انجمن کامپیوتر ایران
Computer Society Of Iran

28th International Computer Conference

Computer Society of Iran

25rd and 26th Jan. 2023, Sharif University of Technology

Sharif University of Technology

PRESENTATION CERTIFICATE

This is to certify that the paper with title

Deceptive Review detection Using GAN enhanced by GPT structure and score of reviews

by

Maryam Tamimi and Mostafa Salehi and Shaghayegh Najari

has been presented in the conference.

Jafar Habibi
General Chair

Alireza Ejlali
Technical Chair

Deceptive Review detection Using GAN enhanced by GPT structure and score of reviews

Maryam Tamimi
Faculty of New Sciences and Technologies, University of Tehran
Tehran, Iran
maryam.tamimi@ut.ac.ir

Mostafa Salehi
Faculty of New Sciences and Technologies, University of Tehran
School of Computer Science, Institute for Research in Fundamental Science
Tehran, Iran
mostafa_salehi@ut.ac.ir

Shaghayegh Najari
Faculty of New Sciences and Technologies, University of Tehran
Tehran, Iran
najari.shaghayegh@ut.ac.ir

***Abstract*—These days almost every people use online platforms, some users check others' reviews before deciding to take any action. According to the importance of submitted reviews, they can cause business success or failure, so there would be an opportunity for deceptive review production, therefore detection and deletion of these reviews from online platforms will be an essential task. Different approaches from Machine Learning (ML) to the recent Neural Networks (NN) have been used to detect deceptive reviews. Along with the power of Generative Adversarial Networks (GAN) in generating data with a distribution close to original data, those have been used to detect deceptive reviews in recent years to obviate the lack of sufficient labeled data. Generative Pre-trained Transformers (GPT) that are powerful in text and language processing have also been used besides GAN architecture for deceptive review detection. Especially, in deceptive review detection, most of the works are based on just textual features without considering other metadata or behavioral features. In this paper, we have proposed a new approach called Score_GPT2GAN to consider the review's scores by the regularization concept to the GAN structure besides textual data. Evaluation results in comparison between our method and SpamGAN_GPT2 have shown an increase in the accuracy of 1.4% on the TripAdvisor dataset and 3.8% on the YelpZip dataset.**

Keywords—Deceptive review detection, Generative Adversarial Network (GAN), Generative Pre-trained Transformer (GPT), text feature, behavioral feature

I. INTRODUCTION

Along with the development of virtual commercial platforms and increasing deceptive reviews, their detection is more important. According to a study [1], the percentage of fake reviews on YelpZip increased from 5% to 25% from 2005 to 2016. In 2018, Washington Post[1] mentioned that 61% of reviews for electronics on Amazon are fake. These kinds of reviews on social media not only can compromise businesses but also can cause wrong decisions by customers. Moreover, deceptive reviews are submitted to deceit users, so it will be hard to discriminate between them, also the spammers may change the way of writing reviews each time so detection algorithms could not find them. Therefore, detecting, and removing deceptive reviews from social media is more complex.

A variety of methods for deceptive review detection have been presented till now, they started extracting behavioral and textual features (or both of them) from the reviews, then classified them based on these features and different machine learning algorithms. We have divided machine learning algorithms into two main categories: classical machine learning and neural network-based algorithms. In this paper, we have used neural network-based ones, because neural networks (NN) can extract the structure of texts automatically [2], [3].

Generative Adversarial Networks (GAN) [4] have been used in this area because they have performed well in Natural Language Processing (NLP) and classification tasks to solve the problem of the lack of labeled data. GANs can generate new data with the distribution close to the main data. For the first time, FakeGAN [5] used the GAN structure in this area. This work used two discriminators with Convolutional Neural Network (CNN) architecture and one generator with Recurrent Neural Network (RNN) architecture. Multiple new methods recently used the GAN structure in this area, such as SpamGAN_GPT2 [6] and ScoreGAN [7]. We want to use a combination of these two frameworks as our base models.

In this area, SpamGAN_GPT2 is a state-of-the-art framework that has two discriminators and one generator built up by a GPT2 structure in all components. The GPT2 is an improvement of the Generative Pre-trained Transformer (GPT) that uses an attention mechanism [8] in its structure. GPT is a pre-trained network, these kinds of networks have performed well recently in different NLP tasks.

FakeGAN and SpamGAN_GPT2 have used the texts of reviews for classification. ScoreGAN is another method that considered metadata and behavioral features in its structure. They have found, adding the behavioral features (precisely the score of reviews) to the textual features could improve the efficiency of the previous works. According to the power of GPT2 in the NLP tasks and the influence of considering the score of reviews in the ScoreGAN, we want to consider both of them in this work.

We further discuss the previous methods used for deceptive review detection in part II; we will divide these methods into two parts: classical ML-based and NN-based methods. Section III will explain our proposed method talking about the regularization concept, and the structure of the generator, classifier, and discriminator respectively. We will discuss the dataset and evaluation strategies, and experimental results in section IV. For the last part, in section V, we will summarize, conclude, and discuss future works.

In this paper, as our contribution, we have proposed a method that uses the scores of reviews besides the texts; these features help the generator to generate more realistic reviews, which cause better classification in the GAN structure. Our GAN structure includes three components that use GPT2 as a network in the generation, discrimination, and classification tasks.

[1] https://www.washingtonpost.com/business/economy/how-merchants-secretly-use-facebook-to-flood-amazon-with-fake-reviews/2018/04/23/5dad1e30-4392-11e8-8569-26fda6b404c7_story.html

II. Related works

In this section, we will divide deceptive review detection methods into two main parts.

A. Classical Machine Learning Approaches

Various classical machine learning approaches have been used to detect deceptive reviews; these approaches have used textual or behavioral features or a combination of them. For the first time in 2008, Jindal and Leo used supervised methods including Naïve Bayes (NB), Support Vector Machine (SVM), and Logistic Regression (LR) with review-centric, reviewer-centric, and product-centric features to create a classifier for this task [9]. Several methods have used Ngram, stylometric, and behavioral features besides SVM for the detection [10], [11], [12], [13]. Another research has used the Support Vector Network (SVN) and different feature extraction methods including Latent Dirichlet Allocation (LDA), Word Space Model (WSM), and LIWC for this task [14]. Khurshid et al [15] used textual features, primal features, and Term Frequency-Inverse Document Frequency (TF_IDF) with five classifiers including Random Forest (RF), NB, AdaBoost, JRIP, and J48 to detect deceptive reviews. Nilizadeh et al [16] have concentrated on anonymous changes in users' profiles and combined them with textual features; they used RF for classification.

B. Neural Network-based Methods

NNs performed well in NLP classification tasks [2], [3]. These models can extract nonlinear relations between different parts of a text as well as, their syntactic, and structures of them. Different NN approaches have been used to detect deceptive reviews; we will discuss some of the most important ones in this section.

CNNs are famous because of their power to find local information in data. Li et al [17] have used CNN in two stages to convert word vectors to sentence embedding, and then to document embedding, they have used the document embedding for each review to detection. [18] Used to combine behavioral features extracted from graph modeling with semantic features extracted by CNN, to detect deceptive reviews. DRI_RCNN [19] and RCNN [20], were used to combine CNN and Recursive models for this task.

RNNs are another type of NNs used; they have a suitable structure for sequential data structures. [21] Have used a three-layered, Multilayer Perceptron (MLP) network with Long Short Term Memory (LSTM) in the second layer for deceptive review detection. Jin et al [22] have combined CNN for feature extraction and Gated Recurrent Neural Network (GRNN) for extracting the order and relation between the words to create a deceptive model.

GAN [4] is another NN structure that includes two parts named generator and discriminator; they pre-train separately then compete with each other in a zero-sum game and help to improve each other. According to labeling challenges in the deceptive review detection area, a lack of labeled data would be a bottleneck; therefore, using GAN to generate data with the same distribution as the main data could be useful. In this regard, there are many works. Aghakhani et al [5] proposed FakeGAN that used one generator and two discriminators. The generator generates reviews, one of the discriminators helps the generator to generate deceptive reviews, and the other one is the classifier of the model. This approach uses LSTM in the generator and CNN in the discriminators' architecture. SpamGAN [23] is another structure that used GAN to augment labeled data; they used RNN in the generator and two discriminators. SpamGAN_GPT2 [6] proposed an improvement to SpamGAN, they changed their previous model by inserting GPT2 in the generator and discriminators instead of RNN. The performance of SpamGAN_GPT2 was better than SpamGAN, due to using GPT2 in its structure.

Most of the NN-based models used in this area have considered the text of the reviews as the input. ScoreGAN [7] used different metadata in the GAN structure besides the text. They added different metadata to the FakeGAN structure and considered their effect of them on efficiency. According to the challenges in the deceptive review detection area, and the efficiency of the SpamGAN_GPT2 structure; we proposed a method that adds the score of reviews (inspired by ScoreGAN) to the SpamGAN_GPT2 structure.

GPT2 is an attention-based [8] pre-trained language model that uses transformer architecture and is trained by an eight million web page dataset. Transformer structures are powerful in sequential data like text; they include encoder and decoder with attention mechanisms in their architecture that overcome the problem of converting long sentences to each other. Transformers performed well in different NLP tasks. GPT2 has about 1.5 billion parameters that are 10 times bigger than GPT1 [25]. GPT3 [26] is the last version that has 100 parameters more than GPT2, we could not use that because of the lack of processing resources. We used the smallest version of GPT2 as SpamGAN_GPT2 did, the generator includes 12 transformer encoders and the discriminator and classifier include 12 transformer decoders.

Most proposed methods in the deceptive review area that uses NNs in their structures have only used the texts of reviews. ScoreGAN [7] added metadata features to the GAN structure in this area using the regularization concept [27], they could outperform their previous method named FakeGAN. The study by ScoreGAN has shown that adding the score of reviews (among the others) to the text of reviews had the best performance. Therefore, in this paper, we added the scores of reviews to the GAN structure using the regularization concept; we have also used the power of GPT2 in our structure.

III. Proposed Method

As labeling as deceptive or real is crucial, the amount of labeled data is spare or the amount of labeled data with deceptive labels is tiny beside the real ones. NNs need a huge amount of data to work with; therefore, due to the power of GAN to generate data near the main data; we could use them to overcome these challenges.

GANs include a generator and discriminator. As is shown in "Fig. 1", our proposed model used two discriminators (we will name one of them as a classifier) and one generator, the same as the FakeGAN and SpamGAN structures. To describe our model, assume that D is the dataset used in this work (including labeled D_l and unlabeled D_u, $D = D_l \cup D_u$). Labeled data D_l include data with $deceptive$ and $non\_deceptive$ labels. We will name the generated data by the generator $fake$ and the rest data in the dataset D $real$. The generator will train using D and learn to generate data

similar to the $real$. The discriminator, on the other hand, will discriminate between $fake$ and $real$. The discriminator helps the generator to generate better by sending feedback to it. The classifier trains with D_l and learns to discriminate the difference between $deceptive$ and $non\_deceptive$ labels.

After the pre-training phase, these three networks will start the adversarial training phase. They play a zero-sum game, send feedback, and help each other to improve, the competition between these three elements causes generating sentences with better quality by the generator and better discrimination quality by the discriminator and classifier. The structure of networks in all three elements is the same as the SpamGAN_GPT2 and they are GPT2 [24].

We will start discussing our architecture by introducing how we added the scores to the structure using the regularization concept, then we will discuss the generator, discriminator, and classifier architecture.

A. *Regularization*

The main goal in GAN structures is to learn the main data distribution $P_{data}(x)$ to the generator. In the adversarial training phase, the generator tends to generate sentences from noises z and improve its generation with the feedback that receives from the discriminator. InfoGAN [27] proposed a regularization concept that adds $\lambda I(s; G(z,s))$ as a regulator to the objective function in the game to keep the constraint s effect during generation ($I(s; G(z,s))$ is the mutual information between the generated $G(z,s)$ sentences and s). The goal is to maximize the I function to achieve the information gain maximization between the scores and the generated reviews. According to the definition of entropy, the mutual information function is defined as follows:

$$I(s,G(z,s)) = H(s) - H(s|G(z,s))$$

$$= \mathbb{E}_{x\sim G(z,s)}[\mathbb{E}_{s'\sim P(S|x)}[\log P(s'|x)]] + H(s)$$

This equation cannot be computed due to the need for x before the generation of it in $P_G(s|x)$. InfoGAN used an auxiliary distribution named $Q(s|x)$ that uses the variational information maximization [28] to delete the posterior variable effect in this equation; they showed that adding the auxiliary distribution $L(G,Q)$ to the min-max equation between the discriminator and generator could help the generator, generate values with constraint s.

ScoreGAN used this concept to generate reviews with the constraint of different metadata; they tried to generate reviews with high information gain with scores and other metadata. The s could be the score of reviews submitted by the users and added to the min-max equation through the auxiliary distribution $Q(s|x)$. ScoreGAN added a fully connected layer to the output of the discriminator to compute the Q(probability that a review has a score of s). By adding the term $\lambda L(G,Q)$ to the objective function of the discriminator. ScoreGAN tries to build a model whose generator can generate reviews with s constraints; therefore, according to these definitions, the objective function of the discriminator is defined as follows. The discriminator tries to maximize it and sends feedback to the generator to help it generate sentences with constraints.

$$\max[\mathbb{E}_{x\sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z\sim p_z} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) + \lambda L(G,Q)]$$

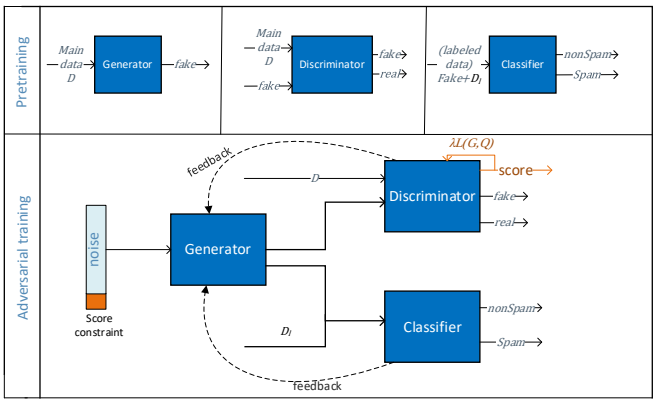

Fig. 1. Score_GPT2GAN framework, pre-training, and adversarial training inputs and outputs.

In practice, to implement this objective function and to calculate Q, a softmax layer is used. The output of the discriminator network is calculated by the softmax to estimate the probability that a review has a score of s, And it is added to the discriminator cost function with the coefficient λ. The softmax function in this implementation is as follows; Q is the probability that a review has a score of s, W is the weight matrix that shows the scores, b is the bias vector, S is the score interval (which here is 1 to 5), and f is the output of the discriminator network after the max-pooling layer.

$$Q = \frac{e^{Wf+b}}{\sum_{s=1}^{S} e^{W_s f+b}}$$

B. *Model Structure*

The proposed generator framework is similar to SpamGAN_GPT2 in architecture, with the difference that the constraints have been added to generate sentences with constraints. The GPT2 structure is used in the generator network to generate sentences. The generated sentences by the generator depend on the parameters of the network θ_g the noise vector z, the review label c, the information related to the position p, and the score constraint s. The z, s, and c values are added to the input at each time interval. This is done so that the class and score constraints for generating new reviews are preserved and have an effect on generating new reviews. The sentence generation process by the generator is the same as spamGAN_GPT2.

The discriminator used GPT2 architecture in its structure. The discriminator structure in input and critic layers is the same as spamGAN_GPT2. Two softmax functions, calculate the result of discrimination for each input. The discriminator estimates whether the input sentence is generated by the generator ($fake$) or the sentence in the main dataset ($real$). The discriminator does this by maximizing its objective function. We added a fully connected layer to the last layer of the proposed discriminator in the spamGAN_GPT2 model to include the scores of reviews (as described in the regularization section) by adding a softmax function. The classifier classifies reviews into $deceptive$ and $non\_deceptive$. The structure of the classifier and discriminator networks are the same with the difference that the softmax layer is not used to estimate the scores in the classifier.

ALGORITHM I. SCORE_GPT2GAN.

Input: labeled data D_l, unlabeled data D_u, $D = D_l \cup D_u$
Output: labeling each review as *spam* or *non-spam*
Parameters: $\theta_g, \theta_d, \theta_c, \theta_{d_{critic}}, \theta_{c_{critic}}$
#Pre-training
Pre-train: generator using D, discriminator using D and *fake* (generated by generator), classifier using D_l and *fake*
#Adversarial training
for `Training-epochs` do:
  for `G-Adv-epochs` do:
    generate sample with s constraint
    for t ∈ 1: T do:
      compute $U(x_{1:t}), V(x_{1:t-1})$
    update θ_g using policy gradient $\nabla_{\theta_g} L_G$ in Eq.3-7
  for `G-MLE-epochs` do:
    update θ_g using MLE
  for `D-epochs` do:
    sample from *fake* and D (*real*)
    update discriminator using Eq.3-2
    compute $U(x_{1:t}, x_t), V(x_{1:t-1})$ for *fake*
    update critic loss using Eq.3-5
  for `C-epochs` do:
    sample from *fake* and D_l
    update classifier using Eq.2-2
    compute $U(x_{1:t}, x_t), V(x_{1:t-1})$ for fake
    update critic loss using Eq.3-5
#Testing
Compute probability of reviews to be *spam* or *non-spam* in $D_l \cup$ fake
Compute generator perplexity

One of the challenges of using the GAN structure is the loss of the effect of changing network parameters, for this aim, many methods use reinforcement learning. In the GAN structure, the generator acts like an agent in reinforcement learning, receiving rewards from the discriminator (which force the generator to generate score-correlated reviews) and classifier. We have used the same method as the SpamGAN_GPT2 to calculate the cost function of our generator.

According to the explanations given in this section, "Algorithm I" has been used to train and test the described structure. As mentioned, in the pre-training phase, all three components in the GAN structure were pre-trained according to their definitions. Then, in the adversarial training phase, they compete with each other and update the weights of their networks. Finally, to evaluate the proposed structure, the efficiency of the classifier and generator networks was calculated.

IV. EXPERIMENTS AND RESULTS

In this section, we will compare and evaluate our proposed model with the two recent works. As mentioned earlier, the method named ScoreGAN, presented in 2020, used the concept of regularization to generate score-correlated reviews and added this concept to the GAN structure, which used LSTM and CNN in their components. Another method called SpamGAN_GPT2, presented in 2020, used the GAN, with GPT2 structure, and used only textual data for detection. Due to the similarity of the structure of these two methods with ours, we have compared these three together.

A. Experimental Setup

The generator in our structure includes 12 transformer decoders and the discriminator and classifier include 12 transformer encoders. This structure includes 12 multi-head masked self-attention layers with 768 units and 12 feed feed-forward with 3072 units. The presented model has used the Adam optimizer. The learning rate for the generator and discriminator was $6.25 * 10^{-6}$ and for the classifier it was $6.25 * 10^{-5}$. The value of weight decay for the generator assigned $1 * 10^{-7}$, for the discriminator $1 * 10^{-6}$, and for the classifier $1 * 10^{-5}$. The number of epochs was 10, 1, and 4 for the pre-train generator, discriminator, and classifier, respectively, and 5 for the adversarial phase.

TABLE I. ACCURACY PERCENTAGE FOR DIFFERENT PERCENTAGES OF LABELED AND UNLABELED DATA.

Method	Percentage of labeled data				
	10%	***30%***	***50%***	***70%***	***90%***
Score_GPT2GAN_0	88.54	83.18	86.55	87.38	87.54
Score_GPT2GAN_50	77.18	84.05	85.62	85.62	84.99
Score_GPT2GAN_70	76.5	84.99	85.3	86.24	82.18
Score_GPT2GAN_100	70.62	83.52	83.74	83.43	86.87
SpamGAN_GPT2_0	83.95	8176	85.1	86.45	87.56
SpamGAN_GPT2_50	77.1	79.37	84.37	84.68	84.37
SpamGAN_GPT2_70	73.74	80.93	83.43	82.18	86.4
SpamGAN_GPT2_100	66.24	80.01	79.99	84.68	84.7

To implement the proposed method, we used ColabPro[2]. There was a limit of 24 hours to run and a memory limit using it. Due to the large size of the network and the limitation of memory in ColabPro, it was impossible to save and restore the network after the pre-training phase. Therefore, we had to do all training and testing processes in one session; so, we inevitably could use a small percentage of the YelpZip [29] dataset for evaluation.

B. Dataset

The datasets used in this work for evaluation were TripAdvisor [10] and YelpZip. TripAdvisor dataset includes 800 deceptive and 800 non-deceptive labeled reviews. To increase its size, 32297 unlabeled data from reviews posted for Chicago hotels were used. The TripAdvisor dataset includes positive and negative semantic labels; it does not include ratings submitted by users. Therefore, in this dataset, to incorporate the reviews score features, we set a score of 5 for reviews with a positive semantic and a score of 1 for reviews with a negative semantic. The YelpZip dataset contains 608,598 restaurant reviews, with about 13% labeled as deceptive. To balance the dataset, we separated 40,219 deceptive and 40,219 non-deceptive reviews. The other 80,438 reviews in that dataset were considered unlabeled reviews for training, which had an equal number of spam and non-spam reviews. This dataset includes metadata features and submitted user scores from 1 to 5. Due to the stated limitation in resources, we could not use the entire YelpZip data in the training and testing process, and only 10% of labeled data and 0% of unlabeled data were used.

In both datasets, 80% of the data was used in the training and the remaining 20% was used in the test phase. In pre-processing phase for both datasets first, the letters in all reviews have been changed to lowercase, then the Byte Pair Encoding (BPE) method was used to represent the texts. The scores of reviews were extracted separately to use in the structure.

C. Results

1) The effect of labeled data on model performance

To compare the performance of the proposed method with previous works, we used a combination of labeled and unlabeled data from the TripAdvisor dataset with different percentages in the training phase. TABLE I shows how did we combine the labeled and unlabeled data in training and the accuracy criteria for each combination.

[2] https://colab.research.google.com

TABLE II. F1 PERCENTAGE FOR DIFFERENT PERCENTAGES OF LABELED AND UNLABELED DATA.

Method	Percentage of labeled data				
	10%	***30%***	***50%***	***70%***	***90%***
Score_GPT2GAN_0	77.85	81.31	86.64	88.12	85.84
Score_GPT2GAN_50	69.54	82.52	86.71	87.16	87.93
Score_GPT2GAN_70	71.64	82.66	86.71	86.91	84.25
Score_GPT2GAN_100	71.58	82.39	84.79	84.11	84.61
SpamGAN_GPT2_0	76.86	83.23	85.13	87.33	85.04
SpamGAN_GPT2_50	75.25	77.22	84.27	86.76	87.06
SpamGAN_GPT2_70	72.96	79.15	80.78	87.1	87.01
SpamGAN_GPT2_100	75.45	80.2	80.99	85.2	85.1

We used different percentages of labeled data including 10, 30, 50, 70, and 90, as training data to analyze the effect of labeled data on the performance of models, as an instance in columns named 30%, 30% of the labeled data was used. The validation data used for training was the remaining percentage of the training data, for example, where the training data was 70%, the validation data was 30%.

We added different percentages of unlabeled data to labeled data in the training phase and used a combination of labeled and unlabeled data to involve and investigate the effect of unlabeled data. According to TABLE I, which shows the accuracy percentage of the models in different percentages combination of labeled and unlabeled data; Score_GPT2GAN_0 means using 0%, Score_GPT2GAN_50, 50%, Score_GPT2GAN_70, 70%, and Score_GPT2GAN_100, 100% of the unlabeled data for training in our proposed model. In the same way, SpamGAN_GPT2_0, SpamGAN_GPT2_50, SpamGAN_GPT2_70, and SpamGAN_GPT2_100 are the percentage of using unlabeled data as 0, 50, 70, and 100 respectively in the SpamGAN_GPT2 model. Likewise, TABLE II shows the F1 criteria for both models on the TripAdvisor dataset with different percentages of labeled and unlabeled data. In these tables, the increase in the efficiency of both models can be seen with the increase in the percentage of labeled data. our proposed model is more efficient than its previous method in many cases.

"Fig. 2" has been drawn to compare Score_GPT2GAN and SpamGAN_GPT2 models side by side on the same amount of unlabeled data in accuracy percentages obtained in TABLE I. To better distinguish, the more the percentage of unlabeled data increases, the more the colors will be deep.

As can be seen in "Fig. 2", with the increase in the percentage of labeled data in all four charts, the accuracy of both models has increased (except for the use of 90% labeled data, which the reason is the decrease in the percentage of validation data). In all four charts and with the same volume of data, our model had a better performance.

To determine the best accuracy for an equal amount of labeled data in TABLE I, we have displayed the highest percentage obtained in each column in gray. As can be seen, our model has shown better performance. Among the different volumes of unlabeled data in both models, using 0% of unlabeled data had the best performance.

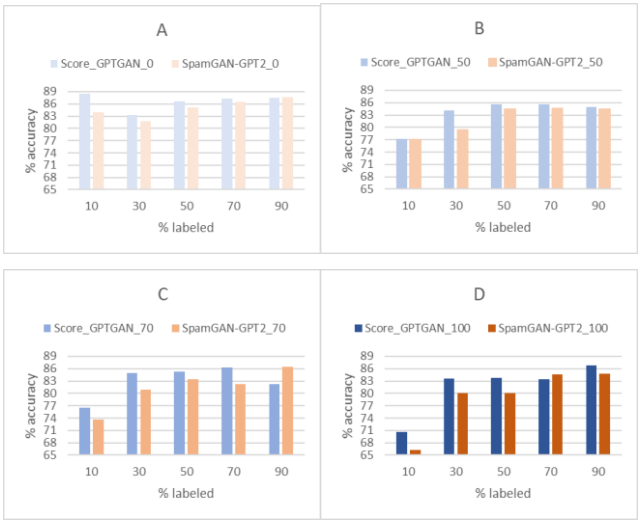

Fig. 2. Comparing the accuracy criteria for different percentages of labeled data and equal percentages of unlabeled data. A. 0% unlabeled data. B. 50% unlabeled data. C. 70% of unlabeled data. D. 100% unlabeled data.

As is shown in "Fig. 2", the figure related to Score_GPTGAN_0 and SpamGAN_GPT2_0, our model performed better in the majority of percentages of labeled data. Especially in using 10% of the labeled data, our model has a difference of 4.59% compared to the SpamGAN_GPT. This highlights the ability of our model to face the challenge of lack of labeled data since our model has been able to have better accuracy in detecting deception with a small amount of labeled data.

2) The effect of unlabeled data on model's performance

To investigate the effect of unlabeled data on the accuracy criteria of the Score_GPT2GAN and SpamGAN_GPT2 on the TripAdvisor dataset, the diagrams of "Fig. 3. A, B" have been drawn from TABLE I. Each figure is a comparison between different unlabeled percentages of data. Both "Fig. 3. A, B" have shown that with the increase in the percentage of unlabeled data, the accuracy of the model has decreased. The reason is that due to the difference in the volume of labeled and unlabeled data, the generator cannot properly learn the distribution of labeled data and generates sentences that are more similar to unlabeled data. Since the data used as labeled and unlabeled are different in terms of distribution so the generator will most likely produce data with a different distribution than the original data, this will not help to improve the classifier in the adversarial training process. The best performance in the accuracy criterion in both models according to "Fig. 3. A, B" is when the model is trained using zero percent of unlabeled data. Furthermore, "Fig. 3. C" was drawn to compare the accuracy of the best of each model with training on zero percent of the unlabeled data. As it is shown in this figure, our proposed model has a better performance than the SpamGAN_GPT2.

3) Perplexity of generated sentences

"Fig. 3. D" shows the best of each model in perplexity criterion for the proposed model and the SpamGAN_GPT2 method in different percentages of labeled and 50% of unlabeled data on the TripAdvisor dataset. Since both models use the GPT2 structure the perplexity criteria for both models are almost close to each other. Moreover, the perplexity for both models decreased with the increase in the percentage of unlabeled data. As expected, the generator was

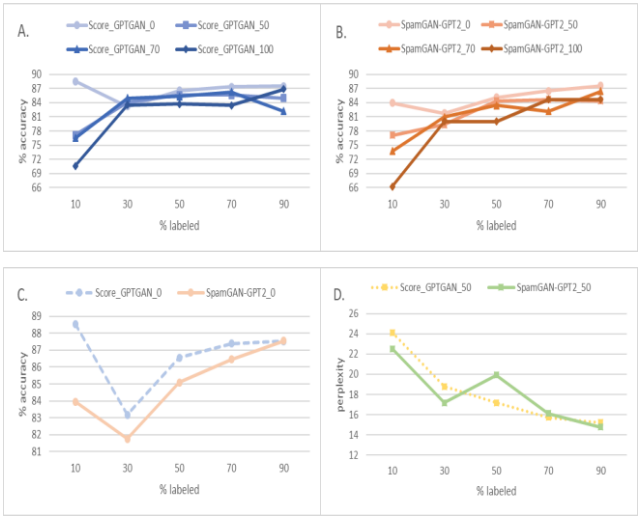

Fig. 3. A. The effect of unlabeled data on the accuracy criterion of the Score_GPT2GAN model, B. The effect of unlabeled data on the accuracy criterion of the SpamGAN_GPT2 model, C. Comparison of the best of both methods in the accuracy criterion for, D. Comparison of the best of both methods in the perplexity of different percentages of labeled data.

able to produce more realistic reviews using larger amounts of data.

4) Comparison of Score_GPT2GAN, ScoreGAN, and SpamGAN_GPT2

TABLE III compares Score_GPT2GAN with ScoreGAN and SpamGAN_GPT2 on the two datasets of TripAdvisor and YelpZip. The data used for all three methods have been the same; the volume of unlabeled data for all three models was equal to zero. In the models, Score_GPT2GAN and SpamGAN_GPT2 for the TripAdvisor dataset, 70% of the training data is used for training and the remaining 30% as validation data, and in the YelpZip dataset, 10% of the training data is used as the validation data.

"Fig. 4" compares all three methods in accuracy and F1 criteria on both datasets. As can be seen in this figure, our proposed method has a better performance than the other two methods. According to this figure, there is a big difference between ScoreGAN and the two other methods, which can show the effect of GPT2 compared to other NN methods used in ScoreGAN.

Our method has a 1.4% improvement on the TripAdvisor dataset and a 3.8% improvement on the YelpZip dataset on the accuracy criterion compared to the SpamGAN_GPT2. This indicates the effect of the score feature in improving the model's performance. As mentioned above, the scores in the YelpZip dataset are the exact scores submitted by users and

TABLE III. PERFORMANCE OF MODELS, SCOREGAN, SPAMGAN_GPT2, AND SCORE_GPT2GAN IN ACCURACY AND F1 CRITERIA ON TWO DATASETS.

dataset	*method*	*Accuracy*	*F1*
TripAdvisor	Score_GPT2GAN_0	88.38	88.12
	SpamGAN_GPT2_0	86.94	87.33
	ScoreGAN	80.26	63.8
YelpZip	Score_GPT2GAN_0	64.17	65.03
	SpamGAN_GPT2_0	60.38	61.65
	ScoreGAN	54.58	41.45

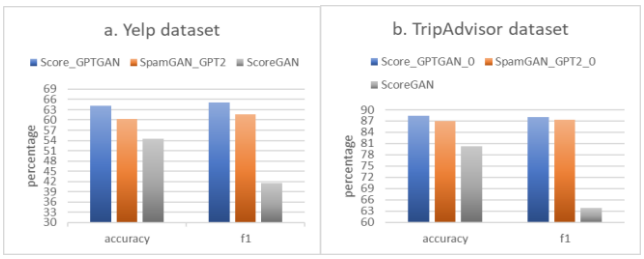

Fig. 4. Comparing the performance of three models in accuracy and F1 criteria. a. Comparison on YelpZip, b. Comparison on the TripAdvisor.

range from 1 to 5, but the scores used for the TripAdvisor data show semantics. Therefore, the effect of the real scores that were presented in the YelpZip dataset is evident in the difference in the efficiency of our method in the two datasets.

5) Performance of the classifier during the training

To check the performance of the classifier during the training process, the accuracy of the classifiers of both models was calculated on the validation data, "Fig. 5" was drawn based on this for the TripAdvisor dataset. As it is illustrated in this figure, the classification accuracy of our proposed model increased earlier than the SpamGAN_GPT2 and approached the ideal number. This indicates that the use of the score feature can create a better classifier with a smaller number of executions, so considering the resource limitations, using our model with a smaller number of executions can be useful.

V. CONCLUSION

We presented a structure to detect deceptive reviews, which, in addition to using the text of reviews, also uses their metadata. The framework used in this work uses the GAN structure, which takes advantage of the power of GPT2. The use of GPT2 for its efficiency in analyzing the text and adding review scores to it has improved the performance of the proposed method in terms of accuracy and F1 compared to previous methods. During the evaluation, our proposed method had an accuracy of 88.3 on the TripAdvisor dataset, while the accuracy was 86.94 and 80.26 for the SpamGAN_GPT2 and ScoreGAN methods, respectively. Moreover, the accuracy for YelpZip data for Score_GPT2GAN, SpamGAN_GPT2, and ScoreGAN models was 64.17, 60.38, and 54.58 respectively. Therefore, combining metadata with texts of reviews can be used in building more powerful models for detection. GPT2 has been used in this work, which has good functionality in the NLP tasks, due to the limitation of resources; working with that was challenging. Therefore, our suggestion is to use other lighter pre-trained networks (such as XLM) than GPT2. The framework presented in this article, with slight changes, could also be used in different classification multilingual tasks. The score for the reviews generated by the generator was considered average in our model; generating metadata in addition to generating reviews can be considered.

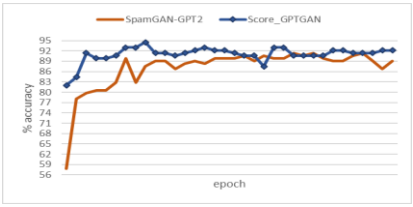

Fig. 5. Classifiers' performance of two models on validation data.

ACKNOWLEDGMENT

Mostafa Salehi was supported by a grant from IPM, Iran (No. CS1401-4-162).

REFERENCES

[1] M. Luca and G. Zervas, "Fake it till you make it: Reputation, competition, and yelp review fraud," *Manage. Sci.*, vol. 62, no. 12, pp. 3393–3672, 2016.

[2] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural Language Processing (Almost) from Scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, 2011.

[3] Y. Ren, R. Wang, and D. Ji, "A topic-enhanced word embedding for Twitter sentiment classification," *Inf. Sci. (Ny).*, vol. 369, pp. 188–198, 2016.

[4] I. Goodfellow *et al.*, "Generative adversarial networks," in *Neural Information Processing Systems*, 2014, pp. 2672–2680.

[5] H. Aghakhani, A. MacHiry, S. Nilizadeh, C. Kruegel, and G. Vigna, "Detecting deceptive reviews using generative adversarial networks," in *Proceedings IEEE Symposium on Security and Privacy Workshops (SPW)*, 2018, pp. 89–95.

[6] A. A. Irissappane, H. Yu, Y. Shen, A. Agrawal, and G. Stanton, "Leveraging GPT-2 for Classifying Spam Reviews with Limited Labeled Data via Adversarial Training," *Submiss. J. Artifical Intell. Res. (JAIR),arXiv Prepr. arxiv-2012.13400*, pp. 1–26, 2020.

[7] S. Shehnepoor, R. Togneri, W. Liu, and M. Bennamoun, "ScoreGAN: A Fraud Review Detector based on Multi Task Learning of Regulated GAN with Data Augmentation," *arXiv preprint arXiv:2006.06561*. 2020.

[8] A. Vaswani *et al.*, "Attention is all you need," in *31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017, pp. 5999–6009.

[9] N. Jindal and B. Liu, "Opinion Spam and Analysis," in *Proceedings of First ACM International Conference on Web Search and Data Mining (WSDM-2008)*, 2008, pp. 219–230.

[10] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock, "Finding deceptive opinion spam by any stretch of the imagination," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, 2011, p. 209_319.

[11] J. Li, M. Ott, C. Cardie, and E. Hovy, "Towards a general rule for identifying deceptive opinion spam," in *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014*, 2014, vol. 1, pp. 1566–1576.

[12] A. Mukharajee, V. Venkataraman, L. Bing, and G. Natalie, "What yelp fake review filter might be doing?," in *Association for the Advancement of Artificial Intelligence*, 2013, pp. 409–418.

[13] S. Shojaee, M. A. A. Murad, A. Bin Azman, N. M. Sharef, and S. Nadali, "Detecting deceptive reviews using lexical and syntactic features," in *International Conference on Intelligent Systems Design and Applications, ISDA*, 2013, pp. 53–58.

[14] Á. Hernández-Castañeda, H. Calvo, A. Gelbukh, and J. J. G. Flores, "Cross-domain deception detection using support vector networks," *Soft Comput.*, vol. 21, no. 3, pp. 585–595, 2017.

[15] F. Khurshid, Y. Zhu, C. W. Yohannese, and M. Iqbal, "Recital of supervised learning on review spam detection: An empirical analysis," in *Proceedings of the 2017 12th International Conference on Intelligent Systems and Knowledge Engineering, ISKE 2017*, 2017, pp. 1–6.

[16] S. Nilizadeh, H. Aghakhani, E. Gustafson, C. Kruegel, and G. Vigna, "Think Outside the Dataset: Finding Fraudulent Reviews using Cross-Dataset Analysis," in *International World Wide Web Conference Committee*, 2019, pp. 3108–3115.

[17] L. Li, W. Ren, B. Qin, and T. Liu, "Learning document representation for deceptive opinion spam detection," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, pp. 393–404, 2015.

[18] X. Wang, K. Liu, and J. Zhao, "Handling cold-start problem in review spam detection by jointly embedding texts and behaviors," in *55th Annual Meeting of the Association for Computational Linguistics*, 2017, pp. 366–376.

[19] W. Zhang, Y. Du, T. Yoshida, and Q. Wang, "DRI-RCNN: An approach to deceptive review identification using recurrent convolutional neural network," in *Information Processing and Management*, 2018, vol. 54, no. 4, pp. 576–592.

[20] J. Z. Siwei Lai, Liheng Xu, Kang Liu, "Recurrent convolutional neural networks for text classification," in *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI-2015)*, 2015, pp. 2267–2273.

[21] C. C. Wang, M. Y. Day, C. C. Chen, and J. W. Liou, "Detecting spamming reviews using long short-term memory recurrent neural network framework," in *ACM International Conference Proceeding Series*, 2018, pp. 16–20.

[22] N. Jain, A. Kumar, S. Singh, C. Singh, and S. Tripathi, "Deceptive Reviews Detection Using Deep Learning Techniques," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, pp. 79–91.

[23] G. Stanton and A. A. Irissappane, "GANs for semi-supervised opinion spam detection," in *IJCAI International Joint Conference on Artificial Intelligence*, 2019, pp. 5204–5210.

[24] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," *OpenAI blog*. 2019.

[25] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-Training," *OpenAI blog*. 2018.

[26] T. B. Brown *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems, arXiv preprint arXiv:2005.14165*, 2020.

[27] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *30th International Conference on Neural Information Processing Systems*, 2016, pp. 2180–2188.

[28] D. Barber and F. Agakov, "The IM algorithm : A variational approach to information maximization," in *Advances in Neural Information Processing Systems*, 2004.

[29] S. Rayana and L. Akoglu, "Collective opinion spam detection: Bridging review networks and metadata," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 985–994.