Domain-Aware Federated Social Bot Detection with Multi-Relational Graph Neural Networks

Huailiang Peng[1,2,3], Yujun Zhang[1,2] ✉, Hao Sun[3], Xu Bai[3], Yangyang Li[4] and Shuhai Wang[5]
[1]Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China
[3]Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[4]National Engineering Research Center for Risk Perception and Prevention, CAEIT, Beijing 100041, China
[5]Shijiazhuang Tiedao University, Shijiazhuang 050043, China
Email: {penghuailiang19b, zhmj}@ict.ac.cn, sunhao0308@gmail.com, baixu@iie.ac.cn,
liyangyang@cetc.com.cn, wsh36302@126.com

2022 International Joint Conference on Neural Networks (IJCNN) | 978-1-7281-8671-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/IJCNN55064.2022.9892366

***Abstract*—Social networks have been the widespread popular tools for communication and socialization, and it also been the ideal platform for bots to publish malicious information. Therefore, social bot detection is essential for the social network's security. Existing methods almost ignore the differences in bot behaviors in multiple domains. Thus, we first propose a Domain-Aware detection method with Multi-Relational Graph neural networks (DA-MRG) to improve detection performance. Specifically, DA-MRG constructs multi-relational graphs with users' features and relationships, obtains the user presentations with graph embedding and distinguishes bots from humans with domain-aware classifiers. Meanwhile, considering the similarity between bot behaviors in different social networks, we believe that sharing data among them could boost detection performance. However, the data privacy of users needs to be strictly protected. To overcome the problem, we implement a study of federated learning framework for DA-MRG to achieve data sharing between different social networks and protect data privacy simultaneously. We conduct extensive experiments on TwiBot-20, and the results demonstrate that the proposed method can effectively achieve federated social bot detection.**

***Index Terms*—social bot detection, federated learning, social network**

I. Introduction

Nowadays, social networks have become the necessary platforms for people to communicate and socialize, which is also essential for publishing information. Due to its widespread popularity and its open nature, it is the ideal platform for bots to achieve malicious goals. These bot accounts spread fake news and promote extreme ideology [1], and they also try to imitate the behaviors of normal users for hiding themselves. Therefore, effective bot detection methods are desperately needed for social networks.

Recent works usually extract the features from the user profile and adopt neural networks, such as RNN and GNN, to obtain user presentations, which would be fitted to a binary classifier (human vs. bot). According to our observation, the bots in different domains always have diverse characteristics. As shown in Fig. 1, the real-world sample bot in the domain of politics focuses on political topics habitually, and its neighbors

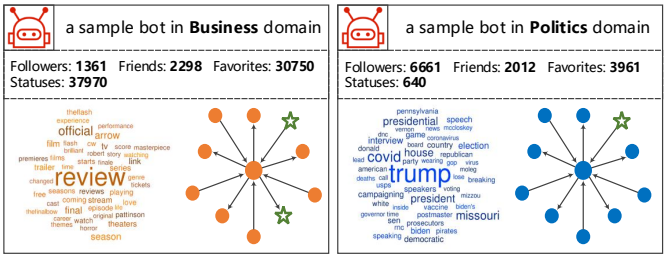

Fig. 1. The real-world bot samples in different domains. Orange dots indicate the users in **business** domain, blue dots indicate users in **politics** domain and green stars indicate users who are in multi-domains.

are in the same domain. In contrast, the sample bot in the business domain is concerned with everyday topics and interacts with business users. Thus, we believe the diversity of bots in different domains can assist us in distinguishing between humans and bots, which is seldom considered in existing methods.

Apart from that, because of the high costs for labeling bots, sharing data among multiple social networks would promote the accuracy of the detection method with the fact that domain-specific bots in different social networks always behave similarly. However, due to the requirement of data privacy protection, it is scarcely possible to implement centralized training by aggregating a large of data from all social networks directly. Therefore, it is necessary to improve the ability of the bot detection methods to exchange information among multiple social networks and protect data privacy simultaneously. As an effective method, Federated learning has been proposed for handling the problem of data silos resulting from data privacy [2] and has gained a noteworthy achievement. Nevertheless, it is not introduced into bot detection in the social network yet, as we know. Therefore, our work focuses on handling federated social bot detection.

To solve the above problems, we propose a domain-aware federated social bot detection method. Firstly, we obtain the user presentation based on multi-relational graph neural

✉ Corresponding author.

978-1-7281-8671-9/22/$31.00 ©2022 IEEE

networks and design domain-aware classifiers to promote detection performance according to our observation in Fig. 1. Secondly, we adopt a federated learning framework to balance data sharing among social networks and data privacy protection. We evaluate the proposed model with a real-world dataset, and the results indicate that our model outperforms comparable baseline models. Significantly, the detection performance of the model in the federated learning framework can achieve almost the same level with centralized training. The main contributions of our work are as follows:

- To the best of our knowledge, we are the first to propose federated bot detection in social networks, which introduces federated learning into social bot detection for sharing data across different social networks while protecting data privacy.
- We construct multi-relational graphs and fuse the domain-aware knowledge of accounts to boost the detection performance. Furthermore, a federated learning framework is adopted with our model to overcome the challenge of data privacy protection.
- Extensive experiments are conducted on TwiBot-20, and the results demonstrate that the proposed method can effectively improve the accuracy of social bot detection. Our model also achieves comparable performance with centralized training and other parallel training in the federated learning framework.

The rest of this paper is organized as follows. The Sec. II review related work for social bot detection and federated learning. The framework and components of our model are described in Sec. III. The dataset, experimental settings and results are shown in Sec. IV. Finally, we summarize this paper in Sec. V.

II. Related Work

In this section, we briefly review the related research works of social bot detection, graph neural network and federated learning.

A. Social Bot Detection

Early bot detection methods extract features from user profiles [3]–[5], tweets [6], etc. and apply the traditional classifiers, such as Random Forest models [7], to detect bots in social networks. In addition, Gao *et al.* [8] identify six features and adopt incremental clustering to distinguish spam campaigns in real-time. Thomas *et al.* [9] design a real-time system to determine whether the URLs in web services are direct to spam content. All of those can be considered feature engineering methods.

Because of the outstanding achievements of deep learning, increasing detection models based on neural networks are proposed. Kudugunta *et al.* [10] utilizes the LSTM architecture to detect bots and considers both tweet context features and contextual features from user metadata. Similarly, Wei *et al.* [11] adopts the BiLSTM with word embeddings to capture features across tweets and distinguish bots from human accounts. SpamGAN [12] is a generative adversarial network that relies on a limited set of labeled data and unlabeled data for opinion spam detection. SATAR [13], a self-supervised representation learning framework for bot detection, jointly leveraging the semantics, property and neighborhood information of the specific user. Recently, researchers have proposed detection methods via the graph structure with multiple relations [14], [15]. Ali *et al.* [16] first attempts to apply graph convolutional networks in bot detection. Feng *et al*. [14] constructs a heterogeneous information network and utilizes relational GNNs to obtain user presentations for bot detection. However, these methods almost ignore the domain information of user, which is important for the task in our observation.

B. Graph Neural Networks

Graph Neural Network (GNN) aims to generate the nodes' representations in low-dimensional space based on a message-passing mechanism. The classical methods, such as GCN [17], GraphSAGE [18], and GAT [19], generate the nodes' representations by aggregating their neighbors' features. Particularly, GraphSAGE samples neighbors to fit the large graph, and GAT adopts the attention mechanism to measure the importance of neighbors. Simultaneously, some works [20]–[23] believe that the local structural features of a graph are valuable to preserve the spatial information in nodes' representations.

Considering varied types of nodes and relationships in graphs, the GNN frameworks of the heterogeneous information network (HIN) and the multi-relational graph have extensively studied and achieved significant success in diverse applications, such as community detection and anomaly detection. LUCE [24] applies HIN to model the multiple relationships between house entities for the property price prediction task, FRAUDRE [25] takes a multi-relation graph as the input to predict the label of fraudsters and normal users, and FinEvent [26] adopts a weighted multi-relational graph neural network for social events detection. Based on the expositions that the multi-relational graph can more explicitly differentiate relation types [27], [28], we construct multi-relational graphs in our model for social bot detection.

C. Federated Learning

For the rising issues of data privacy, Google first proposes the concept of federated learning to build machine learning models based on data sets from multiple devices while achieving data privacy protection [29], [30]. Furthermore, Yang *et al.* [31] summary to the comprehensive federated-learning framework to horizontal federated learning, vertical federated learning and federated transfer learning. With the advance of federated learning, some practical algorithms have been proposed by researchers, such as FedAvg, FedAMP, FedProx, FedNova and FedMV, and have been employed in various areas [32]–[35]. FedAvg [36] is the fundamental and well-studied algorithm of federated learning, where the global model is learned by averaging the parameters of local models trained on private client datasets. FedProx [37] uses a proximal term to generalize and re-parameterize FedAvg. FedAMP [38] proposes to utilize federated attentive message passing to

facilitate pairwise collaborations among clients with similar data. FedNova [39] is a normalized averaging method that eliminates objective inconsistency while preserving fast error convergence. FedMV [40] proposes a federated framework for the multi-view data, where participants have different types of local data availability. Due to the efficiency of FedAvg, in this paper, we mainly follow the idea of FedAvg to implement the federated learning framework for bot detection.

III. Methodology

A. Overview

In this section, we propose a framework named **DA-MRG** for bot detection. We first construct multi-relational graphs from the initial user features and the origin graph for the task. And then, a user representation learning module, consisting of a series of graph embedding layers and semantic attention layers, is designed to obtain the representation for each user. Finally, we propose domain-aware classifiers to discriminate bots from humans. The overall architecture of our methods is shown in Fig. 2. Furthermore, we introduce a federated learning framework for DA-MRG to implement the joint training among multiple participants.

B. Multi-Relational Graph Generator

In a social network, We can obtain a multi-relational graph $G = \{V, X, E, Y\}$ based on the interactive behaviors between users, where $V = \{v_1, v_2, ..., v_n\}$ is the set of user nodes, X is the initial features of all user nodes, and n is the number of users. $e^r_{i,j} \in E$ is an edge between v_i and v_j with a relation $r \in \{1, ..., R\}$, indicating an interactive behavior between user i and user j. Such as user i follows user j or user i comments user j. Y is the set of labels for all users.

Then, We present a multi-relational graph generator to generate the relational graph G_r for the origin graph G. The generator consists of edge separating and feature learning.

Edge Separating. We first generate all relational graphs $\{G_r\}|^R_{r=1}$ by reserving the relation r between users in the whole graph. Thus, the set of edges in G_r is E_r. We add the two nodes of each edge $e^r_{i,j}$ in E_r to the nodes set V_r, as with nodes' features and labels. The relational graph G_r can be denoted as

$$G_r = \{V_r, X_r, E_r, Y_r\}. \tag{1}$$

Feature Learning. Following the assumption that the features of the same user have a different effect on different relational graphs, we learn the features for each relational graph independently:

$$\hat{X}_r = \sigma(W_r \cdot X_r + b_r), \tag{2}$$

where X_r is the initial features of nodes in G_r, and $\sigma(\cdot)$ is non-linearity.

C. User Representation Learning Module

In this module, we obtain the final high-level representation for each user by a series of graph embedding layers and semantic attention layers. Particularly, we first gain the representations for each node in all relational graphs via multiple GNN-based graph embedding layers. Then we aggregate the representations of each node for final embedding based on the semantic attention networks.

Multi-Relational Graph Embedding Layer. We first construct a GNN-based graph embedding layer to obtain the representation for a specific node in each relational graph G_r in this module, which is shown as below:

$$z^{(l)}_r(N^i_r) = mean(\{z^{(l-1)}_r(v_j), \forall v_j \in \{v_i\} \cup N^i_r\}), \tag{3}$$

where N^i_r is the set of v_i's 1-hop neighborhood in G_r, and $z^{(l-1)}_r(v_j)$ is the representation of v_j in the $(l-1)$-th layer of GNNs. And we use the node features $\hat{X}_r$ as the initial representation in the 0-th layer. Then, we gain v_i's presentations in the l-th layer of GNNs as follows:

$$z^{(l)}_r(v_i) = \sigma(W^{(l)}_r \cdot z^{(l)}_r(N^i_r) + b^{(l)}_r), \tag{4}$$

where $W^{(l)}_r$ and $b^{(l)}_r$) are learnable parameters. And $z_r(v_i)$ is used as the final representation for v_i in the GNN-based embedding layer.

Semantic Attention Layer. Each user node's representations in multiple relational graphs are gained via multiple GNN-based embedding layers. Considering the diverse importance of relations, we adopt a semantic attention layer to fuse all representations of each user node.

Firstly, we introduce a relational preference vector $a_r \in \mathbb{R}^{\mathcal{R}*d^{'}}$ for the relation r. For v_i's representation $z_r(v_i)$ in the specific relation r, the weight assigned to $z_r(v_i)$ for its contribution depends on the similarity between a_r and $z_r(v_i)$. To obtain the weight, we first transform d-dimension $z_r(v_i)$ into $d^{'}$-dimension $h_r(v_i)$:

$$h_r(v_i) = \sigma(W_r \cdot z_k(v_i) + b_r), \tag{5}$$

where $\sigma(\cdot)$ is non-linearity and we use $tanh$ in the paper. Then, we calculate the similarity between a_r and $h_r(v_i)$ as follows:

$$w_r(v_i) = \frac{a^T_r \cdot h_r(v_i)}{\|a_r\| \cdot \|h_r(v_i)\|}, \tag{6}$$

where $\|\cdot\|$ is the $L2$ normalization of vectors. The weight assigned to relation r for node v_i is normalized with $softmax$ as follows:

$$\alpha_r(v_i) = \frac{exp(w_r(v_i))}{\sum_{r' \in \mathcal{R}} exp(w_{r'}(v_i))}. \tag{7}$$

Finally, we fuse node v_i's representations in all relations:

$$z(v_i) = \sum_{r \in \mathcal{R}} \alpha_r(v_i) \cdot z_r(v_i). \tag{8}$$

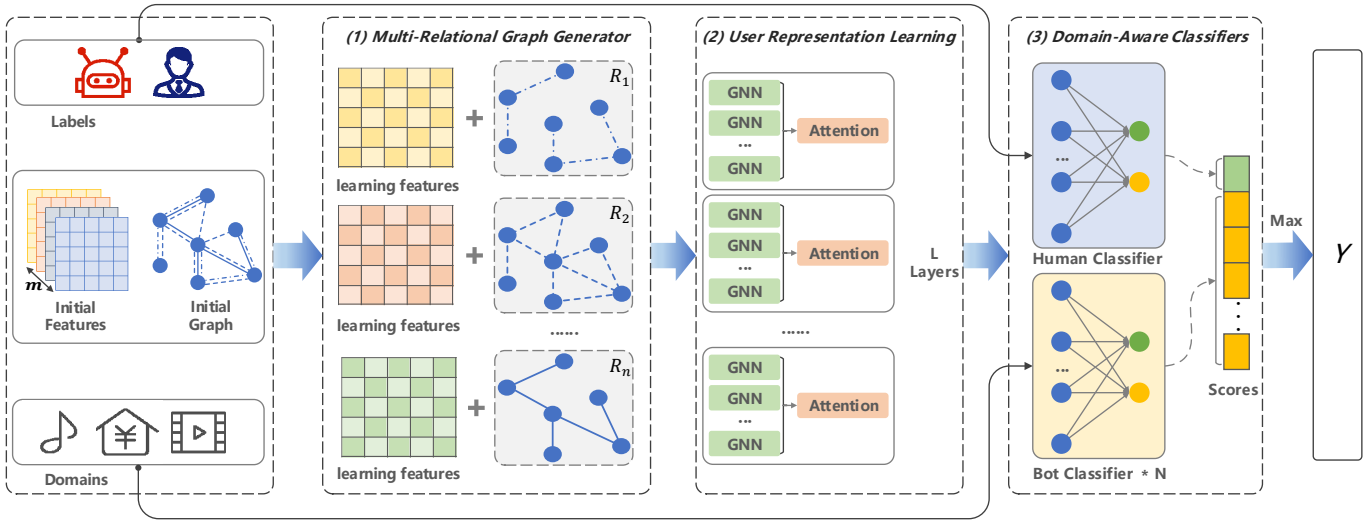

Fig. 2. Overview of DA-MRG. Given the initial graph and features as input, DA-MRG consists of the three steps: (1) Multi-Relational Graph Generator learns features and separates edges for each relation; (2) User Representation Learning module learns users' high-level representations based on a series of GNN-based embedding layers and semantic attention layers; (3) Domain-Aware Classifiers feds the representations and distinguishes bots from humans.

D. Domain-Aware Classifiers

After the user representation learning module, we obtain the final high-level representation z_v for each user node v. Generally, existing methods consider the detection task as a binary classification and the node's representations are fed into a multiple fully connected neural network to gain the prediction:

$$\hat{y}_v = \sigma(MLP(z_v)), \tag{9}$$

where $\sigma(\cdot)$ denotes the activation function and $\hat{y}_v$ is the predicted label of node v. Furthermore, Cross-Entropy is applied as the optimizer. Inspired by the observation that social bots in different domains have obvious differences, we propose the domain-aware classifiers to promote the detection performance. Specifically, we first train a bot classifier for each domain $d \in D_d|_{d=1}^{M}$:

$$P_d(v) = softmax(W_d \cdot z_v + b_d), \tag{10}$$

where $P_d(v)$ denotes v's bot probability in domain d. And then, we acquire the bot probability for v as follows:

$$P_b(v) = Max(\{P_i(v)\}|_{i=0}^{M}), \tag{11}$$

where M is the number of domains. Similarly, we train the human classifier:

$$P_h(v) = softmax(W_h \cdot z_v + b_h), \tag{12}$$

where $P_h(v)$ denotes v's human probability. Thus, we obtain the predicted label as $\hat{y} = argmax([P_h, P_b])$ and determine the final predicted bot probability as:

$$prob = \begin{cases} 1 - P_h, & \hat{y} = 0 \\ P_b, & \hat{y} = 1 \end{cases} \tag{13}$$

E. Federated learning

Due to the data privacy issues, we cannot collect data from multiple Social Network Services (SNSs) for centralized model training. Thus, we introduce a federated learning framework to address the problem. Each SNS, participating in the model training, downloads the global model from the server, trains with its own data, and uploads the trained model to the server, which aggregates models from all participants.

Specifically, suppose that K SNSs are contributing the federated learning in each round, the k-th participant calculates the local gradient of the model in round t according to Eq. 14.

$$g_k = \triangle F_k(\omega_t), \tag{14}$$

where ω_t is the global parameters download from the server in the t-th round and each participant updates its own parameters locally as follows:

$$\forall k, \omega_{t+1}^{(k)} \leftarrow \omega_t - \eta g_k. \tag{15}$$

Then, the server aggregates local parameters uploaded from all participants as Eq. 16:

$$\omega_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} \omega_{t+1}^{(k)}, \tag{16}$$

where n_k is the data size of k-th participant, and ω_{t+1} is distributed to each participant in the $(t+1)$-th round. The details are shown in Fig. 3.

In our work, the federated learning framework combined with the proposed model DA-MRG to implement the social bot detection across multiple social networks. We focus on exploring the influence of the number of participants and the amount of data in each participant. The overall process of our method is shown in Algorithm 1.

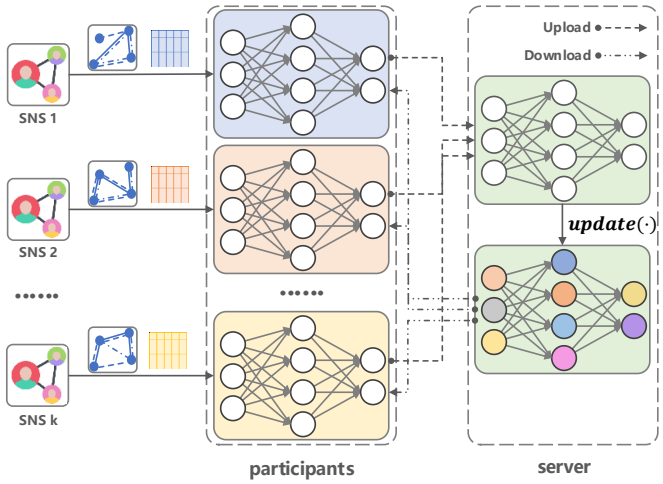

Fig. 3. The architecture of the federated learning framework. Firstly, the global model is initialized in the server sent to all participants. And then, each participant trains the model locally with its own data and uploads the model to the server. Finally, the server updates the global model and distribute it to participants in the next round.

IV. Experiments and Evaluation

A. Dataset

Because DA-MRG requires multi-relational graphs for users, we evaluate our method on TwiBot-20 [41], which is the only publicly available bot detection dataset with user follow relationships as we know. The dataset contains 5,237 human accounts and 6,589 bot accounts with their property items, tweets, and follow accounts. Apart from that, all accounts could be generally split into four domains: politics, business, entertainment and sports.

B. Baseline Methods

Firstly, we compare our proposed model DA-MRG against some existing social bot detection methods.

- **Lee *et al.*** [3] and **Yang *et al.*** [5] both apply random forest classifier with user features to detect bots in social.
- **Miller *et al.*** [4] extracts features from user tweets and properties and proposes a modified stream clustering algorithm to identify bot accounts.
- **Cresci *et al.*** [6] encodes users' online actions as digital DNA sequences and applies DNA analysis techniques to discriminate between genuine and spambot accounts.
- **Botometer** [7] is a public online service that leverages more than one thousand features to classify accounts.
- **Kudugunta *et al.*** [10], **Wei *et al.*** [11] and **SATAR** [13] extract features from user tweets, metadata, etc., and utilize recurrent neural networks to discover bot accounts.
- **Ali *et al.*** [16] and **BotRGCN** [14] construct the graph based on user relationships and gain the user representations with the graph neural networks to distinguish bots from humans.

Especially, the results of these baseline methods come from BotRGCN [14].

Algorithm 1: The Overall Process of Our Method

Input: Initial graph: G, Initial features: X, Layers of User Representation Learning: L, Local Epoch: E, Batch Size: B, Number of Communication: C, Number of Participant: K.
Output: Bot probability of each node v_i, $v_i \in V$.

ModelTraining():
  Construct multi-relational graphs $\{G_r\}|_{r=1}^{R}$ by Eq. (1), (2)
  for $l = 1, 2, ..., L$ **do**
    for $r \in R$ **do**
      $z_r(v) \leftarrow$ Eq. (3), (4);
      $\alpha_r(v) \leftarrow$ Eq. (5), (6), (7);
    $z(v) \leftarrow$ Eq. (8)
  $label,\ prob \leftarrow$ Eq. (10), (11), (12), (13)
  $loss \leftarrow BCELoss()$

LocalTraining(k, ω_c):
  for $e = 1, 2, ..., E$ **do**
    update the local model by ω_c
    for $b = 1, 2, ..., B$ **do**
      $loss \leftarrow$ **ModelTraining**()
      $\omega_{c+1}^{k} \leftarrow$ Eq. (14), (15)
  Return $\omega_{c+1}^{(k)}$ to the server

ServerExecutes():
  for $c = 1, 2, ..., C$ **do**
    for $k \in K$ **do**
      $\omega_{c+1}^{(k)} \leftarrow$ **LocalTraining**(k, ω_c)
    $\omega_{c+1} \leftarrow$ Eq. (16)

Simultaneously, in order to verify the effectiveness of our model in FedAvg, we compare the framework with the following methods.

- **Local Training** means we train DA-MRG locally at each participant by its own data, without any interaction among participants. The data of each participant in our experiments is a subset of the whole training set, generated as Section IV-C.
- **CDS** [42] is a centralized machine learning strategy training DA-MRG on the central server by collecting all participants' data.
- **CIIL** [42], [43] trains the model at each participant locally with consistent training epochs and repeatedly loops through all participants.

C. Implementation Details

We select *follower* and *following* as the multiple relations to construct the multi-relational graphs for DA-MRG. Moreover, Considering the computational cost and the number of parameters in the model, we adopt GraphSAGE [18] as the graph

neural network in the user representation learning module. In addition, to evaluate the federated learning framework with DA-MRG, we randomly assign each participant a unique subset. Firstly, we divide the whole training set into 12 parts evenly and test the training effect of the number of participants at $\{4, 8, 12\}$. And then, in order to test the training effect of the data size, we fix the number of participants at 8 and increase the data size of each participant from 100 to 2000. We adopt Accuracy, F1-score and MCC to estimate the performance of our method in all experiments.

The common parameters of model training are set as learning rate $5e-4$, batch size 256, dropout 0.3, L_2 regularization weight λ $3e-5$ and the dimension of the final embeddings 32. All experiments are conducted on a 64 core Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz with 128GB RAM and a Tesla P100-PICE GPU with 16GB memory. Our model is implemented using pytorch 1.8.1 as the backend.

D. Experiment Results

In this section, We first evaluate the detection performance of DA-MRG on TwiBot-20 [41] and study the effectiveness of domain-aware classifiers and the influence of pivotal experiment settings. Then, comparing with other training strategies, we estimate our model's performance with the federated learning framework. In our experiments, the performances are reported with the best results.

1) Performance Comparison: The results shown in Table I demonstrate that DA-MRG obtains better accuracy, F1-score and MCC than all other compared methods on TwiBot-20. Generally, deep learning-based methods usually perform better than those based on features engineering and traditional machine learning. Although all based on graph neural networks, our method has noticeable improvements over Ali *et al.* [16] and BotRGCN [14], which preliminarily illustrates that DA-MRG better utilizes the domain information of accounts for bot detection.

TABLE I
PERFORMANCE OF DIFFERENT DETECTION METHODS.

Method	Tags	Accuracy	F1 score	MCC
Lee *et al.* [3]	Classic ML	0.7456	0.7823	0.4879
Yang *et al.* [5]		0.8191	0.8546	0.6643
Cresci *et al.* [6]		0.4793	0.1072	0.0839
Miller *et al.* [4]		0.4801	0.6266	-0.1372
Botometer [7]		0.5584	0.4892	0.1558
Kudugunta *et al.* [10]	RNN	0.8174	0.7517	0.6710
Wei *et al.* [11]		0.7126	0.7533	0.4193
SATAR [13]		0.8412	0.8642	0.6863
Ali *et al.* [16]	GNN	0.6813	0.7318	0.3543
BotRGCN [14]		0.8462	0.8707	0.7021
Ours		**0.8698**	**0.8847**	**0.7392**

2) Training Ratio Analysis: The performance of our model with different training ratios is shown in Fig. 4. The result illustrates that our model suffers slight performance degradation with the decrease of training ratio, which implies that a small of labeled data is enough for the model. Specifically, our method outperforms all other compared methods with 60% training data.

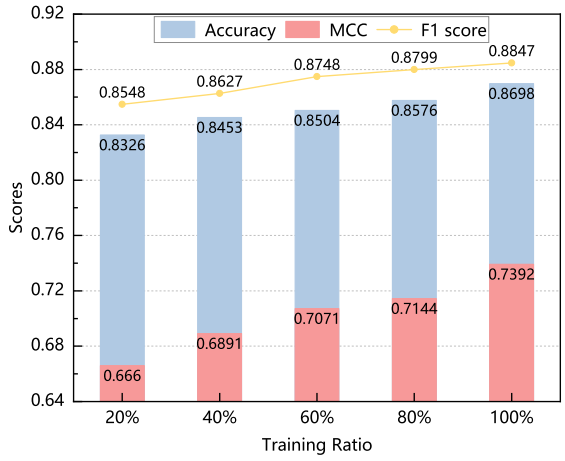

Fig. 4. Model Performance under Different Training Ratio

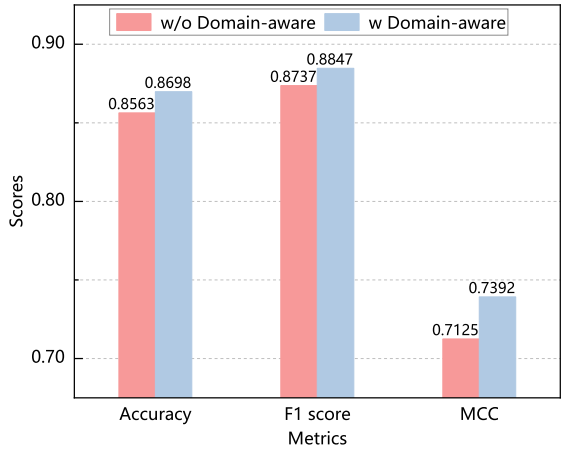

Fig. 5. Ablation studying removing domain-aware module from our model

3) Ablation study: Compared with other GNN-based methods, our model applies the user's domain information to assist the classification. Thus, we conduct an ablation study to evaluate the effectiveness of this module. The results in Fig. 5 show that all metrics would have some loss, especially MCC, when we remove the domain-aware classifiers. It demonstrates that the domain information is indeed valuable to improve detection performance.

4) Parameter Sensitivity: In this section, we investigate how parameters can affect prediction performance. The results are reported in Fig. 6.

Dimension of node embedding. Considering that the dimension of node embedding determines the presentation ability of GNN-based methods, we first explore the impact of various dimension {16, 32, 64, 128, 256}. Fig. 6 (a) indicts that the result achieves the best performance at the dimension of 32 and then degenerates with increases of dimension. We consider that a larger dimension could introduce excessive valueless information.

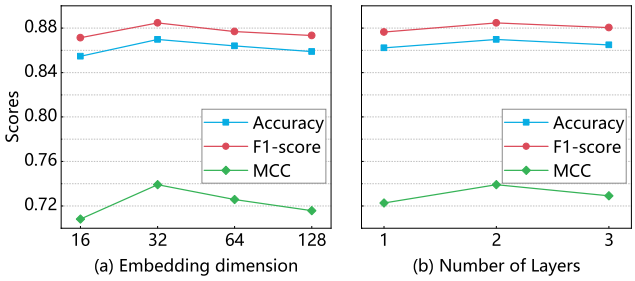

Fig. 6. Parameter sensitivity: Dimension of node embedding and Number of layers.

Number of user representation layer. We vary the layer number in the user representation learning module from 1 to 3 and evaluate its impact on prediction performance. As shown in Fig. 6 (b), 2-layer user representation learning can gain a better result in our model, while 3-layer may be confronted with overfitting.

5) Federated Learning Study: In this section, we evaluate the performance of DA-MRG combined with federated the learning framework in two experiments, fixing the number of participants and fixing the data size of the local training set.

Number of participants. The results in Table II show the prediction effect of increasing participants. Limited by the constant amount of data held by each participant and without any interactive process, local training always has the most unsatisfactory results. At the same time, CDS can achieve the best prediction effect with global data sharing. Generally, the performances of FedAvg and CIIL maintain at a certain level with CDS, which means that our method can well adapt to the parallel training.

TABLE II
RESULTS OF THE FEDAVG EXPERIMENT I.

Number of Participant	Method	Accuracy	F1 score	MCC
4	Local Training	0.8107	0.8346	0.6198
	CDS	**0.8453**	**0.8629**	**0.6892**
	FedAvg	0.8385	0.8582	0.6763
	CIIL	0.8436	0.8622	0.6864
8	Local Training	0.8091	0.8369	0.6201
	CDS	**0.8597**	**0.8781**	**0.7214**
	FedAvg	0.8575	0.8711	0.7123
	CIIL	0.8563	0.8702	0.7104
12	Local Training	0.8191	0.8424	0.6375
	CDS	**0.8639**	**0.879**	**0.7268**
	FedAvg	0.8571	0.8734	0.7133
	CIIL	0.8597	0.8752	0.7181

Size of the local training data. Table III shows the performance of increasing the size of the data set owned by each participant. Similar to the last experiment, the performance of local training is the worst. All other three methods achieve the best results when the size of local data is 1000. However, the prediction performance declines with the increase of local data size, particularly FedAvg. We consider that the model is affected by repeated data among participants. When the data size is greater than 1500, since the sum of data in all participants exceeds the total amount of training data, there are duplicate data among participants.

TABLE III
RESULTS OF THE FEDAVG EXPERIMENT II.

Size of Local Data	Method	Accuracy	F1 score	MCC
100	Local Training	0.7134	0.7349	0.4231
	CDS	**0.8157**	**0.8411**	**0.6321**
	FedAvg	0.8047	0.8280	0.6069
	CIIL	0.8056	0.8276	0.6082
500	Local Training	0.7988	0.8208	0.5942
	CDS	**0.8470**	**0.8648**	**0.6929**
	FedAvg	0.8352	0.8544	0.6688
	CIIL	0.8436	0.8639	0.6880
1000	Local Training	0.8267	0.8530	0.6585
	CDS	**0.8631**	**0.8763**	**0.7241**
	FedAvg	0.8622	0.8745	0.7222
	CIIL	0.8588	0.8753	0.7171
1500	Local Training	0.8233	0.8439	0.6446
	CDS	**0.8597**	**0.8752**	**0.7181**
	FedAvg	0.8555	0.8702	0.7089
	CIIL	0.8538	0.8706	0.7066
2000	Local Training	0.8369	0.8613	0.6790
	CDS	**0.8605**	0.8738	**0.7189**
	FedAvg	0.8588	0.8724	0.7155
	CIIL	0.8580	**0.8739**	0.7148

Both experiments indicate that training the model with federated learning across multiple social networks is an appropriate solution, which promotes the performance obviously than local training at each participant.

V. CONCLUSION

In this paper, we first propose a domain-aware social bot detection method based on the multi-relational graph, DA-MRG, to distinguish bot accounts from human accounts. Firstly, DA-MRG constructs multi-relational graphs with relations between users. Secondly, The model learns each user's high-level presentation via the user representation learning module, consisting of a series of graph embedding layers and semantic attention layers. Lastly, we fed the presentations to the domain-aware bot classifiers. We conduct various experiments to evaluate the model and the results indicate that our method can obtain better detection performance. In addition, the federated learning framework, FedAvg, is introduced to overcome the data privacy problem in data sharing among multiple social networks. And then, we explore the performance of DA-MRG with FedAvg through extensive experiments and demonstrate the efficiency of solving the problem of data islands.

ACKNOWLEDGMENT

The authors of this paper were supported by S&T Program of Hebei through grant 21340301D.

REFERENCES

[1] J. M. Berger and J. Morgan, "The isis twitter census: Defining and describing the population of isis supporters on twitter," *The Brookings Project on US Relations with the Islamic World 3:20*, 2015.

[2] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *arXiv preprint arXiv:1602.05629*, 2016.

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

[4] Z. Miller, B. Dickinson, W. Deitrick, W. Hu, and A. H. Wang, "Twitter spammer detection using data stream clustering," *Information Sciences*, vol. 260, pp. 64–73, 2014.

[5] K.-C. Yang, O. Varol, P.-M. Hui, and F. Menczer, "Scalable and generalizable social bot detection through data selection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 1096–1103, 2020.

[6] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "Dna-inspired online behavioral modeling and its application to spambot detection," *IEEE Intelligent Systems*, vol. 31, no. 5, pp. 58–64, 2016.

[7] C. A. Davis, O. Varol, E. Ferrara, A. Flammini, and F. Menczer, "Botornot: A system to evaluate social bots," in *Proceedings of the 25th international conference companion on world wide web*, pp. 273–274, 2016.

[8] H. Gao, Y. Chen, K. Lee, D. Palsetia, and A. N. Choudhary, "Towards online spam filtering in social networks," in *Proceedings of the 19th Annual Network & Distributed System Security Symposium*, vol. 12, pp. 1–16, 2012.

[9] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Design and evaluation of a real-time url spam filtering service," in *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, pp. 447–462, 2011.

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

[11] F. Wei and U. T. 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*, pp. 101–109, 2019.

[12] G. Stanton and A. A. Irissappane, "Gans for semi-supervised opinion spam detection," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2019.

[13] S. Feng, H. Wan, N. Wang, J. Li, and M. 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*, pp. 3808–3817, 2021.

[14] S. Feng, H. Wan, N. Wang, and M. 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*, pp. 236–239, 2021.

[15] J. Zhao, X. Liu, Q. Yan, B. Li, M. Shao, and H. Peng, "Multi-attributed heterogeneous graph convolutional network for bot detection," *Information Sciences*, vol. 537, pp. 380–393, 2020.

[16] S. Ali Alhosseini, R. Bin Tareaf, P. Najafi, and C. Meinel, "Detect me if you can: Spam bot detection using inductive representation learning," in *Companion Proceedings of The 2019 World Wide Web Conference*, pp. 148–153, 2019.

[17] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of the International Conference on Learning Representations*, 2017.

[18] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

[19] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proceedings of the International Conference on Learning Representations*, 2018.

[20] J. B. Lee, R. A. Rossi, X. Kong, S. Kim, E. Koh, and A. Rao, "Graph convolutional networks with motif-based attention," in *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 499–508, 2019.

[21] H. Peng, J. Li, Q. Gong, Y. Ning, S. Wang, and L. He, "Motif-matching based subgraph-level attentional convolutional network for graph classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 5387–5394, 2020.

[22] Q. Sun, J. Li, H. Peng, J. Wu, Y. Ning, S. Y. Phillip, and L. He, "Sugar: subgraph neural network with reinforcement pooling and self-supervised mutual information mechanism," in *2021 World Wide Web Conference, WWW 2021*, pp. 2081–2091, 2021.

[23] X. Zhao, Q. Dai, J. Wu, H. Peng, M. Liu, X. Bai, J. Tan, S. Wang, and P. Yu, "Multi-view tensor graph neural networks through reinforced aggregation," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[24] H. Peng, J. Li, Z. Wang, R. Yang, M. Liu, M. Zhang, P. Yu, and L. He, "Lifelong property price prediction: A case study for the toronto real estate market," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[25] G. Zhang, J. Wu, J. Yang, A. Beheshti, S. Xue, C. Zhou, and Q. Z. Sheng, "Fraudre: Fraud detection dual-resistant to graph inconsistency and imbalance," in *2021 IEEE International Conference on Data Mining*, pp. 867–876, IEEE, 2021.

[26] H. Peng, R. Zhang, S. Li, Y. Cao, S. Pan, and P. Yu, "Reinforced, incremental and cross-lingual event detection from social messages," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[27] H. Peng, R. Zhang, Y. Dou, R. Yang, J. Zhang, and P. S. Yu, "Reinforced neighborhood selection guided multi-relational graph neural networks," *ACM Transactions on Information Systems*, vol. 40, pp. 1–46, 2021.

[28] Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu, "Enhancing graph neural network-based fraud detectors against camouflaged fraudsters," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 315–324, 2020.

[29] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

[30] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016.

[31] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, 2019.

[32] Z. Liu, L. Yang, Z. Fan, H. Peng, and P. S. Yu, "Federated social recommendation with graph neural network," *ACM Transactions on Intelligent Systems and Technology*, nov 2021.

[33] X. Xu, H. Peng, M. Z. A. Bhuiyan, Z. Hao, L. Liu, L. Sun, and L. He, "Privacy-preserving federated depression detection from multi-source mobile health data," *IEEE Transactions on Industrial Informatics*, 2021.

[34] H. Peng, H. Li, Y. Song, V. Zheng, and J. Li, "Differentially private federated knowledge graphs embedding," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 1416–1425, 2021.

[35] C. He, K. Balasubramanian, E. Ceyani, C. Yang, H. Xie, L. Sun, L. He, L. Yang, P. S. Yu, Y. Rong, *et al.*, "Fedgraphnn: A federated learning system and benchmark for graph neural networks," in *Workshop on Distributed and Private Machine Learning: The International Conference on Learning Representations*, 2021.

[36] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, 2017.

[37] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[38] Y. Huang, L. Chu, Z. Zhou, L. Wang, J. Liu, J. Pei, and Y. Zhang, "Personalized cross-silo federated learning on non-iid data," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 7865–7873, 2021.

[39] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *Advances in neural information processing systems*, vol. 33, pp. 7611–7623, 2020.

[40] S. Che, H. Peng, L. Sun, Y. Chen, and L. He, "Federated multi-view learning for private medical data integration and analysis," *ACM Transactions on Intelligent Systems and Technology*, 2021.

[41] S. Feng, H. Wan, N. Wang, J. Li, and M. Luo, "Twibot-20: A comprehensive twitter bot detection benchmark," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 4485–4494, 2021.

[42] M. J. Sheller, B. Edwards, G. A. Reina, J. Martin, S. Pati, A. Kotrotsou, M. Milchenko, W. Xu, D. Marcus, R. R. Colen, *et al.*, "Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data," *Scientific reports*, vol. 10, no. 1, pp. 1–12, 2020.

[43] K. Chang, N. Balachandar, C. Lam, D. Yi, J. Brown, A. Beers, B. Rosen, D. L. Rubin, and J. Kalpathy-Cramer, "Distributed deep learning networks among institutions for medical imaging," *Journal of the American Medical Informatics Association*, vol. 25, no. 8, pp. 945–954, 2018.