GraphBERT: Bridging Graph and Text for Malicious Behavior Detection on Social Media

Jiele Wu, Chunhui Zhang, Zheyuan Liu, Erchi Zhang, Steven Wilson, and Chuxu Zhang*
Brandeis University, MA, USA
{*jielewu, chunhuizhang, zheyuanliu, erchizhang, stevenwilson, chuxuzhang*}*@brandeis.edu*, **Corresponding author*

***Abstract*—The development of social media (e.g., Twitter) allows users to make speeches with low cost and broad influence. Thus, social media has become a perfect place for users' malicious behaviors like committing hate crimes, spreading toxic information, abetting crimes, etc. Malicious behaviors are covert and widespread, with potential relevance regarding topic, person, place, and so on. Therefore, it is necessary to develop novel techniques to detect and disrupt malicious behavior on social media effectively. Previous research has shown promising results in extracting semantic text (speech) representation using natural language processing methods. Yet the latent relation between speeches and the connection between users behind speeches is rarely explored. In light of this, we propose a holistic model named Graph adaption BERT (GraphBERT) to detect malicious behaviors on Twitter with both semantic and relational information. Specifically, we first present a novel and a large-scale corpus of tweet data to benefit both graph-based and language-based malicious behavior detection research. Then, we design a novel model GraphBERT to learn comprehensive tweet and user representation with the integration of both semantic information encoded by transformers (i.e., BERT) and relational information encoded by graph neural network. GraphBERT further leverages a weight adaption BERT module implemented between transformer layers to refine tweet embedding using relational information for malicious tweet classification. Finally, the adapted tweet embedding is used with the initial tweet representation to generate user embedding for malicious user detection. The extensive experiments on the collected Twitter data show that our model outperforms the state-of-the-art baseline methods for both tasks (i.e., malicious tweet classification and malicious user detection).**

***Index Terms*—Twitter, Graph neural network, BERT, Malicious behavior detection**

I. Introduction

Social media platforms (e.g., Twitter) provide people a place to make free speeches with broad influences [1]. These free and direct connections between people have significantly benefited society but simultaneously become the cradle of malicious behaviors. Among those platforms, online social media (e.g., Twitter) have become a major vehicle for online malicious behaviors, including committing hate crimes [2], making aggressive speech, spreading rumors and misinformation [3], online drug trafficking [4], etc. While we cherish the right of free speech, a number of users get irritated or confused by the spread of inflamed discussions and intentional inducement. As a result, they begin to have malicious behaviors which is an abuse of the freedom.

Under people's concern, online malicious behaviors have been rapidly recognized as a serious problem by the authorities of many countries [5]. Take the most concerning hate crime as an example, efforts have been made to reduce hate speech by abolishing anonymity [5], [6], proposing initiatives [7], studying hate groups and forums [8], and so on. Although these methods have positive impacts on diminishing online hate speech, most of them rely on human intervention. Hence, there is a solid motivation to produce automatic detection methods for online malicious behaviors.

Machine learning methods have been designed for automatic detection based on social media data. Among those methods, Natural Language Processing (NLP) methods have been widely used to detect and reduce online malicious behaviors, especially hateful speech [9]. By obtaining features that can represent the semantic information of a given sentence using artificial neural networks, NLP methods provide convincing results on hate speech detection [10] using machine learning techniques to classify text as hate speech. Fine-tunes of the widely used pre-trained language model from Transformers [11] (e.g., Bidirectional Encoder Representations from Transformers, which also is called BERT) has shown convincing result on specific kind of hate speech detection [11], [12], like racial bias [13]. Although promising progress has been made on hate speech detection, there are two major points that previous researchers have not considered.

The first point is that semantic information is not the only feature in speeches. The speech generated by malicious users often share related characteristics like the same users, topics, named entities, and non-semantic information such as mentions and hashtags. These latent relations provide links between different users and speeches, which can be learned with graph structure. For example, using graph neural network models [14], tweet and user embedding with relational information can be encoded. With the combination of semantic information and user's relational information, a more comprehensive model can be designed for malicious behavior detection. Secondly, while paying too much attention to hate speech detection, the researchers have overlooked the harmfulness of those spiteful users behind the malicious behaviors. Specifically, most of the malicious users may have potentially shared the same idiosyncrasies like unstable emotions, being easily irritated, speaking or acting on hearsay, etc. Since these users are the producers of those malicious tweets, the problem will be largely alleviated by banning those users. Therefore, developing a novel and effective malicious behavior detection technique is essential to solve the problem from the origin.

To validate the above analysis and solve the problem of malicious behavior detection on social media, we first propose a large dataset collected from 13,351 twitter users and 91,500 tweets. Later, we annotate each user and tweet respectively based on their malicious scales. Users are labeled as normal or malicious; tweets are labeled as normal, weakly malicious or strongly malicious. Since graph neural networks (GNNs) are proved to be effective in encoding relational information [15], we construct graph structure based on tweets, users, topics, mentions, hashtags, and other key information. Based on the constructed graph, we propose a novel model called **GraphBERT**, which fuses semantic and relational information to learn both tweet and user representation for malicious behavior detection. GraphBERT consists of three major parts: node feature encoding, weight adaption network, and semantic graph attention network. The node feature encoding module generates user and tweet embedding by GNN for downstream tasks. The weight adaption network utilizes user and tweet embedding from GNN to refine the middle layer embedding of words and sentences according to relational information. Unlike the commonly used early-fusion or late-fusion methods [16], weight adaption network modifies the internal word and sentence embedding of BERT model by adapting semantic internal features to the nonverbal user relational features and finally generates tweet embedding for malicious tweet classification. Based on the previous step's embeddings, we propose a deep fusion network to combine the initial pre-trained tweet feature with the newly adapted feature to generate final user embedding for malicious user detection.

To summarize, our major contributions of this paper are as follows:

- We collect and annotate a large-scale Twitter dataset for malicious behavior detection. Moreover, we further design GraphBERT, a novel model which combines BERT with GNN to learn semantic and relational representation for malicious behavior detection.
- We conduct extensive experiments to evaluate the performance of our proposed model. The result exhibits the superiority of our method in comparison to both Graph and NLP baselines.
- To the best of our knowledge, this is the first attempt at malicious behavior detection on social media using graph-based relation information and transformer-based semantic information.

II. Related Work

A. Machine Learning on Tweets

Twitter is a micro-blogging platform where users can post messages named "tweet" to their friends [17]. It has provided an enormous amount of datasets in name entity recognition [18], sentiment analysis [19], dialect classification [20], and so on. Based on those tweet datasets in multiple research areas, machine learning methods in sentiment analysis [21], recommendation system [22], data annotation system [23] and recently covid related studies [24] emerge and show convincing results. Among those areas, hate speech detection is becoming an increasingly popular research field in recent years. To deal with different types of hate speech, the researchers have presented different methods under disparate backgrounds [1], [11], [25], [26]. Since the core purpose of a hate speech detection task is to find the information owner and stop him from contaminating the internet environment, previous research may have underestimated the importance of seeking malicious users on social media. In this paper, we aim to solve the malicious user problem on Twitter.

B. Language Embedding Models

Learning word representations from large corpora has been a core part of natural language processing (NLP). Language models convert the natural language to features for different downstream applications. Bag-of-words [27] is the first model using a fixed length vector generated by clustering algorithms to represent text. Glove [28] and Word2Vec [29] are trained by machine learning methods and applied for many NLP research. Contextual language representation models trained on large corpora like GPT [30] and ELMo [31] present convincing results on multiple NLP tasks. Base on contextual representation, BERT [32] captures bi-directional context information using multi-transformer encoders with multi-head attention. It is a widely used model pre-trained on a large cross-domain unlabeled corpus. BERT and its improvement methods like XLNet [33] or RoBERTa [34] show breakthrough results in many NLP tasks. Since then, using pre-trained language models on a large amount of data and fine-tuning downstream tasks has become a new paradigm for natural language processing. Furthermore, this new paradigm has shown outstanding results in many research fields. In this paper, we present a novel weight adaption model applied between transformer layers of BERT model to generate tweet embedding with the integration of relational graph information.

C. Graph Neural Networks

With the advance of deep learning, graph neural networks (GNNs) have attracted significant attention [35]–[39]. Unlike the language models mentioned above, GNNs consider data as a graph structure and can aggregate feature information from node's local neighbors via neural networks [14], [36]. For example, Graph Convolutional Network (GCN) [14] proposes a graph-based convolution neural network to propagate embeddings via interaction between nodes in the graph. GAT [40] employs the self-attention method to measure the impacts of different neighbors and combines their impacts to obtain node embeddings. GraphSAGE [36] utilizes neural networks like LSTM, to sample and aggregate neighbors' feature information. GEM [41] for malicious accounts detection has been proposed to obtain better node embeddings for specific tasks. Encouraged by data augmentation for semi-supervised learning in the computer vision research field, graph contrastive learning [42]–[46] emerges to generate representations invariant to specialized perturbations for diverse graph-structured data. Our method uses graph attention network and graphs

contrastive learning respectively to generate representations of users and tweets for fine-tuning of pre-trained BERT.

III. Preliminary

In this section, we first introduce the definition of two kinds of malicious behavior detections: malicious user detection and malicious tweet classification. Later, we elaborate on the collection and annotating methods of the dataset.

A. Problem Definition

Let $G = \{V, E, X\}$ denote a graph data of Twitter data, where V is the set of nodes, $E \in V \times V$ is the set of edges, and X is the node feature set. The nodes include users, tweets, and multiple types of other entities. An edge can be regarded as any type of relationship between two nodes. Given the features X_U of all user nodes $U = (u_1, \ldots, u_N)$, where N is the total number of users. We use $Y_U = (y_{u_1}, \ldots, y_{u_N})$ to represent labels for all users, where y_i = 1 denotes malicious user and y_i = 0 refers to normal user. Each user U_i is linked to a set of tweets T_{u_i}. Semantic information is captured by NLP methods based on text set $T = (t_1, \ldots, t_M)$ and their labels $Y_T = (y_{t_1}, \ldots, y_{t_M})$, where M is the total number of text (e.g., tweet) data. that can combine semantic information using NLP methods and tweet labels with a graph learning model to achieve better detection performance. Formally, the two problems are defined as follows.

Definition 1: **Malicious user detection.** Given a set of Twitter users data $U = (u_1, \ldots, u_N)$. Each user u_i has a corresponding label $y_i = 0$ or 1 (0 for normal and 1 for malicious). The task is to develop a machine learning model $f_\theta : U \to Y_U$ to classify users into different categories, where θ are model parameters.

Definition 2: **Malicious tweet classification.** Given a tweet data set denoted as $T = (t_1, \ldots, t_M)$. Each text t_i has a corresponding category label $y_i = 0$or1 or 2 (0 for normal, 1 for weakly malicious, 2 for strongly malicious). We aim to build a machine learning model $f_\theta : T \to Y_T$ to classify tweets into different categories.

B. Data Collection

We collect more than twenty million tweets from over one million Twitter users. To clean tweets that are useful for our research, we select the data by following constraints:

- We do not consider retweeting in the tweets selection process.
- The length of the natural language words is no less than five and no more than twenty.
- The number of tweets each selected user posted is between five and twenty.
- Under the three constraints above, all data are randomly selected without leaning toward certain groups or topics.

The natural language words in the second constraint refer to words with semantic meanings. Other words or symbols like punctuation marks, emojis, emoticons, and website addresses are not considered in length. After noisy tweets elimination and conditional filtering according to the length of symbols with linguistic meaning, we finally select 91,500 tweets from 13,351 users.

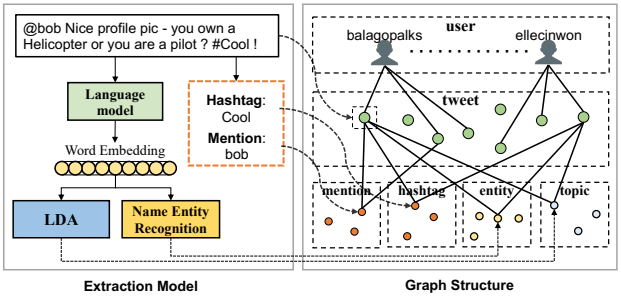

Fig. 1. The illustration of graph generation on Twitter data. The extraction model and example of node extraction from the tweet are shown on the left. The graph structure contains different connections among nodes of users, tweets, mentions, hashtags, topics, and name entities.

C. Data Annotation

We conduct data annotation for both users and tweets in the dataset. For the user label, each annotator considers each user's tweets comprehensively and chooses a label from 0 (normal) and 1 (malicious). Since the definition of malicious user is not universally accepted, we consider malicious users as including at least one of the following behaviors:

- Promote violence, directly attack, threaten or insult other people based on race, ethnicity, national origin, sexual orientation, gender, gender identity, religious affiliation, age, disability, serious disease, or other common characteristics.
- Spread toxic information like obscene, narcotics, terrorism, violence, or abetting crime.
- Use obscene or racial discrimination language for complaining or expressing negative information, which should be prohibited.

For malicious tweet annotation, every annotator decides its malicious state as one of the following: 0 (normal), 1 (weakly malicious) or 2 (strongly malicious). In detail, we define those two special states as follows:

- Strongly malicious: Tweets that attack, threaten or insult a person or group based on national origin, ethnicity, color, religion, gender, gender identity, sexual orientation, or disability. Tweets spread obscene, narcotics, terrorism, violence, or abetting crime.
- Weakly malicious: The tweets that use obscene or racial discrimination language for complaining or expressing vicious information.

For each tweet, we have five independent, well-trained annotators to make annotations. The final label is decided by the majority vote of the five labels. Furthermore, we provide the target labels for all malicious tweets indicating the category of its victim. Furthermore, we propose sentiment labels, including activation and polarity labels for 15,000 tweets. We only use malicious user and tweet annotation in the following model and experiment sections.

TABLE I
STATISTICS OF DATASET.

Component	Group	Type	Number
Node	Info	user	13,351
		tweet	91,500
		topic	80
	Relation	mention	403
		hashtag	493
	Entity	GPE	94
		LOC	7
		ORG	34
		NORP	47
		PERSON	57
	Total	-	106,066
Edge	-	-	215,588

D. Graph Construction

We introduce heterogeneous graph [38], [47]–[49] to represent the collected data. As shown in Figure 1, nodes in the graph contain tweets, users, mentions, hashtags, topics, and entities. We first extract mention (@) and hashtag (#) from raw tweets. Next, we perform the sifting process and filter out the less involved ones. Then we clean all raw tweets and categorize them using LDA model with perplexity criterion [50]. The LDA model clusters the latent semantic structure of all tweets in the document. Therefore, the model provides all tweets with the most likely topics and each topic consists of ten words where each word has its corresponding weight. For name entity generation, we use Name Entity Recognition (NER) method [51] in NLP package 'SpaCy' to get all entities in the tweet. We select the five most meaningful entity types: location (LOC), geopolitical entity (GPE), organization (ORG), person (PER), and nationalities, religious or political groups (NORP). The entities are only recorded as a node in the graph for each entity type when it appears more than ten times. The details of statistics are reported in Table I. Overall, the constructed graph includes 106,066 nodes with 215,588 edges.

E. Challenges of Dataset

Here we listed some challenges of our collected dataset for the research of malicious behavior detection: (1) Our dataset consists of 90,212 negative and 1,288 positive tweets, including 951 weakly harmful tweets and 337 strongly harmful tweets. Besides, our dataset contains 12,293 normal and 1,058 malicious users, so we have to deal with extremely imbalanced data in model training. (2) The tweets in our dataset are randomly selected from Twitter without manually selecting. Each tweet could contain an advertisement or repeated information that may befuddle the machine learning models. (3) Since non-linguistic expressions like abbreviation, emojis, emoticons, and so on are frequently used on Twitter, it is hard for existing language models to encode this information. (4) Previous research may only focus on a certain kind of hate speech/behavior like racial discrimination or a specific topic like Covid-19. In our dataset, we consider hate speech as a whole and aim to detect malicious users whom the platform should prohibit.

IV. THE PROPOSED MODEL

In this section, we describe our proposed GraphBERT model, which contains three major components: node feature encoding, weight adaption network, and semantic graph attention network for malicious behavior detection. The node feature encoding is shown in Fig. 2 (a) where we first use pre-trained BERT to get the initial representation X_t for each tweet node. The initial representations of other nodes are generated based on the initial tweet representation which is specifically introduced in Section. IV-A. After acquiring the initial representation for all nodes, we use a graph attention network to obtain user embedding Z_u and leverage the graph contrastive learning method to generate tweet embedding Z_t. Based on the user and tweet embeddings, we further introduce a weight adaption network applied in the middle layers of BERT (shown in Fig. 2 (b) and (c)) to refine the middle embeddings between BERT layers. The weight adaption network contains word-level and sentence-level adaption, which can be employed separately or simultaneously. The output embedding Z_{adp} of weight adaption BERT is used for malicious tweet classification tasks. Finally, we employ a semantic graph attention network (shown in Fig. 3) to generate the final user representation for malicious user detection.

A. Node Feature Encoding

The constructed graph contains multiple nodes, including tweets, users, mentions, hashtags, topics, and names. As shown in 2(a), the initial feature of tweet nodes is acquired from the pre-trained BERT model. For each tweet node, we use the average feature of all word tokens which is a 768-dimensional vector as the initial tweet representation. The feature X_{u_i} for user i with tweet set T_{u_i} is initialized as:

$$X_{u_i} = \frac{1}{k} \sum_{t \in T_{u_i}} \mathrm{BERT}(t), \tag{1}$$

where k is the number of tweet in set T_{u_i}. For topic node, we first use LDA model to generate topics. We minimize the perplexity of LDA model to determine the best number of topics. Since every single topic in LDA model contains words and weight for each word, we generate the topic feature by calculating the weighted average of word vectors in each topic. As for mention, hashtag, and name entity nodes, their features are represented by the average of their word features.

After getting the initial features for all nodes in the given graph $G = \{V, E, X\}$, we apply graph attention network [40] to generate the user's embedding Z_{u_i} as follows:

$$Z_u = \sigma[W_u \cdot \mathrm{GAT}(V, E, X)], \tag{2}$$

where W_u is the trainable parameters, σ is the ReLU activation function. We further introduce graph contrastive learning method with data augmentation to generate tweet embedding

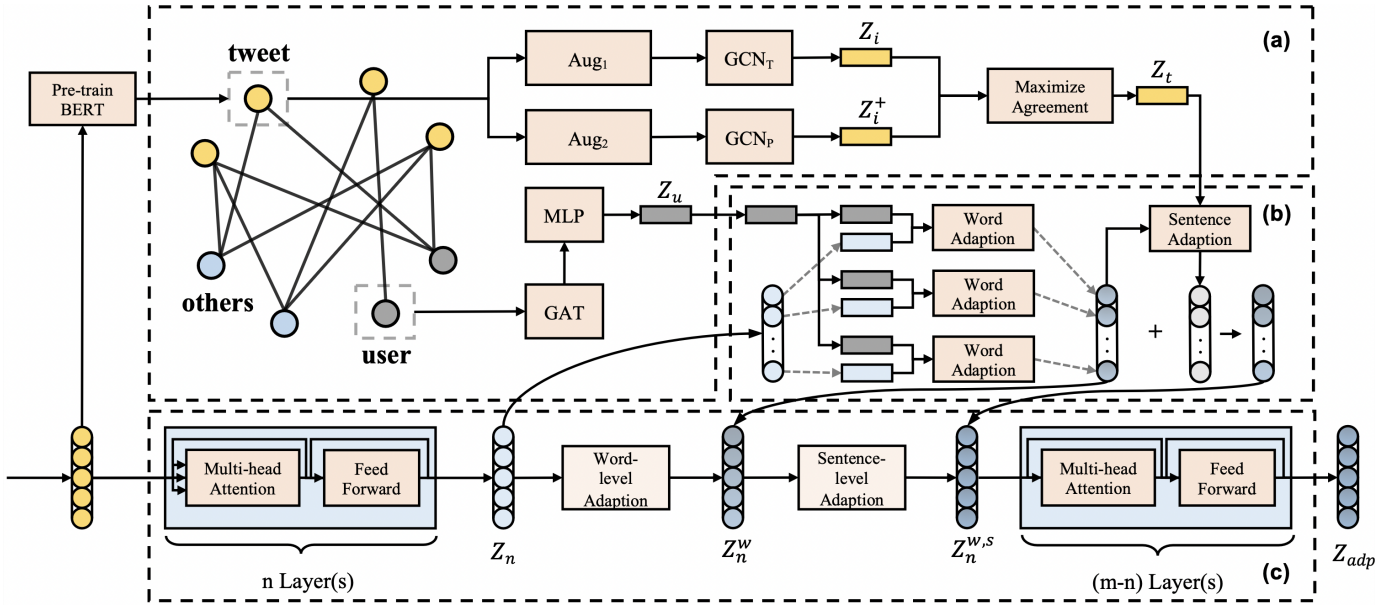

Fig. 2. The illustrations of node feature encoding and weight adaption network based on the constructed graph of Twitter: (a) The node feature encoding module for the user and tweet embedding generation; (b) weight adaption network applied between the transformer layers of BERT to refine the middle embedding for generating better tweet embedding; (c) The backbone of BERT network with m layers of BERT encoder(Transformer).

for later models. The loss to learn node embedding Z of input graph G can be formulated as follows:

$$\mathcal{L}_{CL}(G', G'') = -\sum_{i=1}^{N} \log \frac{\exp\left(\text{sim}\left(Z_i, Z_i^+\right)\right)}{\sum_{j=1, j\neq i}^{N} \exp\left(\text{sim}\left(Z_i, Z_j\right)\right)}, \quad (3)$$

where we apply Z_i and Z_i^+ to represent node i's embeddings, and they are calculated as $Z_i = \text{GCN}_\text{T}(\text{Aug}_1(G))_i$ and $Z_i^+ = \text{GCN}_\text{P}(\text{Aug}_2(G))_i$. $\text{sim}(\cdot,\cdot)$ denotes cosine similarity. $\exp(\cdot)$ denotes exponential function which uses e as base. N means the number of nodes in G. $\text{Aug}_1(\cdot)$ and $\text{Aug}_2(\cdot)$ are the two random augmentations to generate two views G' and G'' for an original input G, $\text{GCN}_\text{T}(\cdot)$ is the target branch and $\text{GCN}_\text{P}(\cdot)$ is the predict branch for processing the input node. Z_i is similar to Z_i^+ since both embeddings are for node i but dissimilar to all other embeddings Z_j for node $j \neq i$. The $\mathcal{L}_{CL}$ maximizes the agreement between Z_i^+ and Z_i which both transformed from one original node. The tweet embedding Z_t generated by the trained graph contrastive model is used for subsequent models.

B. Weight Adaption Network

To effectively combine semantic and relational information, we introduce a weight adaption network for refining tweet embedding. Encouraged by the work of multi-model sentiment analysis that nonverbal behaviors can have an impact on the meaning of words [52]. We introduce a weight adaption network with both word-level and sentence-level adaption to modify the middle layer word embedding of BERT based on the user and tweet embeddings generated in the previous step. In the semantic space, the embedding of each word represents a position point in this latent multi-dimension space. Without the influence of other information, the word is simply put into the space according to the word's own linguistic meaning (non-contextual). From the perspective of words, the changing of word meaning in different contexts can be considered as the influence between semantic information. Yet non-semantic information can also impact the meaning of words and, at the same time, change its position in the semantic space. Our weight adaption network learns the impact of user characteristics on the meaning of the user's tweets and further refines the tweet embeddings.

As shown in Fig. 2 (b) and Fig. 2 (c), the weight adaption network is applied between different middle layers of BERT model and it consists of two levels of adaption: word-level adaptation and sentence-level adaption. The word-level adaption and sentence-level adaption can work either independently or to be used together after any BERT encoder layers. Firstly, to achieve word embedding adaption, for tweet t_j from user u_i with P word tokens, the weight adaption network receives both middle tweet embedding $Z_{n,j}$ generated from tweet t_j after n BERT encoder layers and user embedding Z_{u_i} generated by the graph attention network in the previous subsection. The word adaption vector K_{adp}^{word} is formulated as:

$$K_{adp}^{word} = \Big\|_{p=1}^{P} \sigma[W_{n,u}^{p,adp}(Z_{n,j}^{p} || Z_{u_i})], \quad (4)$$

where $||$ denotes the concatenation operation, $Z_{n,j}^{p}$ is the embedding of the p^{th} word's in tweet embedding $Z_{n,j}$, $W_{n,u}^{p,adp}$ is the trainable parameters, σ is the ReLU activation function. Based on the user embedding Z_{u_i}, we generate adaption weight K_{wgt}^{word} as follows:

$$K_{wgt}^{word} = \Big\|_{p=1}^{P} \sigma(W_{n,u}^{p,wgt} \cdot Z_{u_i}), \quad (5)$$

where $W_{n,u}^{p,wgt}$ is the trainable parameters. Finally, we refine the word embedding $Z_{n,j}$ of tweet t_j by the multiplication of adaption vector and adaption weight vector to get the word-adapted tweet embedding $Z_{n,j}^{w}$:

$$Z_{n,j}^{w} = Z_{n,j} + \alpha(K_{adp}^{word} \cdot K_{wgt}^{word}), \tag{6}$$

where the trade-off weight α is defined as follows:

$$\alpha = \min[\frac{\text{norm}(Z_{n,j})}{\text{norm}(K_{adp}^{word} \cdot K_{wgt}^{word})}, \alpha], \tag{7}$$

where norm$(\cdot)$ represents the L2 norm. The features are finally fed to the downstream layer after a dropout layer and a batch normalization layer.

To further integrate information from tweet embedding Z_t encoded by GCN into word embedding, we introduce a sentence-level weight adaption network (Fig. 2 (b)). For the training on BERT, a [CLS] token is stipulated to be added at the beginning of the sentence. Since [CLS] is a symbol without obvious semantic information, compared with other words in the text, this symbol can evenly integrate the information of each word in the text [32]. For tweet t_j, we combine word feature $Z_{n,j,[CLS]}$ of its [CLS] token with Z_{t_j} to obtain the adaption vector K_{adp}^{sent}:

$$K_{adp}^{sent} = \sigma[W_{n,t}^{adp} \cdot (Z_{n,j,[CLS]} || Z_{t_j})], \tag{8}$$

where $W_{n,t}^{adp}$ is the trainable parameters. The weight K_{wgt}^{sent} for sentence adaptation vector is generated as:

$$K_{wgt}^{sent} = \sigma(W_{n,t}^{wgt} \cdot Z_{t_j}). \tag{9}$$

We finally formulate the sentence embedding by:

$$Z_{n,j}^{s} = \mathop{||}_{p=1}^{P} [Z_{n,j}^{p} + \beta(K_{adp}^{sent} \cdot K_{wgt}^{sent})], \tag{10}$$

$$\beta = \min(\frac{\text{norm}(Z_{n,j,[CLS]})}{\text{norm}(K_{adp}^{sent} \cdot K_{wgt}^{sent})}, \beta), \tag{11}$$

where β is a hyper-parameter. Since the two parts of the weight adaption network are applied in the middle of the BERT, the weight adaption BERT model can be formulated as:

$$Z^{adp} = \text{Layer}_m(\text{Layer}_{...}(\text{Layer}_1(Z_n, Z_u, Z_t, W_m, S_m))), \tag{12}$$

where $\text{Layer}_m(\cdot)$ refers to m-th BERT encoder layer, n refers to the total layer number. W_m and S_m are boolean vectors with length m denoting the position of adapting word-level adaption and sentence-level adaption. In this work, the layer number m of BERT is set to 12. Furthermore, if the word-level adaption and sentence-level adaption are applied after the same BERT encoder layer, the middle embedding Z_n is first sent to the word-level adaption network with output Z_n^w and then sent to sentence-level adaption network with output $Z_n^{w,s}$. Otherwise, the middle embedding Z_n is sent to either the adaptation network or the next layer if both adaption network is not applied. The final output embedding Z_{adp} is used for malicious tweet prediction as follows:

$$\hat{y_t} = \text{softmax}[\sigma(Z_{adp} \cdot W_1) \cdot W_2]. \tag{13}$$

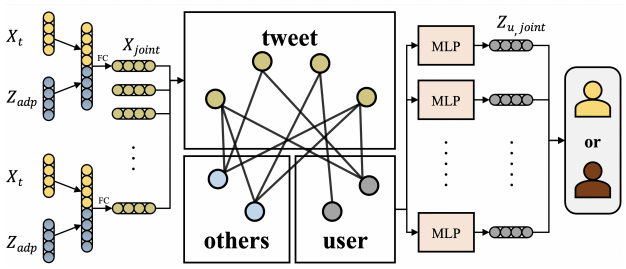

Fig. 3. The illustration of semantic graph attention network for joint representation learning and user embedding generation for malicious user detection.

We adopt cross-entropy loss overall labeled tweets as the final objective function:

$$L_t = -\sum_{t \in T} y_t \log \hat{y_t}, \tag{14}$$

where T is the tweet set, $\hat{y_t}$ is the prediction label of the tweet and y_t is the ground truth label of the tweet. The output embedding Z_{adp} is further sent to the later semantic graph attention network module for malicious user detection.

C. Semantic Graph Attention Network

For malicious user detection, we propose a semantic graph attention neural network based on the output tweet embedding Z_{adp} of the previous subsection (shown in Fig. 3). In this module, the initial representation X_t and the BERT adaption embedding Z_{adp} are acquired by pre-trained BERT and weight adaption BERT respectively. We combine the initial representation with the BERT adaption embedding to form the new representation of the tweet node:

$$X_{joint} = \begin{cases} \sigma[W_{joint} \cdot (Z_{adp_j} || X_{t_j})] & V_j \in T \\ X_j & V_j \notin T \end{cases}, \tag{15}$$

where X_{joint} is the new node representation, W_{joint} is the trainable parameters, $||$ denotes concatenation, σ is the ReLU activation function, T is the set of all tweet nodes and $V_j \in V$. Based on the joint embedding X_{joint}, we train a graph attention network with graph data $G = \{V, E, X_{joint}\}$ to get the joint user embedding $Z_{u,joint}$ and feed it to a multi-layer perceptron for malicious user detection. We adopt the focal loss [53] over all labeled users as the objective function:

$$FL_t = -\sum_{u \in U} \alpha \cdot y_u^{\gamma} \log \hat{y_u}, \tag{16}$$

where U is the user set, $\hat{y_u}$ is the prediction of the user and y_u is the ground truth label of the user. α and γ are set as 0.1 and 3 respectively.

V. Experiments

In this section, we conduct extensive experiments to verify the superiority of our model compared with baseline methods for two tasks: (i) For *malicious user detection*, we conduct experiments on three different amounts of training data to indicate the robustness of our model facing the different data

sizes. (ii) For *malicious tweet classification*, we implement binary and triple classification with multiple baselines. Since GraphBERT consists of several parts, to demonstrate the effectiveness of each component, we also introduce ablation studies. We further propose a few-shot experiment with a small number of training tweets to validate the performance of GraphBERT with few training data. Finally, we perform the embedding visualization of GraphBERT compared to baseline methods.

A. Baseline Methods

We compare our model with multiple baselines including (i) graph neural network baselines for malicious user detection and (ii) natural language processing baselines for malicious tweet classification. We briefly review these baselines in the following three types of models:

Common Baselines. **(1) DNN:** For the deep neural network (DNN) baseline in our experiment, we apply a 3-layer fully connected network with ReLU activation function. **(2) LSTM:** For all Long Short-Term Memory (LSTM) [54] baseline in our experiment, we apply a 2-layer bi-LSTM with ReLU activation function and a linear classification layer.

Malicious User Detection Baselines. **(1) GCN:** Graph Convolution Network (GCN) [14] is a convolution neural network directly used on graph data to extract spatial features of the topological graph. **(2) GAT:** Graph Attention network (GAT) [40] improves GCN by extracting neighbors' information through masked self-attentional layers. Graph attention networks assign different weights to each neighbor node to measure the importance of each neighbor. **(3) GCL:** Graph contrastive Learning (GCL) [43] learns the unsupervised representation of a graph by maximizing the consistency of features from different augmentations. **(4) GraphSAGE:** GraphSAGE [36] is a graph neural network model which generates embedding by sampling and fusion features of local neighbors.

Malicious Tweet Classification Baselines. **(1) Word2Vec:** Word2Vec [55] is a neural network model used to generate word embeddings. We use the Word2Vec embeddings as input and a combination of LSTM with fully connected layers as the classifier. **(2) BERT:** BERT [32] is bidirectional encoder representations generation model composed of multiple transformer layer [56]. It has been proven to be beneficial for many NLP tasks and has displayed promising results. **(3) Fine-tuning BERT:** Fine-tuning BERT has been proven to outperform other deep learning baselines like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) [11] in other fields. We adopt two state-of-the-art fine-tuning BERT in hate speech detection as baselines. **(4) XLNet:** XLNet [33] is an improvement of BERT. As an autoregressive language model, XLNet does not rely on data corruption and eliminates the independence assumption made in BERT. **(5) Roberta:** Roberta [34] is a robust BERT with a large number of model parameters, batch size, and data. The model also applies dynamic masking to improve performance.

B. Experimental Setup

Dataset Splits. For malicious user detection task, we set up three different data splits (60% Training Data, 40% Training Data, and only 20% Training Data). For each split, we randomly select 60%, 40%, and 20% of all tweets as the training set respectively. The rest of the data are split into the validation set and testing set evenly. All experiments on a single split are trained by the same set of training data and tested by the same set of testing data. For malicious tweet classification task, the tweet from training users in 60% training data split is set as the training set, and the validation set and testing set are set in the same way.

Reproducible Setting. For all baselines and our method, we apply grid search to ensure hyperparameters. To make the result more stable and reliable, each result we perform is the mean result of five experiments with same parameters. As for the layer of word adaption and sentence adaption, we construct word adaption after the 1st BERT encoder layer and sentence adaption after the 11th BERT encoder layer. We use Adam as an optimizer with 5e-4 learning rate and 1e-4 weight decay. For models of malicious tweet classification, we set the batch size to 32 and the learning rate to 5e-6. As for the loss function, we use focal loss [53] in graph attention network training with the α set as 0.1, β set as 3 and reduction set as *mean*. For all methods, we select the best model according to the sum of balance accuracy and F1-score on the validation set. An early stop of 5 for transformer-based models and 100 for graph-based models is set to avoid overfitting.

For the proposed node feature encoding, we adopt two layers of GAT to extract user embedding and we apply a contrastive learning model with a two-layer GCN. The hidden size of the two GAT layers is 256 and 64 respectively with a batch normalization layer and a 0.5 dropout applied in the middle to avoid overfitting. For the contrastive learning GCN model, the hidden dimension of two GCN layers is both 32, and a 0.2 dropout is applied after each layer. For hyper-parameter α and β in the weight adaption network, we set both to 1.0.

Evaluation Metrics. For all experiments including three data splits of malicious user detection and binary and triple malicious tweet classification, we report balance accuracy, accuracy, and F1-score. Balance accuracy is the average of recall obtained in each class. Higher values mean better performance for all metrics.

Treatment of imbalanced data. Due to the imbalance between positive and negative data samples, we apply multiple methods to improve the training process and make the experimental results more reliable. First, we conduct experiments on both binary classification and multi-label classification. Since our dataset has three kinds of labels, not harmful, weakly harmful, and strongly harmful, our binary classification experiment merges the weakly harmful and strongly harmful as positive samples. Even so, the imbalance of positive and negative samples is still significant. To mitigate the influence of imbalanced data, we adopt several strategies. First, focal

TABLE II
EXPERIMENT RESULTS FOR MALICIOUS USER DETECTION WITH DIFFERENT TRAINING RATIOS.

Model	60% Training Data			40% Training Data			20% Training Data		
	balance Acc.	Accuracy	F1-score	balance Acc.	Accuracy	F1-score	balance Acc.	Accuracy	F1-score
GCN	66.08	86.22	62.48	65.57	86.50	62.21	62.93	87.11	62.12
GAT	69.09	88.88	63.04	67.78	87.94	63.44	65.72	86.84	62.13
SAGE	63.75	88.49	63.62	63.51	88.11	62.31	63.03	85.69	62.68
GCN+DNN	65.81	87.59	63.69	65.05	87.39	62.79	64.38	85.58	62.71
GCN+GCL+DNN	62.71	78.96	57.63	62.58	76.81	54.83	62.97	73.52	54.55
GCN+GCL+SVM	59.40	88.00	56.44	59.72	87.68	56.86	58.63	87.88	55.96
GAT+DNN	68.07	86.48	62.12	67.92	86.24	62.56	66.92	86.69	63.00
SAGE+DNN	67.04	85.84	62.25	66.56	86.27	62.51	63.56	86.30	62.72
GraphBERT	**74.68**	**90.09**	**67.66**	**70.92**	**89.89**	**65.62**	**68.20**	**88.47**	**63.58**

TABLE III
EXPERIMENT RESULT ON MALICIOUS TWEET CLASSIFICATION. BACC., ACC., F1 DENOTE BALANCE ACCURACY, ACCURACY, F1-SCORE.

Model	Binary			Triple		
	bAcc.	Acc.	F1	bAcc.	Acc.	F1
LSTM	76.06	84.05	50.83	52.55	95.15	54.96
LSTM+DNN	79.92	83.77	50.78	58.15	80.85	54.96
BERT	89.08	96.56	73.57	65.54	96.51	55.43
BERT+CNN	89.06	95.50	73.63	67.29	89.69	48.34
BERT+LSTM	88.29	97.09	75.36	66.72	94.20	52.11
Roberta	90.31	96.01	66.95	65.65	96.69	55.24
XLNet	87.81	96.88	68.00	64.60	95.15	54.96
Ours	**90.99**	**97.78**	**75.77**	**67.71**	**97.35**	**55.51**

loss [53] is used for GNN training to balance. Besides, a weight sampler is used for data selection in malicious tweet classification experiments and training of the weight adaption model. In addition, for evaluation metrics, balance accuracy is used to reflect the model performance on imbalance training.

C. Performance Comparison

Malicious User Detection. As shown in Table II, we adopt research on three GNN baselines including GCN, GAT, SAGE for malicious user detection. In each baseline, we combine graph neural networks with different kinds of classification models (DNN and SVM) to get the result. We also conduct multiple experiments on different amounts of training data including 60%, 40%, and 20% to verify the robustness of models when facing different training ratios. The best results are highlighted in bold. For all cases, our model outperforms all baselines in balance accuracy, classification accuracy, and F1-score. For classification accuracy, our model shows a **+1∼15%** improvement compared to other *GNN-based* baselines, which indicates the superiority of introducing BERT into our model for better utilization of text representation. For balance accuracy, our model shows a great improvement of **+5∼15%** which indicate the effectiveness of our model for extremely imbalanced data classification. For F1-score, our model shows an improvement of **+1∼10%** increasing compared to baselines, which indicates our model considers balancing both precision and recall. Results show that our method is significantly superior to other methods especially reflected in balance accuracy. This result further shows that the semantic and relational information is complementary and can generate robust embedding which is especially effective facing imbalanced data.

Malicious Tweet Classification. As shown in Table III, we also conduct the experiment on the malicious tweet classification task to verify the effectiveness of our model. We compare our model with LSTM and Transformer-based baselines. Compared with LSTM-based methods, our model shows significant improvements in both binary and triple classification, which demonstrate the benefits of the use of a more powerful Transformer based model. Compared to Transformer-based methods including BERT, XLNet and Roberta, our model achieves a **+1∼3%** improvement on balance accuracy, **+1∼2%** improvement on classification accuracy and **+1∼7%** improvement on F1-score on binary classification task and **+2∼3%** improvement on balance accuracy, **+2%** improvement on classification accuracy and slightly improvement on F1-score on triple classification. These results show that related information from GNN which learns useful structure information and improve and perfects tweet embedding learning based on semantic methods. Compared to Fine-tuned BERT baselines including BERT+CNN and BERT+LSTM [11], our model achieves a **+1∼2%** improvement in balance accuracy, classification accuracy, and F1-score on binary classification task and a more obvious improvement on triple classification, which also shows the superiority of our weight adaption method.

D. Ablation Studies

The proposed GraphBERT integrates several crucial components: (i) semantic graph attention network (SG) (ii) Word-level weight adaption network (WF). (iii) Sentence-level weight adaption network (SF). To verify the effectiveness of each component, we conduct ablation studies by removing each component independently. We conduct five different experiments on malicious user detection including **1) GAT** with only graph attention neural network (remove SG, WF, and SF). **2) SG** with semantic graph attention network using

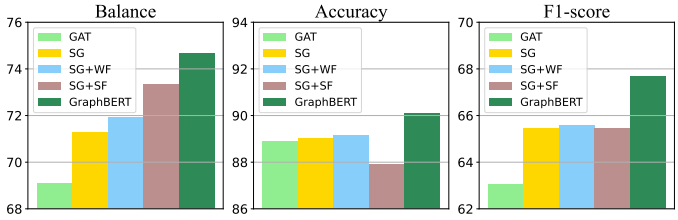

Fig. 4. The result of ablation study.

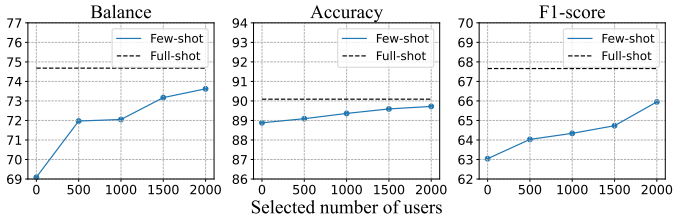

Fig. 5. The result of few-shot study.

semantic tweet embedding from the BERT model (remove WF and SF). **3) SG+WF** with semantic graph attention network using semantic tweet embedding from BERT model with word-level adaption network (remove SF). **4) SG+SF** with semantic graph attention network using semantic tweet embedding from BERT model with sentence-level adaption network (remove WF). **5) GraphBERT** is the full model. As shown in Fig. 4, we find each component has benefits to improvements on both balance accuracy and F1-score. Among them, GraphBERT achieves the best result. For precision, all components show improvements except for **SG+SF**, but SG+SF shows relatively significant improvement in balance accuracy compared to GAT. Ablation results demonstrate that all three components are effective to enhance our model for solving the problem.

E. Few-Shot Performance

Since tweet annotation for semantic representation learning is time-consuming for human annotators, we further conduct a few-shot experiment using a few tweet labels. As it is shown in Fig. 5, we conduct five different experiments using different amounts of labeled tweet data to train our model and explore the performance of few-shot GraphBERT. For all sampled users, we use the labels of tweets they tweeted to train weight adaption BERT. All parameters are set the same as in the full-shot experiment. The few-shot result shows that using tweets from less than 5% of total users (500/13,351) can still achieve +3% improvement in balance accuracy. Furthermore, the result of balance accuracy using labels from 15% of total users (2,000/13,351) achieves a great improvement compared to the baselines (0/13,351) and is only 1% lower than the full-shot result (13,351/13,351). The result on accuracy and F1-score also shows that our model can also achieve promising improvement with a small number of tweet labels.

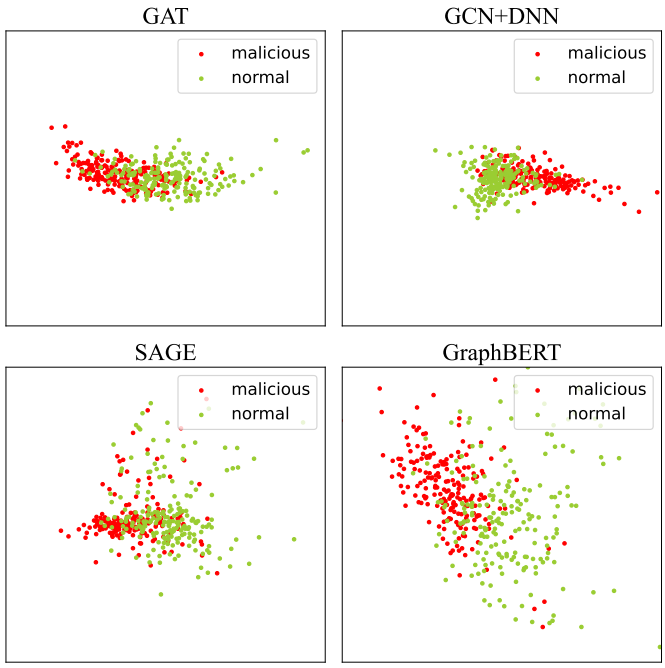

Fig. 6. Visualization of users' embeddings for malicious user detection.

F. Embedding Visualization

To better show the effectiveness of our model, we use t-SNE [57] to visualize user embeddings of four methods: GAT, GCN+DNN, SAGE, and GraphBERT, in Fig. 6. The red point represents the malicious users and the green point represents normal users. We can find that GraphBERT generates the most distinct boundaries and the smallest overlapping rate between malicious users and normal users, which further demonstrates the superiority of our model for malicious user detection.

VI. Conclusions

In this paper, we reveal the necessity of malicious behavior detection on social media and propose a dataset with both user and tweet labels for this research. To integrate both structural and semantic information, we create a graph with multiple nodes including users, tweets, mentions, hashtags, topics, and multiple-name entities. Based on the graph data, we introduce a novel GraphBERT model which integrates both semantic and relational information. The extensive experiments on both malicious user detection and malicious tweet classification tasks show our model outperforms the either graph or textual baselines. Results of few-shot learning also show that using GraphBERT, only a small number of labeled tweets can greatly improve malicious user detection. In the future, one promising direction is to apply a heterogeneous GNN to treat different node types and edge types distinctively. We also start to create the multi-language dataset for malicious behavior detection with the potential to learn cross-language malicious behaviors.

VII. Acknowledgments

This work is partially supported by the NSF under grant CMMI-2146076 and the Brandeis Provost Research Award (2021). Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any funding agencies.

References

[1] S. MacAvaney, H.-R. Yao, E. Yang, K. Russell, N. Goharian, and O. Frieder, "Hate speech detection: Challenges and solutions," *PloS one*, 2019.

[2] K. M. Craig, "Examining hate-motivated aggression: A review of the social psychological literature on hate crimes as a distinct form of aggression," *Aggression and Violent Behavior*, 2002.

[3] M. S. Akhtar, A. Ekbal, S. Narayan, and V. Singh, "No, that never happened!! investigating rumors on twitter," *IEEE Intelligent Systems*, 2018.

[4] Y. Qian, Y. Zhang, Y. Ye, and C. Zhang, "Distilling meta knowledge on heterogeneous graph for illicit drug trafficker detection on social media," in *NeurIPS*, 2021.

[5] L. Silva, M. Mondal, D. Correa, F. Benevenuto, and I. Weber, "Analyzing the targets of hate in online social media," in *ICWSM*, 2016.

[6] H. Sanchez and S. Kumar, "Twitter bullying detection," *NSDI*, 2011.

[7] S. Benesch, "Defining and diminishing hate speech," *State of the world's minorities and indigenous peoples*, 2014.

[8] P. Burnap and M. L. Williams, "Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making," *Policy & internet*, 2015.

[9] P. Fortuna and S. Nunes, "A survey on automatic detection of hate speech in text," *CSUR*, 2018.

[10] A. Schmidt and M. Wiegand, "A survey on hate speech detection using natural language processing," in *SocialNLP*, 2019.

[11] M. Mozafari, R. Farahbakhsh, and N. Crespi, "A bert-based transfer learning approach for hate speech detection in online social media," in *CNA*, 2019.

[12] ——, "Hate speech detection and racial bias mitigation in social media based on bert model," *PloS one*, 2020.

[13] I. Kwok and Y. Wang, "Locate the hate: Detecting tweets against blacks," in *AAAI*, 2013.

[14] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[15] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *TNNLS*, 2020.

[16] S. Wang, X. Liu, E. Zhu, C. Tang, J. Liu, J. Hu, J. Xia, and J. Yin, "Multi-view clustering via late fusion alignment maximization." in *IJCAI*, 2019.

[17] N. F. F. D. Silva, L. F. Coletta, and E. R. Hruschka, "A survey and comparative study of tweet sentiment analysis via semi-supervised learning," *CSUR*, 2016.

[18] D. Küçük and F. Can, "A tweet dataset annotated for named entity recognition and stance detection," *arXiv preprint arXiv:1901.04787*, 2019.

[19] A. Kulkarni, M. Mandhane, M. Likhitkar, G. Kshirsagar, and R. Joshi, "L3cubemahasent: A marathi tweet-based sentiment analysis dataset," *arXiv preprint arXiv:2103.11408*, 2021.

[20] M. Abdul-Mageed, H. Alhuzali, and M. Elaraby, "You tweet what you speak: A city-level dataset of arabic dialects," in *LREC*, 2018.

[21] B. Gaye, D. Zhang, and A. Wulamu, "A tweet sentiment classification approach using a hybrid stacked ensemble technique," *Information*, 2021.

[22] J. Coelho, P. Nitu, and P. Madiraju, "A personalized travel recommendation system using social media analysis," in *BigData Congress*, 2018.

[23] M. Krommyda, A. Rigos, K. Bouklas, and A. Amditis, "An experimental analysis of data annotation methodologies for emotion detection in short text posted on social media," in *Informatics*, 2021.

[24] S. W. H. Kwok, S. K. Vadde, and G. Wang, "Tweet topics and sentiments relating to covid-19 vaccination among australian twitter users: Machine learning analysis," *JMIR*, 2021.

[25] M. Sap, D. Card, S. Gabriel, Y. Choi, and A. N. Smith, "The risk of racial bias in hate speech detection," in *ACL*, 2019.

[26] R. Gomez, J. Gibert, L. Gomez, and D. Karatzas, "Exploring hate speech detection in multimodal publications," in *WACV*, 2020.

[27] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *ECML*, 1998.

[28] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *EMNLP*, 2014.

[29] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *NeurIPS*, 2013.

[30] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[31] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *NAACL-HLT*, 2018.

[32] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[33] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *NeurIPS*, 2019.

[34] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[35] H. Gao, Z. Wang, and S. Ji, "Large-scale learnable graph convolutional networks," in *KDD*, 2018.

[36] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *NeurIPS*, 2017.

[37] C. Zhuang and Q. Ma, "Dual graph convolutional networks for graph-based semi-supervised classification," in *WWW*, 2018.

[38] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous graph neural network," in *KDD*, 2019.

[39] Y. Fan, M. Ju, C. Zhang, and Y. Ye, "Heterogeneous temporal graph neural network," in *SDM*, 2022.

[40] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[41] Z. Liu, C. Chen, X. Yang, J. Zhou, X. Li, and L. Song, "Heterogeneous graph neural networks for malicious account detection," in *CIKM*, 2018.

[42] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," *NeurIPS*, 2020.

[43] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, "Graph contrastive learning with adaptive augmentation," in *WWW*, 2021.

[44] Y. You, T. Chen, Y. Shen, and Z. Wang, "Graph contrastive learning automated," in *ICML*, 2021.

[45] J. Zhao, Q. Wen, S. Sun, Y. Ye, and C. Zhang, "Multi-view self-supervised heterogeneous graph embedding," in *ECML/PKDD*, 2021.

[46] L. Yu, S. Pei, L. Ding, J. Zhou, L. Li, C. Zhang, and X. Zhang, "Sail: Self-augmented graph contrastive learning," in *AAAI*, 2022.

[47] Z. Liu, V. W. Zheng, Z. Zhao, Z. Li, H. Yang, M. Wu, and J. Ying, "Interactive paths embedding for semantic proximity search on heterogeneous graphs," in *KDD*, 2018.

[48] C. Zhang, A. Swami, and N. V. Chawla, "Shne: Representation learning for semantic-associated heterogeneous networks," in *WSDM*, 2019.

[49] W. Zhang, Y. Fang, Z. Liu, M. Wu, and X. Zhang, "mg2vec: Learning relationship-preserving heterogeneous graph representations via meta-graph embedding," *TKDE*, 2020.

[50] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *JMLR*, 2003.

[51] A. Mikheev, M. Moens, and C. Grover, "Named entity recognition without gazetteers," in *EACL*, 1999.

[52] Y. Wang, Y. Shen, Z. Liu, P. P. Liang, A. Zadeh, and L.-P. Morency, "Words can shift: Dynamically adjusting word representations using nonverbal behaviors," in *AAAI*, 2019.

[53] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *ICCV*, 2017.

[54] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, 1997.

[55] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[56] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *NeurIPS*, 2017.

[57] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *JMLR*, 2008.