Social Network Analysis and Mining (2024) 14:218
https://doi.org/10.1007/s13278-024-01379-7

ORIGINAL ARTICLE

CGANS: a code-based GAN for spam detection in social media

Atefeh Rashidi[1] · **Mostafa Salehi**[1,2] · **Shaghayegh Najari**[1]

Received: 19 April 2024 / Revised: 25 October 2024 / Accepted: 26 October 2024

Abstract
In recent years, the growth of social media has significantly increased in people's daily lives. Social media platforms offer various benefits to users. Still, they also come with vulnerabilities, including spam, posing threats to media owners and users, leading to user fatigue, financial and security damages, sensitive information disclosure, and media credibility decline. Due to the high volume of spam, manual detection is challenging. However, employing machine and deep learning methods makes it possible to identify and block them effectively. Also, class imbalances in real-world datasets are prevalent, with more non-spam instances. This poses challenges for models in effectively identifying samples with limited data. In this regard, we propose a neural network model to detect spam based on its textual content. Furthermore, using a code-based Generative Adversarial Networks, the model balances the number of spam and non-spam samples. We used the Wasserstein distance with penalty regularization to measure the convergence between the actual and generated data distributions to address the mode collapse and vanishing gradient issues. Our evaluation of experimental results from two benchmark datasets demonstrates a significant improvement in spam classification and addresses the issue of imbalanced datasets. We compared our model with three other spam detection models, one of which employed data augmentation. In our evaluation, the proposed model achieved an accuracy of 98.2% and an F1-score of 93.0%. In this study, we also assessed the quality of text generated by our proposed model using the perplexity metric to evaluate the fluency and accuracy of the spam content.

Keywords Social spam detection · Deep neural networks · Text classification · Generative adversarial networks · Imbalance dataset

1 Introduction

In the present era, a significant portion of the population actively engages in diverse social media platforms. This widespread adoption facilitates organizations and institutions in effectively reaching out to their target audiences worldwide (Najari et al. 2022). The widespread use of social media has provided a breeding ground for criminal behaviors and exploitation of users and media owners. One of these destructive activities is spamming. Consequently, spammers have exploited these platforms to conduct spam attacks (Gupta et al. 2018b). Spam refers to unsolicited messages sent in the form of bulk messages, phone numbers, malicious web addresses, popular hashtags, images with hidden web addresses, malware dissemination, stock market spam, fake advertisements, fake news, rumors, and more. Spammers propagate these spam messages to earn income, advertise, phishing, violence against women, and other objectives (Bindu et al. 2018).

Various methods have been proposed for spam detection. Due to the large volume and speed of spam production, traditional methods do not provide satisfactory performance (Wu et al. 2017). In recent years, Machine Learning (ML) and Deep Learning-based (DL) approaches have been studied for spam detection (Barushka and Hájek 2018). Barushka and Hájek (2018) argue that ML methods have lower performance when it comes to

M. Salehi, S. Najari have contributed equally to this work.

✉ Mostafa Salehi
mostafa_salehi@ut.ac.ir

Atefeh Rashidi
atefeh.rashidi@ut.ac.ir

Shaghayegh Najari
najari.shaghayegh@ut.ac.ir

[1] Faculty of New Sciences and Technologies, University of Tehran, Tehran, Iran

[2] School of Computer Science, Institute for Research in Fundamental Science (IPM), P.O. Box 19395-5746, Tehran, Iran

Published online: 17 November 2024

high-dimensional data and diverse characteristics of spammers, leading to a decrease in spam detection speed. Therefore, it is necessary to investigate DL methods with efficient feature selection mechanisms to quickly detect large volumes of spam data.

The majority of existing approaches to spam detection fail to achieve optimal performance when faced with imbalanced datasets, where one class lacks in number compared to the other classes (Rao et al. 2023). They often overlook the challenge of class imbalance in real-world datasets (Liu et al. 2017). Various approaches have been proposed to address the issue of imbalanced data classification, including undersampling, oversampling, or a combination of both (López et al. 2013). Based on Table 2, it can be observed that less than 20% of the data in the benchmark datasets HSPAM (Sedhai and Sun 2015) and SMS (Almeida et al. 2011) belong to the spam category. Therefore, if the undersampling technique is employed, a significant portion of the data will be lost.

Detecting and blocking spam messages has become crucial to ensure user satisfaction, maintain security, and prevent harm in social media. Manual identification and removal of spam by experts are slow and inefficient due to the large volume and evolving styles of spam messages. Additionally, imbalanced datasets often lead to classifier bias toward majority classes (Xiaolong et al. 2019).

To tackle this issue, this study proposes a novel approach for spam detection in social media that utilizes Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) to handle imbalanced datasets effectively. GANs are a class of DL models that excel at generating realistic data samples that resemble the training data distribution. Traditional oversampling methods alter original data distribution, while GAN approximates real data distribution by focusing on overall features. Though not identical, generated samples align well with the learned distribution (Jiawei et al. 2023). Specifically, we employ a code-based GAN that operates with text features rather than text content, mitigating challenges associated with text generation due to the discrete nature of text. By leveraging the Wasserstein distance with gradient penalty, our model ensures stability and high-quality data generation. Integrating a joint distribution of real data and latent code enhances the discriminator's feedback, ultimately improving the classification performance using a CNN-BiLSTM architecture with an attention mechanism. Our evaluation uses imbalanced benchmark datasets. Finally, it was revealed that our proposed model, utilizing the balancing and data augmentation approach, outperformed the compared models regarding spam detection.

The main contributions of the presented study are as follows:

1. We customize the code-based GAN approach to address the issue of imbalanced datasets in the domain of spam detection.
2. We use the Wasserstein distance (Leo et al. 2023) with a gradient penalty instead of the Jensen-Shannon divergence (Menéndez et al. 1997) to improve the model's stability and the quality of generated data.
3. We incorporate a joint distribution of real data and its corresponding latent code as input to the discriminator. This architectural selection enhances the discriminator's ability to provide more effective feedback to both the encoder and the generator.
4. We improve the code-based GAN classifier using CNN-BiLSTM (Bhuvaneshwari et al. 2021) with an attention mechanism.

The structure of the paper is as follows: Sect. 2 reviews literature related to our work. In Sect. 3, we present our proposed model and scenarios, Sect. 4 presents the results and analysis, and finally, Sect. 5 concludes the paper and discusses potential future research directions.

2 Related works

Various methods have been proposed so far for detecting spam. Traditional methods such as List-based and Honeypot are time-consuming, cumbersome, and incompatible, while spammers frequently alter their message characteristics. List-based methods involve categorizing specific IPs or users as trusted in a whitelist and marking others as untrusted in a blacklist (Wu et al. 2017). HoneySpam (Hayati et al. 2009) is designed to record spammer behavior, and recorded information can be used in other methods for detecting and limiting spam. Honey-Profile (Lee et al. 2010) is used on Twitter to capture the profile features of spammers during suspicious activities. The spam detection methods of these categories have some limitations, such as feature incompatibility, scalability, and low speed (Zhang et al. 2019).

Therefore, more advanced techniques, such as machine learning and deep learning models, have been proposed. Deep learning methods are more suitable for feature extraction from text, images, videos, and audio (Barushka and Hájek 2018). In general, the classification of these features can be categorized as follows: Methods based on content and methods based on structural graphs. Content-based methods work on textual, multimedia, or metadata content, where metadata content includes account features and statistical features of the content (Rao et al. 2021).

In the realm of spam detection, various studies have been conducted that address the issue of data imbalance. In these studies, diverse methods have been employed to balance the datasets, including overestimation or underestimation.

Nevertheless, some of the research in this domain has not considered data balance and has engaged in analysis and modeling without modifying the original data distribution. Continuing with studies in the field of spam detection, we present a categorized overview Fig. 1.

2.1 Machine learning-based methods

By using text features and user features, and with the help of Random Forest (RF) models, Support Vector Machines (SVM), Gupta et al. (2018a) detected spam messages on the HSPAM dataset (Sedhai and Sun 2015). Mehmood et al. (2018) used TF-IDF features and an RF model to detect spam in the YouTube dataset (Alberto and Lochter 2017). Tajalizadeh and Boostani (2019) tackled the problem of unsupervised learning using DenStream clustering and the Naive Bayes (NB) classifier to cluster and categorize tweets. By using the supervised learning approach and various ML methods, Chen et al. (2015) inferred that the RF achieves the best results when dealing with imbalanced datasets. The mentioned methods do not balance the dataset, leading to biased model training where the classifier becomes overly focused on the majority class, resulting in poor performance and reduced accuracy in identifying minority class instances.

Studies Hosseinpour and Shakibian (2023) and Saxena et al. (2022) applied the SMOTE oversampling technique on TF-IDF vectors of textual data. Subsequently, machine learning algorithms were employed for spam and non-spam classification, including RF, Logistic Regression (LR), NB, and Decision Tree (DT). Yao et al. (2021) used a combination of the SMOTE and Random Under Sampler (RUS) techniques and applied RF, XGBoost, LightGBM, CatBoost, and Gradient Boosting Decision Tree for spam classification on fake reviews. Mustapha et al. (2020) investigated the influence of data distribution imbalance on the efficiency of the proposed XGBoost model for spam detection. Various techniques addressing data imbalance, such as Random over/under-sampling, SMOTE, Borderline SMOTE, ADASYN, SMOTENN, Tomek links, and SMOTE-Tomek links, were applied to the training dataset. The models were trained under identical experimental conditions and tested on a separate imbalanced test dataset. Lu et al. (2018) also proposed an ensemble DT classifier combined with undersampling techniques to balance the dataset for detecting spam in web content. Using features extracted from user reviews, Saumya and Singh (2018) introduced a model based on RF and SVM. They employed SMOTE and ADASYN techniques to balance the dataset. By combining the balancing methods of RUS and Borderline-SMOTE, Li et al. (2022) conducted spam detection on reviews. The analysis was performed on content, reviewer behavior, and deceptive score features. Kumar et al. (2023) employed the SMOTE-ENN technique along with ML methods to utilize user information and tweet features for distinguishing between spam and non-spam content.

The dataset balancing methods mentioned above typically do not consider the original data distribution and may produce synthetic data that differs from the real data distribution. Additionally, the synthetic samples may be overly simplistic or repetitive when dealing with multi-dimensional and complex data, leading to overfitting.

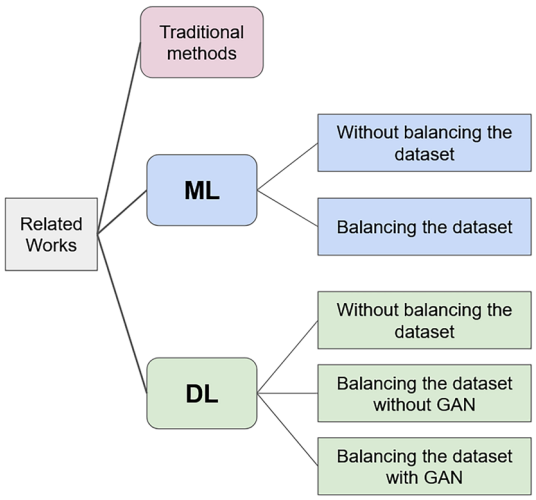


Fig. 1 Categorization of the literature review

2.2 Deep learning-based methods

Machine learning methods are ineffective and slow for spammers' high-dimensional data and diverse nature. To quickly detect spam with a massive volume of data, investigating deep learning methods with efficient feature selection mechanisms is essential (Barushka and Hájek 2018). Wu et al. (2017) achieved better results than ML methods by using a Multilayer Perceptron (MLP) neural network and Word2Vec embedding (Mikolov et al. 2013) on a dataset of Twitter. Jain et al. (2019) introduced three architectures that added a semantic layer using Word2Vec, WordNet, and ConceptNet to Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and a combination of the two neural networks and evaluated them on datasets containing SMS and Twitter messages. Madisetty and Desarkar (2018) proposed a model combining five CNNs with various word embedding techniques and an RF model based on user features. The approach given by Tida and Hsu (2022) employed a pre-trained BERT model (Devlin et al. 2018) on textual data for spam and non-spam classification.

Multimedia content in spam can include images, videos, or audio that have been replaced with images or audio of another person (Tolosana et al. 2020). DeepFakes (Artificial Multimedia Content) can be created using GAN.

Kumar et al. (2018) used a CNN to detect image-based spam (Deepimagespam). Bindu et al. (2018) utilized user, behavioral, content, and graph-based structures features to train DL models for detecting spammer communities. Liu et al. (2023) tackled the issues related to defining, approximating, and employing a novel subgroup structure for spam detection with Graph Neural Network (GNN). Song et al. (2021) leveraged a GNN classification model to identify spammers on social media platforms. They transformed individual user behavior patterns into graphs and extracted graph-based features for training the model.

The GloVe embedding (Global Vectors for Word Representation by Pennington et al. (2014)), NearMiss, and SmoteTomek methods were employed by Rao et al. (2023) for balancing textual datasets. A neural network with Conv1D and bidirectional RNN layers, along with an attention mechanism, was used for the classification model.

Conventional oversampling approaches modify the original data distribution by creating synthetic data points based on the distances between minority instances. In contrast, GAN approximates the genuine data distribution by prioritizing the overall characteristics. Generative Adversarial Networks (GAN) is a generative model introduced by Ian Goodfellow in 2014 (Goodfellow et al. 2014). GAN consists of two competing neural network models: a generator and a discriminator. The generator is responsible for producing fake data similar to the real dataset and learning the distribution of the real data. The discriminator is responsible for determining whether the input data is real or fake. In some common generative models such as Variational Autoencoder (Kingma et al. 2019) and some GANs like vanilla GAN, Jensen-Shannon divergence is used to measure the similarity between the real data and the generated data distributions. Wasserstein GAN (WGAN by Arjovsky et al. (2017)) uses the Wasserstein distance, a more informative measure of the distance between two probability distributions. This distance is also more amenable to gradient-based optimization and provides proper gradients for both the generator and the discriminator. One of the main advantages of WGAN over vanilla GAN is its stability during training. Wasserstein GAN with Gradient Penalty (WGAN-GP by Gulrajani et al. (2017)) is another variant of GANs that addresses some of the limitations of WGAN by adding a gradient penalty to the loss function of the discriminator. Compared to WGAN, WGAN-GP has been shown to provide more stable training, prevent mode collapse, and produce better-quality generated samples.

The SpamGAN architecture (Stanton and Irissappane 2019) was proposed for detecting spam comments. This study addressed the problem of the lack of labeled data and used a semi-supervised learning approach to solve it by employing GAN to generate fake text. Using improved conditional GAN, Wu et al. (2020) aimed to balance the imbalanced dataset of social bots using the number of labels, mentions, hyperlinks, retweets, followers, and following as features for generating synthetic samples. Tamimi et al. (2023) introduced Score_Gpt2ganto model for fake review detection using GAN, Generative Pre-trained Transformer (GPT) and reviews' scores.

Several studies have presented various taxonomies that categorize different aspects. We categorize them as summarized in Table 1.

3 Proposed framework

Our proposed model is based on the modified Adversarially Regularized Autoencoder (ARAE) framework by Zhao et al. (2018), tailored for generating synthetic text to increase the spam data, along with a classifier for detecting spam and non-spam text. The text generation framework consists of two main components: an autoencoder for generating continuous text features and reconstructing text based on the latent code and a GAN for generating synthetic text.

As depicted in Fig. 2, the CGANS (Code-based GAN for Spam) comprises an encoder, a decoder, a generator, a discriminator, and a classifier. Initially, the autoencoder is trained independently without considering the GAN component. The text labeled as "spam" ($X_{spam} \in P_*$) undergoes necessary preprocessing before being fed into the encoder $enc_\varphi : X \mapsto \mathcal{Z}$. The encoder generates a latent code representation $Z_{spam} \in P_Q$. Then, Z_{spam} is passed to the decoder $decp_\psi(\hat{X}|Z)$ to train the reconstruction of text from the latent representation. To enhance the stability of the autoencoder, noise $\hat{S}$ is added to the encoder's output.

$$\hat{X}_{spam} = \arg\max_X p_\psi(X|(enc_\varphi(X_{spam}), \hat{S})). \tag{1}$$

The parameters of the encoder and decoder are optimized by minimizing the cross-entropy reconstruction loss function, as specified in Eq. 2.

$$\min_{\varphi,\psi} \mathcal{L}_{rec}(\varphi, \psi) = -\log p_\psi(\hat{X}_{spam}|(enc_\varphi(X_{spam}), \hat{S})). \tag{2}$$

The generator g_θ takes noise data S sampled from a Gaussian distribution $\mathcal{N}(0, I)$ as input. It learns the distribution of the latent space representing the features of text with the "spam" label by receiving appropriate feedback from the discriminator $d_w(X, Z)$. The generator aims to mimic this learned distribution and generate a fake latent representation $\tilde{Z}_{spam} \in P_z$. The inputs to the discriminator consist of paired distributions: real data input and encoder output as real data (X_{spam}, Z_{spam}), and paired distributions of input and output of generator as fake data $(S, \tilde{Z}_{spam})$. Therefore, the discriminator receives not only the extracted features from

Table 1 Review on methods employed for spam detection problem

Methods	References	Year	Features				Tools
			Textual	Multimedia	Metadata	Network	
ML	Kumar et al. (2023)	2023	x	x	✓	x	SMOTE-ENN
	Hosseinpour and Shakibian (2023)	2023	✓	x	x	x	SMOTE; RF, LR, NB, DT
	Saxena et al. (2022)	2022	✓	x	x	x	SMOTE; RF, LR, NB, DT
	Li et al. (2022)	2022	x	x	✓	x	RUS, Borderline-SMOTE
	Yao et al. (2021)	2021	✓	x	x	x	SMOTE, RUS; LightGBM, XGBoost, CatBoost DT, RF
	Mustapha et al. (2020)	2020	✓	x	x	x	ADASYN, SMOTE, Tomek links; XGBoost
	Tajalizadeh and Boostani (2019)	2019	x	x	✓	x	DenStream; NB
	Lu et al. (2018)	2018	✓	x	x	x	under-sampling; DT
	Saumya and Singh (2018)	2018	x	x	✓	x	SMOTE, ADASYN; RF, SVM
	Gupta et al. (2018a)	2018	✓	x	✓	x	RF, SVM
	Rathore et al. (2018)	2018	x	x	✓	x	ML classifiers
	Singh et al. (2018)	2018	x	x	✓	x	LR
	Liu et al. (2017)	2017	✓	x	x	x	ML classifiers
	Singh et al. (2016)	2016	x	x	✓	✓	ML classifiers
	Chen et al. (2015)	2015	x	x	✓	x	RF
DL	Liu et al. (2023)	2023	x	x	x	✓	GNN
	Rao et al. (2023)	2023	✓	x	x	x	NearMiss; RNN, Conv1D
	Tamimi et al. (2023)	2023	✓	x	x	x	GAN, GPT
	Tida and Hsu (2022)	2022	✓	x	x	x	BERT-CNN
	Song et al. (2021)	2021	x	x	x	✓	GNN, RF
	Elakkiya et al. (2021)	2021	✓	x	x	x	CNN-BiLSTM
	Hao and Zhang (2021)	2021	x	x	x	✓	SDAEs, k-means
	Wu et al. (2020)	2020	x	x	✓	x	improved CGAN
	Stanton and Irissappane (2019)	2019	✓	x	x	x	SeqGAN
	Jain et al. (2019)	2019	✓	x	x	x	CNN, LSTM
	Mehmood et al. (2018)	2018	✓	x	x	x	TF-IDF, RF
	Madisetty and Desarkar (2018)	2018	✓	x	✓	x	CNN, RF
	Kumar et al. (2018)	2018	x	✓	x	x	CNN
	Ban et al. (2018)	2018	✓	x	✓	x	BiLSTM

the latent representations but also the original data itself to determine the authenticity of the input. Based on these inputs, the discriminator calculates the Wasserstein distance function W between the paired distribution of real and fake data and tries to maximize it.

$$\begin{aligned}\max_{w} \mathcal{L}_{dis}(w) = \mathbb{E}_{X_{spam}\sim P_*, Z_{spam}\in P_Q}[d_w(X_{spam}, \\ Z_{spam})] - \mathbb{E}_{\tilde{Z}_{spam}\in P_{\tilde{z}}}[d_w(S, \tilde{Z}_{spam})] \\ + \lambda \mathbb{E}_{\bar{X}\sim P_{\bar{X}}, \bar{Z}\sim P_{\bar{Z}}}[(\left\|\nabla_{\bar{X}} d_w(\bar{X}, \bar{Z})\right\| \\ - 1)^2],\end{aligned} \tag{3}$$

$\mathcal{L}_{dis}$ represents the loss function of the discriminator. The term before the summation sign represents the estimated Wasserstein distance, which should be maximized. To enforce the Lipschitz constraint, a gradient penalty term is added to the loss function. The penalty coefficient λ, with a default value of 10, controls the strength of the gradient penalty. $\bar{X}$ is a linear combination of real data X_{spam} and noise S. $\bar{Z}$ is a linear combination of the latent representation of real data and the generated latent representation. α is selected randomly between 0 and 1.

$$\bar{X} = \alpha X_{spam} + (1-\alpha)S. \tag{4}$$

$$\bar{Z} = \alpha Z_{spam} + (1-\alpha)\tilde{Z}_{spam}. \tag{5}$$

In the final stage of text generation, the discriminator trains the encoder and generator in an adversarial manner. The generator aims to minimize the Wasserstein distance between the real and fake latent representation distributions.

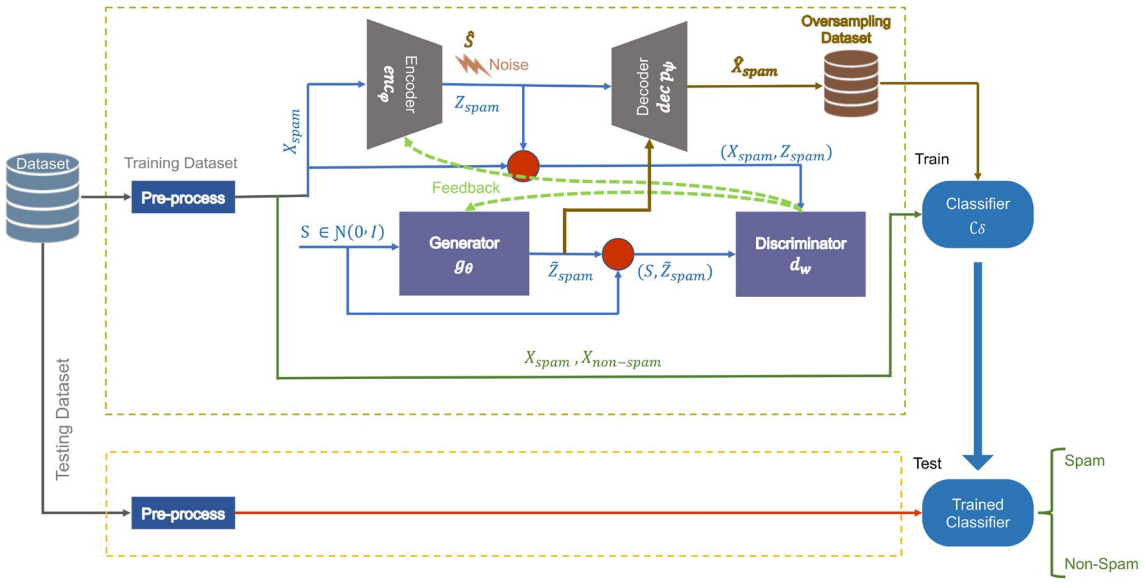


Fig. 2 The proposed CGANS framework

$$\min_{\varphi,\theta} \mathcal{L}_{enc,gen}(\varphi,\theta) = \mathbb{E}_{X_{spam}\sim P_*, Z_{spam}\in P_Q} [d_w(X_{spam}, Z_{spam})] - \mathbb{E}_{\tilde{Z}_{spam}\sim P_z}[d_w(S,\tilde{Z}_{spam})]. \quad (6)$$

The generated fake representations are sent to the decoder to reconstruct the fake text. Finally, the classifier $\mathcal{C}_\delta$ takes as input the expression $w_{1:t} \in D_U$, which consists of t discrete units of text labeled as spam or non-spam. D_U represents the union of the real and the fake dataset generated by the model ($D_{G_{spam}}$). The classifier then determines which class $c \in \mathbb{C}$ the input belongs to.

$$\min_{\delta} \mathcal{L}_{clf}(\delta) = -\sum_{1}^{2} y \log \mathcal{C}_\delta(w_{1:t}), \quad (7)$$

The function $\mathcal{L}_{clf}(\delta)$ represents the binary cross-entropy loss function of the classifier. y denotes the true label of the input $w_{1:t}$, and $\mathcal{C}_\delta(w_{1:t})$ represents the classifier's prediction. We utilize a bidirectional CNN-LSTM network with an attention mechanism to extract important features from the text. The CNN extracts local features and n-gram information, and the resulting sequence of extracted features is fed into a bidirectional LSTM to capture long-term dependencies. The algorithm 1 outlines the flow of our framework.

Algorithm 1 CGANS

Require: Real labeled dataset D_R, Input noise $S \sim \mathcal{N}(0, I)$, enc_φ, $decp_\psi$, d_w, g_θ, $\mathcal{C}_\delta$

Initialize enc_φ, $decp_\psi$, d_w, g_θ, $\mathcal{C}_\delta$ parameters with the random weights
Pre-process and tokenize $X \sim D_R$
for training-iterations **do**
  Sample X_{spam} from dataset D_R
  Generate $Z_{spam} = enc_\varphi(X_{spam})$
  BP $\mathcal{L}_{rec}(\varphi, \psi)$ in Eq. 2
  for discriminator-training-steps **do**
    Sample X_{spam} from dataset D_R
    Sample noise $S \sim \mathcal{N}(0, I)$
    Generate Z_{spam} and $\tilde{Z}_{spam} = g_\theta(S)$
    BP $\mathcal{L}_{dis}(W)$ in Eq. 3
  end for
  for encoder-generator-training-steps **do**
    BP $\mathcal{L}_{enc,gen}$ in Eq. 6
  end for
  for classifier-training-steps **do**
    Sample $(w_{1:t}) \sim (D_R \cup D_{G_{spam}})$
    $label = \text{sigmoid}(\mathcal{C}_\delta(w_{1:t}))$
    BP $\mathcal{L}_{clf}$ in Eq. 7
  end for
end for

4 Experimental results

In this section, we first examine the evaluated datasets. Then, we present the results of our CGANS framework in comparison with the SpamGAN (Stanton and Irissappane 2019), SSCL (Jain et al. 2019), and Bert-CNN (Tida and Hsu 2022) frameworks.

4.1 Datasets

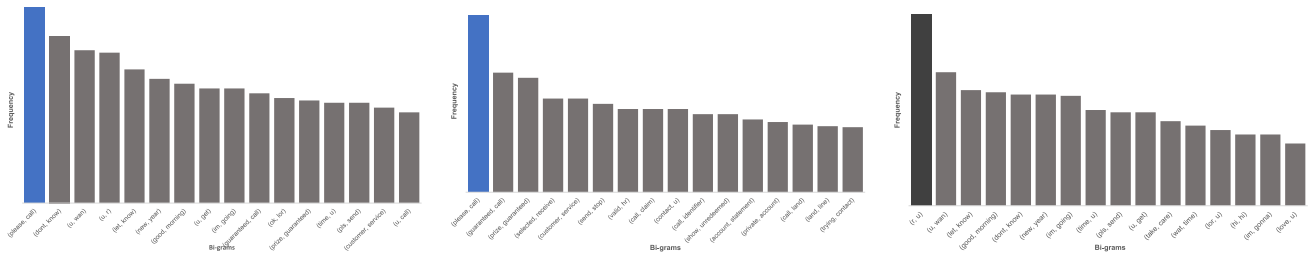
We evaluate our results using the HSPAM (Sedhai and Sun 2015) and SMS Collection datasets (Almeida et al. 2011). The HSPAM dataset consists of 14 million tweets. Approximately 10,680,000 tweets are labeled as non-spam, and 3,300,000 tweets are labeled as spam. However, some of these tweets are inaccessible due to reasons such as tweet deletion or account lockout of the publishing user. From the accessible tweets, we randomly selected and retrieved almost 20,000 tweets.

The distribution of the most frequent bi-grams in the SMS dataset is shown in Fig. 3. The most frequent bi-gram in the entire dataset (Fig. 3a) and the spam-labeled samples (Fig. 3b) are similar and correspond to "(please, call)". This indicates that the most frequently occurring bi-gram among the spam instances has been repeated to such an extent that it has also become the most frequent bi-gram in the entire dataset. Also, in Fig. 5a, the most common spam bi-grams are shown, with the size of each bubble representing the frequency of that corresponding bi-gram. The closeness of the bubbles to each other also indicates the semantic similarity between these bi-grams. The presence of closely located bubbles suggests the prevalent usage of synonymous words within the spam samples.

A similar observation is noted in the HSPAM dataset, as illustrated in Fig. 4. The figure presents the frequency distribution of bi-grams categorized by labels and the dataset as a whole. Notably, the bi-gram "(ive, collected)" appears most frequently in spam samples (Fig. 4b), and it also exhibits the highest count across the entire dataset (Fig. 4a). Similarly, in Fig. 5b, the presence of synonymous bi-grams can be observed by examining the closeness of the bubbles in the spam samples. Information of both datasets is presented in Table 2. As evident from it, the number of non-spam samples significantly outweighs the number of spam ones, indicating an imbalanced distribution in both datasets.

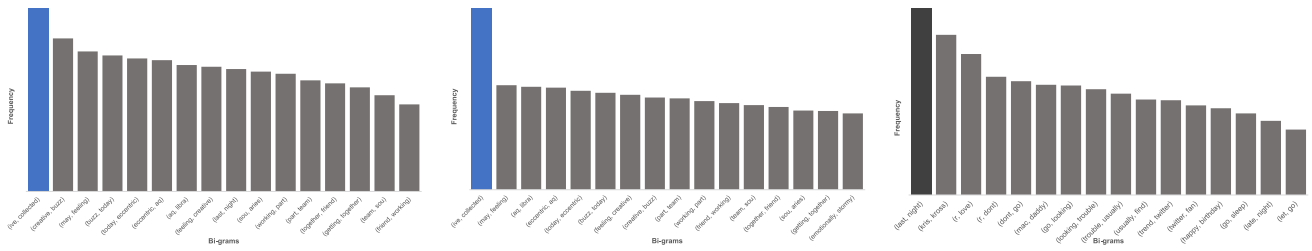
4.2 Preprocess

We preprocess and normalize tweets by applying GloVe (Pennington et al. 2014) embedding. We remove all punctuation marks and white spaces from the text. We replace *hashtags*, *email addresses*, *phone numbers*, *numbers*, *currency units*, *mentions*, and *URLs* with the corresponding tags



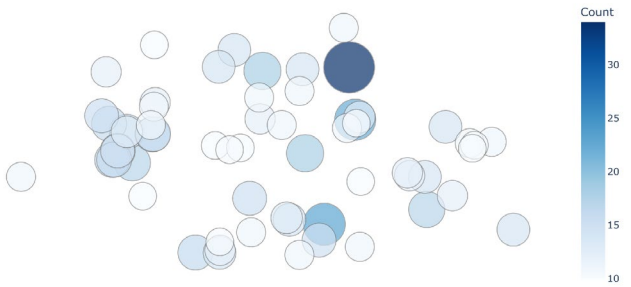
(a) Entire samples (b) Spam samples (c) Non-spam samples

Fig. 3 Distribution of the most frequent bi-grams in the SMS dataset (Almeida et al. 2011)

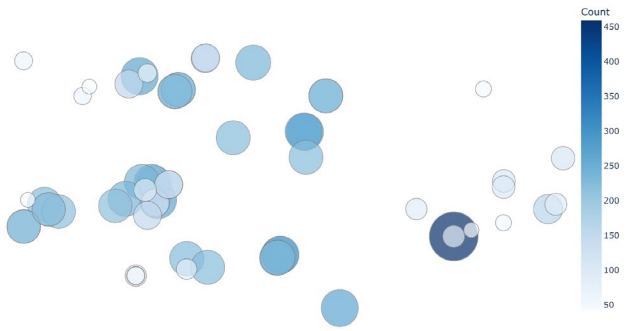


(a) Entire samples (b) Spam samples (c) Non-spam samples

Fig. 4 Distribution of the most frequent bi-grams in the HSPAM dataset (Sedhai and Sun 2015)



(a) SMS (Almeida et al, 2011)



(b) HSPAM (Sedhai and Sun, 2015)

Fig. 5 The number and similarity of the most frequent bi-grams present in spam samples

Table 2 Information of HSPAM and SMS

	Spam	Non-Spam	Total	% Spam
HSPAM	3664	17764	21428	17.1
SMS	747	4825	5572	13.4

<hashtag>, *<email>*, *<phone number>*, *<number>*, *<currency>*, *<mention>*, *<url>*. We also replace some abbreviations and colloquial expressions such as *lol*, *dnt*, and *omg* with their full forms like *laughing*, *do not*, and *oh my god* respectively.

Our approach utilizes an LSTM for both the encoder and decoder. We employ an MLP network with two hidden layers for the generator. The discriminator is implemented as a Recurrent Neural Network (RNN), and for the classifier, we utilize a CNN-BiLSTM network. To assess the performance of our model, we partitioned the dataset into three subsets: the training set, the test set, and the validation set, constituting 80, 10, and 10% of the total dataset size, respectively. Our study was implemented using the Python programming language and the Torch library, utilizing the Graphics Processing Unit (GPU) provided by Google. In our manuscript, we used ChatGPT (OpenAI 2024) to refine the language and structure of the text. Specifically, we utilized it to check for grammatical accuracy, consistency, and clarity. When using ChatGPT, we employed prompts such as "Please correct the grammar in this sentence", "Rephrase this paragraph to improve clarity", and "Suggest a more formal way to express this idea".

4.3 Impact of added noise on autoencoder stability

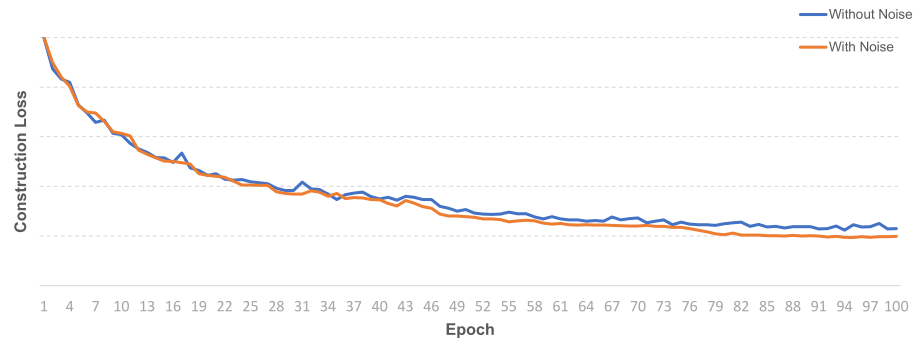
We trained the proposed model's autoencoder twice, with and without noise added to the encoder's output, to evaluate the performance and measure its stability. We compared the reconstruction error for both cases and for both datasets. Considering Fig. 6, the trend of the reconstruction error during training epochs is decreasing for both HSPAM and SMS datasets. Adding noise to the encoder output makes the reconstruction error lower than the case without noise. As a result, instead of precisely reconstructing from a single point in the latent space, the decoder learns to reconstruct from several nearby points in the latent space. It ensures that small changes in the latent space do not lead to drastic changes in the reconstructed output. Ultimately, this indicates the higher stability of the autoencoder in learning.

4.4 Evaluation of generated texts

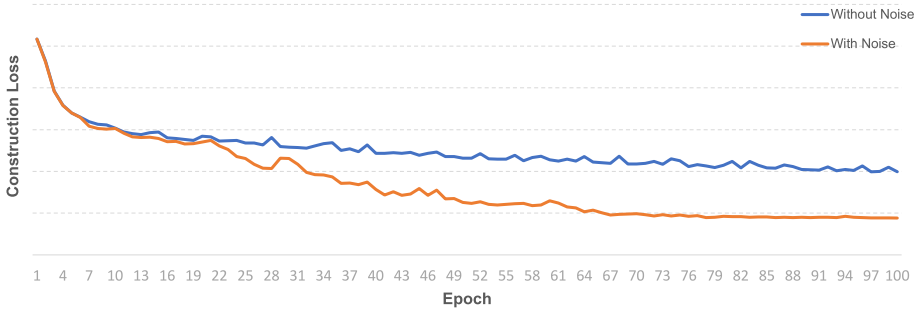
In this section, the quality of text generated by our proposed model is compared with two other models, Sequence Generative Adversarial Net (SeqGAN by Yu et al. (2017)) and ARAE (Zhao et al. 2018), based on perplexity (PPL) and forward perplexity (F-PPL) metrics.

In terms of dataset balancing, models with reduced perplexity tend to yield more coherent and meaningful text structures, thereby enhancing content generation quality. Considering Fig. 7b, in the SMS dataset, the perplexity score has improved in the SeqGAN model compared to the ARAE model, but there is no significant difference in perplexity between the SeqGAN model and our proposed model. However, in the HSPAM dataset (Fig. 7a), our proposed model has achieved a better perplexity compared to the other two models.

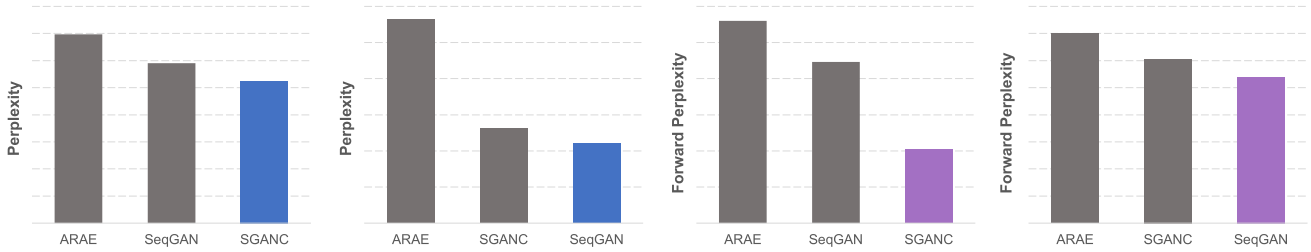
Forward perplexity is a metric used to evaluate the performance of a language model in predicting the next token in a sequence. Artificial datasets were generated using the three discussed models to compute the F-PPL. Subsequently, an RNN was trained on real datasets and evaluated on each of the artificially created datasets. Finally, the perplexity of RNN model was calculated and reported as the forward perplexity. Figures 7c and d demonstrate that our model's F-PPL has improved compared to ARAE in both datasets. In the SMS dataset, SeqGAN performed the best, achieving a lower and better F-PPL. However, both SeqGAN and



(a) HSPAM (Sedhai and Sun, 2015)



(b) SMS (Almeida et al, 2011)



(a) PPL on HSPAM (b) PPL on SMS (c) F-PPL on HSPAM (d) F-PPL on SMS

Fig. 6 Reconstruction loss of autoencoder per epochs with and without noise

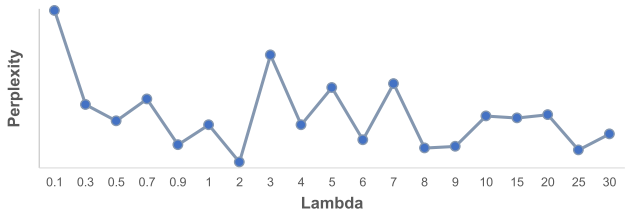
Fig. 7 Perplexity and forward perplexity of generated texts by SeqGAN (Yu et al. 2017), ARAE (Zhao et al. 2018) and CGANS

our proposed model performed relatively well on the SMS dataset.

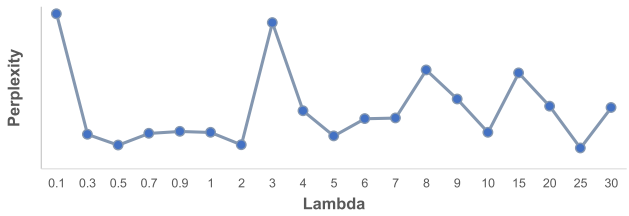
When the discriminator receives a joint distribution of the data and its latent representation, the model is able to generate more refined text. This approach provides the discriminator with both the original data and its abstract features, enabling it to offer more detailed and constructive feedback to the generator and encoder. Although SeqGAN outperforms our model in evaluation metrics such as PPL and F-PPL, it is notably less efficient due to its reliance on Reinforcement Learning and Monte Carlo search techniques for token generation. These additional steps result in a significantly longer execution time. Our model, on the other hand, offers a more efficient alternative without compromising much on the quality of the generated text.

4.4.1 Impact of penalty coefficient on quality of augmented dataset

In this part, we explored how varying the penalty coefficient λ affects the quality of text generated by our proposed model. Figures 8 and 9 illustrate the impact of different values of λ on the PPL and F-PPL metrics, respectively. Our findings indicate that changes in the λ parameter do not significantly alter these metrics, as the PPL and F-PPL values do not show a consistent pattern with either an increase or decrease in λ. This lack of a clear trend can be attributed to the inherent probabilistic nature of the PPL and F-PPL metrics. Additionally, it suggests that the model's stability and performance are relatively robust to the variations in

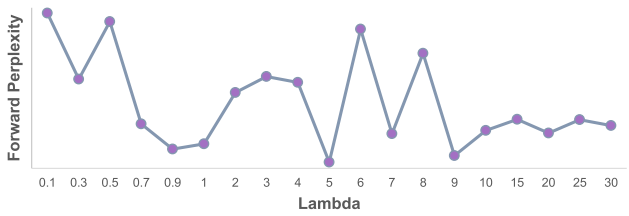


(a) HSPAM (Sedhai and Sun, 2015)

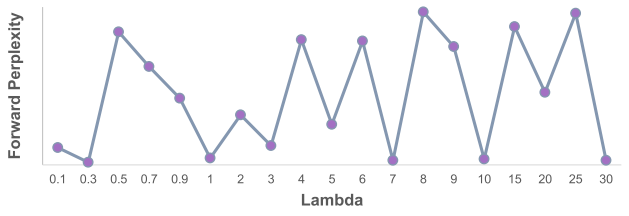


(b) SMS (Almeida et al, 2011)

Fig. 8 Impact of penalty coefficient lambda on perplexity (PPL)



(a) HSPAM (Sedhai and Sun, 2015)



(b) SMS (Almeida et al, 2011)

Fig. 9 Impact of penalty coefficient lambda on forward perplexity (F-PPL)

this regularization term, indicating a degree of resilience in handling different penalty settings.

4.5 Evaluation of spam classification

The proposed model in this study has been compared with SpamGAN-0% (100% labeled) (Stanton and Irissappane 2019), SSCL (Jain et al. 2019), and Bert-CNN (Tida and Hsu 2022) architectures in terms of evaluation metrics, including accuracy, F1-score, precision, and recall applied on two datasets, HSPAM and SMS. The results of the models' classification on the HSPAM and SMS datasets can be observed in Tables 3 and 4. The results indicate that our proposed model has achieved better accuracy and F1-score in spam classification of the HSPAM and SMS datasets regarding the balancing approach. We employ a CNN-BiLSTM architecture and an attention mechanism in our proposed classifier. This contrasts other classifier models that utilize CNN, LSTM, or a unidirectional combination of CNN and LSTM without attention. Furthermore, our proposed model uses a code-based approach for text generation, while SpamGAN relies on reinforcement learning-based methods.

Table 3 Evaluation of the accuracy, F1-score, precision, and recall metrics for SpamGAN, SSCL, Bert-CNN, and CGANS on HSPAM

HSPAM	Accuracy	F1-score	Precision	Recall
SpamGAN	0.873	0.878	0.895	0.862
SSCL	0.897	0.881	0.874	0.906
Bert-CNN	0.864	0.850	0.841	0.859
CGANS	**0.917**	**0.916**	**0.92**	**0.912**

Bolded values indicate that the respective model achieved the highest performance in the listedmetrics (Accuracy, F1-score, Precision, Recall) compared to other models

Table 4 Evaluation of the accuracy, F1-score, precision, and recall metrics for SpamGAN, SSCL, Bert-CNN, and CGANS on SMS

SMS	Accuracy	F1-score	Precision	Recall
SpamGAN	0.896	0.864	0.897	0.833
SSCL	0.972	0.915	0.936	0.88
Bert-CNN	0.945	0.887	0.904	0.871
CGANS	**0.982**	**0.93**	**0.952**	**0.909**

Bolded values indicate that the respective model achieved the highest performance in the listedmetrics (Accuracy, F1-score, Precision, Recall) compared to other models

5 Conclusions

In conclusion, we addressed the challenge of spam detection in social media using a novel deep learning approach with GAN. To tackle the issue of imbalanced datasets, our method uses a code-based GAN to generate synthetic spam samples, effectively mitigating data imbalance and reducing data augmentation time. The generated texts exhibit better quality as per PPL and F-PPL metrics. By incorporating the Wasserstein distance with a gradient penalty, we enhanced the GAN's stability and data quality. Our model also improves feedback through the joint distribution of real data and latent code input to the discriminator and employs

CNN-BiLSTM with an attention mechanism for better classification accuracy.

Experiments on benchmark datasets, HSPAM and SMS, showed our approach significantly outperforms recent deep learning models in accuracy and F1-score. The synthetic samples align well with real data, maintaining high performance even with imbalanced datasets. Key contributions include introducing a novel GAN-based oversampling method, improving model stability through advanced loss functions, and integrating sophisticated deep learning architectures for enhanced classification. These findings highlight GANs' potential to address data imbalance and improve spam detection accuracy in social media contexts.

Future research will explore applying our method to other imbalanced data classification problems, incorporating Graph Neural Networks (GNNs) for improved analysis, and assessing the impact of emojis using Emoji2Vec embeddings (Eisner et al. 2016) during text preprocessing. These advancements are expected to refine spam detection and contribute to more effective dataset balancing.

Acknowledgements Mostafa Salehi is supported by a grant from IPM, Iran (No. CS1403-4-32).

Author contributions These authors contributed equally to this work.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

Alberto T, Lochter J (2017) YouTube spam collection. UCI Mach Learn Repos 45:2. https://doi.org/10.24432/C58885

Almeida TA, Hidalgo JMG, Yamakami A (2011) Contributions to the study of SMS spam filtering: new collection and results. In: Proceedings of the 11th ACM symposium on document engineering, pp 259–262

Arjovsky M, Chintala S, Bottou L (2017) Wasserstein generative adversarial networks. In: International conference on machine learning, PMLR, pp 214–223

Ban X, Chen C, Liu S et al (2018) Deep-learnt features for twitter spam detection. In: 2018 International symposium on security and privacy in social networks and big data (SocialSec), IEEE, pp 208–212

Barushka A, Hájek P (2018) Spam filtering in social networks using regularized deep neural networks with ensemble learning. In: Artificial intelligence applications and innovations: 14th IFIP WG 12.5 International Conference, AIAI 2018, Rhodes, Greece, May 25–27, Proceedings 14, Springer, pp 38–49

Bhuvaneshwari P, Rao AN, Robinson YH (2021) Spam review detection using self attention based CNN and bi-directional LSTM. Multimed Tools Appl 80(12):18107–18124

Bindu P, Mishra R, Thilagam PS (2018) Discovering spammer communities in twitter. J Intell Inf Syst 51:503–527

Chen C, Zhang J, Chen X et al (2015) 6 million spam tweets: a large ground truth for timely twitter spam detection. In: 2015 IEEE international conference on communications (ICC), IEEE, pp 7065–7070

Devlin J, Chang MW, Lee K et al (2018) Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805

Eisner B, Rocktäschel T, Augenstein I et al (2016) emoji2vec: learning emoji representations from their description. arXiv preprint arXiv:1609.08359

Elakkiya E, Selvakumar S, Leela Velusamy R (2021) Textspamdetector: textual content based deep learning framework for social spam detection using conjoint attention mechanism. J Ambient Intell Humaniz Comput 12:9287–9302

Goodfellow I, Pouget-Abadie J, Mirza M et al (2014) Generative adversarial nets. Advances in neural information processing systems 27

Gulrajani I, Ahmed F, Arjovsky M et al (2017) Improved training of wasserstein gans. Advances in neural information processing systems 30

Gupta H, Jamal MS, Madisetty S et al (2018a) A framework for real-time spam detection in twitter. In: 2018 10th international conference on communication systems and networks (COMSNETS), IEEE, pp 380–383

Gupta S, Khattar A, Gogia A et al (2018b) Collective classification of spam campaigners on twitter: a hierarchical meta-path based approach. In: Proceedings of the 2018 world wide web conference, pp 529–538

Hao Y, Zhang F (2021) An unsupervised detection method for shilling attacks based on deep learning and community detection. Soft Comput 25(1):477–494

Hayati P, Chai K, Potdar V et al (2009) Honeyspam 2.0: profiling web spambot behaviour. In: Principles of practice in multi-agent systems: 12th international conference, PRIMA 2009, Nagoya, Japan, December 14-16. Proceedings 12, Springer, pp 335–344

Hosseinpour S, Shakibian H (2023) An ensemble learning approach for SMS spam detection. In: 2023 9th international conference on web research (ICWR), IEEE, pp 125–128

Jain G, Sharma M, Agarwal B (2019) Spam detection in social media using convolutional and long short term memory neural network. Ann Math Artif Intell 85(1):21–44

Jiawei N, Zhunga L, Quan P et al (2023) Conditional self-attention generative adversarial network with differential evolution algorithm for imbalanced data classification. Chin J Aeronaut 36(3):303–315

Kingma DP, Welling M et al (2019) An introduction to variational autoencoders. Found Trends ® Mach Learn 12(4):307–392

Kumar AD, KP S et al (2018) Deepimagespam: deep learning based image spam detection. arXiv preprint arXiv:1810.03977

Kumar C, Bharti TS, Prakash S (2023) A hybrid data-driven framework for spam detection in online social network. Proced Comput Sci 218:124–132

Lee K, Caverlee J, Webb S (2010) Uncovering social spammers: social honeypots+ machine learning. In: Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, pp 435–442

Leo J, Ge E, Li S (2023) Wasserstein distance in deep learning. Available at SSRN 4368733

Li S, Zhong G, Jin Y et al (2022) A deceptive reviews detection method based on multidimensional feature construction and ensemble feature selection. IEEE Trans Comput Social Syst 10(1):153–165

Liu J, Lyu Y, Zhang X et al (2023) Are your reviewers being treated equally? discovering subgroup structures to improve fairness in spam detection. arXiv:2204.11164

Liu S, Wang Y, Zhang J et al (2017) Addressing the class imbalance problem in twitter spam detection using ensemble learning. Comput Secur 69:35–49

López V, Fernández A, García S et al (2013) An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics. Inf Sci 250:113–141

Lu XY, Chen MS, Wu JL et al (2018) A novel ensemble decision tree based on under-sampling and clonal selection for web spam detection. Pattern Anal Appl 21:741–754

Madisetty S, Desarkar MS (2018) A neural network-based ensemble approach for spam detection in twitter. IEEE Trans Comput Soc Syst 5(4):973–984

Mehmood A, On BW, Lee I et al (2018) Spam comments prediction using stacking with ensemble learning. In: Journal of physics: conference series, IOP Publishing, pp 012012

Menéndez ML, Pardo J, Pardo L et al (1997) The Jensen–Shannon divergence. J Franklin Inst 334(2):307–318

Mikolov T, Chen K, Corrado G et al (2013) Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781

Mustapha IB, Hasan S, Olatunji SO et al (2020) Effective email spam detection system using extreme gradient boosting. arXiv preprint arXiv:2012.14430

Najari S, Salehi M, Farahbakhsh R (2022) Ganbot: a gan-based framework for social bot detection. Soc Netw Anal Min 12:1–11

OpenAI (2024) Chatgpt (2024 version) [large language model]. https://chat.openai.com, accessed: 2024-07-26

Pennington J, Socher R, Manning CD (2014) Glove: lobal vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543

Rao S, Verma AK, Bhatia T (2021) A review on social spam detection: challenges, open issues, and future directions. Expert Syst Appl 186:115742

Rao S, Verma AK, Bhatia T (2023) Hybrid ensemble framework with self-attention mechanism for social spam detection on imbalanced data. Expert Syst Appl 217:119594

Rathore S, Loia V, Park JH (2018) Spamspotter: an efficient spammer detection framework based on intelligent decision support system on facebook. Appl Soft Comput 67:920–932

Saumya S, Singh JP (2018) Detection of spam reviews: a sentiment analysis approach. CSI Trans ICT 6(2):137–148

Saxena B, Goyal S, Kumari A et al (2022) Boosting accuracy of fake review prediction using synthetic minority oversampling technique. In: 2022 international conference on computing, communication, and intelligent systems (ICCCIS), IEEE, pp 156–161

Sedhai S, Sun A (2015) Hspam14: a collection of 14 million tweets for hashtag-oriented spam research. In: Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval, pp 223–232

Singh M, Bansal D, Sofat S (2016) Followers or fradulents? an analysis and classification of twitter followers market merchants. Cybern Syst 47(8):674–689

Singh M, Bansal D, Sofat S (2018) Who is who on twitter-spammer, fake or compromised account? a tool to reveal true identity in real-time. Cybern Syst 49(1):1–25

Song Z, Bai F, Zhao J et al (2021) Spammer detection using graph-level classification model of graph neural network. In: 2021 IEEE 2nd international conference on big data. Artificial Intelligence and Internet of Things Engineering (ICBAIE), IEEE, pp 531–538

Stanton G, Irissappane AA (2019) Gans for semi-supervised opinion spam detection. arXiv preprint arXiv:1903.08289

Tajalizadeh H, Boostani R (2019) A novel stream clustering framework for spam detection in twitter. IEEE Trans Comput Soc Syst 6(3):525–534

Tamimi M, Salehi M, Najari S (2023) Deceptive review detection using GAN enhanced by GPT structure and score of reviews. 2023 28th International Computer Conference. Computer Society of Iran (CSICC), IEEE, pp 1–7

Tida VS, Hsu S (2022) Universal spam detection using transfer learning of BERT model. arXiv preprint arXiv:2202.03480

Tolosana R, Vera-Rodriguez R, Fierrez J et al (2020) Deepfakes and beyond: a survey of face manipulation and fake detection. Inf Fusion 64:131–148

Wu B, Liu L, Yang Y et al (2020) Using improved conditional generative adversarial networks to detect social bots on twitter. IEEE Access 8:36664–36680

Wu T, Liu S, Zhang J et al (2017) Twitter spam detection based on deep learning. In: Proceedings of the Australasian computer science week multiconference, pp 1–8

Xiaolong X, Wen C, Yanfei S (2019) Over-sampling algorithm for imbalanced data classification. J Syst Eng Electron 30(6):1182–1191

Yao J, Zheng Y, Jiang H (2021) An ensemble model for fake online review detection based on data resampling, feature pruning, and parameter optimization. IEEE Access 9:16914–16927

Yu L, Zhang W, Wang J et al (2017) Seqgan: sequence generative adversarial nets with policy gradient. In: Proceedings of the AAAI conference on artificial intelligence

Zhang Y, Zhang H, Yuan X et al (2019) Pseudo-honeypot: Toward efficient and scalable spam sniffer. In: 2019 49th Annual IEEE/IFIP international conference on dependable systems and networks (DSN), IEEE, pp 435–446

Zhao J, Kim Y, Zhang K et al (2018) Adversarially regularized autoencoders. In: International conference on machine learning, PMLR, pp 5902–5911