Network Security Situation Prediction Based on Improved WGAN

Jiang Zhu[1,2] and Tingting Wang[1,2(✉)]

[1] School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
1325242@qq.com, 1762089088@qq.com
[2] Chongqing Key Lab of Mobile Communications Technology, Chongqing 400065, China

Abstract. The current network attacks on the network have become very complex. As the highest level of network security situational awareness, situation prediction provides effective information for network administrators to develop security protection strategies. The generative adversarial network (GAN) is a popular generation model, which is difficult to train, collapse mode and gradient instability in this network. A Wasserstein distance as a loss function of GAN is proposed. And a difference term is added on the loss function. The improved Wasserstein-GAN (IWGAN) is to improve the classification precision of the situation value. Compared with other forecasting methods, the results show that the method has obvious advantages.

Keywords: Situational awareness · Situation prediction · Generative adversarial network · Difference · Wasserstein-GAN

1 Introduction

The network security situation prediction is the ultimate goal in the network security situation awareness (NSSA) [1]. NSSA on the premise of extracting and understanding the security element information of real network. Through the observation and analysis of history and current data. Furthermore, the future security trend of the network is speculated.

In the field of network security, situation prediction has become a hot spot. The network security situation prediction is based on the situation value obtained by the network security data in a period of time.

Based on the deepening of machine learning, generative adversarial Network (GAN) [2] is another form based on the micro-generation network, the training of GAN needs to achieve Nash equilibrium, the training of GAN model is unstable. On this basis also made a lot of improvements, such as DCGAN [3] and LSGAN [4]. But in practice this approach does not completely solve the problem. Wasserstein-GAN (WGAN) [5] had a very good effect. In this paper, WGAN is applied to network security, and an improved WGAN situation prediction method is proposed. Taking full account of the dependence of different situation factors, using the correlation of the

H. Song et al. (Eds.): SIMUtools 2019, LNICST 295, pp. 654–664, 2019.
https://doi.org/10.1007/978-3-030-32216-8_64

situation factor time dimension to predict the future network security situation factors, the influence of the historical network security situation on the future situation is more objectively reflected.

2 A Method of Situation Prediction Based on Improved WGAN

2.1 Generative Adversarial Network

GAN consists of two models, generating model G and discriminant model D, random noise z through G generation as far as possible to follow the real data distribution of the sample G(z), discriminant model D can determine whether the input sample is real data x or generate data G(z). Both G and D can be non-linear mapping functions, such as multilayer perceptron. The optimization goal is to achieve Nash equilibrium so that the generator estimates the distribution of data samples. The process of GAN is shown in Fig. 1:

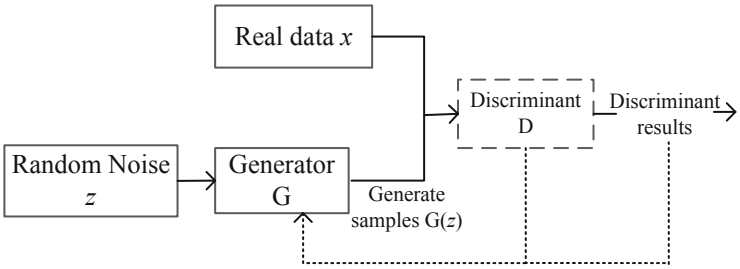

Fig. 1. GAN flow chart.

2.2 GAN Core Principle Description

The discriminant is a two classification model. The training discriminant is the process of minimizing the cross entropy. The $E(\cdot)$ is the calculation of expected value, x is sampled from the real data distribution $p_{data}(x)$, and Z is sampled in a priori distribution $p_z(z)$. In order to learn the distribution of data x, the generator constructs a mapping space $g(z;\theta G)$ by a priori noise distribution $p_z(z)$, and the corresponding discriminant mapping function is $D(x;\theta d)$. The probability of outputting a scalar to represent x as real data is:

$$\min_G \max_D V(D,G) = E_{x-P_{data(x)}}[\log D(x)] + \mathrm{E}_{z-P_{z(z)}}[\log(1-D(G(z)))] \tag{1}$$

It can obtain the optimal state of the discriminant D when the generator G is fixed by the formula (1). For a specific sample x, $P_r(x)$ is the real sample distribution, $P_g(x)$ is the generator produced by the sample distribution, it may come from a real distribution or a generation distribution, and its contribution to the formula (1) Loss function is:

$$-P_r(x)[\log D(x)] - \mathrm{P}_g(x)[\log(1 - D(x)] \tag{2}$$

So that its derivative of D(x) is 0, it concludes:

$$-\frac{P_r(x)}{D(x)} + \frac{\mathrm{P}_g(x)}{1 - D(x)} = 0 \tag{3}$$

The best discriminant for simplification is:

$$D^*(x) = \frac{P_r(x)}{P_r(x) + P_g(x)} \tag{4}$$

This result is intuitively easy to understand, and is to look at the relative proportions of a sample x from the actual distribution and the probability of generating the distribution.

Substituting formula (1), and then a simple transformation can be obtained:

$$\begin{aligned} &E_{x-P_{data(x)}} \log \frac{P_r(x)}{\frac{1}{2}[P_r(x) + P_g(x)]} + \\ &\mathrm{E}_{z-P_{z(z)}} \log \frac{P_g(x)}{\frac{1}{2}[P_r(x) + P_g(x)]} - 2\log 2 \end{aligned} \tag{5}$$

The transformation is to introduce two important similarity metrics for KL divergence [6] and JS divergence [7]:

$$KL(P_1 \| P_2) = E_{x-P_1} \log \frac{P_1(x)}{P_2} \tag{6}$$

$$JS(P_1 \| P_2) = \frac{1}{2} KL\left(P_1 \middle\| \frac{P_1 + P_2}{2}\right) + \frac{1}{2} KL\left(P_2 \middle\| \frac{P_1 + P_2}{2}\right) \tag{7}$$

So the formula (5) can be written as:

$$2JS(P_1 \| P_2) - 2\log 2 \tag{8}$$

Under the approximate optimal discriminant, the loss of the minimized generator is equivalent to minimizing the JS divergence. The gradient (approximate) of the generator is 0, and the gradient disappears. Under the condition of KL divergence, the problems such as the gradient imbalance and the penalty imbalance lead to mode collapse [8].

The problem of mode collapse is caused by the gradient disappearance of GAN, the unbalanced gradient and the unbalanced punishment. In GAN, the Wasserstein distance [9] is introduced as the loss function, because of its superior smoothing characteristic relative to KL divergence and JS divergence, the gradient vanishing problem can be solved theoretically.

2.3 An IWGAN Algorithm Description

WGAN's biggest contribution is to use Wasserstein distance to replace the JS divergence or KL divergence in GAN, greatly alleviate the problem of GAN difficult to train, Wasserstein distance also called Earth-mover (EM) distance, defined as follows:

$$W(P_r, P_g) = \inf_{\gamma \sim \prod(P_r, P_g)} \mathrm{E}_{(x,y)\sim\gamma)}[||x - y||] \tag{9}$$

The $\prod(P_r, P_g)$ is a collection of all possible joint distributions of P_r and P_g combined, The lower bound that can be taken to this expectation in all possible joint distributions is defined as Wasserstein distance.

Since the Wasserstein distance definition formula (9) cannot be directly solved, it can be transformed into the following form with an existing theorem:

$$W(P_r, P_g) = \frac{1}{K} \sup_{\|f\|_L \le K} \mathrm{E}_{x\sim P_r}[f(x)] - \mathrm{E}_{x\sim P_g}[f(x)] \tag{10}$$

This process has been proved by the literature [10]. How this distance is solved. First you need to introduce a concept Lipschitz continuous [11]. It's actually an extra restriction on a continuous function f that requires a constant to satisfy any two elements and within the defined domain:

$$|f(x_1) - f(x_2)| \le K|x_1 - x_2| \tag{11}$$

At this time the Lipschitz constant of the function f is K.

In particular, we can define a series of possible function f_w with a set of parameter w, at which point the solution formula (9) can be approximated to the following form:

$$KW(P_r, P_g) = \max_{w:\|f_w\|_L \le K} \mathrm{E}_{x\sim P_r}[f_w(x)] - \mathrm{E}_{x\sim P_g}[f_w(x)] \tag{12}$$

A discriminant network f_w with the parameter w and the last layer is not a Nonlinear activation layer is constructed, and the loss function of the discriminant is given under the condition that the w does not exceed a certain range:

$$L = -\mathrm{E}_{x\sim P_r}[f_w(x)] + \mathrm{E}_{x\sim P_g}[f_w(x)] \tag{13}$$

The Lipschitz limit is that the gradient of the discriminant does not exceed K (K = 1), and a loss term can be added to the end of the formula to reflect this, which is a difference function, and the loss function is:

$$\begin{aligned} L = &- \mathrm{E}_{x\sim P_r}[D_w(x)] + \mathrm{E}_{x\sim P_g}[D_w(x)] \\ &+ \lambda \mathrm{E}_{x_1\sim P_{\hat{x}}, x_2\sim P_{\hat{x}}} \left[\frac{|D(x_1) - D(x_2)|}{\|x_1 - x_2\|} - 1 \right]^2 \end{aligned} \tag{14}$$

In other words, they are still sampling randomly on distribution $p_{\hat{x}}$, but two at a time, and then ask them to have a line slope of nearly 1. To limit the distance between the true and false samples, the specific difference can play a role, given the following proof.

Theorem: Loss function adding difference term can stabilize gradient value.

Analysis: The discriminant is to try to pull large real sample and the distance between the sample, without the difference limit, usually also want to add a weight Clipping. But it is the overall effect on the sample space, it will inevitably lead to gradient disappear or gradient explosion. The difference is only the true and false sample concentration area, and the gradient is limited to 1 near, controllability is very strong.

Proof: Known $\mathrm{x}_r \sim p_{x_r} \mathrm{x}_r \sim p_{x_r}$; supposing $\varepsilon \sim Uniform[0,1]$ is randomly interpolated in the middle of x_r and x_g, namely:

$$\widehat{x} = \varepsilon \mathrm{x}_r + (1-\varepsilon)\mathrm{x}_g \tag{15}$$

At this time $\widehat{x}$ satisfied with the distribution of $p_{\widehat{x_r}}$, random sampling on $p_{\widehat{x_r}}$, in which to select two different values, for example $x_1 \sim p_{\hat{x}}, x_2 \sim p_{\hat{x}}$, these two values are real samples and generate samples of the concentration of the selection, control the distance between them, to limit it, can prevent the distance too large or too small distance, to distinguish the discriminant has brought good results. A simple comparison between weight clipping and differential was made, and the advantage of difference was obviously seen.

The Fig. 2 shows that the gradient value changes very little after using the difference term, which gives the discriminant an unexpected effect on the distinction between real and generated samples. ■

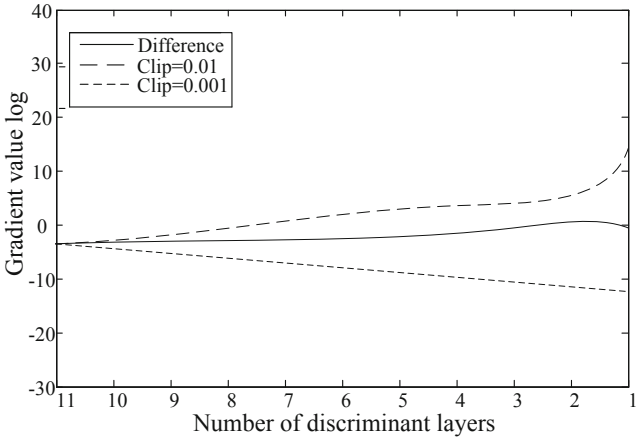

Fig. 2. Gradient value comparison

In conclusion, the IWGAN successfully solves the following problems of GAN: Solving the instability problem of GAN training thoroughly, no longer need to be careful about the training degree of balance generator and discriminant; the problem of collapse mode is solved and the diversity of generating samples is ensured; The problem with WGAN is that The discriminant is a multilayer network. The weight clipping is used directly when dealing with Lipschitz constraints, but this method restricts the parameters to the clip range.

In short, a difference is added to the loss function of the difference term, and by judging whether the new loss function is eligible, the generator and the discriminant will be re-trained until the requirement is reached, and the parameters are updated through the Adam algorithm.

The specific algorithm is as follows:

Algorithm 1: IWGAN algorithm, set in the experiment $\lambda = 10, \alpha = 0.0001, \beta_1 = 0, \beta_2 = 0.9, n = 5, m = 64$ 。

Requirements： λ is Penalty factor， α is Learning rate， n is Number of iterations。 m is batch size， Adam algorithm's higher-order parameters β_1, β_2 ， w is weight initial value， θ is generator initial value

While θ do

for $t = 0, ..., n$ do

Sample $\left\{x^{(i)}\right\}_{i=1}^{m} \sim P_r$ is real data

Sample $\left\{z^{(i)}\right\}_{i=1}^{m} \sim p(z)$ is Generate samples

$\mathrm{x}_r \sim p_{x_r}$, $\mathrm{x}_g \sim p_{x_g}$

$\varepsilon \sim Uniform[0,1]$

$\hat{x} = \varepsilon \mathrm{x}_r + (1-\varepsilon)\mathrm{x}_g$

$$L^i = -\mathrm{E}_{x\sim P_r}\left[D_w(x)\right] + \mathrm{E}_{x\sim P_g}\left[D_w(x)\right] + \lambda \mathrm{E}_{x_1\sim P_x,\ x_2\sim P_x}\left[\frac{\left|D_w(x_1) - D_w(x_2)\right|}{\left\|x_1 - x_2\right\|} - 1\right]^2$$

end for

$$w \leftarrow \mathrm{Adam}\left(\nabla w \frac{1}{m}\sum_{i=1}^{m} L^i, w, \alpha, \beta_1, \beta_2\right)$$

$$\theta \leftarrow \mathrm{Adam}\left(\nabla \theta \frac{1}{m}\sum_{i=1}^{m} -D_w\left(G_\theta(z)\right), w, \alpha, \beta_1, \beta_2\right)$$

end while

3 Flow Chart of Situation Forecast

In the WGAN prediction model, the convolution neural network (CNN) [12] is used in the generator G and discriminant D, which is also a kind of deep convolution countermeasure generation network, the concrete model is shown in Fig. 3:

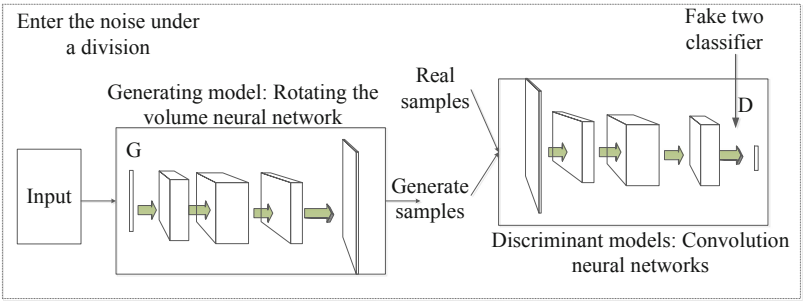

Fig. 3. Deep convolution Confrontation Generation network

This article is the before X's days of data as the input of the generator, generate a distribution. The distribution equivalent to $P_g(x)$. After X's days of data as a real data input discriminant. The final distribution equivalent to $P_r(x)$. The discriminant will be judged in the before X's days of data distribution and after X's days of the data to

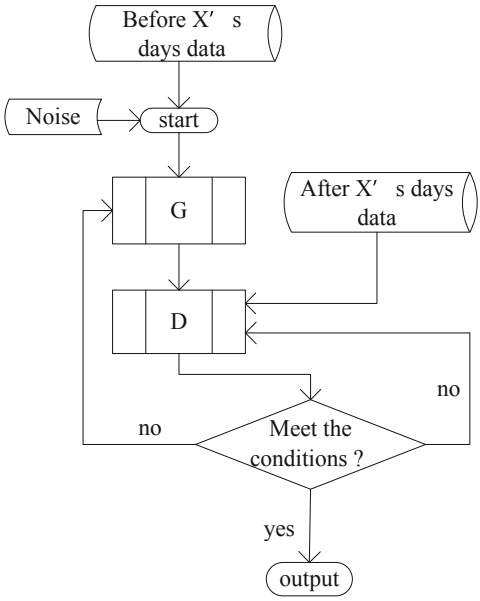

Fig. 4. IWGAN prediction flow chart

distinguish. Constantly update the network to reach the probability of approximately 1 of the state. That is to predict the situation before X's days is not going to develop into the situation after X's days.

IWGAN Security situation forecast Flowchart (Fig. 4):

3.1 Generative Adversarial Network

Analysis of the attack characteristics of security data, uncertainty and continuity, this paper selects a company from July to September 95 days of firewall, IDs and other historical log information as the original dataset [13]. Make a sample of the daily log information.

Because the security situation value is random, the dimensional difference is big, in order to raise the model the training speed, the situation value carries on the extremum standardization processing, the processing formula is as follows:

$$\hat{X} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{16}$$

The upper $X_{\min}$ and $X_{\max}$ are the smallest and largest situation values in the sample. X and $\hat{X}$ are the situation values before and after treatment. The network security situation data after the extreme value standardization is shown in Fig. 5.

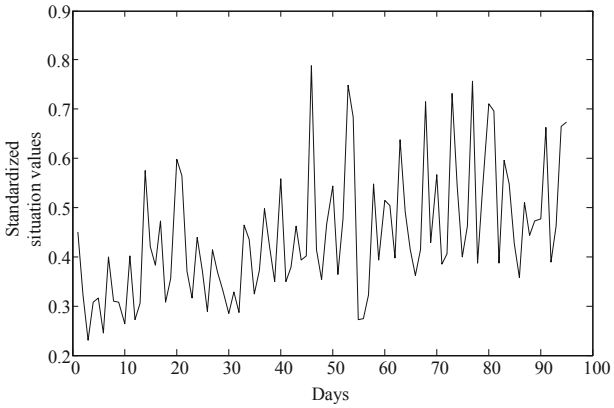

Fig. 5. The value of network security situation after extremum

Figure 5 represents a change curve in the situation value in 95 days of log information. In order to deal with the one-dimensional time series samples which are worth to the situation assessment, the topological order dimension is determined to be 5. The refactoring results are shown in Table 1.

Table 1. Training data refactoring results

Input sample	Output sample
X_1, X_2, X_3, X_4, X_5	X_6
X_2, X_3, X_4, X_5, X_6	X_7
...	...
$X_{74}, X_{75}, X_{76}, X_{77}, X_{78}$	X_{79}
...	...

3.2 Analysis and Comparison of Experimental Results

Convergence Analysis

After the data is refactored, the specification is tanh [−1, 1]. The batch size in Mini-batch training is 64. All parameter initialization is randomly obtained from the normal distribution of (0 0.02) and the slope of the Leakyrelu is 0.2.

A WGAN loss function with a difference item and no difference is added, as shown in Fig. 6. The former is IWGAN, the later is WGAN. It can be seen clearly that the difference function has brought some effect to WGAN.

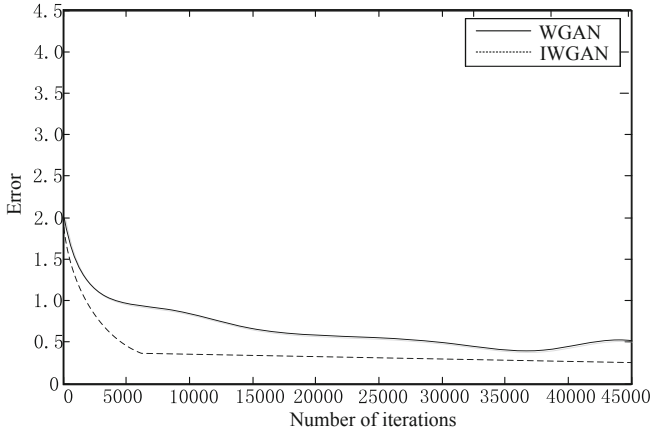

Fig. 6. Variation curve of difference with iterative times

Compared with Other Prediction Methods

Comparing the IWGAN prediction method with the common GAN improvement methods, such as WGAN, DCGAN and LSGAN, the results are shown in Fig. 7.

We can see from Fig. 7 that the IWGAN prediction method works well. This is because it solves the problem of gradient imbalance and collapse mode.

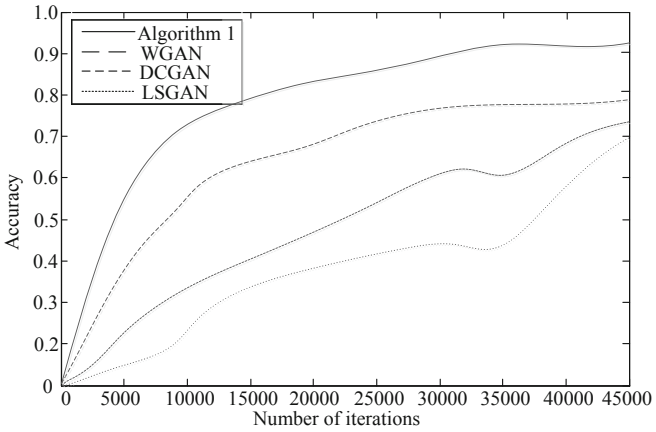

Fig. 7. Comparison of accuracy of different forecasting methods

4 Conclusion

This paper presents an IWGAN network security situation prediction method, establishes the cyclic neural network model for the real network environment, extracts the network security situation factor training model and forecasts the future network security change trend. Using historical log information, such as firewall and IDs 95 days from July to September, as the original data set, the results show that the method is feasible and of high accuracy.

References

1. Wu, D., Si, S., Wu, S., et al.: Dynamic trust relationships aware data privacy protection in mobile crowd-sensing. IEEE Internet of Things J. **PP**(99), 1 (2017)
2. Zhang, H., Xu, T., Li, H.: StackGAN: text to photo-realistic image synthesis with stacked generative adversarial networks. 5908–5916 (2016)
3. Yu, Y., Gong, Z., Zhong, P., et al.: Unsupervised representation learning with deep convolutional neural network for remote sensing images (2017)
4. Gulrajani, I., Ahmed, F., Arjovsky, M., et al.: Improved training of Wasserstein GANs (2017)
5. Zhao, Y., Takaki, S., Luong, H.T., et al.: Wasserstein GAN and waveform loss-based acoustic model training for multi-speaker text-to-speech synthesis systems using a WaveNet vocoder (2018)
6. Wu, D., Yan, J., Wang, H., et al.: Social attribute aware incentive mechanism for device-to-device video distribution. IEEE Trans. Multimed. **19**(8), 1908–1920 (2017)
7. Majors, L., Miller, S., Jensen, S.: Oil spill preparedness for polar bears in Alaska. **2014**(1), 299530 (2014)
8. Srivastava, A., Valkov, L., Russell, C., et al.: VEEGAN: reducing mode collapse in GANs using implicit variational learning (2017)
9. Walczak, S.M.: Wasserstein distance (2017)

10. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein GAN (2017)
11. Jiang, D., Huo, L., Li, Y.: Fine-granularity inference and estimations to network traffic for SDN. PLoS ONE **13**(5), 1–23 (2018)
12. Guo, X., Chen, L., Shen, C.: Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis. Measurement **93**, 490–502 (2016)
13. Jiang, D., Huo, L., Song, H.: Rethinking behaviors and activities of base stations in mobile cellular networks based on big data analysis. IEEE Trans. Netw. Sci. Eng. **1**(1), 1–12 (2018)