Multi-head attention mechanism**

I. Introduction

Popular social media platforms, such as Twitter, connect a large number of users and serve as ideal tools for communication [1]. However, there has emerged with numerous highly anthropomorphic autonomous entities are known as social bots. Social bots imitate normal users to post and spread content through social networks to influence public opinion [2].

Early bot detection conducted feature engineering and machine learning classification algorithms [3], [4]. Although preliminary results have been obtained, bots are evolving, forcing researchers to take new countermeasures. In recent years, bot detection has focused on studying text-based and graph-based detection methods. The recent methods use the deep neural network to utilize the text information, and most of them extract features from single posts for bot detection. For example, Wei et al. [5] employs the Long Short-Term Memory(LSTM) network, and DeepSBD [6] employ the Convolutional Neural Network(CNN) to extract the regularity of the content of single posts. However, some bots utilize deep fake posts or posts created by real users to evade detection. The distinction between posts from new bots and humans has become blurred. As a consequence, the single posts contain insufficient distinguishable and inconspicuous anthropomorphic features. Meanwhile, graph-based methods help the model better understand the implicit relationship between abnormal and legitimate users. These methods take the user's metadata and post content information as node embedding. For post features, almost all Graph Neural Network(GNN) methods use the pre-training language model to encode post information directly [7], [8]. This representation of post content information in the graph method is relatively rough and direct, ignoring the extraction of internal association features of multiple posts.

More adequate information in the content of user posts should be extracted for bot detection. We observe that the new bots show quite anthropomorphic characteristics in their profile and post content. However, they have not yet achieved more sufficient anthropomorphism to evade detection in terms of posting behavioral motivation [2]. In other words, the purpose of bot posting is to promote different content in different periods [9], and its posting often shows abnormal interest changes, while most real users posting is to express their hobbies or daily life, and the interest changes in human posting are relatively stable. Therefore, we extract the feature of interest changes between multiple posts to get a more posts

*Kun Li and Zhou Yan are the corresponding authors.

comprehensive representation.

In this paper, we propose a novel social Bot detection model based on the Window Strategy(BotWS). This model aims to extract single and multiple post features and combine them with the graph structure for bot classification. Specifically, our model consists of two modules, the posts representation module based on the window strategy and the heterogeneous graph classification module. In the first module, we employ the window strategy to obtain the user posting windows. Then, we obtain the internal interest changes among different posting windows through multi-head attention mechanisms. The second module is constructed to capture neighborhood user collaboration types and collaborative content. We summarize our contribution as follows:

- We are the first to propose integrating the internal interest changes of multiple posts.
- We utilize the window strategy and multi-head attention mechanism to effectively capture and characterize the interest changes between bots and humans. Moreover, we extract the attention weights to represent the characteristics of interest change and verify their effectiveness.
- Extensive experiments conducted on three challenging datasets show that our approach achieves state-of-the-art.

II. RELATED WORK

In this section, we briefly review the literature on social bot detection methods.

Feature engineering-based detection. In the early stage of bot detection, traditional feature engineering methods are adopted, and many users' features are proposed and evaluated. The feature engineering method extracts metadata features [3], [10], post content features [11], interactive behavior features [4] and other features [12], and combines machine learning classifiers to achieve bot detection. However, traditional machine learning methods that rely on feature engineering are limited in their ability to extract latent information features between users.

Text-based detection. User posts contain a wealth of information that can be used for detection. With the rapid development of neural network methods, some methods use Long Short-Term Memory (LSTM) and Convolutional Neural Networks(CNN) to mine the differences in user posts [5], [6], [13]. SATAR [14] proposes a self-supervised framework shallowly leveraging user semantic information. GANBot [15] employs generative adversarial networks to detect spam bots. These deep neural network methods focus on extracting the content features of a single post without considering the relationship between posts' sequences.

Text and Graph-based detection. Bots work coordinated and are not usually suspicious when considered individually. In recent years, the bot detection model has adopted combining text and graph neural networks to improve performance. BotRGCN [7] and RGT [8] apply the pre-training Language Model(LM) Roberta to encode user posts and construct a heterogeneous information network to detect. Moreover, RGT also extracts the influence and relationship heterogeneity between users.

III. METHODOLOGY

A. Overview Of Our Methodology

Problem statement. We treat $T = \{t_i\}_{i=1}^{K}$ to denote the user posts, K to indicate the number of posts per user, $t_i = \{w_1^i, ..., w_Q^i\}$ to denote a single post per user. Description $D = \{d_1, ..., d_L\}$ is a sentence about a personal profile, and d_i represents each word of description. User metadata features are divided into user value type features and Boolean type features, using V for value type features and B for Boolean type features. Users are connected through different relationships, using $R = \{r_i\}_{i=1}^{O}$ to denote a user connection social relationships, O indicates the number of categories of edges.

Method introduction. We propose a social bot detection model based on the window strategy. Fig. 1 shows an overview of our approach, which consists of two main components: the posts representation module based on the window strategy and the heterogeneous graph classification module. This model aims to obtain the posts' comprehensive representation and interest changes by window strategy and multi-head attention mechanism and captures the user coordination with a heterogeneous graph.

B. Posts Representation Module Based On The Window Strategy

This module first applies the pre-training Language Model(LM) to obtain the embedded representation of each post. Then, posts' context features are extracted by Bidirectional LSTM (BiLSTM). The posting windows are then obtained using the window strategy, and multi-head attention mechanisms are implemented to extract posting window features and attention weights.

Posts representation. We apply a pre-training language model to encode user posts and obtain semantic information. Specifically, we first use LM to encode token representations of post words. Then, we take the average pooling token vectors to obtain the embedded representation of each post t_i:

$$\{\hat{t}_1, ..., \hat{t}_K\} = AvgPool(LM(\{w_1^i, ..., w_Q^i\}_{i=1}^{K})), \tag{1}$$

$\{\hat{t}_1, ..., \hat{t}_K\}_{i=1}^{K}$ represents the embedding of the user posts.

Window strategy. Firstly, BiLSTM utilizes two LSTMs to learn the embedded representation of each post based on the past and future tokens of the post sequence.

$$\{\tilde{t}_1, ..., \tilde{t}_K\} = BiLSTM(\{\hat{t}_1, ..., \hat{t}_K\}), \tag{2}$$

$\{\tilde{t}_1, ..., \tilde{t}_K\}_{i=1}^{K}$ represents user posts that have been characterized by BiLSTM.

To acquire the interest changes between posting windows, **we first set a window of length n to get N windows of posts. The window containing n posts is called the posting**

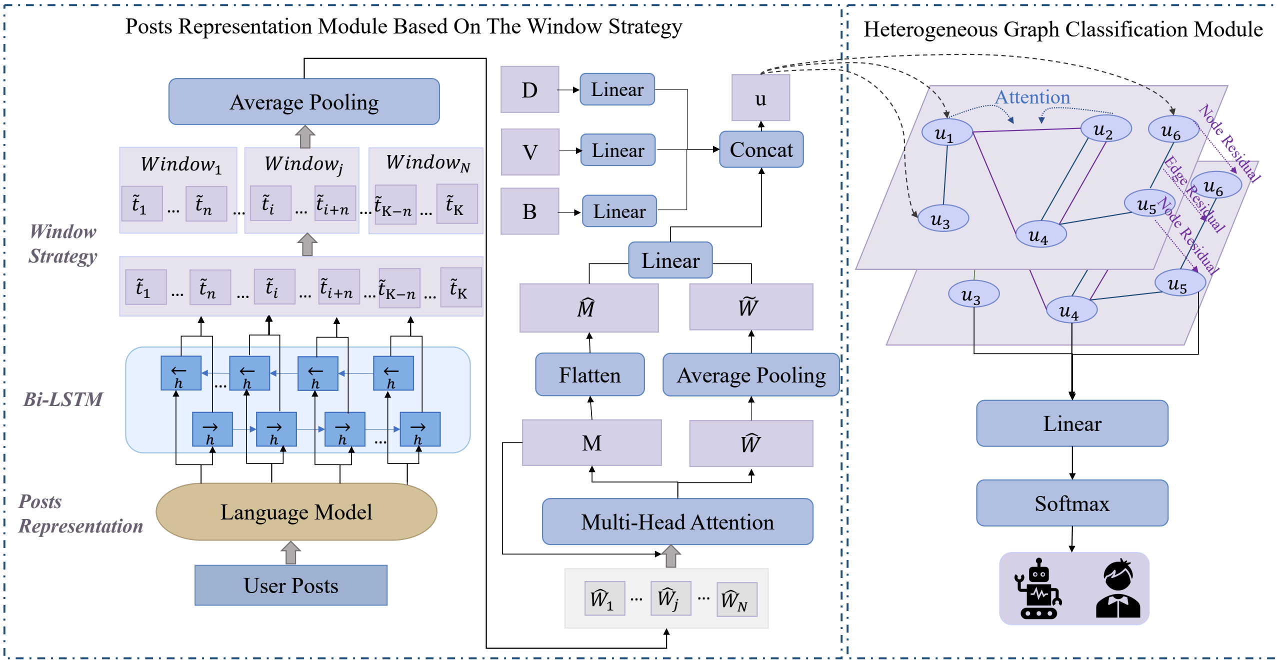


Fig. 1. The framework overview of our BotWS model.

window. Then, the representation of each posting window is obtained by average pooling:

$$\begin{aligned} &Window_j = Window(\tilde{t}_i, ..., \tilde{t}_{i+n}), \\ &\hat{W}_j = AvgPool(Window_j), j = \{1, ..., N\}, \end{aligned} \tag{3}$$

Where n is a hyperparameter, indicating how many posts of a user constitute a posting window. $Window_j$ represents the representation of the combination of n posts. $\hat{W}_j$ indicates a user posting window.

Extract the interest changes. We employ multi-head attention mechanisms to update the embedded representation of the user posting windows and extract the interest changes information.

$$\hat{W}, M = MultiAtt(\hat{W}_j), j = \{1, ..., N\}, \tag{4}$$

M denotes the attention weight matrix among posting window features. $MultiAtt$ represents the MultiHeadAttention. $\hat{W}$ denotes the posting window embedded representation after feature extraction using the multi-head attention mechanism.

We extract the attention weight matrix to represent the interest changes between the user posting windows. After that, the attention weight matrix is flattened to attention weights as a representation of the interest changes:

$$\hat{M} = Flatten(M), \tag{5}$$

Finally, the residual connection is conducted to the posting windows, and the comprehensive representation of user-owned posts is obtained by average pooling:

$$\tilde{W} = AvgPool(Residual(\hat{W})), \tag{6}$$

$\tilde{W}$ represents a comprehensive representation of user all posts.

User representation. The posts($\tilde{W}$), attention weights($\hat{M}$), descriptions(D), Values(V), and Bool features(B) of the user are integrated to be embedded as heterogeneous graph nodes:

$$\begin{aligned} &P = Concat(\tilde{W}; \hat{M}), \\ &Attr = Concat(Linear(D); Linear(V); Linear(B)), \\ &u = Concat(Linear(P); Attr), \end{aligned} \tag{7}$$

P and $Attr$ are intermediate variables that represent the user's post features and the user's description and attribute features, respectively. u denotes the output of the posts representation module based on the window strategy.

C. Heterogeneous Graph Classification Module

This module applies the user social relationship as the edge to build the edge heterogeneity graph and extracts the user coordination. Specifically, we apply Simple Heterogeneous Graph Networks(Simple-HGN) [17] to aggregate features from neighborhoods.

Heterogeneous graph construction. We treat users as nodes, and the user representations come from u in section 3.2. The user's social relationships are treated as edges. At each layer, we allocate a d_l-dimensional embedding $Re_{r_i}^{(l)}$ for each edge type $r_i \in R$.

Relationship and influence heterogeneity. Relationship and influence heterogeneity exist between user nodes [8]. The nodes in the graph structure are connected by different social relationships, which reflects the relationship heterogeneity. The degree of closeness between users is also different, and we consider attention scores to reflect the influence heterogeneity

TABLE I
STATISTICS FOR THE CRESCI-15, CRESCI-17, AND TWIBOT-20, WHERE 'EDGE' INDICATES THE FOLLOWER AND FOLLOWING RELATIONSHIPS BETWEEN USER NODES.

Dataset	*Training*	*Validation*	*Test*	*User*	*Human*	*Bot*	*Tweet*	*Edge*
Cresci-15	3,708	958	535	5,301	1,950	3,351	2,827,757	14,220
Cresci-17	10,053	2,870	1,445	14,368	3,474	10,894	6,637,616	0
Twibot-20	8,278	2,365	1,183	11,826	5,237	6,589	1,999,869	15,434

TABLE II
MAIN RESULTS. '†' REPRESENTS MODEL PERFORMANCE FROM PAPER [16]. '‡' REPRESENTS OUR REPRODUCTION RESULT. BOTWS$_{text-only}$ MEANS THAT ONLY USE TEXT INFORMATION FOR BOT DETECTION.

Model	Cresci-15		Cresci-17		Twibot-20	
	Acc(%)	*F1-score(%)*	*Acc(%)*	*F1-score(%)*	*Acc(%)*	*F1-score(%)*
Cresci et al.† [4]	37.01	1.17	33.49	22.81	47.76	13.69
Yang et al.† [3]	77.08	77.91	92.12	94.61	81.64	84.89
Wei et al.† [5]	96.10	82.70	89.30	78.40	70.23	53.61
Kugugunta et al.† [13]	75.33	75.74	88.23	91.74	59.59	47.26
SATAR† [14]	92.71	94.55	96.22	97.37	84.02	86.07
BotRGCN‡ [7]	96.52	97.30	96.59	97.69	84.62	87.07
RGT‡ [8]	97.15	96.38	96.62	97.72	85.93	87.98
BotWS$_{text-only}$	97.82	98.31	96.66	97.76	77.64	78.32
BotWS	**98.33**	**98.67**	**97.29**	**98.19**	**87.11**	**88.81**

between the user nodes. We compute the attention score using edge embedding and node embedding:

$$\hat{\alpha}_{ij} = \frac{exp(\sigma(W_n([Wu_i][Wu_j][W_r Re_{r_{i,j}}])))}{\sum_{k\in N_i} exp(\sigma(W_n([Wu_i][Wu_k][W_r Re_{r_{i,k}}])))}, \quad (8)$$

$Re_{r_{i,j}}$ denotes the connection relationship between nodes i,j. W_n, W, W_r are trainable parameters. u_i denotes the central user node and u_j, u_k denotes the neighbor node of the user node. We omit the superscript (l) in this equation for the sake of brevity.

Residual connection. Realformer [18] reveals that the residual connection of the attention weight is also beneficial for model prediction. Therefore, we add the residual connection to the attention score:

$$\alpha_{ij}^{(l)} = \hat{\alpha_{ij}}^{(l)}(1-\beta) + \alpha_{ij}^{(l-1)}\beta, \quad (9)$$

Where the hyperparameter $\beta \in [0,1]$is the scaling factor.

We apply the attention score to the neighbor node and make a residual connection to the central node to obtain an updated representation of the central node:

$$\hat{u}_i^{(l)} = \sigma(\textstyle\sum_{j\in N_i} \alpha_{ij}^{(l)} W^{(l-1)} u_j + W_e^{(l-1)} u_i). \quad (10)$$

$\hat{u}_i^{(l)}$ denotes the updated user node representation in layer l and $W_e^{(l)}$ is the trainable parameter.

Classification module. We apply a softmax layer to conduct bot detection based on user nodes representation. Then, we train our model using a cross-entropy loss.

IV. EXPERIMENT

This section will introduce details about experiments and the corresponding analysis.

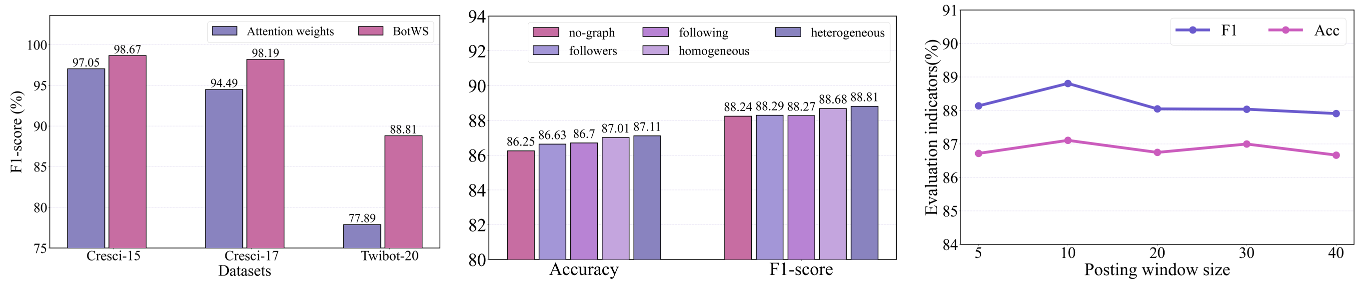
A. Experimental Settings

Dataset and evaluation metric. We compared our model with previous models on the Cresci-15 [19], Cresci-17 [20] and Twibot-20 [21]. Cresci-15 and Twibot-20 are standard Twitter bot detection datasets that support the construction of graph structures. Cresci-17 is also a common bot detection dataset but does not support graph structure. The Cresci-15, Cresci-17, and Twibot-20 datasets all contain user posts. Although the Cresci-17 does not contain edge data, using this dataset can verify the effectiveness of the "Posts Representation Module Based On The Window Strategy" module in our method. The statistics of the dataset are shown in Table I. In our study, we employ accuracy (Acc) and F1-score as our evaluation metrics, which are following the previous studies [7], [8].

Hyperparameters and implementation details. Our model is implemented in PyTorch, and the network weights are optimized by Adam. We use Roberta as the Pre-training language model and the follower and following relationships to construct the graph structure. We set the maximum length of the posts' sentences to 256. The number of heads of the multi-head attention mechanism is set to 4. Dropout is applied to the posts representation module based on the window strategy with a value of 0.1. The batch size is set to 256, the learning rate is 1e-4, and we use cosine annealing to restart the learning rate. In this experiment, we set the hyperparameter n to 10. We train models with up to 100 epochs using Tesla V100 and select the best-performing model on the validation set to output results on the test set.

B. Experiment Results

Overall model analysis. We conduct our approach on the three datasets and compare it with recent techniques of bot detection in Table II. Yang et al. [3] perform bot detection based on user metadata feature. Wei et al. [5] is based on the



(a) Attention weights comparison (b) Relationships Comparison (c) Comparison of posting window size

Fig. 2. Ablation study of BotWS.

TABLE III
F1-SCORE OF ABLATION EXPERIMENT IN POSTS REPRESENTATION MODULE BASED ON THE WINDOW STRATEGY. 'W/O ATTENTION' REPRESENTS REMOVING ATTENTION WEIGHTS, CORRESPONDING TO $\hat{M}$ IN FIG. 1. 'W/O POSTS' REPRESENTS REMOVING POSTS REPRESENTATION, CORRESPONDING TO $\widetilde{W}$ IN FIGURE 1. 'W/O POSTS-ATT' REPRESENTS REMOVING POSTS REPRESENTATION AND ATTENTION WEIGHTS.

Strategy	**Cresci-15** ***F1-score(%)***	**Cresci-17** ***F1-score(%)***	**Twibot-20** ***F1-score(%)***
w/o attention	96.05	97.61	88.28
w/o posts	98.24	97.11	88.67
w/o posts-att	96.19	96.21	86.85
BotWS	**98.67**	**98.19**	**88.81**

characteristics of user posts. Cresci et al. [4], Kugugunta et al. [13] and SATAR [14] are based on user posts and metadata features. BotRGCN [7] and RGT [8] are based on graph structure. Since BotRGCN and RGT do not conduct experiments on the Cresci-17, we employ the same hyperparameters as our approach to reproduce these two models. The performance of the two models on the Twibot-20 is slightly different from that given in the original paper, which we argue may be caused by different experimental environments. Thus, the results of the reproduction are filled in the table II. Since the Cresci-17 lacks edge data, we do not build a graph structure for this particular dataset. Our experiments on the Cresci-17 dataset were conducted up until Section 3.2 of our paper, where we obtained a comprehensive embedding representation of the user and performed a bot/human classification task utilizing cross-entropy loss to train the model.

It is worth mentioning that the F1-score of our model outperformed the RGT method by 2.29%, 0.47%, and 0.83% in the three datasets. In addition, we conduct experiments with $BotWS_{text-only}$. $BotWS_{text-only}$ represents all of the description (D), values (V), and bool features (B), and the heterogeneous graph classification module are removed from BotWS. When using only text information, our BotWS$_{text-only}$ performance is better than Wei et al. The accuracy of text feature extraction is 1.05%, 7.36%, and 7.21% higher, respectively, which proves the effectiveness of our feature extraction strategy of posts. Although the two SOTA models, BotRGCN and RGT, achieved competitive results on the datasets, these two models characterize user posts merely using a language model without considering the influence of relationships between multiple posts. In contrast, our model utilizes more comprehensive user information and performs more efficiently.

C. Ablation study

Ablation study of posts representation module based on the window strategy. To validate the impact of each component of our proposed model, we conducted further experiments on the three datasets. The experimental results in Table III demonstrate that all metrics are lost when different components of this module are removed. According to these findings, posts' comprehensive representation and attention weights are crucial for our approach. Removing the 'posts-att' part of the model has the most significant performance loss, which shows the necessity of the posts representation module based on the window strategy.

Effectiveness of attention weights representation. We conduct experiments on three datasets containing post information to verify attention weights' effectiveness and compare them with the BotWS, as shown in Figure 2(a). Figure 2(a) demonstrates that despite employing simple attention weights for model training and prediction, we achieve competitive performance to BotWS on Cresci-15 and Cresci-17. Of course, we discover comparatively substantial performance gaps on the Twibot-20, possibly because the bots in Twibot-20 are more anthropomorphic. Thus, more useful attributes must be introduced in addition to the attention weights features to detect the new bots.

Ablation study of heterogeneous graph classification module. For the Twibot-20, we validate the impact of different connectivity relationships on the final performance of the model. Figure 2(b) shows that the effect of using the heterogeneous structure is the best, but we discover that the performance gap between different connection relationships is relatively small. This result may be related to the Twibot-20, so we analyze the dataset. Table IV illustrates that the number of followers and following relationships on Twibot-20 is sparse. The average central node is connected to 1.3 neighbor nodes, and some are not connected to edges. Compared with 3.8 neighbor nodes per user in the Cresci-15, the connectivity of the Twibot-20 is relatively sparse. Therefore, the sparse

TABLE IV
SOCIAL RELATIONSHIP ANALYSIS OF THE TWIBOT-20.

Dataset	all nodes	all edges	following edges	following nodes	followers edges	followers nodes	no edges
Twibot-20	11,826	15,434	11,318	8,111	4,116	4,456	1,609

connection of this dataset may lead to fewer performance differences using different relationships [2] and also proves that the performance improvement of BotWS is mainly due to our representation method of posts.

Comparison of posting window size. At the theoretical level, the window size is a key factor in capturing user posts' interest changes over time. Since Twibot-20 is currently the most challenging dataset [21], We set the size of the posting window n to $\{5, 10, 20, 30, 40\}$ and conducted experiments. The results in Figure 2(c) show that when posting window size $n = 10$, the model achieves the best performance, and then as n increases, the performance decreases. Therefore, we take the window length $n = 10$ as the hyperparameter of BotWS.

V. CONCLUSION

In this paper, we propose the bot detection model based on the window strategy, which obtains the comprehensive representation of posts and interest changes through the window strategy and multi-head attention mechanisms. We construct a heterogeneous graph using user social relationships and aggregate the feature from their neighborhood. To demonstrate the efficacy of our proposal, we have worked on three datasets. Results show that our approach achieves state-of-the-art. In addition, we have verified the effectiveness of attention-weight features for social bot detection. This phenomenon proves that detecting the interest changes between posting windows may prevent the bot from escaping. Finally, a specific direction for future work is to explore more effective interest change detection algorithms between posting windows.

ACKNOWLEDGMENT

This paper was supported by the Key Technology Project for the Whole-Chain Governance of Disinformation for Public Opinion Guidance and Intervention Project (E250341114).

REFERENCES

[1] Onur Varol, Clayton A Davis, Filippo Menczer, and Alessandro Flammini, "Feature engineering for social bot detection," in *Feature engineering for machine learning and data analytics*, pp. 311–334. CRC Press, 2018.

[2] Stefano Cresci, "A decade of social bot detection," *Communications of the ACM*, vol. 63, no. 10, pp. 72–83, 2020.

[3] Kai-Cheng Yang, Onur Varol, Pik-Mai Hui, and Filippo Menczer, "Scalable and generalizable social bot detection through data selection," in *Proceedings of the AAAI conference on artificial intelligence*, 2020, vol. 34, pp. 1096–1103.

[4] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi, "Dna-inspired online behavioral modeling and its application to spambot detection," *IEEE Intelligent Systems*, vol. 31, no. 5, pp. 58–64, 2016.

[5] Feng Wei and Uyen Trang Nguyen, "Twitter bot detection using bidirectional long short-term memory neural networks and word embeddings," in *2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*. IEEE, 2019, pp. 101–109.

[6] Mohd Fazil, Amit Kumar Sah, and Muhammad Abulaish, "Deepsbd: a deep neural network model with attention mechanism for socialbot detection," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 4211–4223, 2021.

[7] Shangbin Feng, Herun Wan, Ningnan Wang, and Minnan Luo, "Botrgcn: Twitter bot detection with relational graph convolutional networks," in *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2021, pp. 236–239.

[8] Shangbin Feng, Zhaoxuan Tan, Rui Li, and Minnan Luo, "Heterogeneity-aware twitter bot detection with relational graph transformers," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, vol. 36, pp. 3977–3985.

[9] Ashkan Dehghan, Kinga Siuta, Agata Skorupka, Akshat Dubey, Andrei Betlen, David Miller, Wei Xu, Bogumil Kaminski, and Pawel Pralat, "Detecting bots in social-networks using node and structural embeddings," 2022.

[10] Sweety Kumari, Firdos Khan, Shekh Sultan, and Ruchita Khandge, "Real-time detection of traffic from twitter stream analysis," *Int. Res. J. Eng. Technol.(IRJET)*, vol. 3, no. 04, pp. 2395–0056, 2016.

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

[12] Jorge Rodríguez-Ruiz, Javier Israel Mata-Sánchez, Raul Monroy, Octavio Loyola-González, and Armando López-Cuevas, "A one-class classification approach for bot detection on twitter," *Computers & Security*, vol. 91, pp. 101715, 2020.

[13] Sneha Kudugunta and Emilio Ferrara, "Deep neural networks for bot detection," *Information Sciences*, vol. 467, pp. 312–322, 2018.

[14] Shangbin Feng, Herun Wan, Ningnan Wang, Jundong Li, and Minnan Luo, "Satar: A self-supervised approach to twitter account representation learning and its application in bot detection," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 3808–3817.

[15] Shaghayegh Najari, Mostafa Salehi, and Reza Farahbakhsh, "Ganbot: a gan-based framework for social bot detection," *Social Network Analysis and Mining*, vol. 12, no. 1, pp. 1–11, 2022.

[16] Shangbin Feng, Zhaoxuan Tan, Herun Wan, Ningnan Wang, Zilong Chen, Binchi Zhang, Qinghua Zheng, Wenqian Zhang, Zhenyu Lei, Shujie Yang, et al., "Twibot-22: Towards graph-based twitter bot detection," *arXiv preprint arXiv:2206.04564*, 2022.

[17] Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, and Jie Tang, "Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 1150–1160.

[18] Ruining He, Anirudh Ravula, Bhargav Kanagal, and Joshua Ainslie, "Realformer: Transformer likes residual attention," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 929–943.

[19] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi, "Fame for sale: Efficient detection of fake twitter followers," *Decision Support Systems*, vol. 80, pp. 56–71, 2015.

[20] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi, "The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race," in *Proceedings of the 26th international conference on world wide web companion*, 2017, pp. 963–972.

[21] Shangbin Feng, Herun Wan, Ningnan Wang, Jundong Li, and Minnan Luo, "Twibot-20: A comprehensive twitter bot detection benchmark," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 4485–4494.