Computer Networks and Communications
https://ojs.wiserpub.com/index.php/CNC

Research Article

An Enhanced Detection Method of Hardware Trojan Based on CNN-Attention-LSTM

Nanmin Wang, Haiyan Kang*

Computer School, Beijing Information Science and Technology University, Beijing, China
E-mail: kanghaiyan@126.com

Received: 5 August 2024; **Revised:** 22 October 2024; **Accepted:** 31 October 2024

Abstract: The security issues caused by the insertion of hardware Trojans seriously threaten the security and reliability of the entire hardware device. This article constructs a detection model that combines convolutional neural networks (CNN) and long short-term memory networks (LSTM), and introduces attention mechanism to enhance the model's ability to recognize complex circuits. This method can automatically learn and optimize feature extraction and classification models, reduce reliance on manual experience through training on large amounts of data, and improve the intelligence level of detection. Especially, by combining attention mechanisms and LSTM models, it is possible to more effectively capture small anomalies in circuit design and improve the accuracy and efficiency of hardware Trojan detection. The experimental results show that the proposed CNN-Attention-LSTM model exhibits superior Trojan detection performance and good generalization ability on different datasets, with an precision of 96.3%, a recall rate of 94.7%, and an F1 score of 95.5%.

Keywords: hardware trojan detection, CNN, attention, LSTM

1. Introduction

Hardware Trojans refer to circuits maliciously implanted during chip design or manufacturing, which can be activated under specific conditions, leading to data leakage or other security issues. With the increasing complexity of security threats such as hardware Trojans, traditional detection methods rely on expert knowledge and rule matching, which are no longer sufficient to meet the needs. With the rise of deep learning technology [1], methods based on convolutional neural networks (CNN) have shown great potential in image recognition and pattern detection. Deep learning technology can automatically learn and optimize feature extraction and classification models through training on large amounts of data, reducing reliance on manual experience and improving the intelligence level of detection. Especially, combining attention mechanism (Attention) [2] and long short-term memory network (LSTM) models [3] can more effectively capture subtle anomalies in circuit design and improve the detection accuracy of hardware Trojans. However, these methods often require a large amount of centralized data for training, which is often limited by data privacy and security regulations in practical applications.

Among the myriad of deep learning architectures [4], this paper opted to combine CNN and LSTM due to their complementary strengths in hardware Trojan detection tasks. CNN's robust image recognition and feature extraction capabilities, coupled with LSTM's prowess in handling sequential data, enable our model to effectively capture local and

DOI: https://doi.org/10.37256/cnc.2220245407

temporal anomalies in circuit designs. Furthermore, the introduction of the attention mechanism enhances the model's ability to identify key features. While other architectures such as GANs and ResNeXt [5] excel in their respective domains, our model offers a balanced solution that adapts to the practical needs of hardware security, combining performance and efficiency.

This article proposes a hardware Trojan detection technique based on CNN-Attention-LSTM to address this issue. The integration of the attention mechanism refines the feature extraction process, directing the model to focus on the most critical aspects of the data for more accurate detection. By combining this with the predictive capabilities of LSTM, the model is better equipped to identify complex patterns indicative of hardware Trojans. This paper aims to establish a secure and efficient hardware Trojan detection system, leveraging the strengths of deep learning to enhance detection accuracy.

The main contributions of this article include:

(1) A hardware Trojan detection method that employs an enhanced deep learning technology is proposed. By adding attention mechanism to refine the feature extraction process and direct the model to focus on the most critical aspects of the data, it can improve the model's ability to identify complex circuit anomalies.

(2) A combination of CNN-Attention and LSTM is designed in this article. After extracting local features through CNN and Attention mechanisms, LSTM can further integrate these features to improve the performance and efficiency of the model.

(3) The effectiveness of the proposed method was verified through experiments, demonstrating the high-precision detection potential of CNN-Attention-LSTM in the field of hardware Trojan detection.

The following chapters will provide a detailed introduction to the relevant research on hardware Trojan detection technology, the proposed CNN-Attention-LSTM based detection technology and experimental design and result analysis.

2. Preliminaries and related works

2.1 *Preliminaries*

This section will introduce the preparatory knowledge of the method designed in this article, laying the foundation for future work

2.1.1 *Convolutional neural networks (CNN)*

Convolutional Neural Network (CNN) is a deep learning model that performs well in image and video recognition, classification, and other visual related tasks [6]. CNN can automatically extract features from images and learn by simulating the working principle of the human visual system. The core of CNN is the convolutional layer, which uses a set of learnable filters (also known as convolutional kernels) to scan the input image and capture local features. These filters are automatically adjusted during the training process to recognize patterns such as edges and textures in the image. Convolutional layers are typically used in conjunction with pooling layers, which are responsible for reducing the spatial dimension of features while increasing invariance to image displacement. In addition, CNN also includes Fully Connected Layers, which map the extracted features to the final output, such as category labels, at the end of the network.

Due to its significant performance in image recognition tasks, CNN has become a cornerstone in the field of computer vision and has been widely applied in autonomous driving, medical image analysis, facial recognition, and many other fields. With the continuous advancement of deep learning technology [7], the structure and variants of CNN are also constantly evolving to meet a wider range of application needs.

2.1.2 *Attention mechanism (Attention)*

Attention mechanism is a widely used mechanism in deep learning models. Its core idea is to enable the model to focus on the most important parts of the input data, thereby improving processing efficiency and accuracy.

In traditional neural networks, information is processed in a fixed way, regardless of whether it is crucial for the current task. The Attention mechanism allows the model to dynamically allocate its processing resources and focus on

those parts that are more informative for the current task. For example, when analyzing an image, more attention may be paid to the key objects or features in the image. Attention mechanisms are typically implemented by calculating the correlation between each element in the input sequence and other elements. This correlation can be measured through different methods, such as dot product, multiplication, or convolution.

A key advantage of the Attention mechanism [8] is its ability to provide interpretability for the model's decision-making process. By analyzing attention weights, we can understand how the model makes decisions. In addition, Attention mechanisms can also help improve the performance of models in complex tasks, such as reducing information loss when processing long sequence data, or integrating different types of information in multimodal data.

2.1.3 *Long Short-Term Memory Network (LSTM)*

Long Short-Term Memory Network (LSTM) is a special type of Recurrent Neural Network (RNN) designed to solve the long-term dependency problem encountered by traditional RNNs when processing long sequence data [9]. LSTM introduces three gating mechanisms: Input Gate, Forget Gate, and Output Gate, to control the flow of information and learn long-term and short-term temporal dependencies. The input gate is responsible for determining which new information will be stored in the cellular state, the forget gate determines which old information will be forgotten, and the output gate determines the next hidden state, which is the final output of the network. This sophisticated design enables LSTM to effectively capture important information in sequential data and ignore irrelevant parts, making it perform well in tasks such as recognition and prediction.

Another significant feature of LSTM is its excellent gradient flow, which allows the network to avoid the vanishing or exploding gradient problems in traditional RNNs, thus enabling the training of deeper network structures. With the continuous advancement of deep learning technology, LSTM has become one of the core technologies for processing sequential data and plays an important role in various applications.

2.2 *Related works*

The current development of hardware Trojan detection technology mainly focuses on the following aspects: firstly, detection methods based on side channel analysis [10]. Researchers detect possible hardware Trojans by analyzing abnormal features in side channel information such as power consumption, electromagnetic radiation, and signal delay of chips. The advantage of this type of method is that it does not require a detailed understanding of the internal structure of the chip [11, 12], but its detection effect often depends on the severity of abnormal features, which may be powerless against intricately designed Trojans. The second [13] is a detection method based on circuit behavior, which simulates the behavior of chips under different working conditions, compares the differences with expected behavior, and identifies potential hardware Trojans. This method can more accurately locate the location of the Trojan horse, but it faces the problems of high computational complexity and long detection time [14]. The third [15] is to use machine learning and deep learning techniques to automatically extract and identify Trojan features through big data training models. This type of method has high detection accuracy and adaptability, but requires a large amount of annotated data and computational resources to support.

Especially in recent years, significant progress has been made in the field of hardware Trojan detection research. Scholars have proposed various innovative detection methods and theoretical models, such as hardware Trojan detection based on power consumption analysis, Trojan localization technology based on dynamic path analysis, and automated detection systems combined with artificial intelligence [16]. Hasegawa et al. proposed a Trojan detection method based on logical circuit node features. The authors used different machine learning algorithms. Reference [17] used Support Vector Machine (SVM) to learn five logical features, resulting in high misidentification rates; Reference [18] simultaneously uses SVM and neural network to learn five logical features; References [19, 20] used random forests and multi-layer neural networks to learn 11 logical features, respectively. Reference [19] had a large feature redundancy and an average Trojan detection rate of only 68.32%. Reference [20] attempted different combinations of intermediate layers to find the best combination for detection results. On the basis of detecting node classification in reference [19], reference [21] extracted false negative and false positive neighboring node features for the second node classification, which yielded the best

classification results. However, due to only considering RS232 series circuits, the results are not universal. Reference [22] proposes to combine the inter class distance feature and circuit scale feature (number of primitives, AND gates, OR gates) of SCOAP after K-means clustering to form a 4-dimensional node feature, and then use SVM algorithm to analyze the 4-dimensional feature to distinguish between normal nodes and Trojan nodes.

In short, hardware Trojan detection technology [23] has been continuously developing in recent years, but still faces significant challenges [24]. This article proposes an innovative detection method that combines CNN,Attention and LSTM for hardware Trojan detection, aiming to improve the accuracy and efficiency of detection. In the future, research will focus on exploring the potential of hardware Trojan data sharing [25] in order to achieve new breakthroughs in detection technology. Through more extensive data sharing and technologies such as federated learning and differential privacy, researchers can utilize diverse datasets, which will help improve the accuracy and robustness of the model.

3. Methods

In the field of Hardware Trojan (HT) detection, convolutional neural networks (CNN) play a crucial role. The model proposed in this article is based on CNN, which is composed of multiple convolutional layers, each layer using ReLU activation function. This not only introduces nonlinear characteristics, but also enhances the model's ability to express complex patterns. Following each convolutional layer is the pooling layer, which uses MaxPooling to reduce the spatial dimension of the data while removing redundant information and preserving key features, enabling the network to process more compact data representations.

In order to further improve the performance of the model, this paper integrates an attention mechanism module into CNN. The core of this module is to dynamically evaluate the importance of different features and assign them corresponding weights. By using the softmax function to activate the output of the third convolutional layer and performing self multiplication on these activation values, the model can obtain richer attention information. This mechanism enables the network to concentrate resources on processing features that are crucial for detection tasks, significantly improving the accuracy of detection. The output of the attention module is then reshaped and transposed to match the input requirements of the LSTM layer, ensuring the coherence and effectiveness of the information flow.

The Long Short Term Memory Network (LSTM) is another key component of this model, located after the output of the attention module that has been reshaped and transposed. The design of LSTM allows the network to capture long-term dependencies in time series, which is crucial for understanding complex circuit behavior patterns. The LSTM layer effectively avoids the gradient vanishing problem of traditional RNNs when processing long sequences through its unique gating mechanism, enabling the network to learn deeper feature representations. After the LSTM layer, the network further extracts features through a fully connected layer, and finally outputs a probability distribution through a softmax activation function to represent the probability of the presence or absence of a specific hardware Trojan. This end-to-end design not only improves the accuracy of detection, but also ensures the efficiency of the model in processing complex circuit data.

In this study, an Attention mechanism is introduced to enhance the model's capability to detect hardware trojans. This mechanism primarily consists of several key steps. Firstly, Attention weights are generated by calculating the correlation between input features, which reflect the contribution of each feature to the final classification decision. Then, these weights are applied to emphasize significant features and suppress less critical information. Additionally, the Attention mechanism is utilized to model the dependencies between features, allowing the model to capture deeper data patterns. The role of the Attention mechanism in hardware trojan detection is particularly notable: it not only assists the model in precisely locating areas within circuit designs that may conceal trojans but also enhances the model's generalization against various attack techniques. Through this approach, the model presented in this paper can more effectively identify potential threats from a large amount of normal data, thereby improving the accuracy and efficiency of detection.

The specific process of the hardware Trojan detection method based on CNN-Attention-LSTM proposed in this article can be seen as follows and shown in Figure 1.

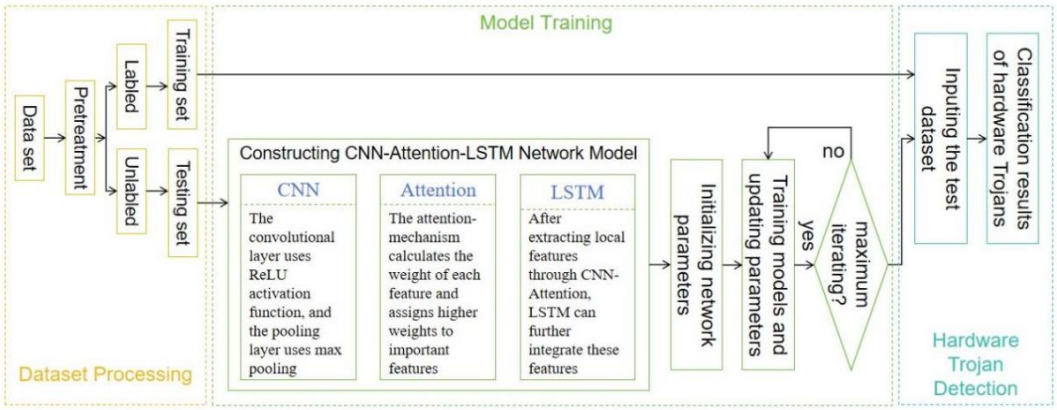


Figure 1. Hardware trojan detection flowchart

Step 1 Data Preprocessing

This paper focuses on identifying two main types of anomalies: combinational logic type Hardware Trojans and sequential logic type Hardware Trojans. Combinational logic type Trojans typically hide at the logic gate level of the circuit and may activate through abnormal signal processing or data flow. Sequential logic type Trojans involve manipulation of the circuit's timing behavior, such as triggering malicious actions by altering clock signals or synchronization mechanisms. To train the model to recognize these complex anomalies, advanced data preprocessing techniques and real Hardware Trojan cases were employed to ensure the model can effectively distinguish these abnormal behaviors from normal operations. The raw data undergoes preprocessing operations such as denoising, filtering, and smoothing to improve data quality. Image data is scaled and standardized to ensure visual consistency and reduce training noise. After data normalization, it is marked based on the presence or absence of trojans, and diversity is increased through data augmentation to solve the problem of data imbalance. Finally, the dataset was divided into training set, validation set, and testing set in a ratio of 7:2:1.

Step 2 Model Training

The model training phase in this study is specifically designed to enhance the model's ability to recognize anomalies caused by Hardware Trojans. The study employs ReLU as the activation function, Adam as the optimizer, and a learning rate of 0.001, which are standard configurations in deep learning due to their effectiveness in accelerating model convergence and enhancing training stability. The model training uses 72 batches and 200 runs, which helps the model to thoroughly learn data features while preventing overfitting, ensuring the model's generalization ability. The combined use of these hyperparameter tunings and settings improves the efficiency and accuracy of the model in Hardware Trojan detection tasks.

The training process is meticulously planned to ensure the model can accurately differentiate between normal circuit operations and anomalies induced by Trojans. By combining batch training methods with cross-validation techniques, the model parameters are finely tuned, effectively preventing overfitting and enhancing model performance. Using a separate validation set, the model underwent detailed evaluation and adjustment until it reached an optimal state. This training process not only ensures high detection accuracy against Hardware Trojans in practical applications but also strengthens the model's ability to recognize unknown attack samples.

Step 3 Hardware Trojan Detection

After the same preprocessing as the training data, the test data is input into the trained model. Optimize classification results using attention layer and Softmax activation function, and set thresholds for classification judgment. By analyzing the classification results through confusion matrix and performance indicators, identify improvement directions, optimize model structure and parameters, and improve detection accuracy.

4. Experiment results and analysis

4.1 *Experimental environment*

This section evaluates the hardware Trojan detection method based on CNN Attention LSTM proposed in this article and designs comparative experiments. The experimental platform used has an operating system of Ubuntu 20.04 LTS, a CPU of AMD Ryzen 7 5800H, a GPU of NVIDIA GeForce RTX 3060, an acceleration library of CUDA11.7, and Pytorch 1.8 for training deep learning models. The system memory is 100 GB, ensuring the needs of large-scale dataset processing and deep learning model training.

4.2 *Experimental dataset*

This experiment selected two groups of netlists from Trust Hub using different standard cell libraries, one of which used the SAED library and the other used the LEDA system 250 nm library. The SAED library consists of 21 netlists, which are embedded with 14 combinational logic hardware Trojans based on 6 host circuits and 7 sequential logic hardware Trojans based on 6 host circuits; The LEDA library has a total of 914 netlists, consisting of 580 combinational logic hardware Trojan embedded netlists based on 8 host circuits and 334 sequential logic hardware Trojan embedded netlists. Due to the relatively small size of the SAED dataset and the fact that most existing work has been implemented on it, this paper will use the SAED dataset for model optimization and result comparison. The LEDA dataset contains a large number of netlists, and this article randomly selects 80 of them to verify the effectiveness of this method.

4.3 *Experimental evaluation indicators*

In this article, the following three indicators are selected as the evaluation criteria for experimental results, namely: Precision, Recall, and F1_score.

The precision measures the proportion of samples that are actually positive in the predicted positive sample, and its formula is

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

The recall rate measures the proportion of correctly identified samples in all positive categories, and its formula is

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

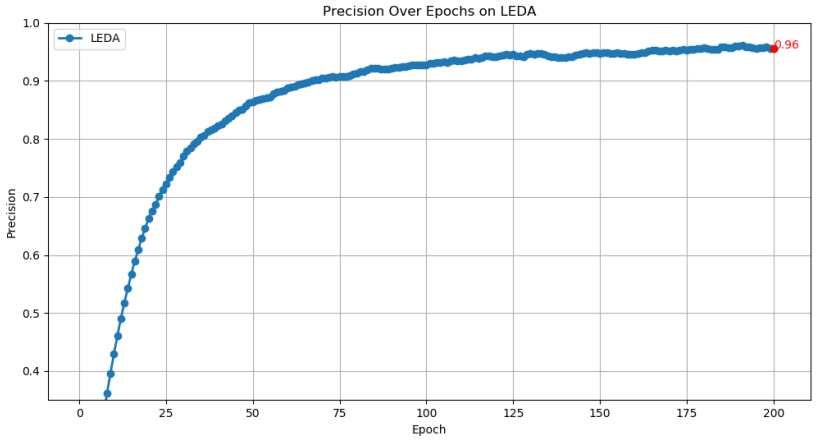
The F1_score is the harmonic mean of Recall and Prec, and its formula is

$$\text{F1\_score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

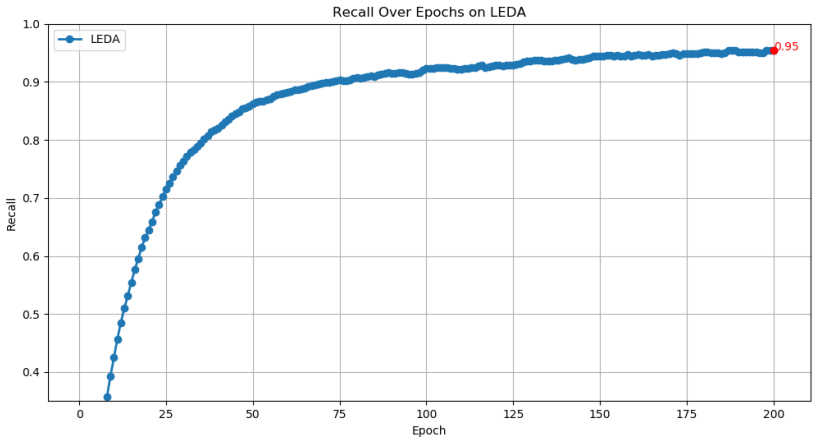
TP, TN, FP and FN represent the number of true cases, true negative cases, false positive cases, and false negative cases, respectively.

4.4 *Experimental results and analysis*

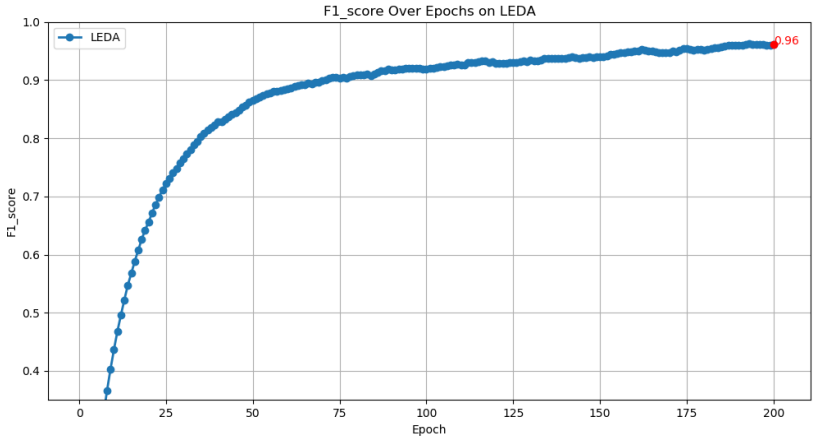
This section conducted experimental verification on the Trojan detection method designed in this article, which was carried out on the LEDA dataset and SAED dataset respectively. The evaluation indicators introduced in the previous section were used to analyze and evaluate the model. The experimental results on the LEDA dataset are shown in Figure 2, and the experimental results on the SAED dataset are shown in Figure 3.



(a) Precision over epochs on LEDA



(b) Recall over epochs on LEDA



(c) F1_score over epochs on LEDA

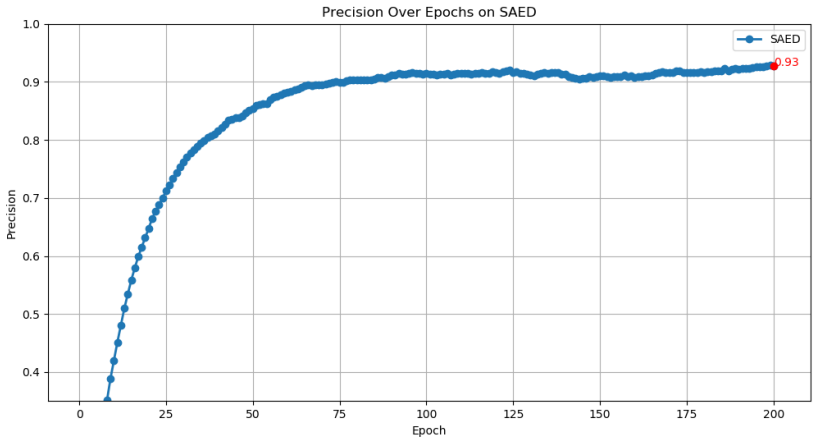
Figure 2. Experimental results on the LEDA dataset

Observing and analyzing the experimental results shown in Figure 2 on the LEDA dataset, the following conclusions can be drawn.

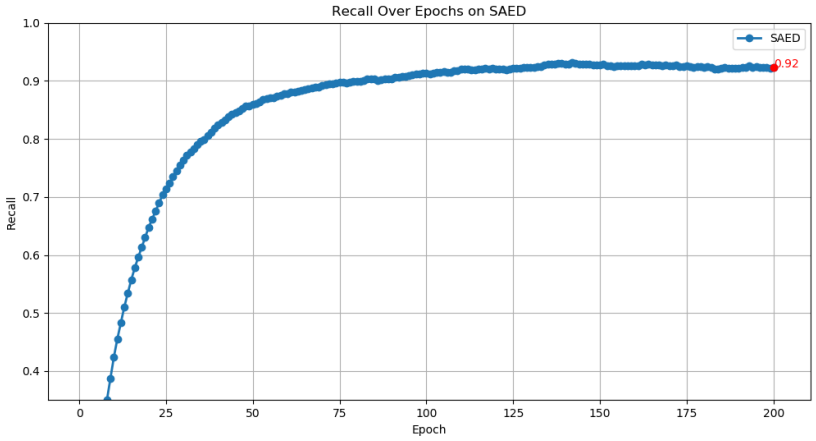
(1) On the LEDA dataset, the model demonstrated high performance with an precision of 96.3%. This result indicates that the model performs well in identifying positive samples and can detect hardware Trojans with extremely high accuracy.

(2) The recall rate of the model is 94.7%, which means that the model can recognize almost all true positive samples, and only a very small number of hardware Trojans may be missed. A high recall rate is crucial for ensuring the efficiency of the model.

(3) The F1 score is 95.5%, which is the harmonic mean of precision and recall, indicating that the model has achieved a good balance between precision and recall. A high F1 score indicates that the model will have high comprehensive performance in practical applications.

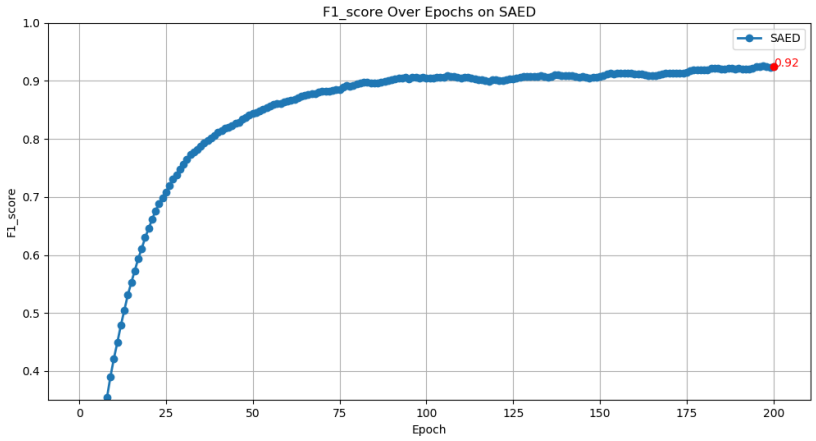


(a) Precision over epochs on SAED



(b) Recall over epochs on SAED

Figure 3. *Cont.*



(c) F1_score over epochs on SAED

Figure 3. Experimental results on the SAED dataset

Observing and analyzing the experimental results shown in Figure 3, the following conclusions can be drawn.

(1) On the SAED dataset, the precision of the model is 93.4%, which is slightly lower than the LEDA dataset, but still shows a high accuracy. This shows that the model has good generalization ability on different datasets.

(2) The recall rate of the model on the SAED dataset reached 91.5%, indicating that although the model may miss a small number of samples when detecting hardware Trojans, it can overall maintain a high detection coverage.

(3) The F1 score is 92.4%, slightly lower than the LEDA dataset on the SAED dataset, but still demonstrating the stability and reliability of the model in detection tasks. The subtle differences in F1 scores may be related to the characteristics of the dataset and sample distribution.

The experimental results on the LEDA and SAED datasets demonstrate the convincing performance of the CNN-Attention-LSTM model proposed in this paper for hardware trojan detection tasks. With an precision rate of 96.3% on the LEDA dataset and 93.4% on the SAED dataset, along with recall rates and F1 scores exceeding 91%, these results not only confirm the model's good generalization capabilities across different circuit design datasets but also indicate the model's efficiency in identifying implanted Hardware Trojans. Obviously, these experimental results can effectively verify the scientificity and feasibility of the model design. The CNN extracts local features, the LSTM handles sequential information, and the attention mechanism further enhances the model's ability to recognize key features. The combination of these designs not only improves detection accuracy but also enables the model to adapt to different circuit designs and Trojan implantation techniques. Therefore, these experimental results provide strong support for the effectiveness of the model design and offer a basis for future research and improvements in hardware security defense strategies.

In the field of HT detection, the CNN-Attention-LSTM model proposed in this paper shares some similarities and distinct differences when compared to the existing models, such as GramsDet and R-HTDetector. The similarity lies in the adoption of deep learning techniques to identify Hardware Trojans, indicating a broad recognition of deep learning's application in this domain. The differences are that our model combines convolutional neural networks, long short-term memory networks, and attention mechanisms, while GramsDet focuses on recurrent neural networks for processing sequential data, and R-HTDetector employs adversarial training to enhance the model's robustness. The uniqueness of our model lies in the integration of the attention mechanism, which helps the model focus on the most relevant features, potentially improving detection accuracy.

Experimental results show that our model achieved an precision rate of 96.3% and an F1 score of 95.5% on the LEDA dataset, demonstrating high accuracy and comprehensive performance, along with a recall rate of 94.7%, indicating its effectiveness in identifying the majority of true hardware trojans. However, the high computational complexity of the model and its dependence on a large amount of annotated data are potential limitations, which may affect the model's

real-time application in resource-constrained environments and pose challenges in practical applications. Future work can explore how to optimize the model to reduce computational costs and study how to decrease reliance on large amounts of annotated data. The comparison of the results between the method proposed in this article and existing Hardware Trojan detection methods is shown in Table 1.

Table 1. Comparison between the results of this article and existing methods

Models	Precision	Recall	F1 Score
GramsDet [26]	32.0%	82.1%	46.1%
R-HTDetector [27]	37.5%	96.8%	56.9%
Random Forest	45.5%	63.6%	77.8%
Model proposed on LEDA	**96.3%**	**94.7%**	**95.5%**

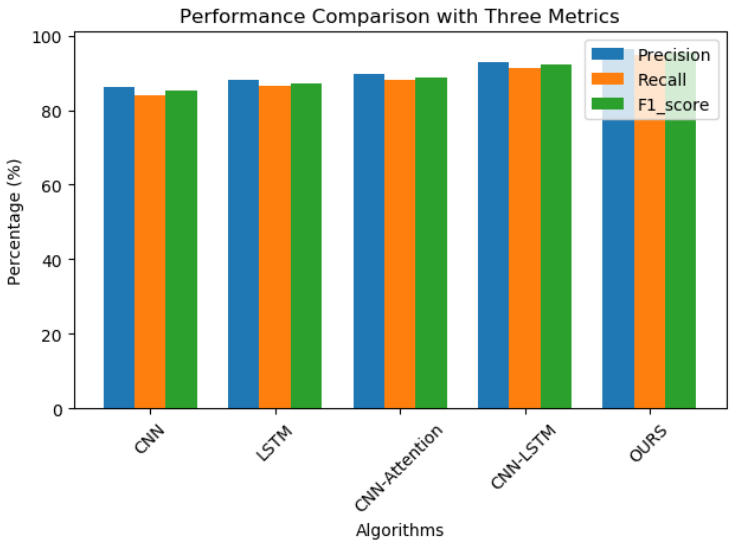
In addition, this section also compares and analyzes the performance of the method designed in this article through comparative experiments and ablation experiments, and evaluates its effectiveness and robustness. The experimental results are shown in the Figure 4.

Observing and analyzing Figure 4, the following conclusions can be drawn:

(1) The ablation experiment shows that the performance of the CNN-Attention-LSTM detection model designed in this paper is superior to the model without any structure, indicating the comprehensive superiority and efficiency of our method in hardware Trojan detection.

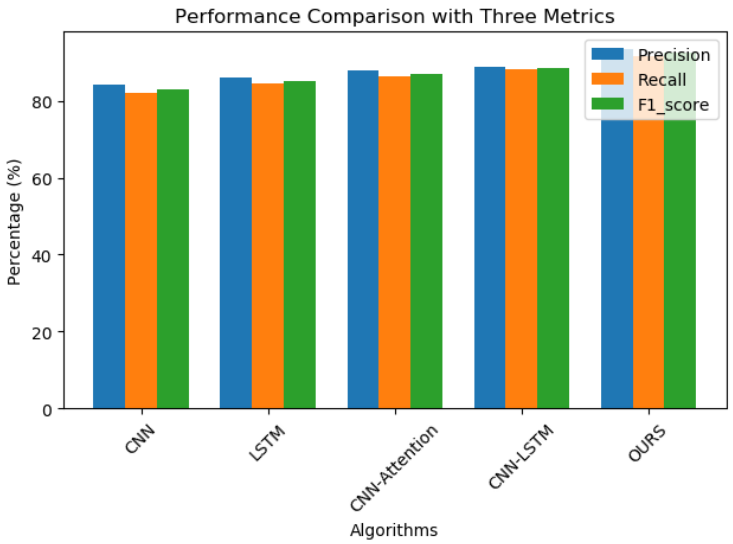
(2) Removing the Attention mechanism can lead to a decrease in model performance, highlighting its core role in identifying key features and improving detection accuracy. Similarly, the lack of LSTM layers can weaken the model's ability to process time series data, thereby affecting overall performance. However, relying solely on CNN or LSTM models for Trojan detection results in significantly lower detection performance compared to CNN-Attention, indicating that the model does not efficiently extract features and capture local spatial features.

(3) These results further confirm the synergistic effect of integrating CNN, LSTM, and Attention mechanisms together. CNN is used for feature extraction, LSTM processes time series data, and Attention mechanism enhances the model's ability to capture key information. This technology fusion strategy enables the model to be more accurate and robust in handling complex and variable hardware Trojan detection tasks.



(a) Results on LEDA

Figure 4. *Cont.*



(b) Results on SAED

Figure 4. Experimental results chart

5. Discussion

5.1 *Complexity analysis*

When analyzing the time and space complexity of the CNN-Attention-LSTM model in depth, this article starts with the three core components of the model: Convolutional Neural Network (CNN), Long Short Term Memory Network (LSTM), and Attention mechanism. Firstly, the CNN part is responsible for extracting features from input data, and its time complexity is mainly affected by convolutional layer operations. For each convolutional layer, the time complexity is approximately $O(n*d*h*w)$, where n represents the number of input data, d represents the depth of the filter, and h and w represent the height and width of the feature map, respectively. The spatial complexity of CNN is also related to the number of parameters, so it is also $O(n*d*h*w)$. Secondly, the LSTM part processes sequence data and captures temporal dependencies, with a time complexity of $O(n*d)$, where n is the sequence length and d is the feature dimension. The space complexity of LSTM is also $O(n*d)$ because it needs to store the state at each time step. Finally, in terms of Attention mechanism, it enhances the model's ability to focus on key information, with a time and space complexity of $O(n*d)$, where n is the number of attention heads. By integrating these parts, the time and space complexity of the entire model can be determined, which are closely related to the size and dimension of the input data. Complexity analysis helps to understand the efficiency and resource requirements of models when processing large-scale datasets, providing important references for future model optimization and applications.

5.2 *Adversarial vulnerabilities analysis*

While the proposed CNN-Attention-LSTM model has demonstrated effectiveness in detecting Hardware Trojans, it is crucial to acknowledge its vulnerability to adversarial attacks. Adversarial samples, which are carefully crafted inputs designed to mislead deep learning models, pose a significant threat to the reliability of our model.

To mitigate the impact of adversarial samples on the proposed CNN-Attention-LSTM model, this study employs several specific defensive strategies. Firstly, adversarial training [27] is conducted by incorporating adversarial samples into the model training process, enabling the model to maintain higher robustness when faced with these carefully designed perturbations. This approach has been proven to enhance the model's generalization ability and reduce the success rate

of adversarial attacks. Secondly, defensive distillation is utilized by transferring the knowledge from a model that has undergone adversarial training to another model, thereby strengthening the model's resistance to adversarial attacks [28]. In addition, input preprocessing is an effective way to improve the model's robustness; operations such as normalizing input data or applying filters can reduce the impact of adversarial perturbations. Lastly, certified defense mechanisms are developed to provide safety certifications, ensuring the model remains correctly classified under a certain range of perturbations. The implementation of these strategies significantly enhances the model's security and reliability in practical applications.

6. Conclusions

With the increasing complexity of hardware Trojan threats, traditional detection methods are no longer able to meet current security needs. The hardware Trojan detection method based on CNN Attention LSTM proposed in this article effectively improves the intelligence level of detection through deep learning technology. The experiment verified the superior performance of this method on different datasets, demonstrating its potential application in the field of hardware security. Future work will further explore and optimize model structures to adapt to a wider range of application scenarios, and continuously improve the accuracy and robustness of detection techniques. We hope that this research can bring new perspectives to the field of hardware security and provide strong technical support for practical hardware Trojan detection.

Acknowledgements

This work is partially supported by the National Social Science Foundation of China (No. 21BTO079), Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing project (No. GJJ-23).

Conflict of interest

There is no conflict of interest for this study.

References

[1] C. Dong, Y. Yao, Y. Xu, X. Liu, Y. Wang, H. Zhang, et al., "A Cost-Driven Method for Deep-Learning-Based Hardware Trojan Detection," *Sensors*, vol. 23, no. 12, p. 5503, 2023.

[2] W. Tang, J. Su, J. He, and Y. Gao, "A Deep Learning Method Based on the Attention Mechanism for Hardware Trojan Detection," *Electronics*, vol. 11, no. 15, p. 2400, 2022.

[3] R. Sharma, G. K. Sharma, and M. Pattanaik, "A Few Shot Learning based Approach for Hardware Trojan Detection using Deep Siamese CNN," in *Proc. 2021 34th Int. Conf. VLSI Design 2021 20th Int. Conf. Embedded Syst. (VLSID)*, Guwahati, India, Feb. 20–24, 2021, pp. 163–168.

[4] J. Wang, G. Zhai, H. Gao, L. Xu, X. Li, Z. Li, et al., "A Hardware Trojan Detection and Diagnosis Method for Gate-Level Netlists Based on Machine Learning and Graph Theory," *Electronics*, vol. 13, no. 1, p. 59, 2023.

[5] S. Chen, T. Wang, Z. Huang, X. Hou, "Detection method of Golden Chip-Free Hardware Trojan based on the combination of ResNeXt structure and attention mechanism," *Comput. Secur.*, vol. 134, p, 103428, 2023.

[6] C. Ravichandran, T. J. Nagalakshmi, P. S. Bharathi, C. Sivakumaran, "Technique for detecting hardware-based Trojans using a convolutional neural network," *Int. J. Inf. Comput. Secur.*, vol. 23, no. 3, pp. 338–347, 2024.

[7] P. Ma, Z. Wang, and Y. Wang, "A Pre-Silicon Detection Based on Deep Learning Model for Hardware Trojans," *J. Circ. Syst. Comput.*, vol. 33, no. 8, p. 2450144, 2023.

[8] W. Tang, J. Su, J. He, Y. Gao, "A Deep Learning Method Based on the Attention Mechanism for Hardware Trojan Detection," *Electronics*, vol. 11, no. 15, pp. 2400–2400, 2022.

[9] C. Dong, Y. Yao, Y. Xu, X. Liu, Y. Wang, H. Zhang, et al., "A Cost-Driven Method for Deep-Learning-Based Hardware Trojan Detection," *Sensors*, vol. 23, no. 12, p. 5503, 2023.

[10] S. Sankaran, V. S. Mohan, and A. Purushothaman, "Deep Learning Based Approach for Hardware Trojan Detection," in *Proc. 2021 IEEE Int. Symp. Smart Electron. Syst. (iSES)*, Jaipur, India, 2021, pp. 177–182.

[11] H. Kang and Y. Yange, "An Enhanced Detection Method of PCB Defect Based on D-DenseNet (PCBDD-DDNet)," *Electronics*, vol. 12, no. 23, pp. 1–22, 2023. https://doi.org/10.3390/electronics12234737.

[12] Y. Yang and H. Kang, "An Enhanced Detection Method of PCB Defect Based on Improved YOLOv7," *Electronics*, vol. 12, no. 9, pp. 1–18, 2023. https://doi.org/10.3390/electronics12092120.

[13] S. Yu, C. Gu, W. Liu, and M. O'Neill, "Deep Learning-Based Hardware Trojan Detection With Block-Based Netlist Information Extraction," *IEEE Trans. Emerg. Top. Comput.*, vol. 10, no. 4, pp. 1837–1853, Oct.–Dec. 2022.

[14] Z. Pan and P. Mishra, "TD-Zero: Automatic Golden-Free Hardware Trojan Detection Using Zero-Shot Learning," *IEEE Trans. Comput. Aided Des. Integr. Circ. Syst.*, vol. 43, no. 7, pp. 1998–2011, July 2024.

[15] V. S. Rathor, D. Singh, S. Singh, M. Sajwan, "Multi-Objective Optimization Based Test Pattern Generation for Hardware Trojan Detection," *J. Electron. Test.*, vol. 39, pp. 371–385, 2023.

[16] K. Wang, H. Zheng, and A. Louri, "TSA-NoC: Learning-Based Threat Detection and Mitigation for Secure Network-on-Chip Architecture," *IEEE Micro.*, vol. 40, no. 5, pp. 56–63, Sept.–Oct. 2020.

[17] K. Hasegawa, M. Oya, M. Yanagisawa, N. Togawa, "Hardware Trojans classification for gate-level netlists based on machine learning," in *Proc. 2016 IEEE 22nd Int. Symp. On-Line Testing Robust Syst. Des. (IOLTS)*, Sant Feliu de Guixols, Spain, Jul. 4–6, 2016, pp. 203–206. https://doi.org/10.1109/IOLTS.2016.7604700.

[18] K. Hasegawa, M. Yanagisawa, and N. Togawa, "A hardware-Trojan classification method using machine learning at gate-level netlists based on Trojan features," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E100.A, no. 7, pp. 1427–1438, 2017. https://doi.org/10.1587/transfun.E100.A.1427.

[19] K. Hasegawa, M. Yanagisawa, and N. Togawa, "Trojan-feature extraction at gate-level netlists and its application to hardware-Trojan detection using random forest classifier," in *Proc. 2017 IEEE Int. Symp. Circ. Syst. (ISCAS)*, Baltimore, MD, USA, May 28–31, 2017, pp. 1–4. https://doi.org/10.1109/ISCAS.2017.8050827.

[20] K. Hasegawa, M. Yanagisawa, and N. Togawa, "Hardware Trojans classification for gate-level netlists using multi-layer neural networks," in *Proc. 2017 IEEE 23rd Int. Symp. On-Line Testing Robust Syst. Des. (IOLTS)*, Thessaloniki, Greece, Jul. 3–5, 2017, pp. 227–232. https://doi.org/10.1109/IOLTS.2017.8046227.

[21] K. Hasegawa, M. Yanagisawa, and N. Togawa, "A hardware-Trojan classification method utilizing boundary net structures," in *Proc. 2018 IEEE Int. Conf. Consumer Electron.*, Las Vegas, NV, USA, Jan. 12–14, 2018, pp. 1–4. https://doi.org/10.1109/ICCE.2018.8326247.

[22] X. Xie, Y. Sun, H. Chen, Y. Ding, "Hardware Trojans classification based on controllability and observability in gate-level netlist," *IEICE Electron. Express*, vol. 14, no. 18, p. 20170682, 2017. https://doi.org/10.1587/elex.14.20170682.

[23] Z. Huang, Q. Wang, Y. Chen, X. Jiang, "A survey on machine learning against hardware Trojan attacks: Recent advances and challenges," *IEEE Access*, vol. 8, pp. 10796–10826, 2020. https://doi.org/10.1109/ACCESS.2020.2965016.

[24] M. Elshamy, G. Di Natale, A. Sayed, A. Pavlidis, M. M. Louërat, H. Aboushady, et al., "Digital-to-analog hardware Trojan attacks," *IEEE Trans. Circ. Syst. I Regular Papers*, vol. 69, no. 2, pp. 573–586, 2022. https://doi.org/10.1109/TCSI.2021.3116806.

[25] H. Kaiyan, J. Yuanrui, and Z. Shuxuan, "Enhanced Privacy Preserving for Social Networks Relational Data Based on Personalized Differential Privacy," *Chinese J. Electron.*, vol. 31, no. 4, pp. 741–751, 2022.

[26] R. Lu, H. Shen, Y. Su, H. Li, X. Li, "GramsDet: Hardware Trojan detection based on recurrent neural network," in *Proc. 2019 IEEE 28th Asian Test Symp. (ATS)*, Kolkata, India, Dec. 10–13, 2019. https://doi.org/10.1109/ATS47505.2019.00021.

[27] K. Hasegawa, S. Hidano, K. Nozawa, S. Kiyomoto, N. Togawa, "R-HTDetector: Robust hardware-Trojan detection based on adversarial training," *IEEE Trans. Comput.*, vol. 72, no. 2, pp. 333–345, 2022.

[28] J. Lee, T. Kim, S. Bang, S. Oh, and H. Kwon, "Evasion Attacks on Deep Learning-Based Helicopter Recognition Systems," *J. Sens.*, vol. 2024, p. 1124598, 2024. https://doi.org/10.1155/2024/1124598.