Hardware Trojan detection research based on MLP

Qiao Ding¹, Shizhuang Yin ¹, Lijun Liu ¹, Chao Wang²

¹Equipment Simulation Training Center. The Army Engineering University of PLA, Shijiazhuang Campus, Shijiazhuang, 050003, China. 644097601@qq.com

²Unit 66389 of PLA, Xian 710100, China. dynasty112@foxmail.com

Abstract. In view of the variety of Hardware Trojan (HT) and the difficulty of obtaining unknown Trojan characteristics in side-channel signals by conventional methods. In this paper, MLP was selected to establish the network model by means of supervised learning, the method took supervised learning ways to build neural network model for feature extraction and discrimination of side channel information. A verification system was set up based on FPGA to obtain side-channel information. The results show that the detection rate of the MLP is more than 1% higher than that of the traditional support vector machine (SVM) method when detecting the hardware Trojan horse with the parent circuit area of 2%.

1. Introduction

When one or more links such as design, production, test and assembly in IC development are not trusted, Trojan circuit may exist [1]. The side-channel analysis method is based on the principle that inserting a Trojan circuit into the FPGA chip will reveal the characteristics of the side channel. Commonly used methods such as power consumption [2], delay [3], electromagnetic radiation [4] and other parameters to detect the existence of Trojan horse circuits. For example, if there are \( N_{\text{origin}} \) logic gates in the “golden circuit” and the measured quiescent current is \( I_{\text{origin}} \), the current change after inserting \( N_{\text{trojan}} \) logic gates in the circuit is \( \Delta I_{\text{trojan}} \) [5]. The hardware Trojan circuit can be distinguished by the change of current. The existing literature has proposed a variety of side-channel analysis methods. Its advantage is that it has a high detection rate for large Trojans. The disadvantage is that the error caused by the processing process and environmental noise is large, making it difficult to detect small-scale circuits.

Machine learning and the combination of the side channel analysis method can effectively improve precision hardware Trojan detection, Jain G et al. [6] proposed a based on support vector machine (SVM) hardware Trojan detection method, based on literature [7] K-L transform to extract the feature vector, the SVM model was constructed in detection of 2% of the female parent circuit hardware trojans, testing accuracy reached 98.1%. But for the side channel data of large sample, support vector machine (SVM) can only solve the problem of classification and cannot extract data features effectively. According to the existing literature, there are few articles combining deep learning with side channel analysis. How to use deep learning method to effectively extract side channel data features and realize correct classification and recognition is an urgent problem to be solved. This paper presents a hardware Trojan horse identification method based on deep learning. MLP was selected to identify the hardware Trojans with the side channel data, and good results were obtained.
2. Multi-layer Perceptron Basic Framework and Network Model

The perceptron model is shown in Figure 1. Each node in the neural network is a perceptron. The electrical signals from the outside world (environment or other cells) are transmitted to the neurons through synapses. The cell is activated and sends electrical signals to the next cell through the axon to complete the processing of outside information. In Figure 4, \( P_1, \ldots, P_n \) are the input data. The output of each layer of the perceptron is fed back to the input of the next layer. The entire network can be expressed as: \( a^l = f(W^l a^{l-1} + b^l) \), where the function is selected as the activation function of the last layer. Suppose the output of the original neural network are \( y_1, y_2, \ldots, y_n \), then the output after regression processing are:

\[
\text{softmax}(y) = \frac{e^{y_i}}{\sum_j e^{y_j}}
\]

(1)

This turns the output of the neural network into a probability distribution.

We choose cross entropy as the loss function. As shown in the following formula:

\[
C = -\frac{1}{n} \sum_y [y \ln a + (1 - y) \ln(1 - a)]
\]

(2)

\( Y \) represents the distribution of real labels, \( a \) represents the predicted label distribution of the trained model. The cross-entropy loss function \( C \) can measure the similarity of \( y \) and \( a \) to achieve classification. We adjust the weights and offsets inside the neural network to make the loss function \( C \) continue to decrease. In this paper, the Adam optimization algorithm is used to iteratively update the network weights.

![Fig.1 Perceptron model](image)

Table 1 Perceptron model structure

| LAYER               | OUTPUT SHAPE     |
|---------------------|------------------|
| InputLayer          | (None, 10000)    |
| dense_1             | (None, 500)      |
| batch_normalization1| (None, 500)      |
| leaky_relu_1        | (None, 500)      |
| dropout_1           | (None, 500)      |

We select 10000 input, 2 output, and 4 hidden layer perceptron models. The number of hidden layer neurons is 500, 500, 100, 100, respectively. The structure of each layer is shown in Table 1. In order to solve the effect of offset and increase of input data, we add the BN (batch normalization) layer before the activation function, and normalize the input of the input activation function. The hidden layer activation function we select the leaky_relu function, which is to solve the problem that the reLu function enters the negative interval and causes the neuron not to learn continue. Due to the large number of single-layer neurons, we added dropout. The purpose is to selectively disable some...
neurons, thereby simplifying the model, improving the model generalization force, and avoiding overfitting. The dropout value is selected as 0.5.

3. Experimental results and analysis

3.1. Physical data sampling
The experiment uses the Japanese SASEBO development board of Xilinx FPGA chip, realizes the 10-round AES encryption algorithm with plaintext and key length of 128bit in the chip[7], and designs a combined hardware Trojan that accounts for 0.3% of the encryption circuit. In the experiment, a set of plain text and its corresponding key are selected and fixed, and a digital oscilloscope (frequency 2.50G Sa/s, duration 0.2μs) is used to repeatedly collect 3K power-consumption side-channel signals for each measurement circuit. The signal sampling in each side-channel length is 10K. We have performed noise reduction processing on the data in advance, each side-channel signal sample is obtained by repeating the sampling 10 times and then averaged. Multilayer Perceptron was built in TensorFlow of 64-bit Win7 system. The experimental schematic diagram is shown in Figure 2.

![Fig.2 Schematic diagram of the experimental platform](image)

3.2. Experimental process and steps
Step 1. We label 3000 current data without Trojan horse as label 0, and 3000 current data with Trojan horse as label 1, merge and shuffle the two sets of data.

Step 2. For 6000 sets of current data samples, 5400 are taken as the training set, of which 540 are used as the verification set, and the remaining 600 are used as the test set.

Step 3. Import the data samples into the multi-layer perceptron model for training.

Step 4. After the model training is successful, input the test set data into the training model, compare with the predicted value and the original label of the data. This process uses one-hot encoding. If the values are the same (both are 0 or 1), the classification is proved correct, the result will be recorded as \( n_{\text{true}} \); if it is different (neither 0 nor 1), the classification is proved wrong, the result will be recorded as \( n_{\text{false}} \). We define accuracy(\( acc \)) as tracking value: \( acc = \frac{n_{\text{true}}}{n} \)

3.3. Experimental results and analysis

![Figure 3 MPL detection effect diagram](image)
The same data set as literature [7] was used for simulation comparison experiment, and the results were shown in Table 2. It can be seen that under the same Trojan horse proportion, the detection accuracy of MLP network models constructed in this paper is higher than that in the literature [7]. MLP uses less detection time and has better detection efficiency.

Table 2 compares results with literature [7]

| Method   | Precision | Epoch | Trojan ratio |
|----------|-----------|-------|--------------|
| MLP      | 99.1%     | 15    | 2%           |
| Reference[7] | 98.1%     | 20    | 2%           |

4. Conclusion

By constructing a classical structure MLP in the neural network and combining with the side channel analysis method, the correct classification of hardware Trojan bypass data is realized. This experiment fully proves that it is feasible to use deep learning to implement feature extraction of the opposite side channel data. Compared with the traditional method, it is more convenient and accurate. It can play a good identification effect for the subtle data differences obtained by the side channel detection.

References

[1] Huang Zhao, Wang Quan, Yang Pengfei. Hardware Trojan Horse: Research Progress and New Trends of Key Issues[J]. Chinese Journal of Computers. 2019.5.
[2] Xiao K, Zhang X H, Tehranipoor M.A clock sweeping technique for detecting hardware Trojans impacting circuits delay. IEEE Design & Test, 2013, 30(2):26-34.
[3] Aarestad J, Acharyya D, Rad R. Detecting Trojans through leakage multiple supply pad IDDQs. IEEE Transactions on Information Forensics and Security, 2010, 5(4):893-904.
[4] Nowroz A N, Hu K Q, Koushanfar F. Novel techniques for high-sensitivity hardware Trojan detection using thermal and power maps. IEEE Transactions on Computers, 2014, 33(12):1792-1805.
[5] M. Tehranipoor, C. Wang [M]. New York: Springer, 2011: 42-45.
[6] Jain G, Raghuwanshi S, Vishwakarma G. Hardware Trojan: Malware Detection Using Reverse Engineering and SVM[C]/International Conference on Intelligent Systems Design and Applications. Springer, Cham, 2017:530-539.
[7] Wei Yanhai, Li Xiongwei, Zhang Yang. A new method of hardware Trojan detection based on K_L transformation and SVM. [J]. Firepower and Command Control. 2019.6.