engineering proceedings

MDPI

Proceeding Paper

PUF Modeling Attacks Using Deep Learning and Machine Learning Algorithms †

Nelakudite Saadvikaa *, Kenneth Jonathan Saketi, Akshitha Gopishetti, Bhavitha Degala and Kiran Kumar Anumandla

Department of Artificial Intelligence and Machine Learning, School of Engineering, Malla Reddy University, Maisammaguda, Dulapally, Hyderabad 500100, Telangana, India; 2011cs020417@mallareddyuniversity.ac.in (K.J.S.); 2011cs020418@mallareddyuniversity.ac.in (A.G.); 2011cs020419@mallareddyuniversity.ac.in (B.D.); drkirankumar@mallareddyuniversity.ac.in (K.K.A.)

* Correspondence: 2011cs020420@mallareddyuniversity.ac.in

† Presented at the 4th International Electronic Conference on Applied Sciences, 27 October–10 November 2023; Available online: https://asec2023.sciforum.net/.

Abstract: The rapid advancement of technology has led to the pervasive presence of electronic devices in our lives, enabling convenience and connectivity. Cryptography offers solutions, but vulnerabilities persist due to physical attacks like malware. This led to the emergence of Physical Unclonable Functions (PUFs). PUFs leverage the inherent disorder in physical systems to generate unique responses to challenges. Strong PUFs, susceptible to modeling attacks, can be predicted by malicious parties using machine learning and algebraic techniques. Weak PUFs, with minimal challenges, face similar threats if built upon strong PUFs. Despite some weaknesses, PUFs serve as security components in various protocols. Modeling attacks' success depends on suitable models and machine learning algorithms. Logistic Regression and Random Forest Classifier are potent in this context. Deep learning techniques, including Convolutional Neural Networks (CNNs) and Artificial Neural Networks (ANNs), exhibit promise, particularly in one-dimensional data scenarios. Experimental results indicate CNN's superiority, achieving precision, recall, and accuracy exceeding 90%, demonstrating its effectiveness in breaking PUF security. This signifies the potential of deep learning techniques in breaking PUF security. In conclusion, this paper highlights the urgent need for improved security measures in the face of evolving technology. It proposes the utilization of deep learning techniques, particularly CNNs, to strengthen the security of PUFs against modeling attacks. The presented findings underscore the critical importance of reevaluating PUF security protocols in the era of ever-advancing technological threats.

Keywords: PUFs; security; cyber security; challenge–response data; deep learning; modeling attacks

Citation: Saadvikaa, N.; Saketi, K.J.; Gopishetti, A.; Degala, B.; Anumandla, K.K. PUF Modeling Attacks Using Deep Learning and Machine Learning Algorithms. *Eng. Proc.* **2023**, *56*, 187. https://doi.org/10.3390/ASEC2023-15948

Academic Editor: Nunzio Cennamo

Published: 9 November 2023

1. Introduction

The advent of technology has been a great blessing, but technology and, by proxy, electronic devices now peer into every little aspect of our lives. This has made it an easy target for enemies who know how to manipulate and steal data from these devices. In cryptography, there exists a concept known as a secret binary key, which offers some solutions to the problems that we face, but unfortunately, in the golden age of technology, physical attacks against the system, such as malware and viruses, can lead to the exposure of the secret binary key and eventually a full security break.

This problem was one of the reasons that led to the creation of Physical Unclonable Functions, also known in shorthand as PUFs. Simply put, when certain challenges C_i are posed to a partially disordered physical system, it yields favorable responses RC_i. This is the definition and the working of a PUF. Unlike conventional digital systems, PUF responses are dependent on the miniscule disorder present in the PUF. This disorder is unique to each PUF and, due to its very small scale, effectively unclonable, which is where

its name comes from. The PUF cannot even be reproduced or duplicated by the person who created it due to its unique id of disorder. Certain error correction methods, like fuzzy extractors [1], are applied to the PUF responses as they tend to be noisy. After the noise is cleared, stable outputs $R'C_i$ are obtained. Simply put, A PUF P can be classified as a singular function F_p that links challenges C_i to responses $R'C_i$ [2].

2. Modeling Attacks for PUFS and How They Are Applicable

Let us assume that a malicious party, for example, Jane, has managed to obtain all the challenge–response pairs of a certain PUF. Now, what Jane tries to achieve is to calculate a model based on the CRP (Challenge Response Pairs) of the PUF, which essentially means that she tries to create an algorithm with high accuracy that can predict the responses to a given challenge data using the CRP data [3] as test and training data. So again, the task is to find out if it is a Weak or a Strong PUF.

(1) Strong PUFs: Strong PUFs mean the class of PUFs for which the modeling attacks are most applicable and were designed. This is due to the fact that they have no way to protect themselves against the attacks from malicious parties like Jane, who can elicit responses from challenges at will from the PUF. So, for Jane to read out and collect challenge–response pairs in a large amount, it is made easy with even a short amount of time for physical access to the PUF. As soon as the model for prediction for a certain PUF has been derived by Jane, it means that the security functionalities of a PUF are no longer in place;

(2) Weak PUFs: Weak PUFs are those PUFs that have very few and constant challenges that are put in place by the author of that PUF. Sometimes, they might have only one challenge in extreme cases. The common consensus is that their responses or a response (in case of a single challenge) are placed inside the hardware that contains the PUF. These PUFs, which are highly resistant to modeling attacks, are not readily accessible to external third parties.

The modeling attacks rarely apply to the Weak PUFs and only under special circumstances, such as if a Strong PUF in a hardware system is used to implement the Weak PUF [3].

In conclusion, this should not lead to the assumption that since most of the Weak PUFs are not applicable to modeling attacks, they are more secure than other PUFs since there are other ways to break the PUF.

3. PUF Modeling Process and Challenges

Now we discuss how to actually best model the PUFs, the process we go through to model them, and the different obstacles faced throughout this process. The modeling of PUFs goes through two phases. The first phase is creating an internal model of the PUF, which consists of describing the behavior of the PUF, especially the challenge–response behavior. The second phase consists of a description of values of different important parameters of the PUF that cannot be replicated [4].

During the second phase, the model of the PUF created in the first phase is used together with an ML algorithm with the sole purpose of creating a model to predict PUF behavior. These first data are known as training data, as they are used to train the model to predict. Now, we use another set of CRP data to test the model against and see its precision and accuracy. This is known as the test data.

The first and most noticeable problem is to find a suitable model for the PUF. Creating a model of a PUF is very easy if we know its parameters and mechanisms, but what is time-consuming and hard is to find the right model for the PUF, which is the most efficient. To do this, we have to experiment and play around with the different parameters and mechanisms and change them or modify them to achieve the best efficiency of the PUF.

Another challenge we have faced is finding an ML algorithm that can effectively and efficiently operate on the specific PUF while maintaining high accuracy in prediction. Since

scaling PUFs to a large amount is not possible due to restrictions on the cost as well as stability of the PUF, even gradual exponential growth is acceptable in some circumstances.

Both these obstacles that we face are correlated, i.e., the problem of finding both a PUF model as well as an ML algorithm best suited for the PUF. Many problems arise because the powerful machine learning algorithms have many demands and requirements from the PUF. Consequently, it is crucial to optimize both the model and the algorithm in unison, enabling them to function seamlessly as a well-coordinated system.

4. Modeling Attacks on PUFS

Let us see the different machine learning techniques as well as deep learning techniques that are applied to these PUFs.

A. Machine Learning Techniques

Many different ML methods have been applied to PUFs in the past [3]. These range from Logistic Regression (LR) all the way to Support Vector Machines (SVM) and, for a brief amount of time, even Sequence Learning and Neural Nets. We have outlined two approaches to modeling for the ML algorithm [5]:

(1) Logistic Regression: Logistic Regression is a widely recognized and extensively researched machine learning technique [6]. When it is applied in the context of PUFs with single outputs because (LR has only a single output), then each challenge $C = a_1 \dots a_n$ is mapped to a probability function $p(C, r \mid w^{\rightarrow})$ such that an output $r \in \{0, 1\}$ is generated.

Logistic Regression [6] has been determined as the most optimal in the earlier literature concerning machine learning applications of PUFs. RProp greatly increases convergence speed and helps keep the algorithm stable.

Logistic Regression does not require the PUF models to be linearly separable as so many powerful machine learning algorithms, such as Support Vector Machines, demand, but ask that the PUF model be differentiable;

(2) Random forest classifier algorithm: Random Forest Classifier is a supervised machine learning algorithm. It is based on the ensemble learning method. It uses different 'decision trees' from various subsets of data and compiles an average of all those trees to give a prediction of the output. The greater the number of trees present, the higher the chance of accuracy.

How we use random forest classifier in regards to modeling attacks to break PUFs is that since our dataset is very large, random forest divides the whole dataset into various smaller subsets of data, trains the data, and takes an average of all the predictions from the aggregate of those outputs and determines the accuracy of the algorithm;

B. Deep Learning Techniques

Deep learning techniques have been tried, although not tested much in the present, since they are fairly new when compared to machine learning algorithms. Deep learning techniques also require more computation and more prerequisites for working in a modeling attack. Let us see one such deep learning algorithm that has been observed by us to give the best accuracy of all the algorithms since used:

(1) Convolutional Neural Network (CNN) Algorithm: Convolutional Neural Network Algorithms are normally used for image and video-based classification, but for the purpose of this paper, we will be using the One-Dimensional (1D) Convolutional Neural Network (CNN) Algorithms [7].

Here is how we use this CNN Algorithm in modeling attacks to break the PUFs.

We divide the model into four layers. First, there are two layers of 1D CNN. Then, a dropout layer for regularization, and finally, a pooling layer. We divided the CNN layers into groups of two as it gives the model a better chance of learning the features from the vast amount of data. The dropout layer is present to slow down the learning process since CNNs were originally intended for 3-D-based datasets. Including the dropout layer gives

us better results when it comes to the accuracy of the predictions. And finally, the pooling layer restricts the features to a smaller size, specifically, a quarter of the original size (1/4), so that it is stripped down to its most essential and important elements;

(2) Artificial Neural Network (ANN) Algorithm: An Artificial Neural Network is a collection of interconnected artificial neurons (also known as nodes or units) organized in layers. Each neuron receives input signals, processes them through an activation function, and produces an output signal. These connections between neurons have associated weights that are adjusted during the learning process, allowing for the network to adapt and learn from the provided data.

Artificial Neural Network Algorithm tries to replicate the working of the biological neural networks that happen in the human brain. The model is divided into nodes or layers which are interconnected, much like the neurons in the brain.

5. Results

As shown in Figure 1, we have seen the machine learning algorithm Logistic Regression. We see that the challenge size used is 64 bits, and the precision, recall, and accuracy of the algorithm are all identical at 74%.

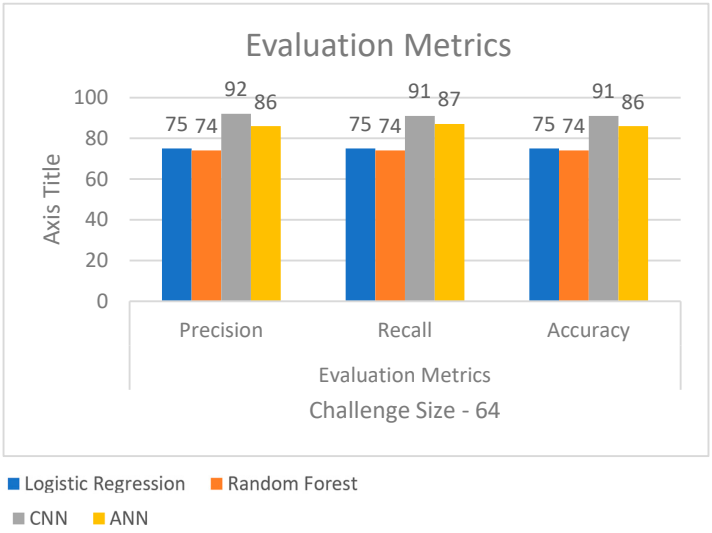

Figure 1. It shows the Evaluation metrics for the classifiers.

Next, we see the random forest classifier, which uses 64 bits as the challenge size. It does not fare as well as expected, with the precision, recall, and accuracy all turning up at 73%. And from Figure 2, we are getting accuracy of 86% for ANN model.

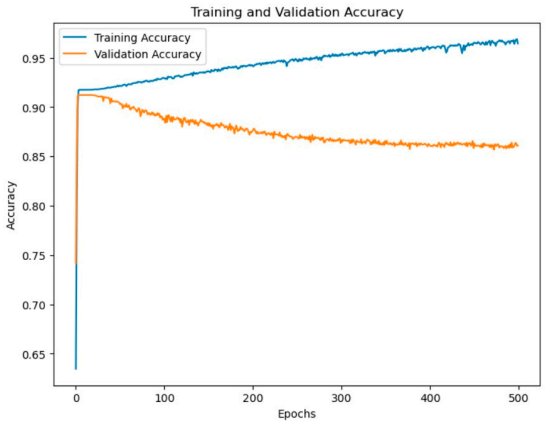

Figure 2. Accuracy Rate of ANN.

From Figure 3, we can see that the training and testing (validation) accuracy rates remain the same relative to each other, and so with the increase in training accuracy, the validation accuracy rate increases as well, resulting in the highest accuracy overall.

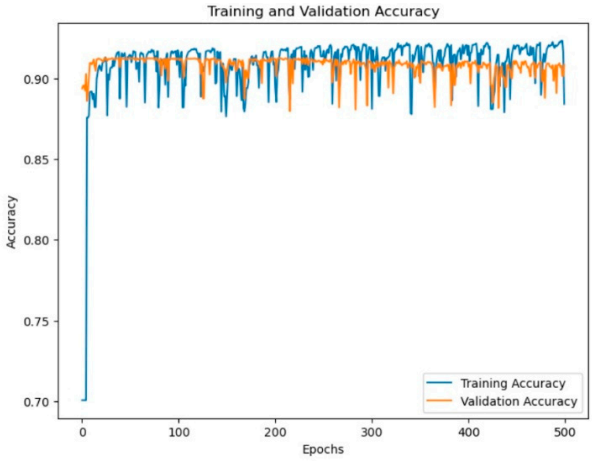

Figure 3. Accuracy Rate of CNN.

However, as we notice, the deep learning technique, Convolutional Neural Networks, has managed to outdo the machine learning techniques used, which are Logistic Regression and Random Forest Classification. It has given the best results out of all the algorithms used, turning up with 91% precision of the model, 90% recall, and 91% accuracy of prediction. From these results, we can come to the conclusion that the most optimal algorithm for the modeling attacks on PUFs is the 1-D Convolutional Neural Network Algorithm.

6. Conclusions and Future Scope

A. Conclusion

From the existing modeling attacks, we are able to come to two direct conclusions, although they are not exactly unbiased. (i) The first is that all Strong PUFs are not secure, and (ii) Secondly, deep learning methods such as Convolutional Neural Networks and Artificial Neural Networks are far more effective in breaking the security of the PUFs.

(i) Coming to the first one, presently, the attacks that are used to break the PUFs are successful in their mission and are able to break several delay-based PUF models, but we look toward the future when PUF designers might be able to fight back on the security of the PUF.

Coming to the second view (ii), we have observed through the results from Figure 1 that the CNN Algorithm has resulted in much higher prediction results when it came to precision, accuracy, and recall.

B. Future Scope

Looking toward the future, we can expect some great showdowns between the coders and the hackers who give it their all to show the other one out, especially in the field of Strong PUFs. First attempts at this process have already begun, albeit with little to no testing against ML techniques.

Those people who build the PUFs may be hard-pressed to find new and innovative ideas to discover the most suitable model for the PUFs that offer the most security and complexity against attacks.

For the ones who are dedicated to breaking the PUFs, an option is to take the attacks mentioned in this paper and improve, enhance, and modify them to make them the most optimized attacks and also try out new and upcoming ML techniques for modeling attacks.

Author Contributions: Conceptualization, K.K.A.; methodology, K.K.A.; software, N.S.; validation, N.S., A.G.; formal analysis, N.S.; investigation, B.D.; resources, A.G.; data curation, K.J.S.; writing—original draft preparation, B.D.; writing—review and editing, B.D.; visualization, K.J.S.; supervision, K.K.A.; project administration, K.K.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: https://github.com/Praneshss/Modeling_of_APUF_Compositions/blob/master/64bit/HalfMillion/4-XOR/4-xorpuf.csv.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dodis, Y.; Ostrovsky, R.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.* **2008**, *38*, 97–139. [CrossRef]
2. Ruhrmair, U.; Solter, J. PUF modeling attacks: An introduction and overview. In Proceedings of the 2014 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, 24–28 March 2014.
3. Gassend, B.L.P. Physical Random Functions. Master's Thesis, MIT, Cambridge, MA, USA, 2003.
4. Brzuska, C.; Fischlin, M.; Schröder, H.; Katzenbeisser, S. Physical Un-clonable Functions in the Universal Composition Framework. In Proceedings of the CRYPTO 2011, Santa Barbara, CA, USA, 14–18 August 2011.
5. Delvaux, J.; Verbauwhede, I. Side channel modeling attacks on 65 nm arbiter PUFs exploiting CMOS device noise. In Proceedings of the 2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), Austin, TX, USA, 2–3 June 2013.
6. Bishop, C.M.; Nasrabadi, N.M. *Pattern Recognition and Machine Learning*; Springer: New York, NY, USA, 2006.
7. Csaba, G.; Ju, X.; Ma, Z.; Chen, Q.; Porod, W.; Schmidhuber, J.; Schlichtmann, U.; Lugli, P.; Rührmair, U. Application of mismatched cellular nonlinear networks for physical cryptography. In Proceedings of the 2010 12th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA 2010), Berkeley, CA, USA, 3–5 February 2010.