Journal of Electronic Testing (2023) 39:129–140
https://doi.org/10.1007/s10836-023-06054-x

Hardware Trojan Detection Method Based on Dual Discriminator Assisted Conditional Generation Adversarial Network

Wenjing Tang[1] · **Jing Su**[1] · **Yuchan Gao**[1]

Received: 25 June 2022 / Accepted: 17 February 2023 / Published online: 29 April 2023

Abstract
Hardware Trojans are usually implanted by making malicious changes to a chip circuit, which can destroy chip functions or expose sensitive information once activated. The hardware Trojan detection method based on side channel information has now become one of the most widely used detection methods. However, due to the influence of the deviation of the acquisition equipment and the noise of the actual chip working environment, insufficient acquisition of useful information of the collected side channel information occurs, affecting the final results. To address the problem, this paper proposes a detection method based on a dual discriminator assisted conditional generation adversarial network (D2ACGAN), which combines the benefits of CGAN, ACGAN, and D2GAN models and can learn a variety of valid information of the tested chip. It can distinguish between side channel data with and without hardware Trojan and classify hardware Trojan using the extended data. Furthermore, to compare the performance of the proposed model, we use the existing CGAN and ACGAN models equally for side channel information expansion and hardware Trojan detection. Finally, the designed hardware Trojan is implanted in an encryption chip for generating data quality evaluation experiments and model method performance experiments. The results show that the average detection accuracy of the D2ACGAN-based hardware Trojan classification model can reach 97.08%, which is better than the detection models based on CNN, SVM, etc. The D2ACGAN model also outperforms the CGAN and ACGAN models in terms of generated data and hardware Trojan classification.

Keywords Hardware trojan detection · Generative adversarial networks · Data augmentation · Side channel information

1 Introduction

With the continued growth of the semiconductor industry, the integrated circuit has become increasingly crucial as the industry's core. The production of IC chips has become a critical component of the industry's supply chain. In addition, the manufacturing of chips is mainly done in third-party foundries, and Hardware Trojans are a security issue that arises from this process [1–3]. Hardware Trojans are usually implanted by malicious alterations to the original circuitry, and if triggered, they can harm the circuit's operation and even disclose confidential information. As a result, the detection of Hardware Trojan needs to be carried out after the chip has been fabricated [4].

From the perspective of whether the chip to be tested is damaged, the existing hardware Trojan detection methods are mainly divided into destructive detection and non-destructive detection [5]. Reverse engineering [6] is an example of destructive detection that uses chemical and physical methods to corrode and polish the chip to be tested, then takes pictures of the original chip layout with a high magnification microscope, extracts the netlist, and compares it to the "golden chip" to detect hardware Trojan. The detection accuracy of this method is great, but the cost is expensive because the chip suffered irreversible damage, and the method's applicability rapidly reduced as the process node declined. Logic testing and approaches based on side channel information analysis are examples of non-destructive detection methods. Logic testing [7, 8] entails feeding the input vectors to the chip one by one and comparing the output vectors to the predefined correct output vectors. However, because it is based

Responsible Editor: M. Tehranipoor

✉ Jing Su
sujing@tust.edu.cn

[1] Artificial Intelligence Academy, Tianjin University of Science and Technology, TianJin 300457, China

on the principle of exhaustive enumeration, it is more difficult to implement for large-scale integrated circuits.

To complete the hardware Trojan classification, the detection method based on side channel information analysis [9, 10] collects the circuit's side channel leakage information (electromagnetic, power consumption, delay information, transient current, etc.) and compares the difference between the information leaked by the "golden chip" and the Trojan chip. Due to its less restricted circumstances and higher detection accuracy, this method is preferred by a large number of researchers both at home and abroad [11] presents a detection technique based on path delay, the main idea of which is to judge the existence of Hardware Trojan based on the relative delay difference between the circuit to be tested and the original circuit [12] proposes a detection technology based on power information. This method obtains different data according to the power information of the original circuit and the circuit to be tested for data preprocessing, and then uses data classification algorithms to analyze the data to be tested to determine whether there are hardware Trojan. After extracting the electromagnetic information, [13] firstly weakens the noise information and reduces the redundancy of the data, then preprocesses the data, analyzes the data to be tested by the data processing algorithm, and completes the detection of the hardware Trojan by comparing and classifying the electromagnetic data. Those methods' fatal flaw is that it is easily influenced by the acquisition equipment, and the signal-to-noise ratio of the side channel information is frequently inadequate, resulting in insufficient acquisition of useful side channel information, model bias, and reduced detection accuracy.

The existing solutions to this problem are mainly from two perspectives: algorithmic model optimization and data enhancement. On the one hand, the algorithmic model optimization focuses on the classifier, the classification capability of the classifier is improved by continuously optimizing the classification model as well as the loss function [14] proposes a deep learning technique for hardware Trojan detection that employs deep learning algorithms to extract controllability and transfer probability values as Trojan features, classifies data using k-means clustering models, and eliminates the need for a "gold chip" as a reference. In [15], a method for detecting hardware Trojan using recurrent neural networks is proposed, which uses n-gram circuit segmentation techniques to model the target circuit and improves the efficiency of Hardware Trojan detection by continuously optimizing the recurrent neural network model. The data enhancement, on the other hand, focuses on the processing of the data to generate more side channel information and mitigate overfitting to some extent. In [16], a random shift method generates new trajectories by shifting the side channel trajectories randomly multiple times and appends these generated trajectories to the original dataset. A Synthetic Minority Oversampling Technique combined with Edited Nearest Neighbor is proposed in [17], which expands the data by applying SMOTE to all but the largest number of classes and then removing the noise in the classes with ENN.

Generative adversarial networks (GANs) have been tried in a variety of applications due to their outstanding performance in computer vision and text analysis. GANs were first proposed in [18], and their biggest advantage is that they can automatically learn the distribution of real data. It consists of a generator that generates fake data that is similar to the original data distribution, completing data augmentation, and a discriminator that distinguishes between real and fake data, both of which are constantly optimized to reach Nash equilibrium [19]. Later, GAN-based models such as CGAN [20], DCGAN [21], ACGAN [22], and WGAN [23] have been proposed. GANs can not only solve the problem of insufficient side channel information acquisition by automatically generating data, but also continuously update the generator and discriminator parameters for classification model training. In this way, both algorithmic model optimization and data enhancement can be achieved simultaneously. As a result, this paper presents an in-depth investigation into the use of GANs and its optimization model in the detection of hardware Trojan. The new D2ACGAN model, which combines the benefits of CGAN, ACGAN, and D2GAN algorithms, is used for side channel information generation and hardware Trojan detection in this paper, and it is compared to existing CGAN and ACGAN methods: The original data is first expanded using the improved D2AGAN model, and the potential feature information between the data is mined while expanding the data set, and the real and false data are used for classification detection by existing machine learning classification methods. Second, the D2ACGAN model is used for the hardware Trojan classification task, and the model is evaluated by combining two factors: time consumption and classification accuracy. The experimental results show that the proposed D2ACGAN method has better classification accuracy compared with the existing machine learning classification methods. Meanwhile, the D2ACGAN model has better data generation capability and classification performance than CGAN and ACGAN methods.

This paper is organized as follows: a brief description of the theoretical knowledge of GANs is given in Sect. 2, a detailed description of the D2ACGAN model is given in Sect. 3, an analysis of the experimental results is given in Sect. 4, and a conclusion is given in Sect. 5.

2 Theoretical Discussion of GAN

Generative adversarial network (GAN), as a generative algorithm, has become a hot research topic in several fields in recent years. Any form of GANs consists of two parts, a generator and a discriminator, which are mostly interpreted by neural

networks [24]. The relationship between the generator network and the discriminator network is similar to that between the counterfeiter and the appraiser, where the generator network is responsible for learning the distribution of the original data to generate as much fake data as possible that will not be identified, and the discriminator is responsible for distinguishing the original data from the fake data as much as possible [25]. In this process, the discriminator gives feedback for each data generated by the generator, and from this feedback, the generator learns how to improve to generate more realistic data. Each of them holds an adversarial idea to try to beat the other and finally achieve Nash equilibrium to accomplish the optimization goal [26, 27]. In this section, we present the original GAN and its variant (Fig. 1) in detail.

The original GAN is the simplest GANs method, the input of the generator is only random noise and use it to generate fake data with similar distribution to the real data, the input of the discriminator includes the real sample data and the generated fake data, the output is the probability of the real and fake data, the optimization function of GAN is as follows.

$$\min_G \max_D V(D,G) = E_{x\sim p_{data(x)}}\left[\log D(x)\right] + E_{z\sim p_{z(z)}}\left[\log\left(1 - D(G(z))\right)\right] \tag{1}$$

where, G denotes the generator, D denotes the discriminator, p_{data} denotes the real data distribution, p_z denotes the data distribution with random noise, $D(x)$ denotes the probability that D discriminates the input data as real data, and $G(z)$ denotes the fake data generated by the G.

CGAN is an extension of GAN to a conditional model by adding some additional information to both discriminator and generator, which can be of arbitrary types, such as data labels, constraints, etc. [28]. The input of its generator becomes splicing of extra information with random noise, and the input is the fake data generated under the constraint of extra information. The input of the discriminator becomes the splicing of real data, fake data, and conditional information. The optimization function of CGAN is given by the following equation.

$$\begin{aligned}\min_G \max_D V(D,G) = {} & E_{x\sim p_{data(x)}}\left[\log D(x|y)\right] \\ & + E_{z\sim p_{z(z)}}\left[\log\left(1 - D(G(x|y))\right)\right]\end{aligned} \tag{2}$$

where, y is the additional information for splicing.

ACGAN is a task-driven GAN that differs most from CGAN in that the whole structure can discriminate not only true and false data but also to classify them. The additional information spliced by its generator is category labeling, which turns the unsupervised problem into a supervised one. The generator consists of two tasks: (1) discriminating real data from generated data, similar to GAN and CGAN (2) completing the task of classifying all data, either as a binary classification problem or as a multi classification problem [29]. The objective function of ACGAN is as follows.

$$L_S = E\left[\log P\left(S = real\middle|X_{real}\right)\right] + E\left[\log P\left(S = fake\middle|X_{fake}\right)\right] \tag{3}$$

$$L_C = E\left[\log P\left(C = c\middle|X_{real}\right)\right] + E\left[\log P\left(C = c\middle|X_{fake}\right)\right] \tag{4}$$

where X_{real} is the real data,X_{fake} is the generated fake data, c denotes the category label, L_S denotes the true and false discriminant loss, and L_C denotes the classification loss. The objective of the discriminator D is to maximize L_S+L_C, and the objective of the generator is to minimize L_S-L_C.

D2GAN creates a very distinctive three-player game scenario. It is highlighted by changing the number of discriminators from one to two, which can solve the problem of insufficient diversity of samples generated by CGAN and ACGAN. The input and output of the generators are the same as those of CGAN, and both discriminators receive true and false data, with one of them giving high scores to the true data and the other to the generated false data. The optimization process uses KL scatter and inverse KL scatter as a unified optimization objective [30]. The optimization function of D2GAN is given by the following equation.

$$\begin{aligned}\min_G \max_{D_1,D_2} V\left(D_1,D_2,G\right) = {} & \alpha \times E_{x\sim p_{data(x)}}\left[\log D_1(x)\right] \\ & + E_{z\sim p_{z(z)}}\left[-D_1(G(z))\right] + E_{x\sim p_{data(x)}}\left[-D_2(x)\right] \\ & + \beta \times E_{z\sim p_{z(z)}}\left[\log D_2(G(z))\right]\end{aligned} \tag{5}$$

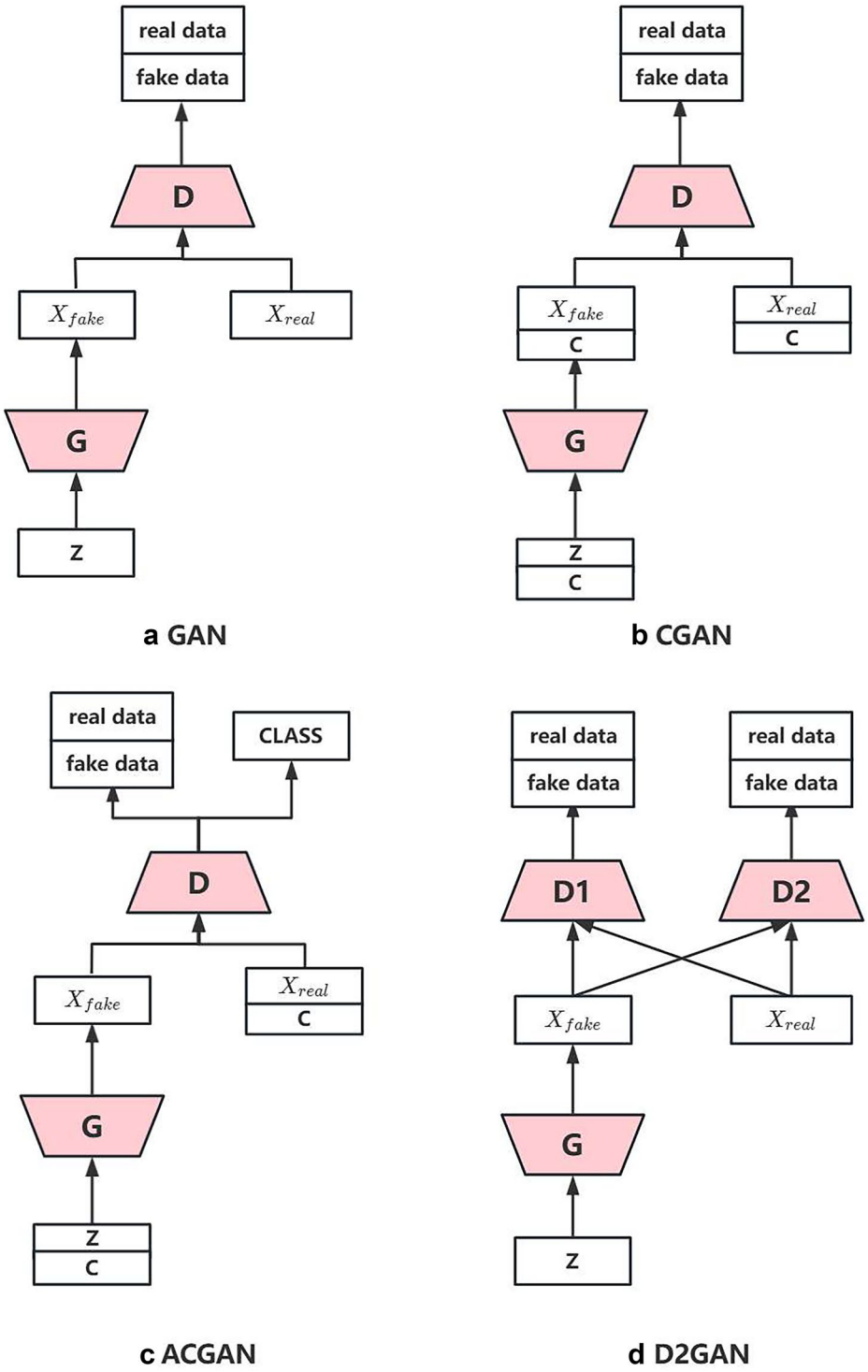
where, D_1 and D_2 represent the two discriminators, α and β(α >0, β<=1) are two parameters used to improve the stability of the learning process and control the influence of KL and reverse KL on the model, respectively.

3 Proposed D2ACGAN Model for Hardware Trojan Detection

3.1 Schematic of D2ACGAN

When used for hardware Trojan detection, traditional GANs models have the drawback of pattern collapse and the need for additional classifier training. When we say pattern collapse, we mean that the generator only generates data with one type of label. If a large number of samples are generated that are already skewed in number, the data imbalance problem is exacerbated. Furthermore, traditional GANs models can often only achieve expanded data, necessitating the use of another classifier to solve the classification problem, and training the two networks separately takes more time and resources. To address the aforementioned flaws, we propose a new D2ACGAN model, as shown in Fig. 2, which combines the benefits of the CGAN, ACGAN, and D2GAN algorithms.

Fig. 1 An illustration of GANs



3.2 D2ACGAN Framework for Hardware Trojan Detection

The framework proposed in this paper aims to synthesize high-quality side channel signals for data enhancement and improve the efficiency of chip hardware Trojan detection to be tested. The whole framework consists of three main parts: data acquisition, detection model training, and chip hardware Trojan detection. The detailed flow of the framework is shown in Fig. 3.

Stage 1: Data Acquisition The hardware Trojan detection and acquisition device consists of five parts: circuit board to be tested, electromagnetic probe, amplifier, oscilloscope, and PC. The test circuit board is a chip with a 1.8V power, the computer

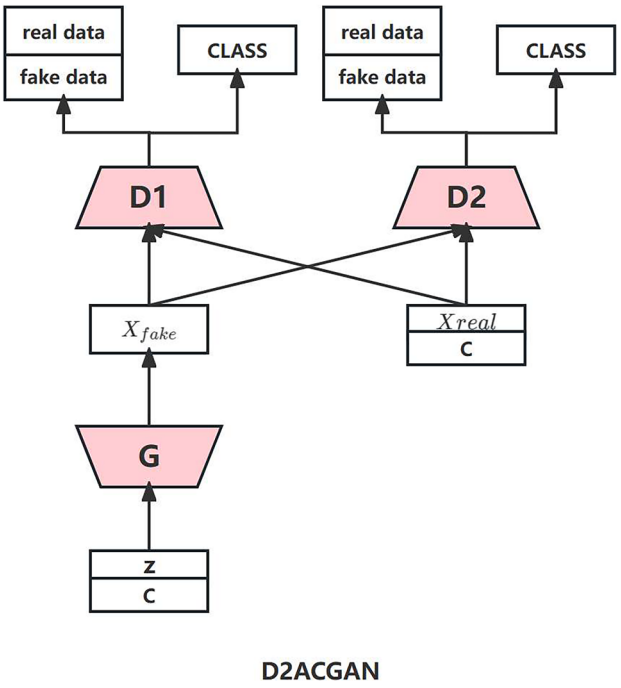


Fig. 2 An illustration of D2ACGAN

sends plaintext and key to the encryption chip, the chip performs AES encryption and returns the cipher text to the computer. During this period, the magnetic field changes on the chip surface are detected by an electromagnetic probe. A low-noise amplifier is connected between the probe and the oscilloscope to provide a certain gain. Then, the electromagnetic signal is measured with the oscilloscope and the data acquisition starts at the moment of detecting the rising edge and is transferred to the computer for storage, which is mainly responsible for communication and further analysis. The encryption and decryption module of FPGA device running AES cryptography algorithm is taken as the research object, i.e., the original circuit. The carrier type hardware Trojan prototype circuit is embedded in the same FPGA platform, and the platform encryption and decryption key information is broadcast externally through AM carrier, so that the Trojan designer can receive encryption and decryption key by wireless receiver. The work of Trojan circuit is to obtain the encryption key while the platform is running the encryption operation, and the key information is modulated and transmitted to the outside by AM carrier. Before hardware Trojan horse detection, the influence of noise information on side channel information should be reduced as far as possible. Generally, the method of multi-volume collection and average value can be adopted, which can eliminate part of the noise information. Then, the depth model of the auto encoder is used to process the side channel data.

Stage 2: Detection Model Training The whole dataset is divided into training and test sets, a D2ACGAN-based data generation and classification model is established, and the network model parameters of the generator and discriminator are initialized. To complete data augmentation, the generator generates fake data by receiving random noise

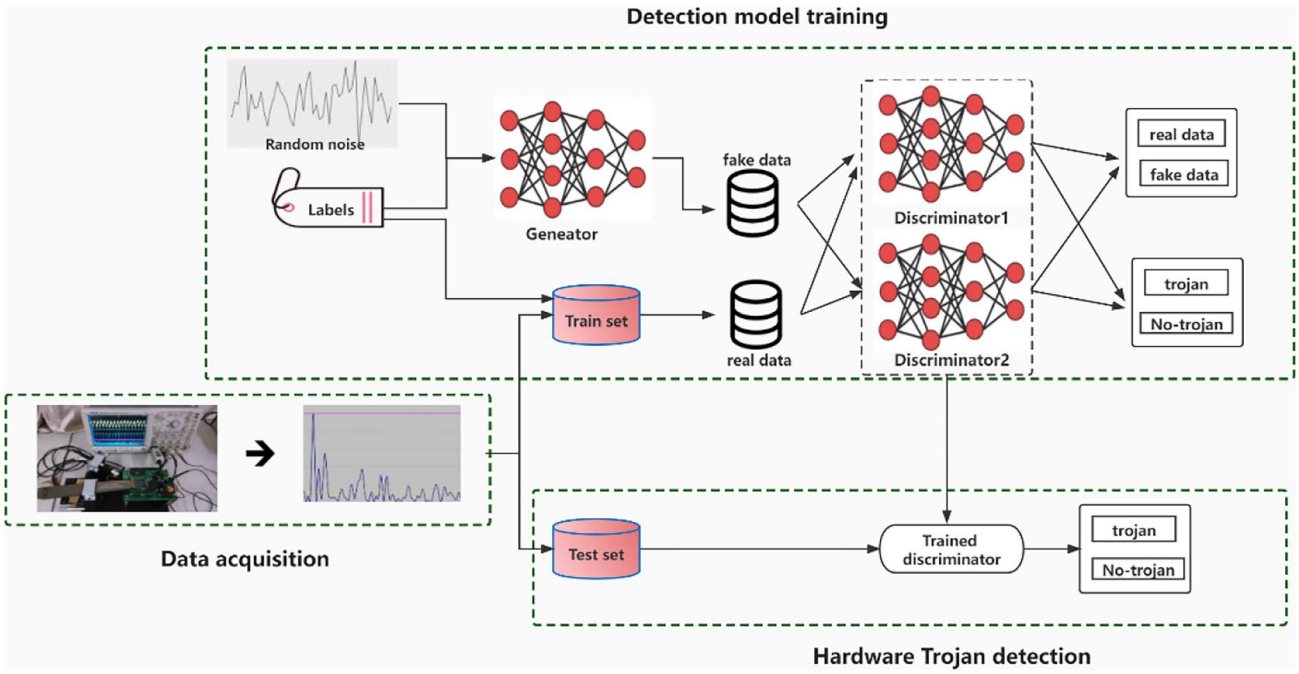


Fig. 3 D2ACGAN for hardware Trojan detection

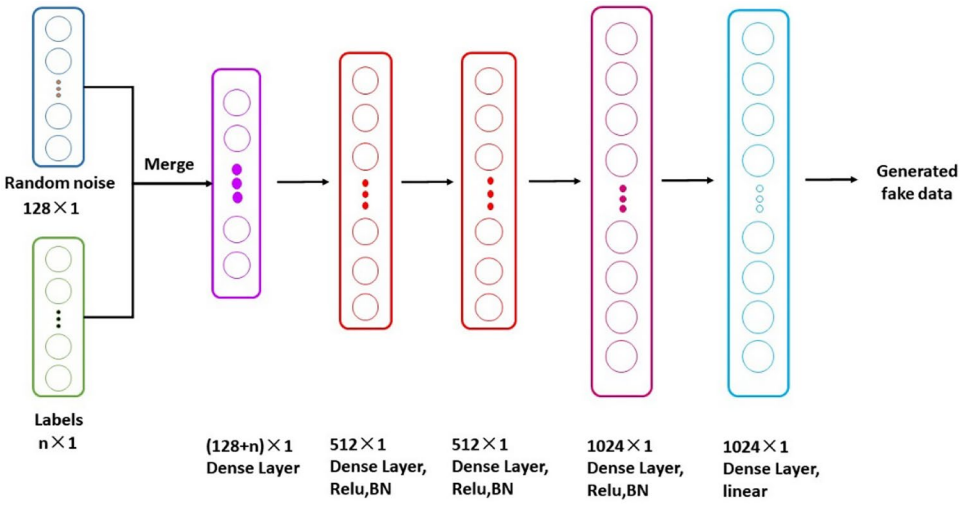


Fig. 4 Generator of D2ACGAN

with hardware Trojan category labels. The discriminator is also fed the real data from the training set as well as the fake data generated during data augmentation. Because the discriminator's two tasks, hardware Trojan classification and false data, share a single feature extraction network, resource competition is unavoidable. As a result, the discriminator's last layer is implemented with two independent fully connected layers to improve the independence of the two tasks and avoid mutual influence. The two discriminators are trained in turn before continuing to train the generator, and Nash equilibrium is achieved by alternating training several times.

Stage 3: Hardware Trojan detection Build the hardware Trojan classification model of the chip to be tested using the augmented dataset. The trained discriminator in Stage 2 is used as the hardware Trojan classification model in this stage, and the test set enters this model to complete the real and fake data discrimination and hardware Trojan detection, and output the classification results.

A deep generative adversarial structure based on D2ACGAN sits at the heart of the entire framework. As input values, random noise is coupled with feature labels containing hardware Trojan categories, and different labels help generate varied side channel information, which aids model convergence. The generative network consists of a 5-layer network structure, as illustrated in Fig. 4, and the random noise is merged with the category label vector in the input layer of the generative network, where n is the number of hardware Trojan categories. There are 512 neurons in the first two intermediate layers and 1024 neurons in the third intermediate layer. The activation functions of the three layers are all Relu, and each layer is connected using batch normalization. The output layer contains 1024 neurons and the activation function is linear. For the dual discriminator, we use the same network structure, as shown in Fig. 5. The true and false side channel

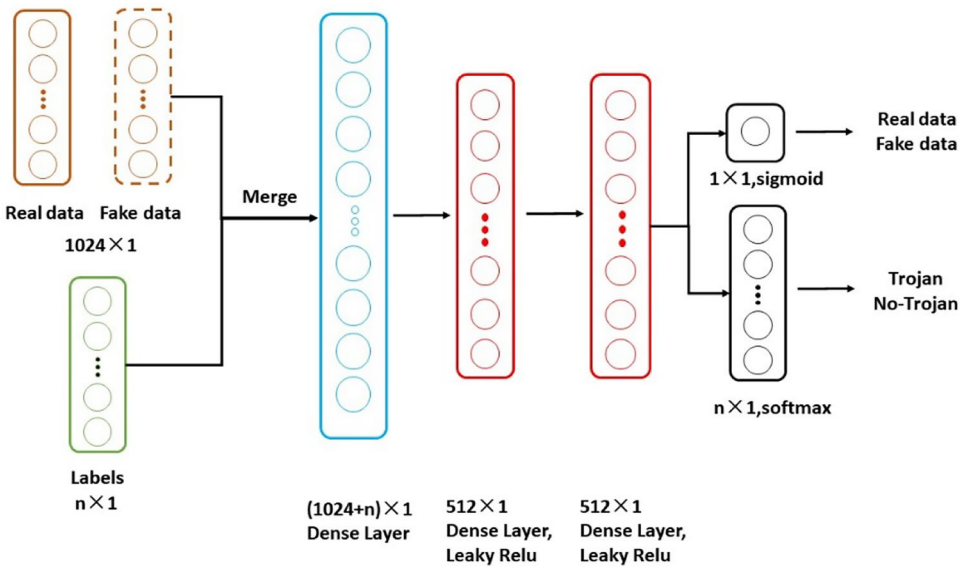


Fig. 5 Discriminator of D2ACGAN

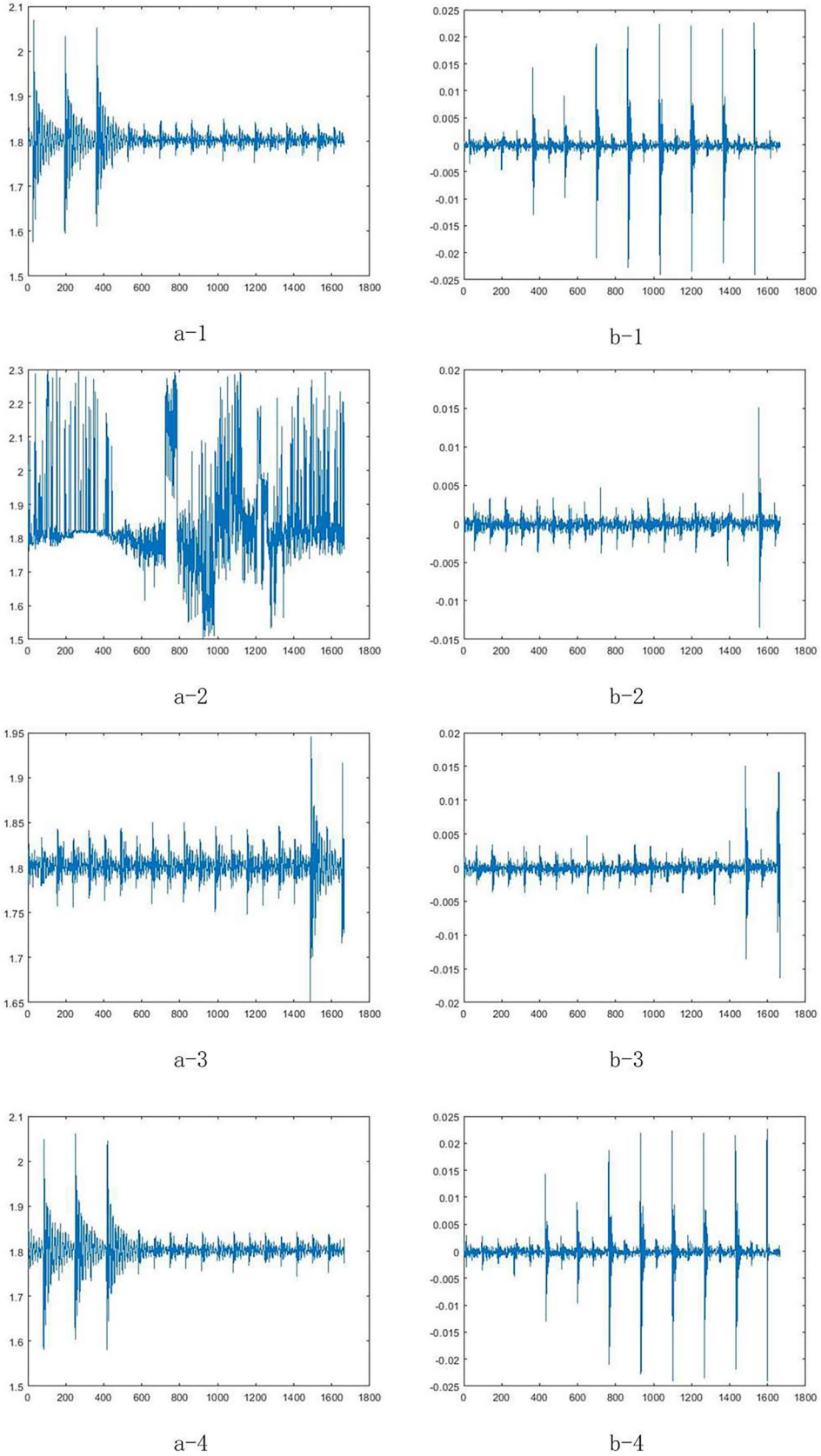


Fig. 6 Visualization of real and generated data

Table 1 Similarity comparison of different generation models on Power and Electromagnetism data

Model	Power			Electromagnetism		
	ED	CS	PCC	ED	CS	PCC
CCAN	7.6978	0.1727	0.0099	3.4893	0.0398	0.2527
ACGAN	5.3291	0.1065	0.7450	1.1673	0.0115	0.7758
D2GAN	5.1001	0.0964	0.7708	1.0492	0.0100	0.7971
D2ACGAN	1.3477	0.0241	0.9614	0.3284	0.0023	0.9786

information is combined with the feature labels into the input layer of the discriminator network, respectively. The middle layer is two fully connected layers with 512 neurons and the activation function is Leaky Relu. The output layer is two fully connected layers with no interference and the activation functions are sigmoid and softmax.

4 Experimental Verification

4.1 Generated Data Results Analysis

In this section, we compare the generation effects of both power consumption and electromagnetic side channel information by different generation models. The original data and 2000 randomly selected sample points from the two types of generated data are acquired and their power consumption and EM side channel information is visualized as shown in Fig. 6.

Figure 6 roughly demonstrates the similarity between the original data and the data generated by different generative models. Specifically, a-1—a-4 are the visualization plots of power consumption information for the original data, CGAN-generated data, ACGAN-generated data, and D2ACGAN-generated data, respectively, and b-1—b-4 are the visualization plots of EM information for the original data, CGAN-generated data, ACGAN-generated data, and D2ACGAN-generated data, respectively. It can be roughly seen from the plots that the three generation models have different learning abilities for the two side channel information; D2ACGAN has the strongest learning ability and the generated data are most similar to the original data, ACGAN is the second, and CGAN has the worst learning ability and the lowest similarity to the original data; in addition, the electromagnetic side channel information is easier to be learned compared with the power consumption information.

Table 2 Classification accuracy of 5 classifiers under different data augmentation strength of 4 generation models

Model		Classification accuracy (%)				
		1000	3000	5000	7000	10000
CNN	CGAN	95.53	95.70	95.99	96.07	96.31
	ACGAN	95.60	95.91	96.27	96.48	96.86
	D2GAN	95.66	96.01	96.33	96.64	96.84
	D2ACGAN	95.92	96.24	96.45	96.72	97.04
GB	CGAN	71.91	71.95	72.07	75.53	79.26
	ACGAN	72.82	76.38	77.27	79.96	83.00
	D2GAN	73.03	75.11	72.26	75.95	80.27
	D2ACGAN	75.01	72.23	72.29	74.04	77.30
LR	CGAN	71.45	72.23	72.29	74.04	77.30
	ACGAN	73.45	73.31	73.80	74.69	78.11
	D2GAN	73.61	73.84	74.09	75.14	78.42
	D2ACGAN	73.72	74.26	75.55	76.40	79.90
RF	CGAN	72.09	75.31	75.13	77.18	79.20
	ACGAN	72.09	75.77	76.24	79.33	83.12
	D2GAN	73.85	75.84	79.58	80.05	83.66
	D2ACGAN	74.90	76.98	80.09	80.12	83.88
SVM	CGAN	71.91	74.31	74.67	75.53	76.94
	ACGAN	73.18	74.92	77.53	79.96	81.25
	D2GAN	73.51	75.28	78.68	80.44	82.65
	D2ACGAN	75.57	77.12	80.48	81.91	84.33

The classification results of the five classifiers on the original data are 69.80%, 69.60%, 70.10%, 69.80%, and 69.30%, respectively

Table 3 Classification accuracy mean value, maximum value, minimum value and variance of different models

Model	Mean value	Max value	Min value	Variance
CGAN+CNN	95.92	96.37	95.46	3.6×10^{-5}
ACGAN	95.90	96.30	95.51	3.1×10^{-5}
D2GAN	96.03	96.52	95.94	2.7×10^{-5}
D2ACGAN	97.08	97.33	96.64	1.1×10^{-5}

We utilize several statistical measures to quantify the similarity of the generated data to the original data in Fig. 4 to more properly compare the capacity of different generative models to generate data. Traditional GANs for image generation mostly use inception score (IS) [31], the Frechet inception distance (FID) [32], and sliced Wasserstein distance (SWD) [33] to measure the generative effect. However, our experimental data is one-dimensional, those methods are only applicable to two-dimensional data, but not to our experimental data. Therefore, we choose Euclidean distance (ED), cosine distance (CS) and Pearson correlation coefficient (PCC) to measure the similarity between the original data and the generated data. Therefore, we choose Euclidean distance (ED), cosine distance (CS) and Pearson correlation coefficient (PCC) to measure the similarity between the original data and the generated data, as shown in Table 1. The two distances between the original data and the real data are represented by ED and CS, with smaller values indicating greater similarity, while PCC represents the linear relationship between the original data and the real data, and larger values indicate the higher similarity.

The statistical metrics in Table 1 show the average performance of the three generation models. As can be seen from Table 1, for both side channel information, the statistical metrics of D2ACGAN generated data are significantly better than the other three generated models, indicating that its generated data have the highest similarity to the original data and the best generation effect, which is consistent with the visualization results of Fig. 6.

Based on the generated dataset, we compared the results of using five classifiers, Convolutional Neural Networks (CNN), Gradient Boosting (GB), Logistic Regression (LR), Random Forest (RF), and Support Vector Machine (SVM), for the data generated by the four generation models, as shown in Table 2.

As shown in Table 2, the hardware Trojan detection based on data enhancement is very effective. The results illustrate two conclusions: First, the classification accuracy of the unadded balanced dataset is significantly lower than that of the added augmented dataset, and the classification accuracy of all five classifiers shows an increasing trend with the increasing intensity of data augmentation, indicating that the more generated data are added, the better the classification effect is; Second, regardless of the data enhancement intensity, the classification accuracy of the classifiers trained by the generated data of CGAN, ACGAN, D2GANand D2ACGAN tends to increase gradually, implying that D2ACGAN can solve the pattern collapse problem more effectively, further validating the previous results of Fig. 6 and Table 2. In summary, the data generated by the D2ACGAN model is most similar to the original data and can obtain the best classification accuracy on different classifiers.

4.2 Hardware Trojan Classification Results Analysis

The model used in this paper aims to expand the side channel information and improve the detection accuracy of hardware Trojan. Therefore, after comparing the ability of recursive GANs to generate data, we further compare their hardware Trojan classification ability. Since CGAN does not complete the classification task, we choose the CNN method with the best classification effect in Table 2 combined with CGAN and ACGAN, D2GAN, D2ACGAN to compare the hardware Trojan classification results, and the specific results are shown in Table 3.

After the network training is completed, Table 3 illustrates the mean, maximum, minimum, and variance of the accuracy of the three different models. D2ACGAN has the highest mean, maximum, and minimum values, indicating that it has the best classification effect, D2GAN has the second-best classification effect, CGAN+CNN has the third-best classification effect, and ACGAN has the worst classification effect; additionally, D2ACGAN has the smallest variance, indicating that it is more stable than the other three models.

In addition, we compare the training and classification time consumption when different models reach stability in the hardware Trojan classification scenario, and the results are shown in Table 4.

Table 4 Consuming time (s) of different models

Consuming time(s)	Model				
	CGAN+CNN	ACGAN	D2GAN	D2ACGAN	SVM
Training time	14302	8184	8658	9250	6336
Classing time	5.7×10^{-5}	5.9×10^{-5}	6.2×10^{-5}	6.3×10^{-5}	8.8×10^{-5}

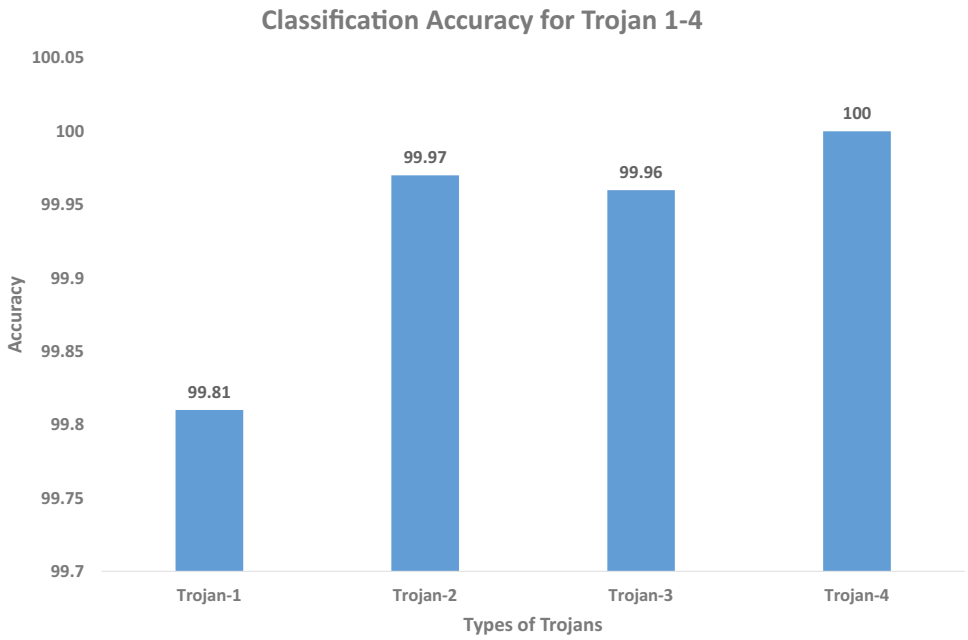


Fig. 7 D2ACGAN classification accuracy for Trojan 1-4

The analysis results show that the CGAN+CNN method, which requires training two models, consumes the longest time, with a training time of 14302s alone, D2ACGAN takes a little longer than ACGAN because it has to train two discriminators, and SVM, which does not involve the training process of deep learning, has the shortest training time.

Combining the results of Table 3 and Table 4, SVM takes the least time but has the worst classification effect; CGAN+CNN has a higher classification accuracy but also the highest time cost; ACGAN takes less time and has an average accuracy; D2ACGAN takes the middle time and has the best classification effect, so D2ACGAN is the better model under the consideration of both time and classification effect.

To further verify the classification ability of D2ACGAN, we used it for a multi-classification hardware Trojan classification task. The experimental dataset contains four hardware Trojans, and the results are shown in Fig. 7.

Figure 7 shows that the D2ACGAN method has the same high detection results for multi-classification hardware Trojan detection. Especially for Trojan-4, the classification accuracy reaches 100%.

As shown in Table 5, this paper also makes a comparison with network-related methods. The same dataset that was used in the experiment in this paper was used in all comparative experiments. Additionally, it is seen to be the case that, compared with the literature [34–36], the proposed detection model improves the detection accuracy effectively.

Table 5 Classification results of network-related methods

Model Name	Accuracy	Precision	Recall	F1 Score
D2ACGAN	92.41	90.12	90.75	89.92
[34]	75.10	76.41	71.26	76.45
[35]	81.25	80.38	79.14	80.73
[36]	85.63	83.02	79.44	81.82

5 Conclusion

This paper proposes a new hardware Trojan detection method based on D2ACGAN model. This method combines the network structure of ACGAN and D2AGN, which can improve the quality of generated data and the detection accuracy of the hardware Trojan, and solve the problems of insufficient acquisition of side channel information and low detection rate during hardware Trojan detection. We conducted several validation experiments on the data set, and the experimental results show that, measured mathematically, our method can generate fake data closer to the original data than CGAN, ACGAN, and D2GAN. At the same time, the data generated by D2ACGAN is used to train the classification model for hardware Trojan detection, and the accuracy rate can reach 97.04%, which proves that our method has better scalability. In addition. We also use different models to complete binary and multi-classification Trojan detection, and compare with the existing detection methods. The results show that the detection accuracy of D2ACGAN reaches 92.41%, which is significantly higher than the other three methods.

Acknowledgments We thank all the reviewers for their helpful feedback and advice. This work was supported by Natural Science Foundation of Tianjin (No. 19JCYBJC15300).

Data Availability The data that support the findings of this study are not openly available due to their involvement in military projects and are available from the corresponding author upon reasonable request (sujing@tust.edu.cn).

Declarations

Conflict of Interest We declare that we have no conflict of interest.

References

1. Yeh A (2012) Trends in the global IC design service market. DIGITIMES Res
2. Bhunia S, Tehranipoor MM (2017) The Hardware Trojan War: Attacks, Myths, and Defenses. 1st ed. Springer, Heidelberg. https://doi.org/10.1007/978-3-319-68511-3
3. Hayashi Y, Kawamura S (2020) Survey of hardware trojan threats and detection. In: International Symposium on Electromagnetic Compatibility-EMC EUROPE, pp. 1–5. Rome. https://doi.org/10.1109/EMCEUROPE48519.2020.9245675
4. Khamitkar R, Dube RR (2022) A Survey on Using Machine Learning to Counter Hardware Trojan Challenges. In: ICT with Intelligent Applications, pp.539-547. Singapore. https://doi.org/10.1007/978-981-16-4177-0_53
5. Jain A, Zhou Z, Guin U (2021) Survey of Recent Developments for Hardware Trojan Detection. In: IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1-5. Daegu. https://doi.org/10.1109/ISCAS51556.2021.9401143
6. Wang X, Narasimhan S, Krishna A, Bhunia S (2012) Side-channel analysis based reverse engineering for post-silicon validation. In: 25th International Conference on VLSI Design, pp. 304-309. IEEE, Hyderabad. https://doi.org/10.1109/VLSID.2012.88
7. Zhou Z, Guin U, Agrawal VD (2018) Modeling and test generation for combinational hardware Trojans. In: 36th VLSI Test Symposium, pp. 1–6. IEEE, San Francisco. https://doi.org/10.1109/VTS.2018.8368626
8. Farahmandi F, Huang Y, Mishra P (2017) Trojan localization using symbolic algebra. In: 22nd Asia and South Pacific Design Automation Conference (ASP-DAC), pp. 591-597. Chiba. https://doi.org/10.1109/ASPDAC.2017.7858388
9. Rad R, Plusquellic J, Tehranipoor M (2009) A sensitivity analysis of power signal methods for detecting hardware Trojans under real process and environmental conditions. IEEE Trans Very Large Scale Integr (VLSI) Sys 18(12):1735-1744. https://doi.org/10.1109/TVLSI.2009.2029117
10. Hossain FS, Shintani M, Inoue M, Orailoglu A (2018) Variation-aware hardware Trojan detection through power side-channel. In: IEEE International Test Conference, pp. 1–10.IEEE, Phoenix. https://doi.org/10.1109/TEST.2018.8624866
11. Nejat A, Hely D, Beroulle V (2015) Facilitating side channel analysis by obfuscation for Hardware Trojan detection. In 2015 10th International Design & Test Symposium (IDT), pp.129-134. IEEE, Amman. https://doi.org/10.1109/IDT.2015.7396749
12. Xue M, Bian R, Liu W (2018) Defeating Untrustworthy Testing Parties: A Novel Hybrid Clustering Ensemble Based Golden Models-Free Hardware Trojan Detection Method. IEEE Access 7:5124–5140. https://doi.org/10.1109/ACCESS.2018.2887268
13. He J, Liu Y, Yuan Y, Hu K, Xia X, Zhao Y (2019) Golden Chip Free Trojan Detection Leveraging Electromagnetic Side Channel Fingerprinting. IEICE Electronics Express 16(2):1–8. https://doi.org/10.1587/elex.16.20181065
14. Reshma K, Priyatharishini M, Nirmala Devi M (2019) Hardware trojan detection using deep learning technique. In: Soft Computing and Signal Processing, pp. 671-680. Springer, Singapore. https://doi.org/10.1007/978-981-13-3393-4_68
15. Lu R, Shen H, Su Y, Li H, Li X (2019) Gramsdet: Hardware trojan detection based on recurrent neural network. In: 28th Asian Test Symposium (ATS), pp. 111-1115. IEEE, Kolkata. https://doi.org/10.1109/ATS47505.2019.00021
16. Pu S, Yu Y, Wang W, Guo Z, Liu J, Gu D, Wang L, Gan J (2017) Trace augmentation: What can be done even before preprocessing in a profiled sca? In: International Conference on Smart Card Research and Advanced Applications, pp. 232-247. Springer, Cham. https://doi.org/10.1007/978-3-319-75208-2_14
17. Picek S, Heuser A, Jovic A, Bhasin S, Regazzoni F (2019) The curse of class imbalance and conflicting metrics with machine learning for side-channel evaluations. IACR Trans Cryptographic Hardware Embedded Sys (1):1-29. https://doi.org/10.13154/tches.v2019.i1.209-237
18. Generative Adversarial Networks. https://arxiv.org/abs/1406.2661
19. Creswell A, White T, Dumoulin V, Arulkumaran K, Sengupta B, Bharath AA (2018) Generative adversarial networks: An overview. IEEE Signal Processing Magazine 35(1):53–65. https://doi.org/10.1109/MSP.2017.2765202
20. Conditional Generative Adversarial Nets. https://arxiv.org/abs/1411.1784
21. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. https://arxiv.org/abs/1511.06434
22. Conditional Image Synthesis With Auxiliary Classifier GANs. https://arxiv.org/abs/1610.09585
23. Wasserstein GAN. https://arxiv.org/abs/1701.07875v2
24. Kusiak A (2020) Convolutional and generative adversarial neural networks in manufacturing. International Journal of Production Research 58(5):1594–1604. https://doi.org/10.1080/00207543.2019.1662133
25. Data Synthesis based on Generative Adversarial Networks. https://arxiv.org/abs/1806.03384v2
26. Pan Z, Yu W, Yi X, Khan A, Yuan F, Zheng Y (2019) Recent progress on generative adversarial networks (GANs): A survey. IEEE Access 7:36322–36333. https://doi.org/10.1109/ACCESS.2019.2905015
27. Shaker AM, Tantawi M, Shedeed HA, Tolba MF (2020) Generalization of convolutional neural networks for ECG classification using generative adversarial networks. IEEE Access 8:35592–35605. https://doi.org/10.1109/ACCESS.2020.2974712
28. Douzas G, Bacao F (2018) Effective data generation for imbalanced learning using conditional generative adversarial networks. Expert Systems with applications 91:464–471. https://doi.org/10.1016/j.eswa.2017.09.030
29. Kamal S, Mujeeb A, Supriya MH (2022) Generative adversarial learning for improved data efficiency in underwater target classification. Eng Sci Technol Int J 30:101043. https://doi.org/10.1016/j.jestch.2021.07.006
30. Dong F, Zhang Y, Nie X (2020) Dual discriminator generative adversarial network for video anomaly detection. IEEE Access 8:88170–88176. https://doi.org/10.1109/ACCESS.2020.2993373

31. Improved Techniques for Training GANs. https://arxiv.org/abs//1606.03498
32. Heusel M, Ramsauer H, Unterthiner T, Nessler B, Hochreiter S (2017) Gans trained by a two time-scale update rule converge to a local nash equilibrium. Adv Neural Inform Proc Sys 30
33. Computational Optimal Transport. https://arxiv.org/abs/1803.00567
34. Madden K, Harkin J, McDaid L, Nugent C (2018) Adding Security to Networks-on-Chip using Neural Networks. In: Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1299–1306. Bangalore, India. https://doi.org/10.1109/ssci.2018.8628832
35. Reshma K, Priyatharishini M, Nirmala Devi M (2019) Hardware Trojan Detection Using Deep Learning Technique. In: Soft Computing and Signal Processing; Advances in Intelligent Systems and Computing, pp. 671–680. Springer: Singapore. https://doi.org/10.1007/978-981-13-3393-4_68
36. Hu T, Dian S, Jiang R (2020) Hardware Trojan detection based on long short-term memory neural network. Eng 46:110–115. https://doi.org/10.19678/j.issn.1000-3428.0055589

Wenjing Tang received her B.S. degree from the School of Data Science and Software Engineering, Qingdao University, Qingdao, China and is currently a postgraduate student. Her research interests are intelligent information processing and Hardware Trojan detection.

Jing Su received her B.S. and M.S. degrees in School of Computer Science and Information Engineering, Tianjin Normal University, Tianjin, China; received her Ph.D. degree from School of Information, Tianjin University, Tianjin, China. In 2003, she joined the School of Computer and Information Engineering, Tianjin University of Science and Technology. She is currently an associate professor at the School of Artificial Intelligence. Her research interest covers intelligent information processing and Hardware Trojan detection.

Yuchan Gao received her B.S. degree from the School of Computer and Information Engineering, Tianjin University of Science and Technology, Tianjin, China and is currently a postgraduate student. Her research interests are intelligent information processing and Hardware Trojan detection.