SeGA: Preference-Aware Self-Contrastive Learning with Prompts for Anomalous User Detection on Twitter

Ying-Ying Chang, Wei-Yao Wang, Wen-Chih Peng

National Yang Ming Chiao Tung University, Hsinchu, Taiwan
cindy88409.cs10@nycu.edu.tw, sf1638.cs05@nctu.edu.tw, wcpeng@cs.nycu.edu.tw

Abstract

In the dynamic and rapidly evolving world of social media, detecting anomalous users has become a crucial task to address malicious activities such as misinformation and cyberbullying. As the increasing number of anomalous users improves the ability to mimic normal users and evade detection, existing methods only focusing on bot detection are ineffective in terms of capturing subtle distinctions between users. To address these challenges, we proposed SeGA, preference-aware self-contrastive learning for anomalous user detection, which leverages heterogeneous entities and their relations in the Twittersphere to detect anomalous users with different malicious strategies. SeGA utilizes the knowledge of large language models to summarize user preferences via posts. In addition, integrating user preferences with prompts as pseudo-labels for preference-aware self-contrastive learning enables the model to learn multifaceted aspects for describing the behaviors of users. Extensive experiments on the proposed TwBNT benchmark demonstrate that SeGA significantly outperforms the state-of-the-art methods (+3.5% $\sim$ 27.6%) and empirically validate the effectiveness of the model design and pre-training strategies. Our code and data are publicly available at https://github.com/ying0409/SeGA.

1 Introduction

The exploration of anomalous user detection has broad applicability across various domains. Whether it involves developing strategies for network security (Bilot et al. 2023), financial transactions (Chai et al. 2022), or social media analytics (Agarwal et al. 2022; Wang and Peng 2022), these scenarios can be effectively framed as anomalous user detection systems characterized by intricate relations between users. Twitter serves as one of the most widely-used social media platforms; however, the widespread features in recent years have also led to the rapid growth of anomalous users. For instance, abnormal users often initiate campaigns to pursue malicious goals, which violates the principles of healthy online discussions on social media platforms (Alieva, Moffitt, and Carley 2022). One of the most common types of anomalous users is bots, identified at 9% to 15% of active users (Varol et al. 2017). In late 2017, the US Congress disclosed a list of 2.7k Twitter accounts that were identified as Russian trolls (Zannettou et al. 2019), which is an emerging type of anomalous user. As the variety of anomalous user types increases, and their potential negative impact on social media becomes more pronounced, it is crucial to emphasize the significance of advancing anomaly detection methods capable of identifying not only bots but also trolls.

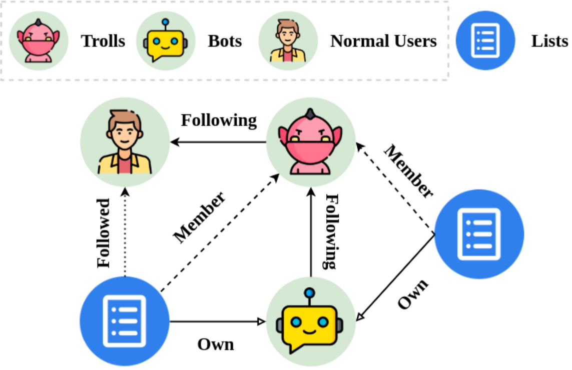


Figure 1: An example of anomalous user behavior in social media, where the edges connecting entities represent diverse relationships.

Previous works primarily focused on identifying bots and achieved effectiveness by leveraging the topology with graph neural networks (GNNs) (Kipf and Welling 2017; Velickovic et al. 2018) and heterogeneous information networks (HINs) (Feng et al. 2021c, 2022a). However, as the diversity of anomalous users and the evolutionary strategies for malicious activities continue to expand, such as spreading fake news through lists[1], existing methods lack the ability to effectively distinguish various types of anomalous users like both bots and trolls. The main difference in detecting trolls and bots is that the former is controlled by humans, resulting in similar behavior to normal users compared with the latter.

Figure 1 illustrates an example of anomalous user behavior with other entities on social media, where anomaly users are able to not only interact with other users, but also own lists and add other anomalous users as members to carry out malicious activities such as spreading fake news. For example, a bot might create a list and add trolls as members. If a

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1] A list is a curated selection of Twitter accounts for organizing.

normal user follows this list, they may receive misinformation that is spread by anomalous users within the list. Therefore, a challenging problem arises with this problem: *How to capture the subtle difference between users when troll users may behave similarly to normal users?*

To tackle these issues, we propose a novel framework SeGA, a preference-aware self-contrastive learning approach for anomalous user detection on Twitter, which encodes the heterogeneous relations of various entities on Twitter with the heterogeneous encoder. We introduce self-contrastive learning with pseudo-labels to discern subtle differences between users based on user preferences via the corresponding posts. Specifically, we construct an HIN that incorporates various edge types between node types including users and lists to model users with diverse activities. In order to learn about discrepancies between users, the pre-training strategy incorporates the knowledge of large language models (LLMs) to capture user-preferred topics and emotions for preference-aware self-contrastive learning with prompts. To evaluate the effectiveness of our proposed method, we propose a new anomalous user detection benchmark, TwBNT, which demonstrates a significant improvement of at least 3.5% compared with the state-of-the-art baselines in terms of F1 score.

Our main contributions are summarized as follows:

- We propose SeGA to address the challenging but emerging anomalous user detection task on Twitter. To the best of our knowledge, this is the first work that jointly distinguishes normal, troll, and bot users on social media.
- We introduce preference-aware self-contrastive learning to learn user behaviors via the corresponding posts. In addition, we incorporate prompt templates with user preferences as pseudo-labels to capture the user-preferred topics and emotions.
- We collected a large-scale Twitter dataset named TwBNT for anomalous user detection including normal users, troll users, and bots. Extensive experiments were conducted to evaluate the performance of our proposed model, which demonstrated superior improvement of between 3.5% and 27.6%.

2 Preliminaries

2.1 Related Work

Anomalous User Detection on Twitter Early Twitter anomalous user detection models focused on detecting bot accounts with user features or tweets (Miller et al. 2014; Cresci et al. 2016). With the advent of graph neural networks, an increasing number of graph-based bot detectors have been proposed by representing users and their interactions as a social graph, and utilizing aggregation techniques to gather information from neighboring nodes. For example, GCN (Kipf and Welling 2017) aggregates features equally from neighboring users to learn representations, while GAT (Velickovic et al. 2018) models user influence using attention mechanisms. On the other hand, heterogeneous graphs are also utilized for bot detection due to their effectiveness in representing social networks with diverse node and edge types. Lv et al. (2021) adopted different strategies to enhance GAT on the heterogeneous graph, and Feng et al. (2022a) proposed relational graph transformers to model heterogeneous relations and influence heterogeneity between users. Nonetheless, previous methods failed to detect another important facet of anomalous users: trolls, which react similarly to normal users to avoid detection. Therefore, we introduce preference-aware self-contrastive learning to differentiate user preferences from posts with pseudo-labels.

Self-Supervised Learning In recent years, self-supervised learning (SSL) has been proven to be a powerful and effective approach in various domains for learning contextualized representations via pretext tasks, such as natural language processing (Gao, Yao, and Chen 2021) and computer vision (Lv et al. 2021; Bardes, Ponce, and LeCun 2022). Predictive learning serving as a branch of SSL has also been applied for bot detection with promising results. Feng et al. (2021a) incorporated follower count as a self-supervised learning signal to enhance the model's performance in bot detection. However, relying on a single feature as a self-supervised indicator overlooks the diversity among anomalous users and falls short of adequately representing different users. For instance, troll users are able to construct artificial follower counts followed by other troll users. Therefore, we adopt user preferences acquired by LLMs with the corresponding posts to summarize multifaceted behaviors (topics and emotions in this paper) to describe users for self-contrastive learning.

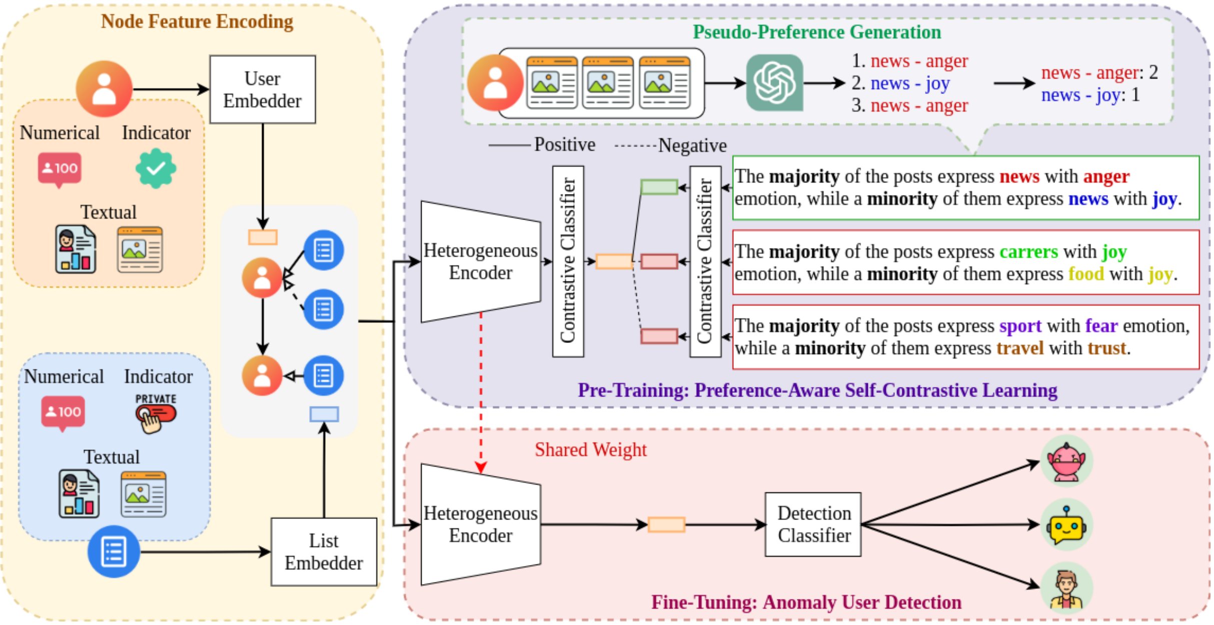
2.2 Problem Formulation

We denote $G = \{V, E, A, R, \phi, \psi\}$ as the input heterogeneous information network (HIN) with different types of nodes V and edges E, and it is associated with a node mapping function $\phi : V \rightarrow A$ and an edge mapping function $\psi : E \rightarrow R$, where A and R denote the sets of all node types and edge types. The nodes in HIN are categorized into either user type u or list type l as $v_i^A \in V$, where $A = \{u, l\}$ is the node type of node i representing user and list, respectively. The edges are denoted as $(v_i^A, r, v_j^A) \in E$, where $r \in R$ represents the relation between node i and node j.

For each node i, node features are categorized into three types: indicator features $C_i = \{c_{i_1}, c_{i_2}, ..., c_{i_k}\}$ indicating if matching the indication, numerical features (e.g., number of posts) $N_i = \{n_{i_1}, n_{i_2}, ..., n_{i_m}\}$ and textual features $T_i = \{des_i, twe_i\}$, where $des_i = \{w_{i_1}^{des}, w_{i_2}^{des}, ..., w_{i_s}^{des}\}$ denotes the description with s words for a user or a list, and $twe_i = \{twe_{i_j}\}_{j=1}^{q}$ denotes q tweets posted from a user or contained in a list, while each tweet j contains L words denoted as $twe_{i_j} = \{w_{i_{j_1}}^{twe}, w_{i_{j_2}}^{twe}, \cdots, w_{i_{j_L}}^{twe}\}$. We formulate anomalous user detection as a multi-class classification problem, where users are classified into three classes based on past activities and relations with society. Given a learned HIN G, we aim to learn an anomalous user detection function $f(G) \rightarrow \hat{y}$ to predict whether the user v_i^u is a troll, bot, or normal user.

3 The Proposed Method

The overview of SeGA is shown in Figure 2, which consists of three stages: node feature encoding, the pre-training



The **majority** of the posts express **news** with **anger** emotion, while a **minority** of them express **news** with **joy**.

The **majority** of the posts express **carrers** with **joy** emotion, while a **minority** of them express **food** with **joy**.

The **majority** of the posts express **sport** with **fear** emotion, while a **minority** of them express **travel** with **trust**.

Figure 2: Overview of our proposed framework, SeGA, consisting of three stages: 1) Node feature encoding to initialize node embeddings from heterogeneous information; 2) The pre-training stage to pre-train the encoder with pseudo-aware self-contrastive learning to capture the differences in user preferences; and 3) The fine-tuning stage to classify anomalous users.

stage, and the fine-tuning stage. SeGA leverages users and lists as nodes within an HIN to capture diverse relationships among these entities. We first encode three types of node features (i.e., indicator, numerical, and textual) to encode representations for each node type. In the pre-training stage, the encoder is pre-trained with preference-aware self-contrastive learning to learn the user preferences summarized by LLMs from posts. In the fine-tuning stage, we fine-tune the pre-trained model to classify anomalous users.

3.1 Model Architecture

Node Feature Encoding Similar to the process of feature encoding as (Feng et al. 2022a), we first concatenate each indicator feature for each node i with node type A:

$$x_i^{A ind} = c_{i_1} \oplus c_{i_2} \oplus \cdots \oplus c_{i_k}, \tag{1}$$

where $\oplus$ is the concatenation operator, and $x_i^{A ind} \in \mathbb{R}^k$ is the indicator embedding concatenated by k features.

Similarly, the numerical features are concatenated as:

$$x_i^{A num} = n_{i_1} \oplus n_{i_2} \oplus ... \oplus n_{i_m}, \tag{2}$$

where $x_i^{A num} \in \mathbb{R}^m$ is the numerical embedding for node i concatenated by m features.

To encode textual features, we applied pre-trained RoBERTa (Liu et al. 2019) for encoding with s words:

$$x_i^{A des} = RoBERTa(\{w_{i_j}^{des}\}_{j=1}^{s}), \tag{3}$$

where $x_i^{A des} \in \mathbb{R}^{d_{des}}$ denotes the description embedding. Likewise, we obtain the tweet embedding $x_{i_j}^{A twe} \in \mathbb{R}^{d_{twe}}$ by averaging all embeddings encoded from each tweet with pre-trained RoBERTa:

$$x_{i_j}^{A twe} = RoBERTa(\{w_{i_{j_y}}^{twe}\}_{y=1}^{L}), \tag{4}$$

$$x_i^{A twe} = AVG(x_{i_1}^{A twe}, x_{i_2}^{A twe}, \cdots, x_{i_q}^{A twe}). \tag{5}$$

Afterwards, we transform them separately:

$$x_i'^{A_F} = \sigma(W_x \cdot x_i^{A_F} + b_x), F \in \{ind, num, des, twe\} \tag{6}$$

where $x_i'^{A_F} \in \mathbb{R}^{d_h}$, W_x, b_x are trainable parameters, and σ is denoted as LeakyReLU.

For each user or list node i, we obtain the user/list embeddings $x_i^A \in \mathbb{R}^{4*d_h}$ by concatenating the indicator $x_i'^{A ind}$, numerical $x_i'^{A num}$, description $x_i'^{A des}$ and tweet $x_i'^{A twe}$ embeddings:

$$x_i^A = x_i'^{A ind} \oplus x_i'^{A num} \oplus x_i'^{A des} \oplus x_i'^{A twe}. \tag{7}$$

We then transform the user/list embeddings x_i^A as the initial node embedding $z_i^{A,(0)} \in \mathbb{R}^{4*d_h}$ for the heterogeneous encoder:

$$z_i^{A,(0)} = \sigma(W_I \cdot x_i^A + b_I), \tag{8}$$

where W_I and b_I are trainable parameters.

Heterogeneous Encoder To model the various entities and their diverse relation with different importance to enrich the embeddings of users, we applied a relational graph transformer (RGT) following (Feng et al. 2022a) and an MLP with activation function as the heterogeneous encoder:

$$z_i^{u,(g)}, z_i^{l,(g)} = RGT^{(g)}(z_i^{u,(g-1)}, z_i^{l,(g-1)}), \tag{9}$$

$$z_i^u = \sigma(W_z \cdot z_i^{u,(g)} + b_z), \tag{10}$$

where $z_i^{u,(g)}, z_i^{l,(g)} \in \mathbb{R}^{d_{out}}$ is the i-th node embedding learned from the g-th layer, $z_i^u \in \mathbb{R}^{d_u}$ is the i-th user representations, and W_z and b_z are trainable parameters. We note that as the task is to classify user categories, we adopt aggregated user embeddings z_i^u for preference learning and anomaly detection.

3.2 Pre-Training Stage: Learning User Preferences via Posts

User preferences represent the behavior of an individual as they showcase a person's choices in various aspects, which is beneficial for detecting users' behavior patterns. Therefore, we introduce the pseudo-preference generation, which is summarized from LLMs based on users' historical posts. In order to describe and learn user preferences, we designed a prompt template to represent the majority and minority topic-emotion pairs of user posts for preference-aware self-contrastive learning.

Pseudo-Preference Generation Given the success paradigms of leveraging LLMs in natural language applications (Wu, Zhang, and Huang 2023), we incorporate the powerful knowledge of LLMs to retrieve user preferences from posts. Specifically, we opt for the preferred topic and emotion to represent the preference of each user from the corresponding posts since anomalous users may exploit them to achieve malicious intentions (Ghanem, Buscaldi, and Rosso 2019; Balasubramanian et al. 2022). The 10 recent posts of user i are used as the prompt for LLM to generate the topic t and the emotion e used in each tweet j:

$$\{(t^u_{i_j}, e^u_{i_j})^{10}_{j=1}\} = LLM(P(twe^u_{i_1} \oplus \cdots \oplus twe^u_{i_{10}})), \tag{11}$$

where $P()$ is an instruction prompt to acquire the user preference from LLMs described in Appendix A.1, and LLM is ChatGPT (Liu et al. 2023) in this paper. The topics are derived from 16 categories based on Twitter topics (Jim and Ann 2020), and the 8 emotions are based on Plutchik's emotions (Ghanem, Buscaldi, and Rosso 2019).

Preference-Aware Self-Contrastive Learning After obtaining the topic-emotion pairs for each user, we aim to pre-train the model with this pseudo-information to describe user preferences. An intuitive approach is to predict all topics and emotions of each post by a user, which can be formulated as a multi-label classification task. However, this fails to effectively capture the user-preferred topics and emotions since the model treats all labels equally without considering their relative importance. For example, a user may post mainly on a news topic with anger and seldom post on a news topic with joy.

To tackle this issue, we propose a preference-aware self-contrastive learning approach that incorporates preferences with prompts for learning enhancement. We define the preference of a user as the integration of the most frequently used topic-emotion pair $(t^u_{i_{max}}, e^u_{i_{max}})$ and the least frequently used pair $(t^u_{i_{min}}, e^u_{i_{min}})$ to reflect the user's interest and emotional behavior. We then generate the pseudo-label p^u_i for each user i by leveraging the designed prompt template $PT()$ filled with the most and the least frequently used topic emotion pairs as:

$$p^u_i = PT((t^u_{i_{max}}, e^u_{i_{max}}), (t^u_{i_{min}}, e^u_{i_{min}})). \tag{12}$$

The prompt template is designed as: *"The majority of the posts express $t^u_{i_{max}}$ with $e^u_{i_{max}}$ emotion, while a minority of them express $t^u_{i_{min}}$ with $e^u_{i_{min}}$."*. Experiments with different templates are discussed in Section 4.4.

Afterwards, pseudo-labels of user i are encoded with the prompt encoder using SimCSE RoBERTa (Gao, Yao, and Chen 2021), which learns sentence embeddings from contrastive learning:

$$p'^u_i = SimCSE(p^u_i), \tag{13}$$

where $p'^u_i \in \mathbb{R}^{d_p}$ is the pseudo-label embedding of user i. Then, we transform the user embedding z^u_i and pseudo-label embedding p'^u_i with the contrastive classifier:

$$\tilde{z}^u_i = W_{\tilde{z}} \cdot z^u_i + b_{\tilde{z}}, \tag{14}$$

$$\tilde{p}^u_i = W_{\tilde{p}} \cdot p'^u_i + b_{\tilde{p}}, \tag{15}$$

where $\tilde{z}^u_i \in \mathbb{R}^{d_a}$ denotes the anchor user embedding for user i and $\tilde{p}^u_i \in \mathbb{R}^{d_a}$ denotes the corresponding positive sample embedding, $W_{\tilde{z}}$, $b_{\tilde{z}}$, $W_{\tilde{p}}$ and $b_{\tilde{p}}$ are trainable parameters.

To compute the contrastive loss L_{pre}, we define the embedding $\tilde{p}^u_i$ as the positive pair of the anchor user's embedding $\tilde{z}^u_i$, and embeddings transformed from other pseudo-labels are considered as the negative pairs of the anchor user:

$$L_{pre} = -\sum_{i \in U} log \frac{exp(sim(\tilde{z}^u_i \cdot \tilde{p}^u_i)/\tau)}{\sum_{j \in S(i)} exp(sim(\tilde{z}^u_i \cdot \tilde{p}^u_j)/\tau)}, \tag{16}$$

where U is the set of indices of all user nodes, $S(i)$ is the set of a positive pair, and negative pairs are randomly sampled from all negative pairs. When minimizing the contrastive loss, the user embeddings with the same pseudo-label tend to be closer together, while simultaneously encouraging the encoder to learn user-preferred topics and emotions.

This generic pre-training objective with pseudo-labels and the prompt template reinforces the model to learn user behavior with at least two advantages:

- User preferences can also be utilized in other social media-related tasks, such as user recommendation (Wang et al. 2022) and community detection (Wu et al. 2021), since posts play a crucial role in user interactions within social media.
- Incorporating user post preference enables an effective representation of users, as posting content is the primary avenue through which individuals pursue their objectives and express themselves on social media.

3.3 Fine-Tuning Stage: Anomalous User Detection

To achieve the goal of anomalous user detection, the user embedding z^u_i from the pre-trained model is used in the detection classifier with a softmax layer to predict the class $\hat{y}$ for each user i:

$$\hat{y}_i = softmax(W_y \cdot z^u_i + b_y), \tag{17}$$

where W_y, b_y are trainable parameters.

Finally, we jointly fine-tune the pre-trained embedders, pre-trained encoder, and the detection classifier for anomalous user detection. The fine-tuning process incorporates both the cross-entropy loss and an L2 regularization term as follows:

$$L_{fine} = -\sum_{i \in U} [y_i \log(\hat{y}_i)] + \lambda \sum_{\omega \in \theta} \omega^2, \tag{18}$$

where U is the set of indices of all user nodes, y_i represents the ground-truth label of user i, λ is a hyper-parameter, and θ encompasses all the trainable parameters.

4 Experiments

4.1 Experiment Settings

Proposed Benchmark: TwBNT Dataset Since there is no public dataset consisting of trolls, bots, and normal users, we proposed a new dataset, TwBNT, by extending the bot detection benchmark Twibot-22 (Feng et al. 2022b) with automatic troll annotations. Twibot-22 is a bot detection dataset that provides graph structures with various entities and relations within the Twitter network, which evaluates graph-based approaches to bot detection. Nonetheless, Twibot-22 annotates users as either bots or humans, but excludes the critical category for trolls. To this end, we sample users from Twibot-22 using a breadth-first search algorithm for user collection following (Feng et al. 2021b) to ensure the sampled users include different types of bots, trolls, and normal users. Then, the list nodes connected to the sampled users are employed to construct the TwBNT dataset. Since trolls are controlled by real human users, we automatically identify them by obtaining a troll score $scr \in [0, 1]$ for each user labeled in Twibot-22 with the widely recognized platform Bot Sentinel[2]. Users with a score greater than the threshold value $scr = 0.5$ are labeled as trolls, while others are labeled as normal users following the spreading intent detection (Zhou et al. 2022). Table 1 summarizes the statistics of our collected datasets.

Baselines Due to the lack of baselines for the proposed anomaly user detection tasks that also distinguish troll users, we compared the proposed model with 6 baselines of bot detection methods to verify its effectiveness: GCN (Kipf and Welling 2017), GAT (Velickovic et al. 2018), SimpleHGN (Lv et al. 2021), SATAR (Feng et al. 2021a), BotRGCN (Feng et al. 2021c), and RGT (Feng et al. 2022a). More details of baselines are given in Appendix B.

Implementation Details The numbers of indicator features k for the user and list node are 3 and 1, and the numbers of numerical features m for the user and list nodes are 5 and 4. Detailed features are described in Appendix A.2. The numerical features N are applied with z-score normalization. The dimensions of d_{des}, d_{twe} and d_p are 768, the dimension of d_h is 32, the dimension of d_{out} is 128, and the dimensions of d_u and d_a are 64. The number of layers g of the relational graph transformer is 2. Following (Feng et al. 2022a), the max number of tweets q for each user and list is set to 20 for representing the recent 20 tweets from each user and list. The max lengths for description s and each tweet L are set to 50 words, and padding with zeros is applied if the length is insufficient. In the pre-training stage, we ask ChatGPT to classify each tweet into 16 topics and 8 emotions by providing the instructions depicted in Appendix A.2. As there remain some results that do not belong to these categories, we treat them as "others". We randomly sample 100 prompts with other pseudo-labels as negative samples and set the temperature τ as 0.1 for computing the pre-training loss. In the fine-tuning stage, λ is set to 3×10^{-5}. The training epochs of the pre-training and fine-tuning stages are set to 100 and 150, respectively. The dropout rate is set to 0.3. We employ the AdamW optimizer (Loshchilov and Hutter 2019) using the learning rate of 0.001 and the batch size is set to 2048. All experiments were conducted with a Nvidia RTX A5000 GPU and all parameters were tuned based on the validation set. The analysis of the hyper-parameter is discussed in Appendix C.1.

Types	Item	Count	Total
2 node types (A)	# user # list	100,001 20,788	120,789
5 edge types (R)	# following # followers # membership # followed # own	751,927 148,250 365,593 39,258 7,901	1,312,929
3 classes	# troll # bot # normal	896 5,047 94,058	100,001
3 splits	# train # valid # test	70,000 20,000 10,001	100,001

Table 1: Statistics of the proposed benchmark TwBNT.

Evaluation Metrics As the numbers of troll and bot users are more imbalanced than normal users, as shown in Table 1, macro F1 scores are utilized as principal indicators to evaluate the overall performance of the model. Additionally, precision and recall are employed to evaluate the results of anomaly user classification. We evaluate all baselines and our model by computing the mean and standard deviation for the results obtained with 3 different random seeds.

4.2 Quantitative Performance

Table 2 summarizes the overall performance of anomalous user detection methods, demonstrating that our proposed model surpasses all baselines, which are extended to classify trolls. Quantitatively, SeGA achieves at least a 3.5% improvement in F1-score compared with the best-performing baseline. We summarize the observations as follows:

Compared with the homogeneous graph-based methods, GCN and GAT, SeGA achieves a significant improvement on Macro F1 with 27.6% and 19.6%, demonstrating the capability of heterogeneous graphs to classify various anomalous and normal users. Additionally, SeGA outperforms other heterogeneous graph-based methods such as SimpleHGN, BotRGCN, and RGT, ranging from 3.5% to 26.6%, which is attributed to considering diverse entities not only from edges but also from nodes. This also highlights the capability of leveraging topic-emotion pairs via posts with prompts as pseudo-labels to model user preferences.

Although SATAR also adopts the follower count as a self-supervised objective, we can observe that the performance of SATAR deteriorates substantially since troll users can manipulate the follower count followed by other troll users to act as normal users. The comparison of SeGA and SATAR

[2] https://botsentinel.com/

Methods	Attention Mechanism	Edge Heterogeneity	Node Heterogeneity	Self Supervision	Precision	Recall	F1
GCN					67.43±1.24	42.76±1.14	47.57±1.54
GAT	✓				66.40±2.19	45.54±0.84	50.75±0.66
SimpleHGN	✓	✓			79.84±2.54	45.68±1.19	52.22±1.54
SATAR	✓	✓		✓	55.31±7.69	44.77±2.35	46.98±1.06
BotRGCN		✓			65.85±12.55	43.51±5.20	47.94±7.20
RGT	✓	✓			75.55±1.27	52.10±1.25	58.61±0.92
SeGA (Ours)	✓	✓	✓	✓	68.13±0.97	56.58±1.50	**60.69±0.72**

Table 2: Performance of all baseline methods and our proposed SeGA. All methods are evaluated by precision, recall, and Macro F1. The best results are highlighted in boldface, and the second-best results are underlined. The strategies employed by each method are marked as ✓.

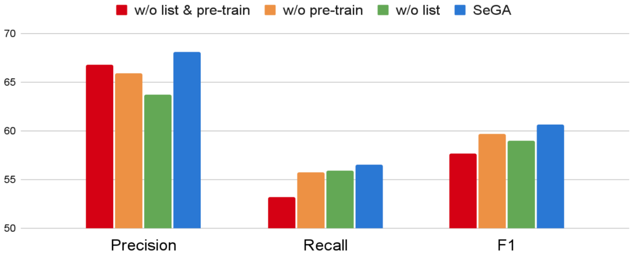


Figure 3: Ablation study including removing the list nodes and the pre-training stage from SeGA.

	Methods	Precision	Recall	F1
A1	RoBERTa	68.32	55.39	59.84
A2	Short	65.86	54.16	58.26
	Topic	66.03	55.53	59.27
	Emotion	65.59	55.19	58.80
	Tandem	72.87	53.53	59.83
A3	Multi-label	71.95	52.59	58.74
	SeGA (Ours)	68.13	56.58	**60.69**

Table 3: Performance of SeGA with different pre-training settings, including modifications in the prompt encoder (A1), prompt design (A2), and pre-training task (A3).

reveals the importance of considering user preferences for distinguishing anomalous users, where the user preference not only behaves as a task-agnostic objective but also describes the user profile based on the corresponding posts.

4.3 Ablation Study

To analyze the contributions of different designs of SeGA, we conducted an ablation study by removing different node types and the pre-training stage as shown in Figure 3. As expected, removing both types of information (w/o list & pre-train) significantly hinders the performance. Moreover, we can observe that the F1 score decreases when removing list information (w/o list) for constructing the graph, which signifies that considering user relations with lists helps in detecting anomalous users with various malicious activities. The deleterious performance of removing the pre-training stage (w/o pre-train) suggests the advantage of learning user preferences as the pre-training objective.

4.4 Pre-Training Strategies Study

To further validate the effectiveness of the pre-training design, we conducted extensive experiments with three aspects: prompt encoder (A1), prompt design (A2), and pre-training task (A2), as illustrated in Table 3.

A1: Variant of Prompt Encoder We changed the prompt encoder in Eq. (13) from SimCSE RoBERTa to RoBERTa for preference-aware self-contrastive learning, which can be observed that replacing the prompt encoder with RoBERTa slightly decreases the performance. As shown in Figure 4, we further compute the cosine similarity between prompt embeddings using these prompt encoders separately to analyze the discrepancies of different topic-emotion prompts. The result shows that SimCSE RoBERTa provides more diverse discrepancies with different topic-emotion pairs in the same prompt template, while RoBERTa is inferior to separate different representations of topic-emotion pairs. In addition, the minimum cosine similarity is 0.9795 with pre-trained RoBERTa and 0.3083 with SimCSE RoBERTa. These results indicate that SimCSE RoBERTa is able to capture subtle differences between prompts arising from the replacement of emotions, whereas RoBERTa produces similar representations with different pseudo-labels. The distinguishable embeddings enable performance enhancement for self-contrastive learning, which again raises the need for incorporating this for contrastive learning with prompts.

A2: Variants of Prompt Design To analyze different designs of the prompt template for self-contrastive learning, we evaluate SeGA with four variants: 1) **Short prompt**: Majority: $t^u_{i_{max}}$ - $e^u_{i_{max}}$, minority: $t^u_{i_{min}}$ - $e^u_{i_{min}}$. 2) **Topic prompt**: The majority of the posts express $t^u_{i_{max}}$, while a minority of them express $t^u_{i_{min}}$. 3) **Emotion prompt**: The majority of the posts express $e^u_{i_{max}}$, while a minority of them express $e^u_{i_{min}}$. 4) **Tandem prompt**: The majority of the posts express $t^u_{i_{max}}$, while a minority of them express $t^u_{i_{min}}$. The majority of the posts express $e^u_{i_{max}}$, while a minority

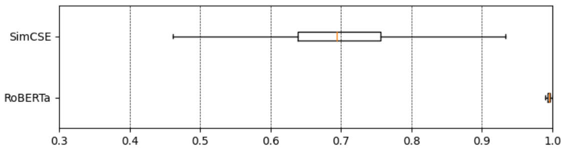


Figure 4: The first, second, and third quartiles of the cosine similarity results between prompt embeddings from RoBERTa and SimCSE.

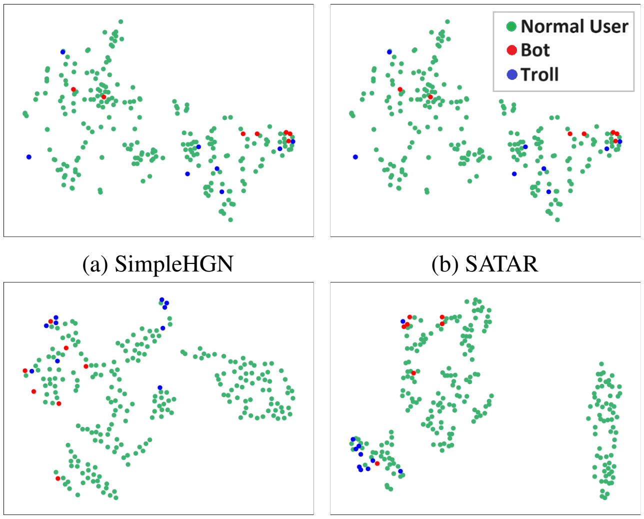
of them express $e^u_{i_{min}}$. For topic and emotion prompts, we calculate the corresponding preference frequencies independently, and the tandem prompt is an integration of topic and emotion prompts, which can be viewed as the prompt based on independent frequencies.

Adopting the short prompts for learning leads to an inferior performance compared with the proposed design, which implies that taking the semantics of natural language into account to form a prompt is essential for preference-aware self-contrastive learning compared to forming the template naively and structurally. From topic and emotion prompts, the inclined effects indicate that considering the information of either topic of emotion to describe users is insufficient to differentiate normal, troll, and bot users. Moreover, the performance decreases when applying tandem prompts, which illustrates that capturing the paired relationship between topics and emotions is more meaningful than considering them separately. Meanwhile, our proposed prompt still outperforms the tandem prompt, which showcases the strength of not only jointly considering the frequencies, but also describing topic-emotion pairs together to form a prompt.

A3: Variant of Pre-Training Task To delve into the learning strategies for incorporating topic-emotion pairs via posts, we modify the preference-aware self-contrastive learning to the multi-label classification task (Multi-label), which aims to predict all potential topic-emotion pairs of a user with the same model architecture. We can see that the proposed self-contrastive learning outperforms predictive learning, which highlights that multi-label classification fails to capture users' preferred emotions associated with specific topics since it treats all labels as being equally important. In contrast, preference-aware self-contrastive learning mitigates this limitation through the prompt design, leading to a substantial improvement in anomaly user detection.

4.5 Case Study

In order to investigate the efficacy of embeddings acquired from preference-aware self-contrastive learning related to anomaly user detection, we sampled users who primarily expressed anger in their general posts (i.e., others) to reflect the normal behavior of a user. We note that extensive topics related to news are discussed in Appendix C.2. We visualize the t-SNE embedding results of our approach against three baselines: RGT, SimpleHGN, and SATAR as illustrated in Figure 5. The embeddings of SATAR and RGT show ambiguous collocation for groups of normal, bot, and troll users, while troll and bot users of SimpleHGN have more distinguishable distances but mix with normal users. Nonetheless, our proposed SeGA is able to describe different categories of users with clearer boundaries, which is attributed to preference-aware self-contrastive learning to describe user behaviors via the corresponding posts. It is worth noting that the generic preference-aware objective is not directly intended to address the anomaly user detection task, but definite boundaries help group various user behaviors.



(a) SimpleHGN

(b) SATAR

(c) RGT

(d) SeGA (Ours)

Figure 5: The visualization of user embeddings produced by the heterogeneous encoder and baselines.

5 Conclusion

This paper proposes SeGA, novel preference-aware self-contrastive learning with pseudo-preference generations for anomalous user detection on Twitter, including more challenging troll users. Distinct from existing works that only focus on bot detection, our proposed method is able to distinguish various anomalous and normal users by learning similarities and discrepancies of topic-emotion pairs from posts of users summarized by LLMs, allowing the capability of capturing user-preferred behaviors. Meanwhile, our prompt design considers the context of the multifaceted preferences of users to avoid the model being biased by only considering the most-appearance preferences. We propose a new benchmark for distinguishing anomalous and normal users on Twitter, which shows that SeGA significantly outperforms state-of-the-art approaches by between 3.5% and 27.6%. We believe that SeGA serves as a general framework for social media due to the flexible design for incorporating user preferences from external knowledge of LLMs as well as for the self-contrastive learning approach, and multiple interesting directions could be further explored within the framework, such as more metadata for user preferences, few-shot examples, etc.

References

Agarwal, P.; Srivastava, M.; Singh, V.; and Rosenberg, C. 2022. Modeling User Behavior With Interaction Networks for Spam Detection. In *SIGIR*, 2437–2442. ACM.

Alieva, I.; Moffitt, J. D.; and Carley, K. M. 2022. How disinformation operations against Russian opposition leader Alexei Navalny influence the international audience on Twitter. *Soc. Netw. Anal. Min.*, 12(1): 80.

Balasubramanian, S. K.; Bilgic, M.; Culotta, A.; Hemphill, L.; Nikolich, A.; and Shapiro, M. A. 2022. Leaders or Followers? A Temporal Analysis of Tweets from IRA Trolls. In *ICWSM*, 2–11. AAAI Press.

Bardes, A.; Ponce, J.; and LeCun, Y. 2022. VICRegL: Self-Supervised Learning of Local Visual Features. In *NeurIPS*.

Bilot, T.; Madhoun, N. E.; Agha, K. A.; and Zouaoui, A. 2023. Graph Neural Networks for Intrusion Detection: A Survey. *IEEE Access*, 11: 49114–49139.

Chai, Z.; You, S.; Yang, Y.; Pu, S.; Xu, J.; Cai, H.; and Jiang, W. 2022. Can Abnormality be Detected by Graph Neural Networks? In *IJCAI*, 1945–1951. ijcai.org.

Cresci, S.; Pietro, R. D.; Petrocchi, M.; Spognardi, A.; and Tesconi, M. 2016. DNA-Inspired Online Behavioral Modeling and Its Application to Spambot Detection. *IEEE Intell. Syst.*, 31(5): 58–64.

Feng, S.; Tan, Z.; Li, R.; and Luo, M. 2022a. Heterogeneity-Aware Twitter Bot Detection with Relational Graph Transformers. In *AAAI*, 3977–3985. AAAI Press.

Feng, S.; Tan, Z.; Wan, H.; Wang, N.; Chen, Z.; Zhang, B.; Zheng, Q.; Zhang, W.; Lei, Z.; Yang, S.; Feng, X.; Zhang, Q.; Wang, H.; Liu, Y.; Bai, Y.; Wang, H.; Cai, Z.; Wang, Y.; Zheng, L.; Ma, Z.; Li, J.; and Luo, M. 2022b. TwiBot-22: Towards Graph-Based Twitter Bot Detection. In *NeurIPS*.

Feng, S.; Wan, H.; Wang, N.; Li, J.; and Luo, M. 2021a. SATAR: A Self-supervised Approach to Twitter Account Representation Learning and its Application in Bot Detection. In *CIKM*, 3808–3817. ACM.

Feng, S.; Wan, H.; Wang, N.; Li, J.; and Luo, M. 2021b. TwiBot-20: A Comprehensive Twitter Bot Detection Benchmark. In *CIKM*, 4485–4494. ACM.

Feng, S.; Wan, H.; Wang, N.; and Luo, M. 2021c. BotRGCN: Twitter bot detection with relational graph convolutional networks. In *ASONAM*, 236–239. ACM.

Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *EMNLP (1)*, 6894–6910. Association for Computational Linguistics.

Ghanem, B.; Buscaldi, D.; and Rosso, P. 2019. TexTrolls: Identifying Russian Trolls on Twitter from a Textual Perspective. *CoRR*, abs/1910.01340.

Jim; and Ann. 2020. twitter topics – the mega list. https://inboundfound.com/twitter-topics-list/. Accessed: 2023-08-08.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR (Poster)*. OpenReview.net.

Liu, Y.; Han, T.; Ma, S.; Zhang, J.; Yang, Y.; Tian, J.; He, H.; Li, A.; He, M.; Liu, Z.; Wu, Z.; Zhu, D.; Li, X.; Qiang, N.; Shen, D.; Liu, T.; and Ge, B. 2023. Summary of ChatGPT/GPT-4 Research and Perspective Towards the Future of Large Language Models. *CoRR*, abs/2304.01852.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *ICLR (Poster)*. OpenReview.net.

Lv, Q.; Ding, M.; Liu, Q.; Chen, Y.; Feng, W.; He, S.; Zhou, C.; Jiang, J.; Dong, Y.; and Tang, J. 2021. Are we really making much progress?: Revisiting, benchmarking and refining heterogeneous graph neural networks. In *KDD*, 1150–1160. ACM.

Miller, Z.; Dickinson, B.; Deitrick, W.; Hu, W.; and Wang, A. H. 2014. Twitter spammer detection using data stream clustering. *Inf. Sci.*, 260: 64–73.

Varol, O.; Ferrara, E.; Davis, C. A.; Menczer, F.; and Flammini, A. 2017. Online Human-Bot Interactions: Detection, Estimation, and Characterization. In *ICWSM*, 280–289. AAAI Press.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *ICLR (Poster)*. OpenReview.net.

Wang, S.; Xu, X.; Zhang, X.; Wang, Y.; and Song, W. 2022. Veracity-aware and Event-driven Personalized News Recommendation for Fake News Mitigation. In *WWW*, 3673–3684. ACM.

Wang, W.; and Peng, W. 2022. Team Yao at Factify 2022: Utilizing Pre-trained Models and Co-attention Networks for Multi-Modal Fact Verification (short paper). In *DE-FACTIFY@AAAI*, volume 3199 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Wu, D.; Zhang, J.; and Huang, X. 2023. Chain of Thought Prompting Elicits Knowledge Augmentation. In *ACL (Findings)*, 6519–6534. Association for Computational Linguistics.

Wu, J.; Zhao, C.; Yu, T.; Li, J.; and Li, S. 2021. Clustering of Conversational Bandits for User Preference Learning and Elicitation. In *CIKM*, 2129–2139. ACM.

Zannettou, S.; Caulfield, T.; Cristofaro, E. D.; Sirivianos, M.; Stringhini, G.; and Blackburn, J. 2019. Disinformation Warfare: Understanding State-Sponsored Trolls on Twitter and Their Influence on the Web. In *WWW (Companion Volume)*, 218–226. ACM.

Zhou, X.; Shu, K.; Phoha, V. V.; Liu, H.; and Zafarani, R. 2022. "This is Fake! Shared it by Mistake": Assessing the Intent of Fake News Spreaders. In *WWW*, 3685–3694. ACM.