Variational Autoencoder Based Enhanced Behavior Characteristics Classification for Social Robot Detection

Xiaolong Deng[1](✉), Zhengge Dai[1], Mingdong Sun[2], and Tiejun Lv[1]

[1] Key Lab of Trustworthy Distributed Computing and Service of Education Ministry,
Beijing University of Posts and Telecommunications, Beijing 100876, China
shannondeng@bup.edu.cn, {daizhengge,lvtiejun}@bupt.edu.cn
[2] BeiJing THUNI Soft Corporation Limited, Beijing 100084, China
sunmd@thunisoft.com

Abstract. With the development of Internet and online social communication tools, malicious social bots have become a problem which cannot be ignored. They are intentionally manipulated by some organizations or people and always disseminate malicious information on the Internet, which greatly impact network environment. As a result, the detection of malicious social bots has become a hot topic in machine learning and many kinds of classification methods are widely used to detect social bots. However, the current classification methods are limited by the unbalanced dataset in which the samples of human users always outnumber samples of social bots. Given by the nature of binary classification problem of detecting social bots, the imbalance of the dataset greatly impacted the classification accuracy of the social bots. Aiming to promote the classification accuracy, this article has proposed the use of Variational Autoencoder (VAE) to generate samples of social bots basing on existing samples and thus mitigate the problem of imbalance dataset before feeding the original dataset into classifiers. In this way, the classification accuracy of social bots and normal human users can be efficiently improved basing on selected six categories of user account features. In order to verify the advantage of VAE in generating social bot samples, other traditional oversampling algorithm named SMOTE and SMOTE based SVM are also used to generate samples of social bots and compared in the contrast experiments. The experimental results indicated that the proposed method in this article has achieved better classification accuracy compared with other scenarios using the same datasets.

Keywords: Malicious information · Social bots · Classification accuracy · Dataset training · Machine learning

1 Introduction

With the development of the Internet, social network has become a significant part of ours daily life. Generally speaking, every social network account corresponds to one user and this user uses his social network account to connect with others. However, there are a

S. Yu et al. (Eds.): SPDE 2020, CCIS 1268, pp. 232–248, 2020.
https://doi.org/10.1007/978-981-15-9129-7_17

large number of social network accounts which are not used by real human beings. They are named social robots a.k.a. social bots. Social bots are usually manipulated by some software systems which are designed to control those social bots. The primary goals of maintaining these accounts are to disperse malicious information, which could be advertisement, malicious public opinion, fake news or even illegal contents. As a result, it is very important to identify these social bots in social network so that the service provider of social network can restrict the malicious behaviors of these social bots.

In this case, methods focus on detecting the social bots from a large dataset are proposed by researchers. Because the process of deciding whether a social network account is a robot or a human being is a binary classification problem, the final classification accuracy of these methods are largely depended on balance of the positive and negative samples of the dataset, which means the ratio of social bots and human users. However, the number of human being users often greatly outnumbers the one of social bots. In this scenario, this article proposes the use of Variational Autoencoder (VAE) to address the imbalance problem of the dataset.

The main contribution of this article is to use VAE to learn the statistical attribution of social bots in probability space and then uses these high-level features to generate new samples of social bots while given some random noise input. In this way, the imbalance of the dataset is mediated and the final classification accuracy of social bots' detection is improved. The whole process consists of the following operations: taking the minority classes from the Twitter user Dataset as the input of VAE; the encoder component of VAE taking the projecting the data into high dimension vector space; the decoder adding random noise to the features in high dimension space as initial value; the decoder uses these initial values to projecting them into Twitter social account data. In this way, some new social robots accounts data are generated. Then these data are added into the original dataset. In this article, a neural network technology was used as the classifier of the revised dataset. It will take the revised dataset, which include half social bots and half normal human twitter account, as training dataset. Then a random testing sub-dataset will be selected from the original dataset to test the effectiveness of the classifier. In order to prove the performance of this article, the classifier will also be trained on (1) original dataset with unbalanced human and social bots' samples (2) revised dataset which is added social bots samples generated by another popular oversampling algorithm. By proving the improvement of the neural network detection accuracy using VAE over the two scenarios as mentioned above, the effectivity of VAE can be proved.

The designed whole process of social bots detection can be found in Fig. 1. Variational Autoencoder is used to generate new social bots samples and built a balanced dataset before classifier training.

2 Related Work

The research on detecting online social bots basically began in 2010. At that time, the basic machine learning models and algorithms were used in it. For example, in 2010, Chu and Gianvecchio [1] firstly established a standard detection process to detect social bots with a set of measurements over 500,000 twitter accounts dataset. Their detection process including four parts: an entropy-based component calculating the latent regularity of the

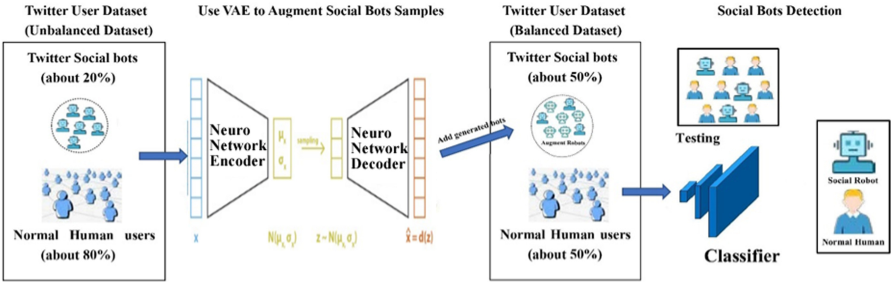


Fig. 1. Detection system design overview

twitter posted by a particular user, a machine-learning-based component classifying social bots from overall twitter dataset, an account properties component which can decide the personal information of an account, and a decision maker able to decide whether the input twitter account is a normal human. It also firstly used the features extracted from twitter user account to determine whether the account is a normal human user or a social bot. In 2011, Yang and Wilson [2] deployed the support vector machine (SVM) for uncovering social network social bots based on account data in the Renren Online Social Network. It utilized the user network features including average ratio of the number of incoming friend requests accepted and outgoing friend requests which are sent in one hour and collusion coefficient between different large groups. In this paper, Sybil graph and novel types of Sybil features were firstly applied to online social network.

From 2012, several papers focused on using social graph to depict the social network and using graph related algorithms to classify the social bots of online social network. For instance, in 2012, aiming to detect fake accounts in large scale social network, Cao and Sirivianos [3] designed a tool named SybilRank which marked impersonation possibilities of users by using social graph attributes. This social graph properties based algorithm will give each user a score according to their similarity of being sake and proved to be reliable under a large-scale data canter. In 2013, Beutel and Xu [4] focused on analyzing social graph between Facebook users and pages as well as the times at which the edges (links by other users) in the graph were established. They proposed two new algorithms to distinguishing the malicious behaviour according to graph structure and edge constraints.

From 2015, with the application of much more machine learning algorithms and mathematic models in this area, researchers transfers their researching focal point to the effectivity of detecting process by using less features of Twitter accounts. In 2015, Zafarani and Liu [5] derived a classification method with limited information given the realistic situation that the information of social bots may vary. It shows that lower to 10 bits data is sufficient to help decide whether an account is social bot. In this paper, the features of account are divided into five categories. The classification framework is proved to be robust under different datasets and algorithms. However, with the development of social bots on imitating normal human users and the increasing complexity of online social network, social bots are becoming more and more hidden and different to distinguish from normal human users. In 2016, Clark and Williams held the view that former detection methods which mainly based on user metadata cannot achieve great accuracy when facing social bots with powerful capacities [6]. They proposed to utilize the content of tweet as a main benchmark when classifying twitter accounts. Three linguistic features of a user's text are used in the detection process which including the average URL count per tweet, the average language dissimilarity between a user's tweeting, and the decay rate of text introduction of one user during different time-ordered tweets. In the same year, DARPA held a competition with the name of “The DARPA twitter bot challenge” [7], applying machine learning methods to identify social robots. During this competition, multiple teams compete on detecting a particular set of social bots of a certain topic including five features of twitter accounts: tweet syntax, cached real-time network statues, twitter semantics properties, cached behaviour features and user profile. Those teams who having achieved high detection accuracy have figured out the efficacious combinations of different features and applied specifically optimized machine learning methods to train on these features.

With the rise of deep learning, from 2016, researchers have employed more models in deep learning to this area. In 2016, Chavoshi [8] proposed a Twitter bot detection system called DeBot which utilized unsupervised machine learning algorithm to group up correlated users accounts in social network. The key technique they used to realize it is a novel lag-sensitive hash mapping algorithm based on synchronism properties of user accounts. Their final classification accuracy achieve up to 94% which can be validated on thousands of Twitter bots per day. In 2017, Cai and Li [9] used the combination of CNNs (convolutional neural networks) and LSTM (long short-term memory) model to extract and build connection between semantic information of user in twitters. It also built a latent timing model to take the textual features as well as behaviour features of users into detection process. Also, in 2017, Varol and Ferrara [9] made comprehensive conclusion on the existing research on social bots and applied new classification algorithms to detect social bots. They leveraged more than thousand features from public data and meta-data of social network users including such as friends, tweet content, sentiment patterns, network statues, activity time slots and so on. In their article, machine learning algorithms regarding random forest, decision trees, AdaBoost classifier and logistic regression are used to detect non-human twitter accounts while random forest is the best classifier for the accuracy criteria named the area under curve (AUC). Also, in 2017, Gilani and Farahbakhsh [11] collected large scale Twitter dataset and define various metrics based on metadata of users. Basing on these metrics, the twitter accounts were divided into four categories and then several questions are asked to distinguish certain relationships between the identity of these accounts (whether they are social bots are real human users) and their features including content view times, the source of twitter, account age, content viewed and account profile. In 2018, in order to extract more information from very limited data and improve the efficiency of detecting social bot, Kudugunta and Ferrara [12] used the sole twitter content to decide whether a twitter account is a social robot. They utilized LSTM to figure out the contextual information which takes the user metadata and twitter content as input. Another contribution point is that they devised a technique based on synthetic minority oversampling in order to produce large scale dataset. This measure highly improves the balanced level of the dataset by generating minority samples and thus increases the final classification accuracy. The proposed detection model achieved perfect precision up to 99% for the AUC and solved the negative effects brought by limited minority samples data. Last but not least, in 2018, B. Wu, L. Liu [13] applied Generative Adversarial Networks (GAN) to generate social bots samples and thus effectively improved the accuracy of social bots detection, which inspires this paper to dig deeper in using deep learning model to address the imbalance of social bots training dataset.

For the part of machine learning algorithm VAE, it was firstly proposed by Diederik and Max [14] in 2014. In this paper, VAE was used as inference and learning tool in directed probabilistic models to generate pictures based on MNIST dataset and VAE generates relatively ideal pictures. Although the original application of VAE is in the field of generating images, VAE also presents promising potentiality as generation model in other fields. The application of VAE in generating textual data is rising in recent years. Because the previous standard RNNLM (recurrent neural network language model) can generate sentences one word at a time and does not produce an explicit semantic representation of the sentence. In 2016, Vinyals and M [15] proposed VAE generation model based on RNN, which can create the potential semantic representation information at the sentence level by just sampling from the priors yields well-structured and diverse sentences. In 2019, Shen and Celikyilmaz [16] proposed various multi-level network structures for VAE (the ML-VAE model), which are expected to make use of high-level semantic features (themes and emotions) and fine-grained semantic features (specific word selection) in order to generate globally coherent long text sequences. In their project, hierarchical RNN decoder was proposed as a generation network to take advantage of the expression of sentence level and word level. Moreover, the author also found that it is better to transmit hidden variables to RNN decoder at high level and to output a mesh to RNN decoder at lower level to generate word than to directly transmit hidden variables to RNN decoder. Shen and Celikyilmaz also evaluated ML-VAE with language model for unconditional and conditional text generation tasks. The complexity of language modeling and the quality of the generated examples are greatly improved over some of the baseline levels. Additionally, in 2019, Zhang and Yang proposed a syntax-infused VAE which is able to integrate grammar trees with language sentences increase the performance of generating sentences. An unsupervised machine learning model based on SIVAE-i is proposed and presents syntactically controlled sentences production. The model of the system of Zhang and Yang was tested on real human words trees dataset and produced satisfied results for producing sentences with greater loss and little grammar mistakes. As a result, We can find that it is feasible to apply VAE to generate natural language information and twitter account information in textual format and the generated VAE social bots dataset may promote our detecting accuracy.

3 Designed Framework and Implementation

3.1 Twitter Account Features Selection

A twitter account has many features. For example, features that normal users can percept, including the number of friends and followers, the content of tweets produced by the users, profile description and user settings etc. Also, there are other implicit features which involving personal emotion and social network connection in the back-end dataset.

These features have already been thoroughly analyzed and categorized. Generally, these features are divided into following six categories which can be found in Table 1. In our project, eleven features having better classification effect for social bots are selected from these six categories which presented in Table 1.

Table 1. Features and their categories.

Category	Features used
User-based features	Number of favorite topics, most frequent words used in twitter, the punctuation using habit, the length of the twitter, number of forwarding other twitters
Friends features	Number of the ratio of followers and followed other accounts
Network features	Number of URL linkage, the resource of the tweets
Temporal features	Number of mentions of other users
Content and language features	Topic tags
Sentiment features	The sentiment similarity of twitters

Number of Favorite Topics: this feature represents how many topics a twitter user interested in. Normally, a human user is interested in several specific topics and keeps following these topics by posting their own comments, interacting with other twitter users under the same topic or thumbing up other twitters. Yet for twitter social bots, their main job is to extend their influence and target topics that they are designed to. Thus, their favorite topics may remain the same and unrelated to each other.

Most Frequent Words Used in Twitter: When normal human users post twitters, they have their own typing style and language usage habit. Some words and expression will be more frequent used by a particular human user. However, because the content of social robots is generated by system, their language habit is usually more random and uncertain.

The Punctuation Using Habit: The reason using this feature is the same as the most frequent words used in twitter. It is defined as follows:

$$\mathrm{d}_n = \Sigma_{\mathrm{i}=1}^{N} v_i \tag{1}$$

Where v_i denotes the variance of the happening of a specific word's usage.

The Variance of the Length of the Twitter: The length of twitter posted by normal human users normally fluctuates a lot. It depends on what the users what to express and how intense of their felling. This feature not likely exists in social bots account for they focus on one topic and act as a particular role. It is defined as follows:

$$\sigma = \frac{\Sigma\left(\mathrm{d}_n - \overline{d_n}\right)^2}{N}, \ \overline{d_n} = \frac{\sum \mathrm{d}_n}{N} \tag{2}$$

Where $\overline{d_n}$ means the average of the twitter length, N stands for the total number of the twitter account and d_n is the numerical value of the nth twitter's length.

Number of the Ratio of Followers and Followed Other Accounts: For normal users, they have limited time so they don't follow too many other twitter accounts. In this case, the ratio of followers and followed other accounts is normally no larger than 10. But for a social robot it is normal that they follow a large crowd of people in order to disperse their influence in larger people set.

Number of URL Linkage: This feature represents the average number of URL linkage in a twitter account. For normal human, their main intention of using social network is to communicate with other people and express their own opinions and posting an external URL is not a common operation for them. Social bots are always intended to direct other normal users to other sources, for example inducing others to fishing website, selling illegal sources or posting fake advertisements. So the frequency of using extra URL is larger than normal users.

The Resource of the Tweets: It represents the average ratio of twitters that pushed from other sources of the total twitters of the user. One main task of social network for human users is providing a good place to share their actions on other applications (for instance, sharing one's favorite song from Spotify). So, some tweets from normal users are from external resource, which uses official interface provided by Twitter. On the contrary, social bots is unlikely to share this information from other legal external applications. Even if they do, it is highly possible that they use illegal interface.

Number of Mentions of Other Users: This feature represents the average number of mention of other users in a user' twitter. The form is used as "@name" in twitter. Because the social bots need to influence more normal users, generally they mention more users than human being.

Topic Tags: This feature represents the average number of tag usage for a user. Each tweet can be selected to correlate with a topic by using the symbol "#topic". Generally normal human user will not use this tag frequently or use more than one tag. Yet in order to expand their influence, social bots are more inclined to use more tags which could be a good feature to distinguish normal users from bots.

Number of Forwarding Other Twitters: It represents the average ratio of the number of retweeting and self-edited tweet. Normally, twitters forwarded by human users are high correlated with their interested topics with their comments. While for social bots, they tend to retweet more frequently without their own opinions so that to disperse them more quickly.

The Sentiment Similarity of Twitters: This feature represents the latent sentiment and semantic similarity of one user's twitter. This feature is realized by using "Vector Semantic Space" (VSS), which can analyze the potential relatedness between words and documents. The similarity of the distribution of words between two documents a and b can be calculated as follows:

$$s = \frac{\Sigma_{\mathrm{i}=1}^{N}(a_i \times b_i)}{\sqrt{\sum_{i=1}^{N}(a_i)^2} \times \sqrt{\sum_{i=1}^{N}(b_i)^2}} \tag{3}$$

Where a and b denote the row vector of document of word A and word B, they can be expressed as $\mathrm{a} = \{\mathrm{a}_1, \mathrm{a}_2, \ldots, \mathrm{a}_\mathrm{n}\}$ and $\mathrm{b} = \{\mathrm{b}_1, \mathrm{b}_2, \ldots, \mathrm{b}_\mathrm{n}\}$. The closer the value of s. 1, the more similar the word A and B is. Human users usually have a common sentiment pattern when posting twitters with a larger value of s close to 1. However, social bots post sentiment dissimilar twitters with lower value s which is close to 0.

3.2 VAE

Variational Autoencoder was firstly proposed in 2014 as a generated model and the main target of VAE is to construct a function using the latent variable to generate the target data. More specifically, it tries to map the original data into latent variables and supposes the latent variables obey a certain type of distribution (for example normal distribution in this case). Then it uses the latent variables to generate new data with random noise added. As the results, the ultimate goal of this algorithm is to better make the generated distribution close to the real distribution of the dataset. The goal of VAE is to better make the generated distribution close to the real distribution of the dataset. In order to realize it, what VAE do is to make transformations between different distributions using neural network. What VAE trying to build this is: supposing there is a N dimension vector, which stands for N features that are used to solely determining one twitter user account. For each feature, there is a distribution of it. And what the VAE do is to sampling from these distributions and use a deep neural network to rebuild a simulated twitter account.

The structure of VAE consists of an encoder, a decoder, and a loss function which can be found in Fig. 2. The encoder takes the twitter account data as input and mapping every input data into latent attributes. Because these latent variables obey normal distribution, each of them has two attributes including mean value and variance. The decoder uses the latent value to regenerate twitter account information. In order to use VAE to generate new samples, the latent attributes will be modified each time in generating a new account. In order to generate new samples which possess original accounts' properties but not totally the same, before directly takes the mean value attribute as its input, the decoder adds random noise to the variance attribute and then adds the revised variance to the mean value. That being said the latent attributes will be modified each time when generating a new account. In this project, the latent attributes are assigned random noise to make sure it can generate different and thorough twitter social bots accounts. Specifically, the mathematic deduction is as follows: suppose the input data samples denote as $\{\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_\mathrm{n}\}$ with the distribution p(X). However, we cannot directly calculate this distribution. Then we look into the latent variable p(Z), which is mapped by the encoder from the p(X). Now we can suppose p(Z|X) is the exclusive posterior probability distribution of X_n and p(Z|X) obeys normal distribution. The decoder can recover the twitter account data by sampling from p(Z|X). Formula (4) is used to calculate the distribution of Z:

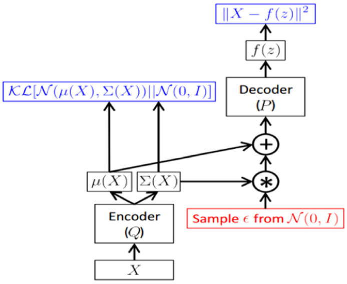

$\|X - f(z)\|^2$
$f(z)$
$\mathcal{KL}[\mathcal{N}(\mu(X), \Sigma(X))||\mathcal{N}(0, I)]$
Decoder
(P)
+
*
$\mu(X)$
$\Sigma(X)$
Encoder
(Q)
Sample ϵ from $\mathcal{N}(0, I)$
X

Fig. 2. The structure of VAE

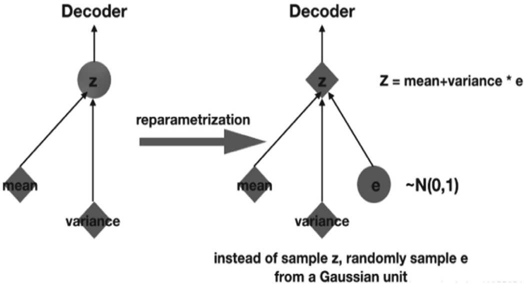


Fig. 3. Flow of reparameterization

$$p(Z) = \sum_X p(Z|X)p(X) = \sum_X \mathcal{N}(0, I) * p(X) = \mathcal{N}(0, I) \sum_X p(X) = \mathcal{N}(0, I) \quad (4)$$

In this way, the distribution of Z can be kept in standard normal distribution whose mean value is denoted as $\mu(X)$ and variance is denoted as $\Sigma(X)$. Then random noise (in this case realized by sample a value from $\mathcal{N}(0, I)$) is added into $\Sigma(X)$ and the revised $\Sigma(X)$ is added to $\mu(X)$ as the final input of decoder. In this way, the decoder is able to continually generate new bots' samples. Finally, the model is trained multiple epochs based on the loss function which has two parts: (1) Kullback–Leibler divergence, also known as KL divergence, between the distribution of $\mathcal{N}(\mu(X), \Sigma(X))$ and $\mathcal{N}(0, I)$ (2) Mean-square error, also known as MSE, between the input data X and the output data $f(z)$. Finally, the flow chart of VAE is shown in Fig. 2.

In formula (4), for the reason that Z is a random variable, in the process of calculating the back propagation in neural networks, Z will be an undifferentiated variable. And a method named reparameterization will be used to calculate the back information of Z. The basic idea of this method is to introduce a Gaussian distribution ε, which can transform Z to a fixed value which can be found in Fig. 3 while circles represent random variables and squares represent variables.

3.3 Social Robot Classification Method

The problem of detecting social bots is defined as follows: $D = \{d_1, d_2, \ldots, \mathrm{d}_n\}$ represents the information of a single twitter account where d_1 to d_n represents the N key features of this account. The dataset $W = \{D_1, D_2, \ldots, D_m\}$ consists of M accounts while each account is either a normal human user or a robot. The category R consists of two parts: R_r (robots) and R_h (human). The essence of this problem is to decide whether a twitter account D belongs to R_r or R_r. What this project done is to improve the performance of a function which can mapping a twitter account D based on its N features d_1 to d_n to the correct categories R_r or R_h. The definition of this mapping is as follows ($\varphi(A_n)$ represents the mapping function):

$$\varphi(A_n) = \begin{cases} 0 & A_n \in R_r \\ 1 & A_n \in R_h \end{cases}, \text{ where } R = \{R_r, R_h\} \quad (5)$$

In fact, this process is simplified as a binary classification problem. Considering the binary decision problem on pure numeral dataset is very mature and holds high recognition accuracy, we select the most common and accurate method– simple neural network. So, what limited the performance of the final classification accuracy of this neural network is the balance and abundance of the training dataset. In the following sections, we will focus on how to address the imbalance problem of dataset.

3.4 The Overall System Design

Figure 4 shows the overall system design of this project. Firstly, eleven key and highly-discriminative features are selected and extracted from the original dataset. Then they are transformed into pure numeral data to facilitate the training and classification process. Secondly, the dataset are split into two parts: testing and training dataset under the ratio of 1:4. In each sub-dataset, social bots accounts and human user account are separated (the ratio of social bots accounts and human users is about 1:4. Thirdly, the VAE takes the social bots accounts of the training dataset as input and output generated more social bots accounts. The newly generated social bots accounts are added to the previous training dataset to solve the imbalance problem. Fourthly, the neural network classifier uses the revised training dataset as its training dataset and the final classification accuracy will be tested by the testing dataset.

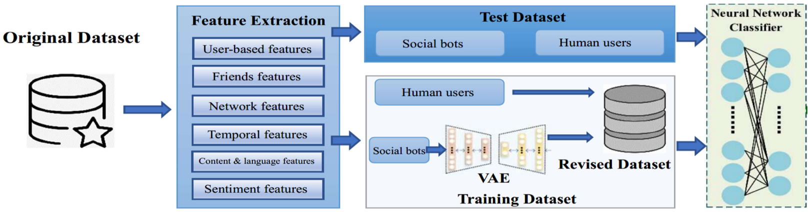


Fig. 4. Overall system design

3.5 Detection Algorithm Process

The flow chart of the detection process is shown in Fig. 5. The main process can be listed as the following steps:

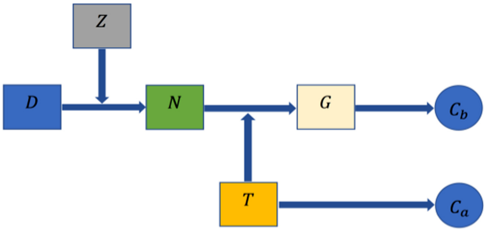


Fig. 5. Detection process

1. Divide the original dataset into training dataset T and testing dataset E. The twitter account neural network classier C_a is trained on the training dataset T.
2. Divide the training dataset into two parts: pure social bot dataset B and pure human user dataset H. The VAE model is trained on B to generate new social bots accounts.
3. The decoder D and the encoder N could generate new twitter account with given random noise Z, which is added to the input of N.
4. Add G into the training dataset T and used the revised dataset to train a new neural network classifier C_b.
5. Compared the performance of C_a and C_b.

4 Results and Discussion

4.1 Dataset

Because detection of social bots has become a hot topic these years, a number of public datasets are available online and bot repository is selected by us as the original dataset resource. Bot repository (https://botometer.iuni.iu.edu/data/index.html) is a centralized website to share annotated datasets of Twitter social bots and normal human users. It provides dataset of different magnitude, crawled from twitter website in different years. In our experiment, cresci-2017 folder is selected as the dataset of training and testing data. It provides over ten thousand twitter accounts information each with 23 features. As discussed in Sect. 3.1, eleven features are finally selected and used. Additionally, in order to decrease the computational complexity and training time while keeping sufficient data, the final dataset is composed of 1970 human user accounts and 463 social bots account.

4.2 Compared Algorithm

In order to prove the performance of VAE in generating social bots accounts, traditional over-sampling algorithm named Synthetic Minority Over-sampling Technique (SMOTE) is selected as the compared algorithm. SMOTE algorithm is a revised scheme of random oversampling while random oversampling simply copies samples randomly to add new samples into the original dataset. SMOTE algorithm is design to solve the over fitting problem of the classification model so that the information the model learns become too specific rather than general. To solve this, SMOTE algorithm analyses the minority samples and composites new samples according to original samples. The SMOTE process is as follows:

(1) For every samples in the minority category, calculate the Euclidean distance between it and every other minority samples;

(2) Define parameter called sampling rate N according to the unbalanced level of the original dataset. For every sample belongs to the minority category, randomly select several samplings using k-Nearest Neighbor algorithm. Denote the neighbor sample selected as $\mathrm{X_n}$;

(3) For every selected the neighbor sample $\mathrm{X_n}$, using formula (6) below to generated new samples:

$$x = x + rand(0, 1) \, * \, (\widetilde{x} - x) \tag{6}$$

The compared algorithm is realized by Imblearn Python library (https://pypi.org/project/imblearn/). Same as VAE, SMOTE algorithm also takes the training dataset as input data. The new samples generated by SMOTE are also added into the original dataset. The augmented dataset will be used to train the simple neural network classifier and final accuracy will be calculated using the testing dataset.

4.3 Experimental Process and Results Analysis

Parameter Settings: After designing basic structure of VAE, selecting appropriate parameters is also a vital part in constructing neural network. In machine learning, these parameters are called hyperparameters. Hyperparameters includes a set of settings includes the layer of the neural network, the number of the neuron of each layer, the learning rate, the coefficient of regression, etc. And these hyperparameters should be revised continuously during the training time until the neural network can achieve optimal results. In this paper, the selection of these hyperparameters including layers, learning rate, number of the neurons referenced the settings in previous work [13] which utilized GAN to generate social bots samples. The settings of the neural network used in this article are listed below:

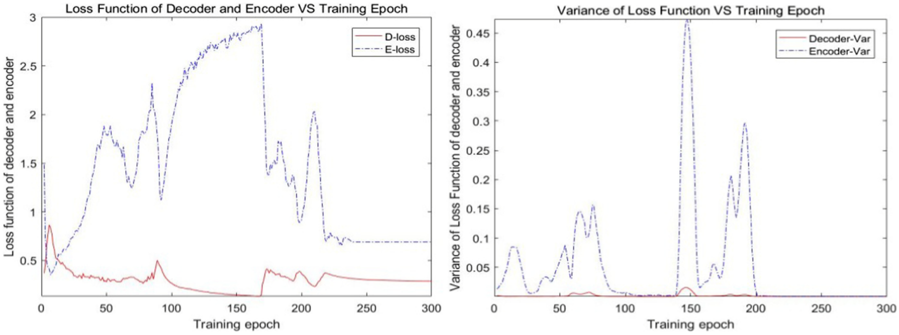
1) Layers: According to related research focus on make prediction based on pure data, two to four layers is the common settings. After testing and optimizing, each neural network component including encoder, decoder and our classifier used three layers in this article.
2) Activation function: activation function performs as the role of processing function between each previous layer and later layer of neural network. In this project, two activation functions were used:
   Sigmod function: $f(x) = \max(0, x)$ and ReLU function: $\sigma(x) = \frac{1}{1+\mathrm{e}^{-x}}$.
   In this case, Sigmod function is the ideal function for activation function before the last layer output. For the input layers, because the final equivalent function of neural network is highly possible non-linear, ReLU is the ideal non-linear function.
3) Optimization function: Machine learning algorithm is always focus on optimization model and optimize the loss function (the optimized objective function). According to the original VAE algorithm and open source research projects about using VAE to generate text data, Adam algorithm is selected as optimization algorithm of encoder and the classical SGD (Stochastic Gradient Descent) algorithm is selected as optimization algorithm of both decoder and neural network classifier.
4) Learning rate: To get accurate predictions, a learning algorithm called gradient descent updates the weights as it moves back from the output to the input in neural network. The gradient descent optimizer estimates the good value of the model weight in multiple iterations by minimizing a loss function (L), and this is where the learning rate comes into play. It controls the speed at which the model learns.

$$\boldsymbol{L} = -\left[\boldsymbol{y}\log\left(\hat{\boldsymbol{y}}\right) + (1-\boldsymbol{y})\log\left(1-\hat{\boldsymbol{y}}\right)\right] \tag{6}$$

Where y denotes expected output and y^ is the actual output of the neural network. According to the original VAE algorithm and other researched using VAE to generate text data, our learning rate in VAE and neural network classifier is between 0.0001 to 0.0005 which can help to reach the most optimal learning effect.

5) Number of neurons: In this project, according the settings of the neurons in related researches and continuous optimization of the parameters during the practical experiments, the number of the neurons of each layer can be determined. For the first layer of the encoder, the number of the neurons should be equal to the input data's dimension. For the last layer of the decoder, the number of the neurons should also equal to the dimension of the twitter account data. Besides, the number of the neurons of other layers ranges from 10 to 30 depends on the nature and final performance of experiments. For the decode, the function of it is to learn the latent features of the twitter accounts which means data compression, so the neurons of it are set between 10 to 25. For the decoder and neural network classifier, the function of them is to regenerate the data and accurately decided whether an account is a social bot. In this case, the neurons are set between 20 and 30 for better distinguishing performance.

Experimental Results and Analysis: Firstly, the VAE should be trained on the social bots samples of the original dataset. The encoder would use the neural network to convert the input twitter samples data into low-dimension vector and then the decoder should add random noise to the features extracted and regenerate high-dimension twitter account data. Then the output data would be calculated its loss between input real account data using SGD algorithm. Thus, the loss function of the decoder, encoder in this project is a logarithm function. After 2000 epochs training, the final curve of the loss function and the variance of the loss function are presented in Fig. 6.

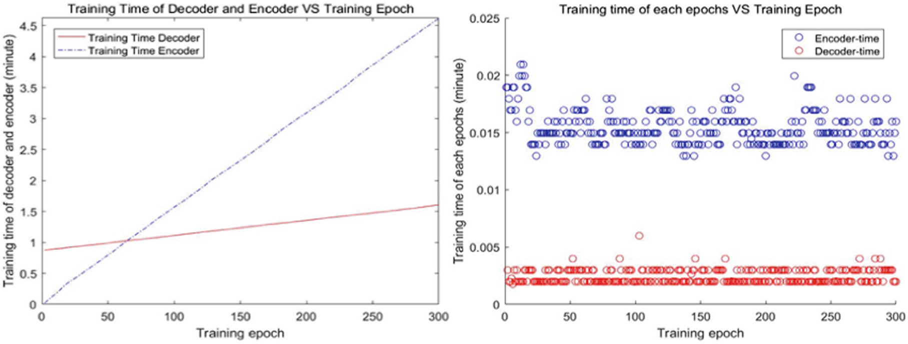


(a) loss function curve for decoder and encoder (b) variance of the loss function

Fig. 6. Loss function of VAE components in training process

The loss function fluctuates a lot at the beginning of the whole training process. For the reason that the encoder has not learnt the basic features of the incoming social bots samples in the lack of training epochs, in the front 225 epochs, the VAE is adjusting its internal neural network to conform its distribution to the social bots. Also, in the training process, the loss of encoder changed more violently and keeps higher value than the decoder. It represents that the beginning statue of encoder could be further than decoder and takes more dramatically change to the final state. After about 260 training rounds, the loss function of both decoder and encoder declined to be smooth, which means that the features have been basically extracted and incorporated in VAE's neural network.

Additionally, the total training time is also recorded in the experiment. The time of training each epoch is recorded and the accumulative training time is also recorded in Fig. 7. As the result presented in the graph, it takes much longer time to train the encoder than the decoder, which also proves the early speculation that the encoder is more time-consuming to train according to the dramatically change of loss function.



(a) accumulative total training time (b) scatter diagram of each training epochs

Fig. 7. Training time of VAE components in training process

In order to prove the effectivity of VAE in learning the latent features of social bots tweets and generating new samples, we need to test the dataset mixed with the samples VAE generated on the neural network classifier. If the final detection accuracy increases compared to neural network classifier trained on original dataset, it can prove that the VAE is effective. Furthermore, if the final detection accuracy increases compared to the same structure classifier trained on dataset mixed with twitter bots samples generated by other oversampling method which in this project refers to SMOTE algorithm, we can draw the conclusion that VAE is more effective than the traditional oversampling method. For the experiment design of this project, in order to control variables, the same dataset and minority samples are used when training the VAE and SMOTE model and the neural network classifier used the same hyperparameters when trained on augmented dataset generated by VAE and SMOTE algorithm. Additionally, the same test set is used as the criteria of final accuracy of the classifier.

The VAE, SMOTE and SMOTE based SVM algorithms will generate some social bots until the number of social bot samples is equal to the normal human user samples. Then the neural network classifier will train on according revised dataset. The result diagram can be found in Fig. 8.

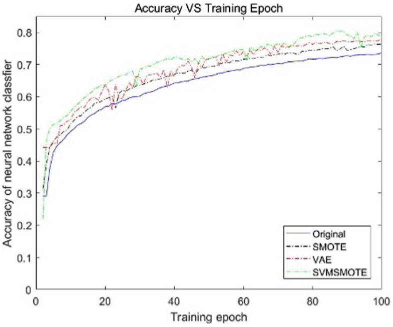


Fig. 8. Accuracy comparation of classifiers

From Fig. 8, it can be seen that the final accuracy of the neural network classifier trained on VAE achieves highest accuracy in the final training epoch even if the original accuracy realized by SMOTE algorithm is high enough up to 76%. The detection accuracy of VAE is higher than the one of SMOTE 1.5% and higher than the original dataset without any augmented on social bots' samples 15%. This proves that both the datasets revised by VAE and SMOTE algorithms could make the neural network classifier detect more social bots because its accuracy exceeds the one of using original dataset. It also verifies the previous assumption proposes: the accuracy of neural network classifier when solving binary classification problem is highly affected by the number of the samples belongs to two categories and what is more important, the balanced level of the dataset which is the ratio of the positive and negative samples. In this place, the original unbalanced ratio is ameliorated by the samples added by VAE and SMOTE accordingly. The result shows that if the ratio is closer to 1, the accuracy of the classifier will better be assured. Additionally, the result presents that the accuracy improvement made by VAE overpasses the accuracy made by SMOTE algorithm by 1.5 per cent and closed to which of SMOTE based SVM algorithm. It demonstrates that compared to SMOTE algorithm, the VAE can better help the neural network classifier to learn the features of social bots' samples and thus improves its detection ability. This is brought up by the strong capability of VAE in extracting the latent properties of social bots and VAE is better in imitating these samples. In fact, it is decided by the nature of VAE which used neural network model which can better extract features from existing dataset while SMOTE algorithm which realized oversampling on a simple and too random method.

5 Conclusions and Future Work

This article aims at using VAE to solve the problem of unbalanced data between normal human users and social bots by generating social bot samples. In order to prove the effectiveness of VAE in generating new samples, two compared experiments are conducted including training the classifier on original dataset and training the classifier on dataset added with new social bots generated by SMOTE algorithm. The experimental result shows that dataset revised by using the social bot data generated from VAE effectively improves the accuracy compared to the compared algorithms.

However, there are some problems existing in the usage of VAE. Firstly, for the accuracy improvement with the increase of the training epochs, the result reached by VAE presents fluctuation compared to the smooth result curve of the compared experiments and it may be caused by the mean square error between the generated data and the original data by VAE. This will cause the uncertain ambiguity of the generated data which will affect the accuracy when training the classifier in some epochs. Secondly, the accuracy improved by VAE over traditional oversampling SMOTE algorithm is obvious and close to the better algorithm of SMOTE based SVM. It may because the total number of the original dataset is not large enough so the advantage of VAE cannot be fully presented in this case. In future work, in order to overcome the problem of the uncertain ambiguity of the generated data, adversarial nets can be combined with VAE which means to apply adversarial nets to train it while keeping the basic structure of VAE. Moreover, the VAE can be deployed in large scale dataset to better present its advantage over traditional oversampling algorithm.

Thanks to the support of National Key Research and Development Program of China under Grant (NO. 2017YFC0820603, NO. 2018YFC0831301 and NO. 2018YFC0831302).

References

1. Chu, Z., Gianvecchio, S., Wang, H., Jajodia, S.: Who is tweeting on twitter: human, bot, or cyborg?. In: Proceedings of the 26th Annual Computer Security Applications Conference, pp. 21–30, 6-10 December 2010
2. Yang, Z., Wilson, C., Wang, X., Gao, T., Zhao, B.Y., Dai, Y.: Uncovering social network sybils in the wild. In: Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference, pp. 259–268, 2–4 November 2011
3. Cao, Q., Sirivianos, M., Yang, X., Pregueiro, T.: Aiding the detection of fake accounts in large scale social online services. In: Proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation, p. 15, 25–27 April 2012
4. Beutel, A., Xu, W., Guruswami, V., Palow, C., Faloutsos, C.: "CopyCatch: stopping group attacks by spotting lockstep behavior in social networks. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 119–130, 13-17 May 2013
5. Zafarani, R., Liu, H.: 10 bits of surprise: detecting malicious users with minimum information. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 423–431, 18-23 October 2015
6. Clark, E.M., Williams, J.R., Galbraith, R.A., Jones, C.A., Danforth, C.M., Dodds, P.S.: Sifting robotic from organic text: a natural language approach for detecting automation on twitter. J. Comput. Sci. **16**, 1–7 (2016)
7. Subrahmanian, V.S., et al.: The darpa Twitter bot challenge. Computer **49**(6), 38–46 (2016)
8. Chavoshi, N., Hamooni, H., Mueen, A.: DeBot: Twitter bot detection via warped correlation. In: Proceedings of the 16th IEEE International Conference on Data Mining, pp. 817–822, 12-15 December 2016
9. Cai, C., Li, L., Zengi, D.: Behavior enhanced deep bot detection in social media. In: Proceedings of IEEE International Conference on Intelligence and Security Informatics, pp. 128–130, 22-24 July 2017
10. Varol, O., Ferrara, E., Davis, C.A., Menczer, F., Flammini, A.: Online human-bot interactions: detection, estimation, and characterization. In: Proceedings of the Eleventh International AAAI Conference on Web and Social Media, pp. 280–289, 15-18 May 2017

11. Gilani, Z., Farahbakhsh, R., Tyson, G., Wang, L., Crowcroft, J.: Of bots and humans (on Twitter). In: Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 349–354, 31 July–3 August 2017
12. Kudugunta, S., Ferrara, E.: Deep neural networks for bot detection. Inf. Sci. **467**, 312–322 (2018)
13. Wu, B., Liu, L., Dai, Z., Wang, X., Zheng, K.: Detecting malicious social robots with generative adversarial networks. KSII Trans. Internet Inf. Syst. **13**(11), 5515–5594 (2019). https://doi.org/10.3837/tiis.2019.11.018
14. Kingma, D.P.: Max welling, "auto-encoding variational Bayes". In: The 2nd International Conference on Learning Representations (ICLR2014), p. 14 (2014)
15. Vinyals, O., Dai, A.M., Jozefowicz, R., Bengio, S.: Generating sentences from a continuous space. Comput. Sci. (2015)
16. Shen, D., Celikyilmaz, A., Zhang, Y., Chen, L., Wang, X., Gao, J., et al.: Towards generating long and coherent text with multi-level latent variable models (2019)
17. Zhang, X., Yang, Y., Yuan, S., Shen, D., Carin, L.: Syntax-infused variational autoencoder for text generation (2019)