Hindawi
Security and Communication Networks
Volume 2022, Article ID 1164125, 11 pages
https://doi.org/10.1155/2022/1164125

Research Article

Detecting Fake Reviews with Generative Adversarial Networks for Mobile Social Networks

Zheng Qu,[1] **Qingyao Jia**,[2] **Chen Lyu**,[1] **Jia Liu**,[3] **Xiaoying Liu**,[4] **and Kechen Zheng**[4]

[1]*School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai 200433, China*
[2]*Hwabao WP Fund Management Co., Ltd., Shanghai 200120, China*
[3]*Center for Strategic Cyber Resilience Research and Development, National Institute of Informatics, Tokyo 101-8430, Japan*
[4]*School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, Zhejiang 310023, China*

Correspondence should be addressed to Chen Lyu; lyu.chen@sufe.edu.cn

Received 8 September 2022; Accepted 6 October 2022; Published 10 November 2022

Academic Editor: Jianbo Du

With the growth of mobile social networks (MSNs), crowdsourced information could be used for recommendation to mobile users. However, it is quite vulnerable to Sybil attacks, where attackers post fake information or reviews to mislead users for business benefits. To address this problem, existing detection models mainly use graph-based techniques or extract features of users. However, these approaches either rely on strong assumptions or lack generalization. Therefore, we propose a novel Sybil detection model based on generative adversarial networks (GANs), which contains a feature extractor, a domain classifier, and a Sybil detector. First, the feature extractor is proposed to identify the rich information in the review text with the neural network model of TextCNN. Second, the domain classifier is implemented by a neural network discriminator and is able to extract common features. Third, the Sybil detector is utilized to discriminate the fake review. Finally, the minimax game between the domain classifier and Sybil detector forms a GAN and enhances the overall generalization ability of the model. Extensive experiments show that our model has a high detection accuracy against Sybil attacks.

1. Introduction

A growing number of mobile social networks (MSNs) in recent years have focused on the contents of social activities, such as eating, traveling, and shopping. For a specific activity or product, each user can give a review or rate it. With the Internet and mobile social platform, this information can be posted in real time. If other users are interested in it, they may collect information based on the reviews provided by the platform and adjust their consumption behavior according to the rating from other users. With the commercialization of MSNs, they have become popular platforms to share information and recommend products. Users can easily post reviews about merchants and obtain other people's reviews on the platforms.

Despite the convenience offered by MSNs, product reviews face severe security threats on the platforms such as Yelp and Dianping. On the one hand, users' reviews are usually posted individually and anonymously. It is difficult to access any information about users in the real world, making it hard to authenticate them. On the other hand, users are hard to verify the validity of a review based on the content of the review alone. Users post reviews on social networks based on their personal consumption experience, and MSNs make personalized recommendations based on the user's situation as well. Actually, merchants with high scores are more likely to capture customers, which may attract merchants to maliciously post fake reviews to improve their scores. This makes users' reviews become the targets of Sybil attacks, which is a major concern for many operators of MSNs.

The concept of Sybil attack was first applied to computer security which creates a large number of false identities (i.e., user accounts) and has a significant security threat to a system. Sybil attackers often manipulate social media through misinformation, defamation, spam, malware, or even just unrelated noise [1]. According to Yelp's 2021 Trust

and Safety Report, Yelp generated more than 19.6 million reviews in 2021, of which only about 71% were recognized as recommended reviews to be displayed by the platform. About 29% of reviews were considered non-compliant reviews, which have various issues including possible conflicts of interest, false, useless, or unreliable. Therefore, a reliable Sybil attack detection scheme is essential for the MSNs.

Unlike traditional Sybil attacks launched by fake accounts in online social networks, Sybil attacks in MSNs deceive customers by recruiting real users to generate fake content, which makes many existing Sybil detection approaches based on user behaviors fail (e.g., [2, 3]). Figure 1 illustrates a typical Sybil attack in Dianping. For a Sybil attack, an agent often hires real users and includes relevant requirements in the task posting, which specify the object and aspect of an attack. Compared to traditional online social networks (e.g., Twitter or Weibo), MSNs also greatly diminish the impact of user relationships as their users are not closely connected. This is because the main purpose of users is to learn about a product or merchant through other users' reviews, rather than communicating directly. Therefore, this feature makes previous graph-based approaches [4–8] ineffective. Other studies [9–11] have demonstrated that text features could provide good results for fake news detection. However, unlike news, features extracted from reviews are relatively scarce and variable, leading to these approaches being less efficient. In addition, reviews in MSNs are related to the product category, making the text features highly correlated with it. Hence, they lack the generalizability of detection of Sybil attacks.

In this work, we propose a Sybil attack detection model based on generative adversarial networks (GANs) to improve the accuracy and generalization of MSNs. Inspired by the idea of GANs, our model has three significant components: a feature extractor, a domain classifier, and a Sybil detector. First, to construct the feature extractor, we make use of the neural network model of text convolutional neural network (TextCNN) to extract text features of reviews, which would be input to the Sybil detector and domain classifier. Second, we make use of a neural network discriminator to design the domain classifier. The discrimination loss is set to be maximized, and therefore the learned features are common features unrelated to the product category. Third, based on the extracted text features, we design the Sybil detector with a fully connected layer to detect the fake reviews of Sybil attacks. Finally, we constitute a GAN using the minimax game between the domain classifier and the Sybil detector. Based on the real data crawled from Dianping, we validate our model and compare it with 9 state-of-the-art approaches. The extensive experiments show that our model has the best performance against Sybil attacks.

As far as we know, our model is the first Sybil detection model with GANs in MSNs. Our contribution can be summarized as follows:

(i) We design a Sybil detection model based on GANs to provide the generalizability of the model for MSNs, which includes a feature extractor, a domain classifier, and a Sybil detector.

(ii) We make use of the neural network model of TextCNN to construct the feature extractor and extract the text features of reviews.

(iii) We introduce the domain classifier with a neural network discriminator, which is able to learn common features.

(iv) We propose a GAN using the minimax game between the domain classifier and the Sybil detector. Through extensive experiments on Dianping, our GAN model effectively improves the detection accuracy of Sybil attacks.

The organization of the rest of the paper is as follows. We present the related work in Section 2. The description of data crawling, preprocessing, and annotation is given in Section 3. In Section 4, we present the construction of our model. Extensive experiments are conducted and analyzed in Section 5. At last, we conclude our work in Section 6.

2. Related Work

We list the literature related to our study and classify them into two categories based on their research focus. The first concentrates on the detection of Sybil attacks, and the second targets GANs.

2.1. Sybil Attacker Detection. Most previous research has focused on detecting Sybil attackers in online social networks (OSNs), such as fake accounts and spammers on Twitter. They mainly construct a graph of user relationships in the social networks using graph-based techniques. In the graph, nodes of the graph represent users and edges of the graph represent relationships.

Wei et al. [4] relied on social network graphs and proposed a mechanism of Sybil defense, which uses the metric to measure the relationship of users and thus decrease the number of edges of Sybil attacks. Effendy and Yap [5] made use of strongly connected graphs and strengthened the defense by decreasing the number of edges of Sybil attackers. Experimental results show that the defensibility of their method can be recovered once suspicious edges are removed. Furutani et al. [6] gave an explanation about the task of Sybil detection in terms of signal processing of graphs and proposed a general framework to design an approach for Sybil detection with both belief propagation and random wandering. Zhang et al. [8] improved the detection rate of Sybil attacks by integrating local structural similarity matching, regularization algorithms, and graph pruning in the graph networks. To detect Sybil attacks, Xue et al. [7] proposed a combination of graph edges and user feedback information for social networks. All these methods making use of graph models rely on the strong assumption that users are closely connected to each other, which only applies to OSNs [12]. However, the relationship between users is quite sparse for recommendations in the MSNs, since most users are not connected to others. Hence, building such an effective graph model is impossible for users in MSNs.

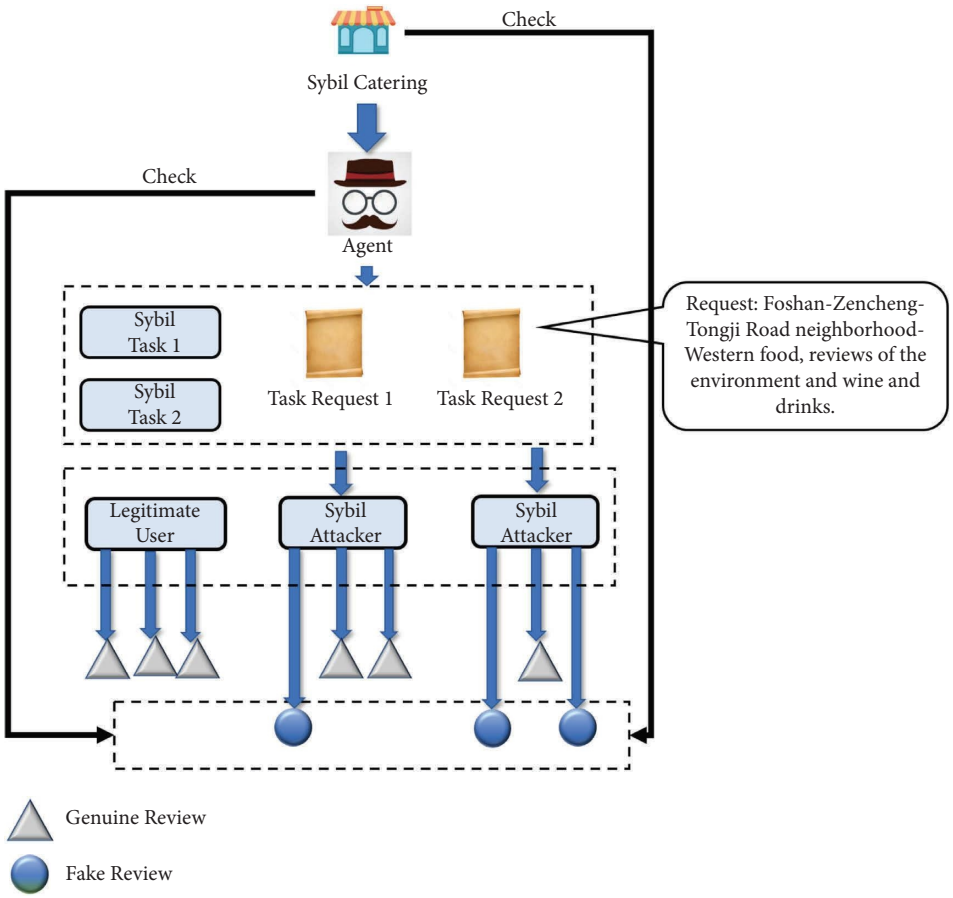

Figure 1: The flow of a Sybil attack.

Another important approach for Sybil detection is to deploy feature extraction techniques. Rahman et al. [13] took the impact of Sybil attackers into account and then made use of the parameters of user impact weights for Sybil detection. Egele et al. [14] proposed a model to identify anomalous users during a short time which is able to detect account theft. These two approaches mainly extract user features from textual content for Sybil detection. For OSNs, Ramachandran et al. [3] extracted user behavior features (e.g., replies) and network-based features (e.g., IP addresses) and then proposed a spam detection system. In Twitter, Song et al. [2] designed a method for fake review detection utilizing the feature of retweeting. In MSNs, Zhang et al. [15] used behavioral features of users and location-based features to detect Sybil attacks. Lyu et al. [16] introduced spatial-temporal features and users' preference features and combined them with traditional features to improve the detection accuracy of Sybil attackers.

All these methods based on user features have two drawbacks for detecting Sybil attacks in MSNs. On the one hand, since Sybil attackers often try to imitate the behaviors of real users in MSNs, these methods cannot distinguish real users and Sybil attackers by only extracting user features. On the other hand, they lack generalizability due to the natural limit of feature engineering. In this work, we make use of a neural network model to construct a detection model based on GANs, aiming to improve the accuracy and generalization against Sybil attacks.

2.2. Generative Adversarial Networks. Our work is inspired by the idea of GANs [17]. Existing GANs usually generate images that match the observed samples by means of a framework of minimax game.

Makhzani et al. [18] proposed a probabilistic self-encoder that deploys GANs to perform variational inference by matching the posterior of the self-encoder with an arbitrary prior distribution to ensure the distribution of the generated samples. Lipton and Tripathi [19] made use of a simple, gradient-based timely cropping technique that combines with GANs to transform potential vectors into visually plausible images. The robustness of their method was verified through experiments on unseen images. Ganin and Lempitsky [20] proposed a GAN-based deep learning framework in the absence of task-specific labeled data. They used few standard layers and a simple new gradient inversion layer for data augmentation to obtain better performance for small samples.

Pu et al. [21] designed a new GAN for joint distribution matching. Unlike other methods that only learn conditional distributions, their proposed model is able to learn the joint distribution of multiple random variables (domains), which establishes minimax games between event discriminators

and multimodal feature extractors. In particular, since the multimodal feature extractor is forced to learn a static representation of events in order to deceive the discriminator, it eliminates the tight dependence on specific events in the collected dataset and achieves a better generalization capability on unseen events. Since GANs perform outstandingly well for image and text processing, we explore the core idea of GANs and leverage it for detecting Sybil attacks in the MSNs.

3. Dataset

3.1. Dataset Description. Dianping is the largest and most popular mobile social network for recommendation in China. According to the official data, Dianping has over 250 million active users and over 150 million reviews per month in China. When a user visits Dianping, it suggests a list of local merchants (e.g., restaurants) based on keywords entered by the user or his current geographic location, which is usually sorted by the merchant's rating. According to Dianping's rules, the star rating of a merchant is a combination of the overall rating of the site's users and is automatically updated by the system based on a scientific formula without any human intervention. Users score the merchant's taste, environment, and service according to criteria ranging from one star to five stars. The system averages all users' scores and then adjusts them according to several predetermined indicators (including the number of reviews, review time, member/merchant's reputation, and so on).

A merchant with a large number of positive reviews on Dianping is a valuable advertisement, since a top-ranked merchant on the praise list tends to attract more users to visit that merchant. As a result, the platform of Dianping has been under constant threat of Sybil attacks, and both the number of positive reviews and ratings are often purposefully manipulated by Sybil attackers. Dianping has established its own review filtering mechanism. When we crawled the data from Dianping, we found that users' reviews are divided into normal reviews and hidden reviews. The hidden reviews are not shown on the default store page or user page, but we can still obtain them using a crawler. Reasons for becoming a hidden review may include that the review lacks sufficient informativeness (default positive reviews or reviews are too short) or that the platform believes the review may be a suspicious review posted by a Sybil attacker. However, the details of Dianping's review filtering algorithm are not available to the public. Moreover, despite the platform's filtering algorithm, fake reviews with commercial fraudulent nature are not completely eliminated.

3.2. Dataset Annotation. In this work, our target is to build a Sybil attack detection model for the review/comment data from Dianping. We crawl the data related to Dianping and acquire the data in the following steps. First, we manually select 12 merchants that have been officially confirmed to have Sybil attacks and crawl the reviews posted under these merchants. Second, based on the list of users in the reviews, we crawl out the personal information of these users and all the reviews they have posted. Finally, we collect a total of 918,373 user reviews.

For the hidden comments, we hire five undergraduate students as annotators to flag Sybil or real but low-quality comments. The annotators were also given full freedom to make use of any relevant information or their own intuition. In terms of some controversial cases, we deployed voting to determine the final outcome. Therefore, a review is marked as Sybil when and only when the results of five votes are SSSLL, SSSSL, or SSSSS, while S stands for Sybil and L stands for the legitimate review that contains low information or invalid positive reviews. The average annotation consistency based on Cohen κ is 0.74, which indicates the consistency property of annotation [22].

4. Our Methodology

In this section, we first introduce the three components of the model proposed in this paper: a feature extractor, a domain classifier, and a Sybil discriminator. Then, we describe how to integrate these three components to establish a generalized learning representation model. The flowchart of our model is shown in Figure 2.

4.1. Feature Extractor. For the Sybil attack in MSNs, we first choose a text feature extractor to extract the text features. Unlike common fake reviews, Sybil attacks are organized. Some Sybil attackers often give verbal hints to show the advantages of products or services, and these reviews are different for various types of products and services. Hence, we choose a text feature extractor to identify the rich information in the review text. Our feature extractor makes use of a convolutional neural network (CNN) as the main feature input module, which was first proposed due to the need for work on images and has been widely used in areas such as image processing [23, 24]. In the year of 2014, Zhang and Wallace [25] first proposed using CNNs to implement sentence classification. The initial TextCNN network has only one convolutional layer and one maximum pooling layer, and the output is connected to softmax for multiple classifications. The general structure diagram of TextCNN is illustrated in Figure 3. In this work, we capture text features of different granularity by adjusting the size of the convolutional window.

In terms of text feature extraction, we first preprocess the raw text of the reviews. We remove non-Chinese and unrecognizable reviews (e.g., text containing only emojis and special symbols) because these samples play no role in model training. We then eliminated information such as punctuation marks or emoticons in the sentences and split the review text using jieba. Jieba is a Python-based Chinese splitting component that can be used for word segmentation, lexical annotation, and keyword extraction. After preprocessing, we remove useless information such as conjunctions in the splitting result according to jieba and finally repatch the text at the end of the splitting to get the split words.

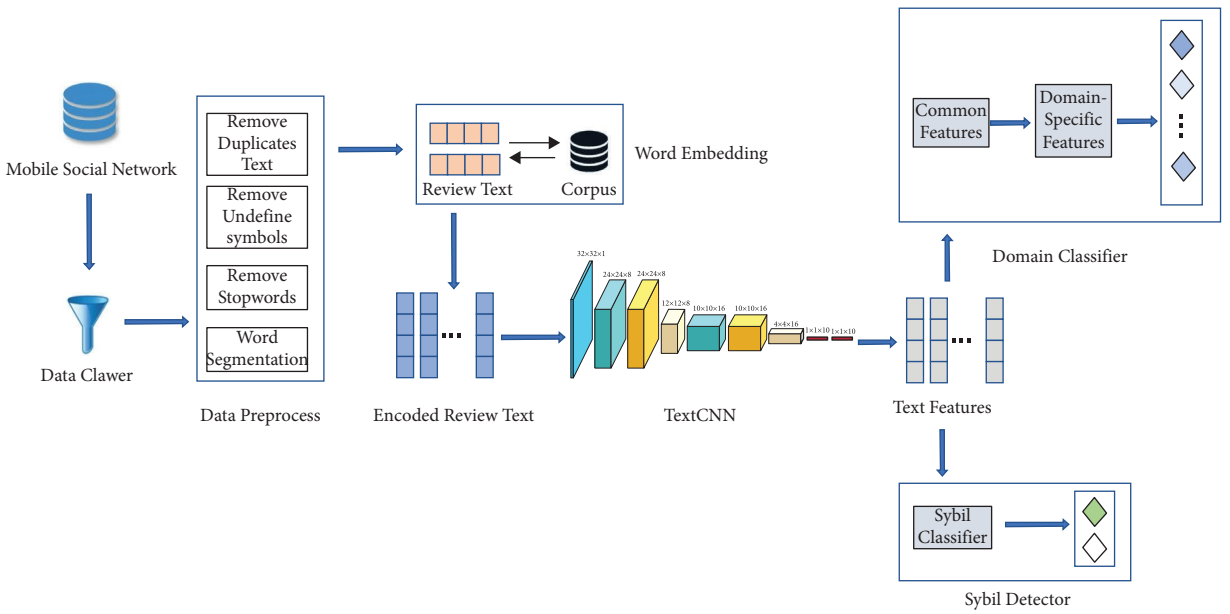

Figure 2: The flowchart of our model.

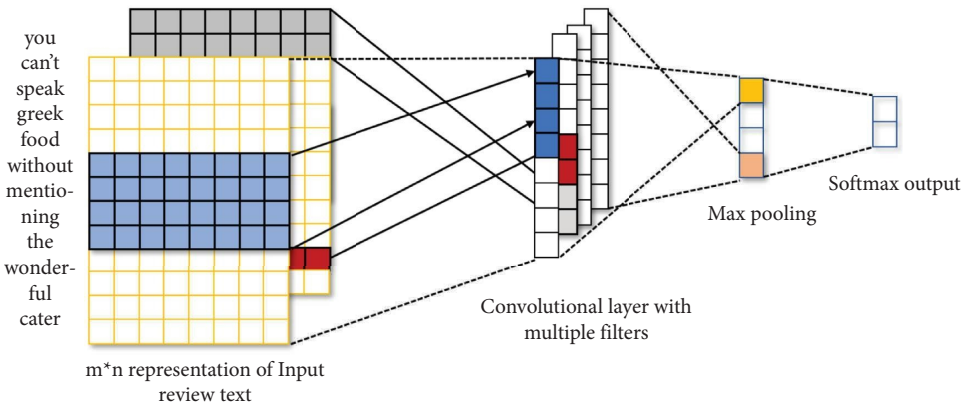

Figure 3: The structure of TextCNN.

Subsequently, we use word2vec to encode the processed text. Previous studies [26, 27] have shown that using a word embedding model can improve the performance of convolutional neural networks to a greater extent than the TextCNN structure adaptation. To obtain the word vector encoding, we choose a pretrained word embedding model that generates a 32-dimensional word vector corresponding to a word. The pretrained model is constructed by skip-gram through an existing lexicon. Compared to large-scale pretrained language models, the word embedding model has lower dimensions and is more suitable for convolutional neural networks. As a result, each word is encoded as a vector. For a given text $D_i$containing the sentence S_j, $S_j = \{w_1, w_2, \ldots, w_{n_j}\}$, when all words w are in the dictionary of the word embedding model, we give the representation of the sentence:

$$V_{S_j} = \text{concate}\left(V_{w_1}, V_{w_2}, \ldots, V_{w_{n_j}}\right), \tag{1}$$

where concate means concatenation of vectors. Similarly, the word embedding of the comment text D_i is represented as

$$V_{D_i} = \text{concate}\left(V_{S_1}, V_{S_2}, \ldots, V_{S_{m_i}}\right). \tag{2}$$

Since the vast majority of the text in the review is within 100 characters in length, we do not consider the long-term dependency between sentences. After getting the input text embedded, the convolution filter of window size z outputs the filtered word vectors based on the input vectors. For word w_k, the output vector after convolution is

$$\text{cov}_z\left(w_k\right) = \text{Relu}\left(W_c \cdot V_{\left[w_{k-\frac{z}{2}}: w_{k+\frac{z}{2}}\right]}\right), \tag{3}$$

$$\text{Relu} = \max\left(0, x\right), \tag{4}$$

where W_c is the weight of the filter and *Relu* is the activation function. The filter converts all words of sentence S_j to a feature vector:

$$cov_z(S_j) = \left[\text{cov}_z(w_1), \ldots, \text{cov}_z\left(w_{n_j-z+1}\right)\right]. \quad (5)$$

We then use maxpooling to extract the maximum value of the features. Maxpooling can reduce the number of model parameters and help to reduce the problem of model overfitting. After the pooling operation, the 2-D or 1-D array is often converted into a single value. For the subsequent convolution layer or fully connected hidden layer, the number of parameters of a single filter or the number of neurons in the hidden layer is reduced. The variable length of inputs can be collapsed into fixed-length inputs. The model of CNN often ends up with a fully connected layer, and its number of neurons needs to be fixed in advance. The text features after the maxpooling operation are denoted as R_f. A fully connected layer is used to obtain the final text features: $F_f(W_f \cdot R_f)$, where W_f is the weight matrix of the fully connected layer.

We denote the text feature extractor as $F_f(R_f; \theta_f)$, where θ_f denotes the parameter to be learned. The output of the feature extractor is used as the input features for the subsequent generation of the adversarial model.

4.2. Domain Classifier. The main purpose of the domain classifier is to learn the category to which a review belongs. During the data processing, we classify review data into K categories, and there are some differences in the text corresponding to different products. The domain classifier determines the category to which the reviews belong by dealing with the output of the feature extractor. In our task, we want to identify fake reviews into different domains by text, which is also able to extract Sybil text features with commonality.

The domain classifier G_d consists of a neural network discriminator with a network structure consisting of a three-layer fully connected neural network and using Relu as the activation function. We give the loss function of the domain classifier as follows:

$$L_d\left(\theta_f, \theta_d\right) = \sum_{k=1}^{K} \log\left(G_d\left(F_f\left(R_f; \theta_f\right)\right); \theta_d\right). \quad (6)$$

We could learn the parameter of loss function of domain classifier θ_d by

$$\left(\widehat{\theta}_f, \widehat{\theta}_d\right) = \arg\max_{\theta_f, \theta_d} L_d\left(\theta_f, \theta_d\right), \quad (7)$$

where the loss L_d is calculated by the cross-entropy function. The loss is used to estimate the variability of the different domain distributions. When the loss is large, the difference between reviews' domain distributions is small and the learned features are approximated. This means that the common features of all domain texts are extracted. Therefore, in our model, we prefer the domain loss function to be as large as possible, that is, to maximize the discriminative loss $L_d(\theta_f, \theta_d)$ by finding the optimal parameter θ_f. With this condition, the Sybil classifier of text can find all the Sybil reviews as possible.

4.3. Sybil Detector. In this part, we introduce the Sybil detector, which uses softmax to deploy a fully connected layer to determine whether a review is a Sybil review or not. Our detector is based on the text features extracted from the feature extractor F_f. We denote the Sybil detector as $G_s(F_f; \theta_s)$, where θ_s denotes all parameters included in the detector. Given a review D_i, the probability that this review belongs to Sybil reviews is $P_s(D_i)$:

$$P_s(D_i) = G_s\left(F_f\left(R_f; \theta_f\right); \theta_s\right). \quad (8)$$

We use cross-entropy to calculate the loss of the model:

$$L_s\left(\theta_f, \theta_s\right) = \sum_{i=1}^{N} [y \log\left(P_s(D_i)\right) + (1 - y)\log\left(1 - P_s(D_i)\right)], \quad (9)$$

where L_s donates the loss of Sybil detector and N donates the total number of reviews. For a single Sybil review detector, we only minimize the loss function by finding the optimal parameter θ_s:

$$\left(\widehat{\theta}_f, \widehat{\theta}_s\right) = \text{argmin}_{\theta_f, \theta_s} L_s\left(\theta_f, \theta_s\right). \quad (10)$$

The minimization loss can capture class-specific-based representations. However, such features lack generalization. Therefore, we need a generalized learning representation model that captures common features across categories.

4.4. Model Combination Optimization. To establish the generalized learning model, we need to remove the uniqueness of each domain feature. This is completed by measuring the variability of feature representations across domains and removing them to capture feature representations across domains. Therefore, it leads to a minimal and maximal game between the domain classifier and the Sybil detector. On the one hand, the domain classifier tries to trick the detector to maximize the discriminative loss. On the other hand, the Sybil detector aims to discover event-specific information contained in the feature representation to identify the Sybil review. Hence, we construct our model with GANs using the minimax game [28–30]. The overall loss of these two classifiers is expressed as

$$L_{\text{all}}\left(\theta_f, \theta_d, \theta_s\right) = L_d\left(\theta_f, \theta_d\right) - \lambda L_s\left(\theta_f, \theta_s\right), \quad (11)$$

where λ is a parameter that regulates the importance of two classification tasks. Larger λ indicates a higher importance of the domain classification task, and smaller λ indicates a higher importance of the Sybil review detection task in a specific domain. In the experimental part, we investigate the optimal value of λ. For the minimax game, the parameter set we seek is the saddle point of the final objective function:

$$\left(\widehat{\theta}_f, \widehat{\theta}_d, \widehat{\theta}_s\right) = \text{argmin}_{\theta_f,\theta_d,\theta_s} L_{\text{all}}\left(\theta_f, \theta_d, \theta_s\right). \tag{12}$$

We make use of stochastic gradient descent to find the saddle point. We fix the learning rate r and update the loss in each step:

$$\begin{aligned}
\theta_f &:= \theta_f - r\left(\frac{\partial L_d}{\partial \theta_f} - \lambda \frac{\partial L_s}{\partial \theta_f}\right), \\
\theta_d &:= \theta_d - r\frac{\partial L_d}{\partial \theta_d}, \\
\theta_s &:= \theta_s - r\frac{\partial L_s}{\partial \theta_s}.
\end{aligned} \tag{13}$$

In the experimental section, we compare our model with other approaches and also discuss the effect of the learning rate and the optimization on our model.

5. Experiments

In this section, we separate the data for training and testing and set up evaluation for our model. First, the raw data are analyzed and the distribution of length of reviews is illustrated. Second, a series of basic methods are presented for comparative evaluation. Third, a large number of experiments are done and the performance evaluation of our Sybil detector is shown in detail. Fourth, an ablation study is performed to demonstrate the validity of our model. Finally, we validate our model on different parameters. Our model is evaluated on a server with Intel CPU Xeon W-2123 3.9 GHz and 64G RAM. The GPU of the server is NVIDIA Tesla v100 with CUDA version 10.2. Our model is implemented by Python 3.7, with PyTorch 1.10.0.

5.1. Data Analysis. We analyze the distribution of length of reviews, and the results are shown in Figure 4. The text length analyzed here is the length of the raw review text, which contains emotions and punctuation marks. Hence, the length will be greatly reduced after preprocessing. We choose a criterion of every 30 words and divide the text length into 10 levels. It can be found that more than 1/4 of the comments have no more than 30 words. At the same time, nearly half of the comments are less than 60 words in length, while comments longer than 120 words only account for 30% of the total.

We also count the length distribution of text after symbol removal, since these symbols have no meaning in feature learning of text. The results are shown in Figure 5. More than 30% of valid characters are less than 30 characters in length. The proportion of less than 60 words is more than 50%, and the effective comments longer than 120 words only account for 1/4. By comparing before and after symbol cleaning, we demonstrate the previous hypothesis that text lengths are generally shorter in MSNs. Useless symbols account for a large proportion and are not suitable for general detection models of long text.

In order to reduce the impact of distribution of review length and then reduce the training cost of the model, we choose reviews with higher quality (longer length and relatively obvious features) to form our dataset.

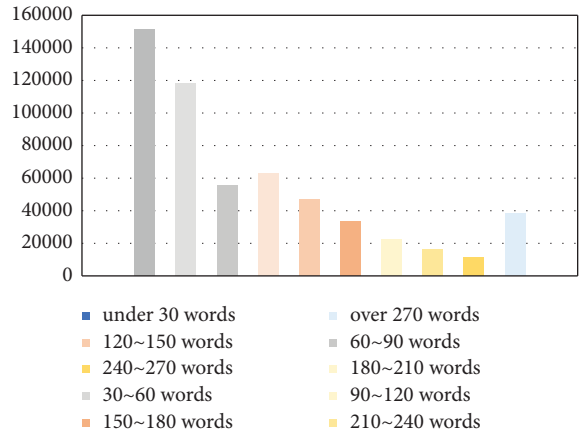

Figure 4: The distribution of length of original review data.

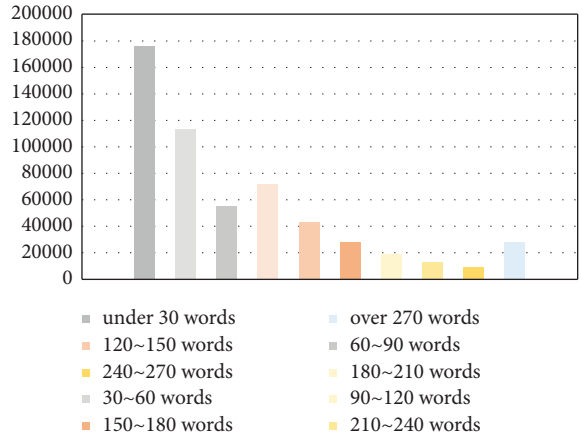

Figure 5: The distribution of length of processed review data.

5.2. Baseline Methods. Benchmark 1 (Classical Machine Learning Model). In the experiment, we select some classical models that are widely used for text classification. To get the best results, we also choose the optimal parameters for each of the models as much as possible.

We select tree-based models, including random forest (with the number of estimators as 20 and max depth as 5), XGBoost (with the learning rate of 0.05 and max depth as 4), and AdaBoost (with the number of estimators as 200, learning rate as 0.05, and estimators as CART decision tree). The input of all these models contains all the text features we extracted.

We choose the logistic regression (with Lasso) as a model for comparison. We also compare our model with SVM (with RBF kernel and penalty parameter as 1) and KNN (with Euclidean distance).

Benchmark 2 (Relevant Model). We also choose another three detection schemes for comparison as they all are deployed in MSNs. The traditional feature model (TFM) makes use of statistical features of users and text features for Sybil detection. Since the feature dimension is low, we use

SVM as the classifier. Zhang et al. [31] utilized both location-based features and traditional features of users to detect Sybil attacks in Dianping. Lyu et al. [16] made use of location-based features, users' preference features, and spatial-temporal features for fake review detection in Dianping.

To evaluate the performance of these models, we will use the following metrics: precision, recall, F1-score, and AUC (area under the ROC curve), which are commonly used in the classification system.

5.3. Model Evaluation. In the experiments, we make the following setups. For the feature extractor, we set the dimension of word embedding k as 32 and the number of filters as 20. In terms of the model of TextCNN, we set the window size of filters from 1 to 5. The hidden size of the fully connected layer in the feature extractor is set to 32. For the Sybil detector, the hidden size of the fully connected layer is set to 64. The domain classifier consists of two fully connected layers. The hidden size of the first layer is set to 64, and the hidden size of the second layer is 32. For all baselines and our proposed model, we use a batch size of 100 in the training phase. The epochs are set to 100, and the learning rate is set to $5e-4$.

We compare our scheme with various baseline methods and give the results in Table 1. For different machine learning models, we choose different methods of text encoding based on the dimension of the input features, since the input dimension will largely affect the performance of the model. For example, in terms of tree-based models, we choose word2vec + TextCNN as the method of text encoding. By comparing different machine learning models, the tree-based models based on TextCNN + word2vec generally perform quite well, which proves the effectiveness of text feature extraction. Among the tree-based models, random forest obtains the best results, outperforming the other models in F1-score, precision, and recall.

Our model outperforms Benchmark 1 and Benchmark 2 using features in terms of almost all the indicators. The results of our model exceed random forest by 3% in precision and 6% in AUC, which illustrates the usefulness of adversarial networks in MSNs. In Benchmark 2, Zhang et al.'s method and Lyu et al.'s method both perform slightly worse than our method despite the use of location-based features. This further demonstrates that the interference of the difference of domain distribution affects the final results. Compared to the other models, our model does not use any additional features, and the generalization of our model is enhanced.

5.4. Ablation Study. In this part, we perform ablation experiments for our model. The ablation experiments compare two additional models: the non-text feature model (nTFM) and the non-adversarial model. The nTFM model contains user features and interaction information, while the non-adversarial model does not implement a GAN, but deploys a Sybil detector only using the TextCNN model combined with a neural network classifier.

The experimental results are shown in Figure 6. The overall performance of our proposed model is better than that of the non-adversarial model in terms of AUC, precision, recall, and F1-score. The AUC of our model with GANs is similar to the model of nTFM. However, all other metrics are improved significantly, which indicates that our model has better generalizability for detecting Sybil reviews. For all the metrics, the results of the non-adversarial model are lower than the results of the adversarial networks, but they perform slightly better than the model of nTFM in terms of generalization.

In order to show the difference between GANs and the non-adversarial model for text feature extraction, we use t-SNE [32] to reduce the dimensionality of the classification results and then perform visualization. The final results are shown in Figure 7. Compared to GANs, the classification results of the non-adversarial model are more compact and the distances between positive and negative samples are closer. This means that there is little difference between a normal Sybil review and a normal review in the non-adversarial model. In contrast, the discriminability of the text features learned by GANs is better, with a larger interval between samples with different labels. This is because the domain classifier tries to eliminate the dependency between the feature representation and the product category during the training phase. With the assistance of the minimax game, the Sybil detector can learn invariant features in different categories and obtain the capability of generalization.

5.5. Parameter Analysis. In this section, we discuss the impact of various parameters on our model. For the feature extractor of our model, we discuss the effect of window size z . For our final model based on GANs, we focus on the impact of the loss weight λ and the learning rate r in the loss function on the overall model.

5.5.1. The Impact of Window Size. The experimental results for different window sizes are shown in Table 2. When the window size is set from 1 to 5, we can best extract text features with different granularity. For other window sizes, either the results will be degraded due to missing features or the effective features will be degraded due to too large window size. For a specific window size, our model has n different filters. Considering the size of the training set, we set n to be 20 in order to reduce the training time while ensuring the training results.

5.5.2. The Impact of Loss Weight. In our GANs, λ is the critical parameter that regulates the importance of the two classification tasks. Larger λ indicates a higher importance of the product category classification task, while smaller λ indicates a higher importance of the Sybil review detection task. The value of λ determines the overall performance of the model, so we research the optimal value through a large number of experiments, which are shown in Table 3.

Based on our experiments on λ, we can study the impact of the auxiliary classification task of GANs on the overall

Table 1: The comparison results of different models.

Methods	Precision	Recall	F1-score	AUC
LR (TF-IDF)	0.49	0.59	0.53	0.55
XGBoost (word2vec + TextCNN)	0.63	0.64	0.63	0.66
AdaBoost (word2vec + TextCNN)	0.70	0.59	0.64	0.62
Random forest (word2vec + TextCNN)	0.76	0.77	0.77	0.76
SVM (TF-IDF)	0.74	0.71	0.73	**0.82**
KNN (TF-IDF)	0.73	0.75	0.74	0.66
TFM	0.66	0.68	0.67	0.79
Zhang et al.'s method [31]	0.70	0.68	0.69	0.79
Lyu et al.'s method [16]	0.72	0.74	0.73	0.80
Our model (word2vec + TextCNN + GANs)	**0.79**	**0.80**	**0.80**	**0.82**

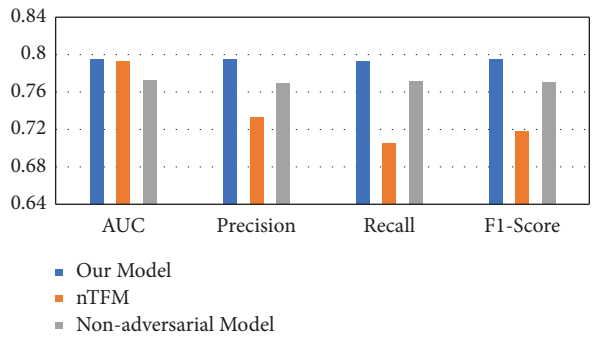

Figure 6: Results of ablation study on different models.

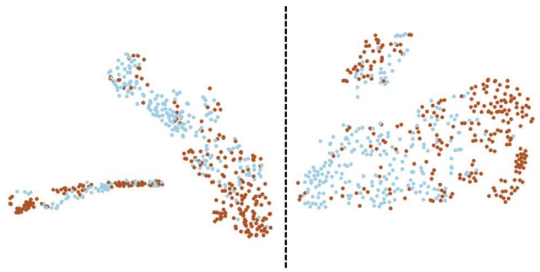

Figure 7: The final extracted features of the non-adversarial model (a) and GANs (b).

Table 2: Results on different window sizes.

Size of window	AUC	Precision	Recall	F1-score
1	0.652	0.63	0.643	0.636
2	0.7	0.72	0.723	0.721
3	0.725	0.721	0.725	0.723
4	0.722	0.723	0.721	0.722
1, 2	0.750	0.752	0.753	0.752
1, 2, 3	0.780	0.777	0.772	0.774
1, 2, 3, 4	0.782	0.78	0.779	0.779
1, 2, 3, 4, 5	0.785	0.782	0.783	0.782
1, 2, 3, 4, 5, 6	0.778	0.78	0.773	0.776
2, 3, 4, 5	0.775	0.768	0.78	0.774

model. When λ is set to zero, it means that the module of domain classifier fails, and its classification result has no effect on the feature extraction based on TextCNN. Hence,

Table 3: Results on different model weights.

λ	AUC	Precision	Recall	F1-score
0	0.773	0.771	0.772	0.771
0.1	0.776	0.775	0.776	0.775
0.5	0.778	0.776	0.78	0.778
0.8	0.78	0.78	0.779	0.779
0.9	0.788	0.788	0.781	0.784
0.95	0.795	0.796	0.800	0.798
1	0.782	0.780	0.781	0.780
1.05	0.780	0.780	0.780	0.780
1.1	0.778	0.78	0.773	0.776
1.5	0.772	0.771	0.771	0.771
2	0.76	0.758	0.76	0.759
10	0.71	0.703	0.701	0.702
100	0.62	0.617	0.62	0.618
10000	0.433	0.435	0.433	0.434

the overall model can be considered as a simple TextCNN for Sybil detection task. It can be found that even a simple TextCNN model can achieve good results for Sybil attack detection. As λ gradually increases, the experimental results of the model are gradually improving. If $\lambda = 0.95$, our model obtains the best classification performance, which proves that the introduction of the domain classification task helps improve the generalization ability of our model. When λ continues to increase, our model pays more attention to the accuracy of the classification task of review domains, resulting in insufficient information for the Sybil detection task. When the value of λ is particularly large, it is equivalent to the model completely turning into a domain classification model. In this case, the objective becomes to classify reviews into different categories, which leads to poor final results of the model.

5.5.3. The Impact of Learning Rate. We also experimentally determine the optimal learning rate of the model. As an important parameter in supervised learning and deep learning, the learning rate determines whether and when the objective function converges to a local minimum. The convergence process will be slow when the learning rate is small, while the gradient may vibrate when the learning rate is large. A suitable learning rate can make the objective function converge to a local minimum in a suitable time.

Table 4: Results on different learning rates.

Learning rate	AUC	Precision	Recall	F1-score
1	0.522	0.513	0.520	0.516
0.1	0.676	0.677	0.678	0.677
0.01	0.748	0.750	0.749	0.749
0.001	0.780	0.780	0.776	0.778
5e − 4	0.788	0.788	0.781	0.784
1e − 5	0.788	0.789	0.780	0.784
Learning rate decay (from 1e − 3 to 1e − 5)	0.787	0.785	0.783	0.784
Adam	0.792	0.788	0.790	0.789
Adam + learning rate decay (from 1e − 3 to 1e − 5)	0.795	0.796	0.800	0.798

To compare the effects on the model results between different learning rates, the parameters other than the learning rate are consistent with the optimal results discussed above, and our experimental results are shown in Table 4. In addition to the fixed learning rate approach, we also discuss two optimization methods based on learning rate decay (learning rate decay and Adam + learning rate decay). The dynamic learning rate decay method decreases the learning rate as the epoch increases, which reduces the possibility of model oscillations and allows the gradient to converge to a stable range. Adam optimizer adaptively adjusts the learning rate and optimizes the results according to the gradient changes. In terms of learning rate decay, we set the initial learning rate to 1e-3 and make it decrease gradually with the number of iterations. The experimental results show that using learning decay alone cannot improve the model performance. By adding the Adam optimizer, all the performances of metrics improve, with F1-score improving by 1.5% compared to the optimal fixed learning rate model. Therefore, we suggest using Adam + learning decay as the optimizer.

6. Conclusion

In this paper, we propose a novel Sybil detection model based on GANs for MSNs, which contains a feature extractor, a domain classifier, and a Sybil detector. First, we construct the feature extractor with the neural network model of TextCNN, which is able to extract the text features of reviews. Second, we introduce the domain classifier to learn common features of reviews in different domains. Third, we design the Sybil detector to detect the Sybil review. Finally, we design our model based on GANs using the minimax game between the domain classifier and the Sybil detector. We also examine the effect of the two classification tasks of GANs and then find the optimal adversarial parameters for our model. Based on the dataset from Dianping, we experimentally validate that our model has excellent generalizability and achieves better detection accuracy than other Sybil detection models as well. In the future research, we will try to introduce graph neural networks to provide more properties for our model.

Data Availability

The data used to support the findings of this study are available from the corresponding author or first author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by JSPS KAKENHI (grant no. JP20K14742), Project of Cyber Security Establishment with Inter University Cooperation; National Natural Science Foundation of China (grant nos. 62102303, 61902351, and 61902353); Key R&D Program of Shaanxi Province (grant no. 2021KWZ-04); the Zhejiang Provincial Natural Science Foundation of China (grant nos. LY21F020022 and LY21F020023); Fundamental Research Funds for the Central Universities (no. 2022110161).

References

[1] D. Yuan, Y. Miao, N. Z. Gong et al., "Detecting fake accounts in online social networks at the time of registrations," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1423–1438, London UK, November 2019.

[2] J. Song, S. Lee, and J. Kim, "Crowdtarget: target-based detection of crowdturfing in online social networks," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 793–804, Denver Colorado USA, October 2015.

[3] A. Ramachandran, N. Feamster, and S. Vempala, "Filtering spam with behavioral blacklisting," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pp. 342–351, Alexandria Virginia USA, October 2007.

[4] W. Wei, F. Xu, C. C. Tan, and Q. Li, "Sybildefender: defend against sybil attacks in large social networks," in *Proceedings of the 2012 IEEE INFOCOM*, pp. 1951–1959, IEEE, Orlando FL USA, March 2012.

[5] S. Effendy and R. H. Yap, "The strong link graph for enhancing sybil defenses," in *Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pp. 944–954, IEEE, Atlanta GA USA, June 2017.

[6] S. Furutani, T. Shibahara, K. Hato, M. Akiyama, and M. Aida, "Sybil detection as graph filtering," in *Proceedings of the GLOBECOM 2020-2020 IEEE Global Communications Conference*, pp. 1–6, IEEE, Taipei Taiwan, December 2020.

[7] J. Xue, Z. Yang, X. Yang, X. Wang, L. Chen, and Y. Dai, "Votetrust: leveraging friend invitation graph to defend against social network sybils," in *Proceedings of the 2013 Proceedings IEEE INFOCOM*, pp. 2400–2408, IEEE, Turin, Italy, April 2013.

[8] H. Zhang, J. Zhang, C. Fung, and C. Xu, "Improving sybil detection via graph pruning and regularization techniques," in *Proceedings of the Asian Conference on Machine Learning. PMLR*, pp. 189–204, Hong Kong, November 2016.

[9] M. Bao, J. Li, J. Zhang, H. Peng, and X. Liu, "Learning semantic coherence for machine generated spam text detection," in *Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, Budapest, Hungary, July 2019.

[10] H. Ahmed, I. Traore, and S. Saad, "Detecting opinion spams and fake news using text classification," *Security and Privacy*, vol. 1, no. 1, p. 9, 2018.

[11] F. Zhang, L. Qiu, P. Qi, and H. Luo, "A novel text features jointing model for review spam filtering of Chinese," in *Proceedings of the 2020 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 2051–2056, IEEE, Limassol, Cyprus, June 2020.

[12] J. Ding, Z. Liu, S. Xiao et al., "Beyond the click: a first look at the role of a microblogging platform in the web ecosystem," *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 743–754, 2019.

[13] M. Rahman, B. Carbunar, J. Ballesteros, G. Burri, and D. H. Chau, "Turning the tide: curbing deceptive yelp behaviors," in *Proceedings of the 2014 SIAM International Conference on Data Mining SIAM*, pp. 244–252, Philadelphia, PA, USA, April 2014.

[14] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna, *Compa: Detecting Compromised Accounts on Social Networks*Carnegie Mellon University, Pittsburgh, PA, USA, 2013.

[15] X. Zhang, H. Xie, and J. C. Lui, "Sybil detection in social-activity networks: modeling, algorithms and evaluations," in *Proceedings of the 2018 IEEE 26th International Conference on Network Protocols (ICNP)*, pp. 44–54, IEEE, Cambridge, UK, September 2018.

[16] C. Lyu, D. Huang, Q. Jia et al., "Predictable model for detecting sybil attacks in mobile social networks," in *Proceedings of the 2021 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, IEEE, Nanjing, China, March 2021.

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[18] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," 2015, https://arxiv.org/abs/1511.05644.

[19] Z. C. Lipton and S. Tripathi, "Precise recovery of latent vectors from generative adversarial networks," 2017, https://arxiv.org/abs/1702.04782.

[20] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proceedings of the International Conference on Machine Learning. PMLR*, pp. 1180–1189, Lille, France, July 2015.

[21] Y. Pu, S. Dai, Z. Gan et al., "Jointgan: multi-domain joint distribution learning with generative adversarial nets," in *Proceedings of the International Conference on Machine Learning. PMLR*, pp. 4151–4160, Stockholmsmässan, Stockholm Sweden, July 2018.

[22] Z. Qu, C. Lyu, and C.-H. Chi, "Mush: Multi-Stimuli Hawkes Process Based Sybil Attacker Detector for User-Review Social Networks," *IEEE Transactions on Network and Service Management*, 2022.

[23] J. Gu, G. Wang, J. Cai, and T. Chen, "An empirical study of language cnn for image captioning," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1222–1231, Cambridge, MA, USA, June 1995.

[24] K. Chandrasegaran, N.-T. Tran, and N.-M. Cheung, "A closer look at fourier spectrum discrepancies for cnn-generated images detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7200–7209, Seattle, WA, USA, June 2021.

[25] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," 2015, https://arxiv.org/abs/1510.03820.

[26] B. Guo, C. Zhang, J. Liu, and X. Ma, "Improving text classification with weighted word embeddings via a multi-channel textcnn model," *Neurocomputing*, vol. 363, pp. 366–374, 2019.

[27] C. Zhang, R. Guo, X. Ma, X. Kuai, B. He, and W-textcnn, "W-TextCNN: a TextCNN model with weighted word embeddings for Chinese address pattern classification," *Computers, Environment and Urban Systems*, vol. 95, Article ID 101819, 2022.

[28] L. Liu, M. Zhao, M. Yu, M. A. Jan, D. Lan, and A. Taherkordi, "Mobility-aware multi-hop task offloading for autonomous driving in vehicular edge computing and networks," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[29] J. Feng, L. Liu, Q. Pei, and K. Li, "Min-max cost optimization for efficient hierarchical federated learning in wireless edge networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 11, pp. 2687–2700, 2021.

[30] H. Han, L. Fang, W. Lu, W. Zhai, Y. Li, and J. Zhao, "A gcica grant-free random access scheme for m2m communications in crowded massive mimo systems," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 6032–6046, 2022.

[31] X. Zhang, H. Zheng, X. Li, S. Du, and H. Zhu, "You are where you have been: sybil detection via geo-location analysis in osns," in *Proceedings of the 2014 IEEE Global Communications Conference*, pp. 698–703, IEEE, Austin, TX, USA, December 2014.

[32] L. Van der Maaten and G. Hinton, "Visualizing non-metric similarities in multiple maps," *Machine Learning*, vol. 87, no. 1, pp. 33–55, 2012.