Machine Learning and its Application to Side-Channel Analysis

Amal Almuhawwis
dept. Computer Science
George Mason University
Fairfax,VA,USA
aalmuhaw@gmu.edu

Aabha Bothera
dept. Computer Science
George Mason University
Fairfax,VA,USA
abothera@gmu.edu

Chris Limson
dept. Computer Science
George Mason University
Fairfax,VA,USA
climson@gmu.edu

***Abstract*—Machine Learning (ML) represents powerful techniques in the field of Side-Channel Analysis (SCA). This paper justifies the integration of ML techniques within SCA based on an extensive review of the literature and scientific papers. It presents a brief review of various ML techniques used in the context of SCA. The paper also contrasts traditional versus ML-based approaches in SCA, highlighting the effectiveness of ML in most cases. It focuses on techniques like Support Vector Machine (SVM), Random Forest, and Naive Bayes, along with Deep Learning (DL) approaches such as Multi-Layer Perceptron (MLP) and Convolutional Neural Network (CNN). Moreover, the use of ML is emphasized by demonstrating it practically on Advanced Encryption Standard (AES) implementation. In addition, we mount an investigation on other ML and DL applications for SCA further than testing or designing new attacks. Finally, the best practices and guidelines to enhance the effectiveness of ML and DL in SCA context were illustrated based on the recommendations from industry experts and academic researchers.**

Keywords— Side-Channel Analysis, Machine Learning, Deep Learning, Advanced Encryption Standard.

I. Introduction

Side-Channel Analysis (SCA) has become one of the feasible ways of attacking systems. This attack might target any device that contains a secret by utilizing its physical information. Therefore, it is vital to protect devices with a strong countermeasure to SCA. Ensuring security means not only focusing on software but also the physical hardware. SCA emphasizes the importance of securing the complete system. As ML is a promising domain nowadays, it provides powerful techniques that are effective for many SCA applications when they properly adopted. This paper aims to investigate the use of ML, specifically Deep Learning (DL) applications in SCA. Indeed, one of the main motivations for the use of ML is its prominent results in different fields. Further, an extensive review of the literature and scientific papers will be conducted in order to compare and classify different ML techniques according to their performance within SCA. In addition, the difference between the traditional analysis techniques and ML techniques for SCA will be illustrated.

This study aims to provide a summary of several ML techniques that have been used by SCA and are currently being investigated. In addition, it is not limited only to the traditional ML methods but also includes newer DL approaches. This paper focuses on Side-Channel Analysis and then extends to represent ML applications to SCA including Support Vector Machine (SVM), Random Forests, and Naive Bayes, as well as DL applications including Multi-Layer Perceptron (MLP) and Convolutional Neural Networks (CNN).

The remaining of this paper is organized as follows. In Section II, where an overview of Side-Channel Analysis is presented. Section III introduces traditional ML techniques applied to SCA. Section IV elaborates on Deep Learning applied to SCA. Section V demonstrates an experiment performed to illustrate the principles of DL's application to SCA. Section VI presents other applications of SCA further than testing or designing new attacks. Section VII illustrates guidelines to standardize the presentation of ML techniques in SCA research, making it easier for further research of ML in SCA domain

II. Side-Channel Analysis

Side-channel analysis is an attack that involves exploiting information leaked by a cryptographic device during its operation to reveal secret information, such as cryptographic keys. This secret information on sensitive variables is usually discovered thanks to unintended physical leakages through channels such as electromagnetic radiation, power consumption, or timing variations. This paper focuses on power analysis where an attacker can exploit cryptographic devices based on leakage of power consumption traces. In this regard, the following lines justify two types of attacks based on prior knowledge of the targeted device. [1,6]

A. Profiling SCA

In profiling attacks, an attacker has prior knowledge about the target device by having an additional programmable device that is highly identical to the target device and it is called a profiling device. An attacker can use this profiling device to exploit leakage measurements in order to extract the secret key of the target device. The most common applications of profiling attacks are template attacks and stochastic models.

B. Non-Profiling SCA

Non-profiling attacks assume that an attacker has only a limited number of leakage measurements (side-channel traces) for a fixed unknown key value from the target device, and then uses some SCA methods in order to infer sensitive information such as cryptographic keys [1,10]. The most common techniques used for Non-Profiling SCA are Simple power analysis (SPA) and other statistical analysis methods such as Differential power analysis (DPA), and Correlation Power Analysis (CPA).

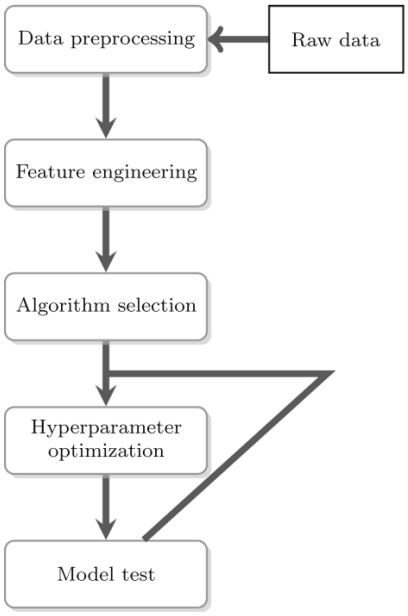


Figure 1: Standard Process of ML

III. Traditional Machine Learning Techniques for SCA

Machine learning models use mathematical functions to represent a powerful set of techniques for predicting and classifying input data based on known properties learned from the training data. Classification of the input data is a major application of ML; therefore, ML techniques can be effectively adapted to SCA. In this section, the standard of the ML process in the context of SCA will be covered, followed by a review of several ML algorithms applied to SCA.

A. Standard Process of ML

In a usual ML application, the model goes under five main phases as shown in Fig. 1, this section demonstrates these phases in the context of SCA.

1) Preprocessing: The published papers can be classified into three categories [6] regarding data preprocessing in SCA applications, it is either used to:

- Transform the data set into another representation by using wavelet transform or normalization.
- Reduce the noise in the traces by lowering the dimension or by computing average traces.
- Verify that the measurement points are aligned properly.

However, many papers don't even mention preprocessing steps; thus, the authors in [6] assume the preprocessing was either entirely disregarded or considered unnecessary for those approaches. On the other hand, there are a significant number of papers that found the preprocessing step affects the effectiveness of ML models. In general, it is difficult to recommend how to apply the preprocessing step in the context of SCA, but choosing the appropriate preprocessing method should depend on the used classifier.

2) Feature Engineering: In the SCA domain, feature selection refers to the process of choosing a subset of sample points from the leakage traces corresponding to manipulations of the target variable such as a cryptographic key or intermediate value. Numerous methods exist within the classical SCA community for identifying those points, but the most prominent one is the Pearson correlation which is similar to CPA. In this context, Pearson correlation can be used to select a subset of information-rich features (side-channel measurements) based on the correlation between the target output of the S-box of the AES and the power consumption measured at a specific time point. Some works introduced other techniques applied to ML-based attacks such as the Sum Of Squared Pairwise Differences (SOSD) and Sum Of Squared Pairwise T-Differences (SOST). It has been demonstrated that SOST outperforms SOSD on noisy datasets and that applying the metric increases the classification model's overall efficiency. Principal component analysis (PCA) is also another commonly applied technique. In the reviewed papers, PCA served for both trace preprocessing and feature selection. In general, most of the previous contributions find that the selection of proper features is essential to the success of the attack [1,6,7].

3) Algorithm Selection: The process of selecting an appropriate ML for an algorithm is typically a difficult task since it depends on different factors. The authors in [6] have examined many published works and given many observations. They found that the choice between supervised and unsupervised techniques is affected by the class of attacked cipher implementation as well as by the attacking model. To illustrate further, in profiling attacks where the adversary possesses labeled trace data, supervised classifiers are better, Whereas, in the case of non-profiling attacks, unsupervised techniques can still be used.

Overall, SVMs and MLPs are the most widely used ML techniques used in SCA applications. SVM models were first chosen because of their ease of use, strong mathematical foundation, and positive results in other domains. More observations regarding how to select an appropriate algorithm were presented in [1,5,6] with a high emphasis on the importance of proper algorithm selection during ML-based SCA. In this regard, DL techniques select features automatically from data. Therefore, feature selection has less impact on SCA when DL is applied [4].

4) Hyperparameter Optimization: Hyperparameter optimization is the process of finding the best set of parameter configuration settings for an ML model that maximizes an algorithm's accuracy. Many different hyperparameters can affect the overall behavior of the model, such as the loss function, the number of layers in NN, and the activation function [6]. However, selecting the best hyperparameter combinations is not a trivial task. The importance of loss function in the SCA domain has been recognized by many authors of the reviewed papers. In the most recent work [8,9], the authors provide systematic comparisons of the efficiency of many loss functions among different SCA contexts. Therefore, selecting the loss function, which is one of the

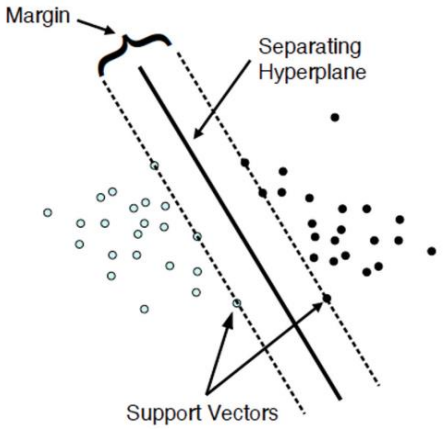


Figure 2: SVM: distance to the binary hyperplane for two sets of training data[1]

tunable hyperparameters, plays a significant role in the efficiency of training a DL model.

5) Model test: Based on the literature, ML and DL classifiers in the SCA applications use different evaluation metrics; hence comparing the conducted experiments and evaluating the quality of the models is hard. Prior contributions on the subject concentrated on classifying the secret key or individual bits of the intermediate bits. Other works attempted to extract the HW of specific key bytes (or the full key bytes). The efficiency of an attack was also measured in many different methods, where the most common metrics of such measures are key guessing entropy and accuracy. Furthermore, some studies take into account how various noise levels might affect the analysis, while others do not. Therefore, the researchers in the SCA community have shown some efforts to standardize the evaluation process of ML-based applications for SCA[1,8,6].

B. *Types of ML Used for SCA*

This section presents the types of ML techniques that are used in SCA for template and profiling attacks. Indeed, SCA frequently uses ML algorithms for classification, especially on datasets with high signal-to-noise ratios. The majority of work presented during recent years has investigated ML techniques for SCA for the aim of predicting discrete labels, which are subkey bytes in this case. Therefore, supervised learning techniques that are used in SCA will be investigated, while omitting the details about Unsupervised and Semi-Supervised learning algorithms due to the lack of usage in this field.[1,6,12]

1) Supervised ML Techniques: In supervised learning, the ML algorithm is fed with a training dataset (labeled dataset), where each input data (features) is associated with a corresponding output (labels). During the training process, the algorithm learns from the mapping of the input-output pairs provided in the labeled dataset. The main objective is to create a prediction model that performs well on unseen data, meaning it minimizes the difference between the predicted outcomes and the correct outcomes [6]. The following ML techniques are the most common supervised learning methods that are used in SCA for profiling and template matching.

a) Support Vector Machines: SVM is one of the most supervised learning algorithms used in SCA. It is also called a max-margin classifier since it creates a discriminative classifier in N dimensions by creating an optimal binary hyperplane based on the labeled training data. An optimal hyperplane is a decision boundary that separates data points of different classes as shown in Fig.2. To illustrate further, an optimal hyperplane is the one with the largest minimum distance to the training points; hence, noisy data will be correctly classified. Therefore, it maximizes the margin between training data from different classes in which the new examples mapped to the same class membership are predicted based on which side of the margin examples fall on [1][6]. SVMs models can analyze data and recognize patterns; thus, it is used for classification and regression analysis. As SVM can classify new samples to one class or the other based on training samples, it is considered as a non-probabilistic binary linear classifier. Indeed, most real-world problems involve nonlinear separable data, thus SVMs are extended to handle non-linear classification by using the so-called kernel trick, which transforms the input data into a higher-dimensional space. Indeed, the use of kernel tricks is well-established in most SCA settings [5,7,12].

b) Random Forests: Random Forest (RF) is an ensemble algorithm that consists of several decision tree classifiers. Its randomness is generated during its construction by using a bootstrapped dataset and feature selection in every node splitting. Each tree builds with a random subset of available features in every internal node of a decision tree. Typically, the feature selection for splitting is the square root of the total number of features or only one feature. Classification of new samples after training is done by making a voting-based decision among the grown trees. In fact, RF has shown successful results for many SCA implementations, especially for profiling attacks. This is because RF can assess the class probabilities accurately, which is crucial for figuring out the secret key [5,6,12].

c) Naive Bayes: The Näıve Bayes is a probabilistic ML algorithm used mainly for classification which is grounded in Bayes' theorem that assumes that features are conditionally independent given the class label. This conditional probability model assumes x = (x1,.....,xn) as a vector of values of n features (independent variables) that need to be classified. Then, it assigns probabilities for every vector p(ck | x1,.....,xn) for each of the K possible classes. In Naive conditional independence assumptions, the set of classes(c1,.....ck) is mutually exclusive and exhaustive. Using Bayes' theorem, the posterior conditional probability can be decomposed as p(ck |x) = p(ck). p(x|ck)/ p(x). The Na¨ıve Bayes classifier combines the Na¨ıve Bayes probability model with a decision rule. In the SCA field, the Näıve Bayes classifier has been applied widely because of its high computational efficiency[1,5].

2) Unsupervised ML Techniques: Other ML paradigms have been used in SCA besides supervised learning, including unsupervised learning and semi-supervised learning. Fig.3 depicts the main difference between these learning paradigms. In unsupervised learning, the algorithm is fed with an

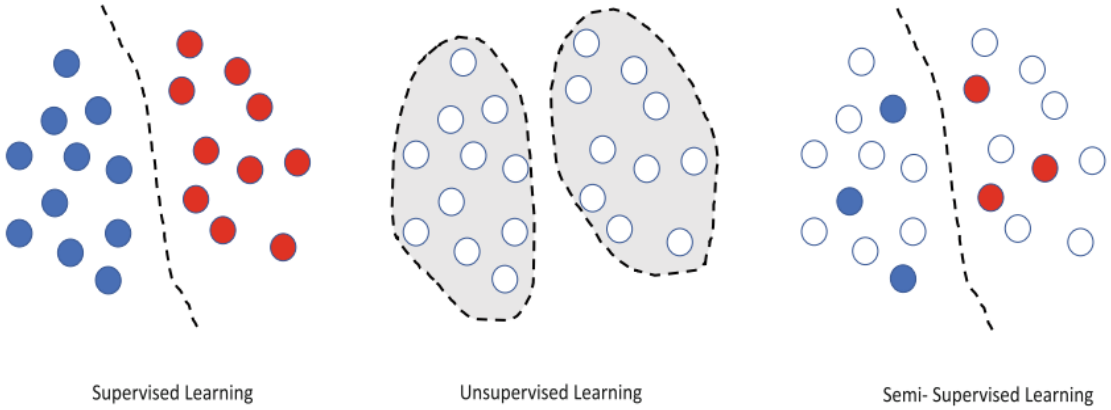


Figure 3: Machine Learning Models [1]

unlabeled dataset and the goal is to uncover hidden patterns, similarities, differences, relationships, or clusters within the data without using class labels. The main challenge in unsupervised learning is the difficulty of assessing the performance of the learned model due to the absence of labels [1].

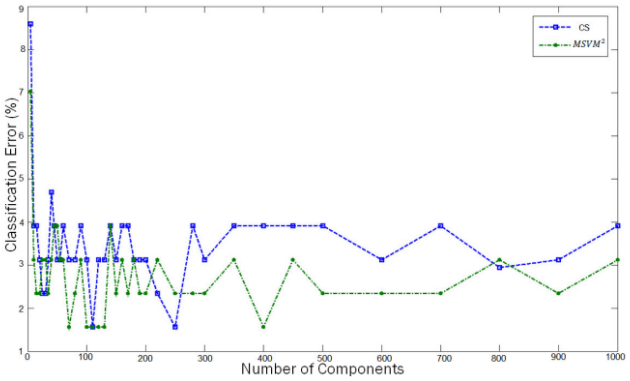
3) Semi-Supervised ML Techniques: This model lies in the middle ground between supervised and unsupervised learning. In semi-supervised learning, the training dataset is a mix of labeled and unlabeled data due to the limitation of labeled data resources.

C. Evaluation of Traditional ML Techniques in SCA

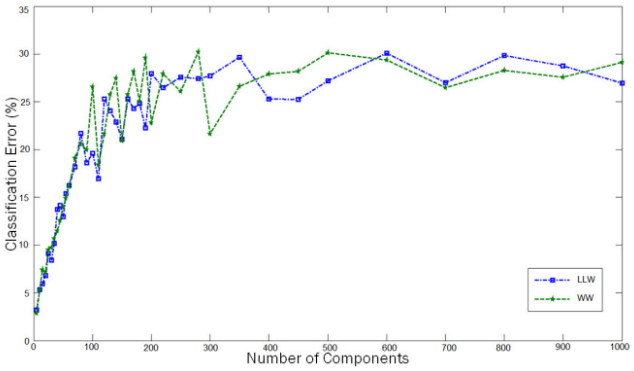
In recent years, researchers have been doing significant work to apply various ML algorithms for SCA in order to improve the effectiveness of the attack and its countermeasures. The current work aims to evaluate different scenarios such as different ML algorithms, different cryptographic algorithm implementations, different dataset dimensions, different countermeasures, and different dataset noise levels in order to improve SCA through the appropriate utilization of ML approaches. Indeed, it is challenging to compare the analyzed work because distinct procedures were used, different assessment measures were used, and experiments were barely conducted over the same datasets. To analyze and evaluate applications of ML in SCA, some of the recent papers that compare the traditional template attack to traditional ML-based SCA are summarized. Indeed, most of the published work in this area is based on profiling attacks; hence this section will present different applications of profiling attacks by support of traditional ML.

Saeedi and Kong [13] reported an attack on data leakage of FPGA implementation of Elliptic Curve Cryptography(ECC). They investigated the power of dimensionality reduction with Principal Component Analysis(PCA) in reducing noise and complexity of input data. They also found that when applying SVM with RBF or polynomial kernels, about 600 components out of 2500 sample points are sufficient to reach an accuracy of about 95% as seen in Fig.4. In conclusion, they recommend that the Gaussian RBF kernel function would provide the highest level of multi-class classification accuracy for SCA applications.

The authors of [12] carried out a study to evaluate the performance of ML compared to Template Attack(TA). They compared TA with many ML scenarios by using SVM and RF as ML techniques. The first experiment was conducted in the perfect profiling model (where an attacking model is identical to the target device) and the leakage samples contain useless points. In this scenario, if the useful leakage points are independent of useless ones, ML attacks cannot outperform TA.In the second case, which assumes imperfect profiling as the attack affected by errors during profiling there is variation in the number of useless points. They observed that when the size of useless samples in leakage traces increases and/or the profiling set gets too limited, ML-based attacks gain interest as compared to TAs. Additionally, it has been demonstrated that RF has good performance due to its randomization-based feature selection.



A: classification based on CS and M-SVM2 algorithms



B: classification based on LLW and WW algorithms.

Figure 4: The error ratio of different multi-class classifiers as a function of principal components[13]

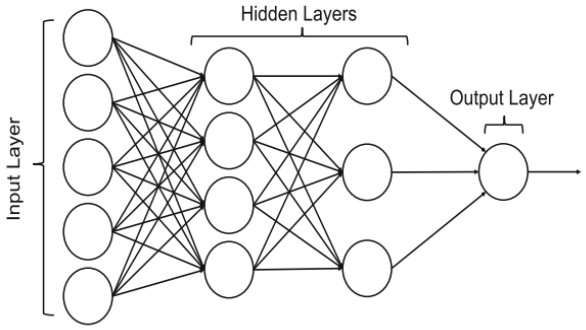


Figure 5: Multilayer Perceptron Architecture [1]

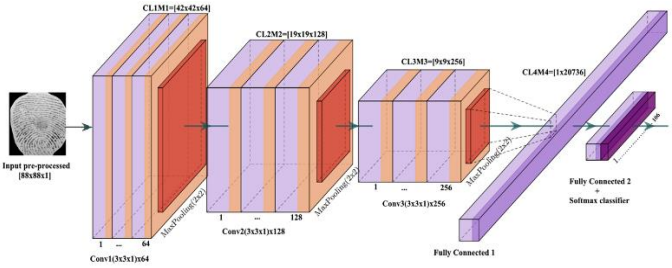


Figure 6: Convolutional Neural Networks Architecture [1]

IV. Deep Learning Techniques for SCA

This section presents the types of DL techniques that are used in SCA. The advantages of deep learning in contrast to classic techniques and traditional machine learning. The subsection A, introduces Multilayer Perceptron (MLP) along with its advantages in respect to SCA. The subsection B, discusses Convolutional Neural Network (CNN) and how it is used in profiled SCA. Subsequently, subsection C describes various enhancing techniques of DL.

A. Multilayer Perceptron (MLP)

The simplest kind of neural network known as Perceptron, is a linear binary classifier. This learning algorithm works on mimicking the actions of neurons in the human brain. It only works for linearly separable data, i.e., where a hyperplane can separate positive from negative points [11]. To handle nonlinear problems, more layers of perceptron were added to form Multilayer Perceptron(MLPs). MLPs are feed-forward neural networks that map input to output sets. It consists of multiple layers: an input layer, a hidden layer, and an output layer as shown in Fig.5. Each layer is fully connected to the next layer. MLPs work by forwarding an input vector of the training dataset and then the output and hence the error is computed. Then back-propagation technique is used to reduce error by adjusting weights. This method is used until the training data is learned. Despite their effectiveness, MLPs are considered "black boxes" due to challenges in fully understanding and interpreting their internal work.

In the context of SCA, MLPs were used for tasks like AES key classification. Initially, MLPs provide crisp classifications based on different candidate keys. In later works, they adopted to produce probabilistic outputs, more suitable for attack success metrics like guessing entropy in context of SCA evaluation. MLPs are known for their potential in breaking mask-protected implementations [11], as they can implicitly perform complex combinations of features, akin to higher-order SCA.

B. Convolutional Neural Networks (CNNs)

The Convolutional Neural Network (CNN) has structural properties that make it more robust in comparison with MLPs. A typical CNN architecture is formed by superimposing the convolutional layer and pooling layers before a fully connected layer. The convolutional layer specializes in linear operations using shared weights to produce feature maps as shown in Fig.6. These feature maps are then processed through an activation function. A pooling layer is usually placed after a convolutional layer and activation function. Its main objective is to reduce the size of the feature map. In CNN, the superposition of pooling layers and convolution layers plays the role of extracting features. CNN is better in terms of robust data distortions, especially in the case of image recognition, mainly because CNN considers data topology whereas MLP just uses numerical values.

In SCA, CNNs are useful due to their resilience against noise and countermeasures. Maghrebi et al. [11] were the first ones to apply CNN architecture to SCA, effectively countering jitter-based hiding methods in AES encryption without pre-processing. Despite having fewer weights than MLPs, CNNs need extensive training data to learn invariant features. Hettwer et al. [6] improved CNN performance in Deep Learning Side-Channel Analysis (DLSCA) by introducing Domain Knowledge (DK) neurons to enhance feature classification/regression. It was discovered that using round key information in the form of labels was more effective than using S-box output. Alternative input methods, such as transforming power traces into spectrograms and adding artificial noise, have been proposed for increasing robustness. CNN has been successfully applied to profiled SCA against secure public key cryptosystems, demonstrating their versatility.

C. Evaluation of DL Techniques in SCA

The integration of Deep Learning in the Side Channel Analysis has led to research on taking leverage for enhancing attack from well-known techniques such as regularization, visualization, hyper-parameter optimization, and model interpretation.

1) Regularisation Techniques in SCA: Regularization techniques mainly evolved to tackle the overfitting problems during training. The techniques include data augmentation, noise addition, weight decay, dropout layers, and early stopping. Data Augmentation methods, such as random traces and trace wrapping enable CNNs to adapt to side-channel measurements with desynchronization issues, which is a common issue faced in protecting AES implementation. Kim et al. [14] explored it further by using additional noise as a regularization method. This helped to understand the advantage of convolutional layers for leakage detection.

2) Hyper- Parameter Optimisation and Architectural Development: One of the most crucial parts of training is the selection of hyper-parameters in Deep Neural Networks. State-of-the-art CNN architectures from image recognition, like VGG-16, ResNet-50, and Inception-v3, are being used and fine-tuned for SCA, aiming at the optimization of hyper-

parameters. It helps in the development of CNN architectures specifically updated to SCA, where initial models are refined to suit dataset characteristics.

3) Visualisation and Model Interpretability: Visualization techniques in profiled SCA are mainly used to identify the main input features that the trained model considers the most crucial for classification. Masure et al.[8] uses gradient visualization. Singular Vector Canonical Correlation Analysis (SVCCA) has gained attention in model interpretability. SVCCA is employed to understand what neural network is learning from Side-channel datasets.

It has been seen that deep learning techniques in SCA have significantly affected advancing and making the system more robust in comparison to traditional methods. It offers enhanced attack performance through better regularization, hyper-parameter optimization, and improved model understanding by visualization and interpretability tools.

V. Experiment

One application of employing ML for SCA is recovering encryption keys. The hardware component used for encryption is fed the plaintext data, and the secret key hardwired internally is to be extracted. Analysis of the power consumption in order to derive the secret key for an implementation of AES which is unprotected on the example board is demonstrated, exposing terminals that can be probed to determine key bits from electrical current. The experiment performed by researchers at Google's cybersecurity is replicated and explained [3].

A. High Level Overview

SCA is an effective way to attack secure hardware because it targets the implementation which is harder to secure than the algorithm. AES is an example of this which is vulnerable to SCA [10].

This experiment uses the open-source platform TensorFlow's model to recover AES keys from a TinyAES implementation running on an ARM CPU (STM32F415) from its power consumption traces obtained with ChipWhisperer. Datasets of oscilloscope electrical current readings for given plaintext inputs are collected. The datasets are used to train a convolutional neural network model. Because the datasets were large and training would require GPU resources of significant scale, a pre-computed model would be used with only CPU-optimized usage. The trained model is then selected to run a trace. That trace predicts each bit's attack point value, recovering the key bit and combining predictions for the keys in the dataset. Metric computations to evaluate attack efficiency in the worst case for the implementation the attacker can recover about 40% of the key with a single trace, and four traces achieve 100%.

B. Algorithm

For each byte of the key to recover, 1 to 16:

- Load its associated data from the dataset and the signals related to that part of the key. This involves collecting the traces associated with the one byte of the key, including the transitions detected for all rounds of computation in AES.
- Load the convolutional neural network model from the library, previously selected from options that the TensorFlow platform provides.
- The model predicts intermediate factors used to compute the actual value, based on the encryption algorithm and output from the last layer. The actual current byte of the key is then computed from the previous intermediate predictions.
- Accumulating probability, by accounting for each trace performed. Order the aggregate of predictions by probability. Again, this will only be from the four traces necessary to obtain an accurate value. Because only four probabilities are needed, the fundamental lack of efficiency here by sorting may be neglected. Select the strongest guess for the key's byte.

These procedures are looped sixteen times, each time deriving the correct key byte based on all its related traces. The recovered key is compared to the real key for demonstrative purposes, as the real key is included in the datasets obtained for the supervised ML. A consistent diagonal on the confusion matrix shows the model performs equally on all the predicted values versus the actual values as shown in Fig.7.

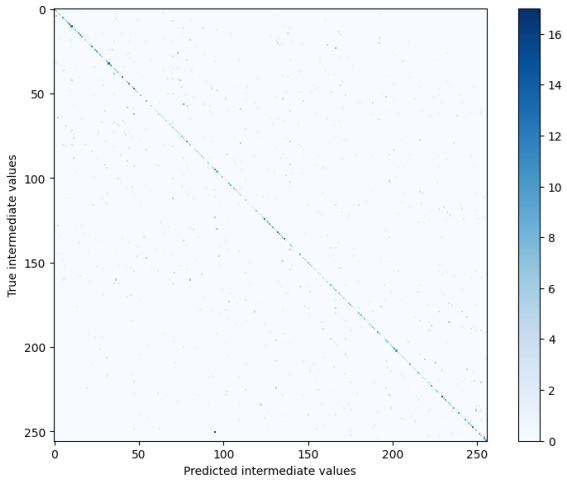


Figure 7:Prediction Confusion Matrix

Another graph of performance in Fig.8. shows that four traces are sufficient to recover the key accurately.

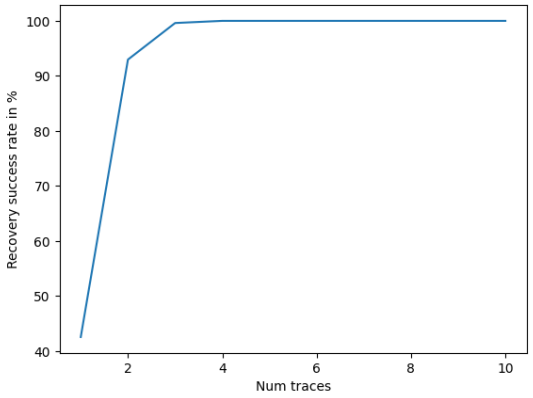


Figure 8: Recovery Performance in TinyAES

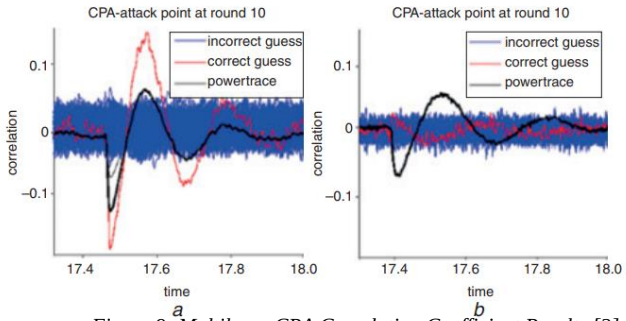


Figure 9: Multilayer CPA Correlation Coefficient Results [2]

a) Unprotected AES(500 traces)
b) Protected AES (100,000 traces)

VI. Different Applications of ML(and thereby DL) to SCA

The practicality of the ML and DL for SCA goes further than testing or designing new attacks. This section outlines the various ML and DL applications on SCA which are different from applying them as a trained classifier.

A. *Using ML as a countermeasure for SCA:* Shan and Zhang [2] utilize the efficiency of ML by proposing an SCA resistance methodology. They implemented their experiment in the AES-128 circuit and trained an ML power compensation module to compensate for the probability of Hamming Distance (HD) of the intermediate data in order to make it incapable of differentiating between the correct and incorrect sub-key; hence protecting against SCA. HD is used mostly to find the distance between strings of equal length. In particular, the ML technique is applied to determine the optimal HD redistribution mapping by using neural dynamic programming. Their results on protected AES, and analysis of 100,000 traces can provide 200× measures to disclosure with no sign of discovering the correct sub-key as shown in Fig.9. They also considered their application as a digital design that does not affect the original AES's maximum frequency. Furthermore, it has zero frequency overhead, low power, and area overhead; thus making it suitable for hardware implementation of SCA countermeasures.

B. *DL Against Public-Key Implementations:* The public key implementations (e.g. RSA or ECC) are usually protected by randomization techniques that make single trace attacks the only feasible side-channel solution. In [17], the authors show the result of multiple successful DL attacks (especially the CNN models) against protected RSA implementation. Remarkably, they stressed the need for analyzing and developing efficient countermeasures.

C. *Deep Neural Network as a Side-Channel Distinguisher:* In 2019, Timon introduced the first non-profiled DL techniques to attack secure AES implementations [18]. The conclusion of his analysis that in a non-profiled settings, a trained Deep Neural Network (DNN) can be used as a key distiniguisher. To do that, for every key byte candidate, the authors train an identical DNN architecture. Then, conducting separate training by using training traces labeled in accordance with the current key guess. He considered the divide-and-conquer strategy since it is commonly used for AES; thus in order to recover a single key byte, the analysis would need 256 training phases at least. He also illustrated that the complexity is appropriate for some particular targets. Beyond the significant contribution of this work, there is a risk that the complexity will grow out of control if it takes too long to train a single model for a single key-byte candidate.

VII. Guidelines for conducting ML(and thereby DL) in SCA

In order to compare and reproduce the results of ML techniques in SCA, the researchers should follow the following guidelines. Indeed, these guidelines aim to standardize the presentation of ML techniques in SCA research, making it easier for further research of ML in SCA domain [6].

1) *Clarification of Preprocessing Techniques***:** All the techniques used in preprocessing techniques on raw data should be specifically stated.

2) *Detailed Description of Hyperparameters:* All details about hyperparameters used in ML algorithms, especially for complex models such as neural networks should be specified. The description should include architectural parameters and training processes such as optimizer, learning rate, batch size, and number of epochs.

3) *Explanation of Optimal Parameter Selection:* In cases when hyperparameters are searched the optimal parameter range and selection methods should be correctly specified. This information could be used to enhance the performance of other models too.

4) *Use of Multiple Performance Metrics:* Performance metrics in both ML and SCA fields should be used to measure model performance which makes it easier in case of comparison with related work.

5) *Application of Cross-Validation***:** For estimating final model performance, cross-validation methods should be applied.

6) *Utilization of Learning Curves or Restricted Attacker Framework:* Instead of fixing the number of training traces, learning curves should be used to understand the performance with varying data, and a restricted attacker framework [16] should be used for in-depth analysis.

7) *Identification of Sensitive Components:* A study should be conducted to further successively remove sensitive components from algorithms. It can be used to identify where the power of a certain approach comes from pre-processing[15].

8) *Quantification of Attacker Overhead:* All the information about the attacks should be mentioned. The resources required for an attack should include the minimum number of traces needed, time for training the model, and computing resources.

9) *Publication of Experiments and Data***:** It is encouraged to share software experiments and publish associated data on open-source platforms to stimulate further research in the ML community in the SCA domain.

VIII. Conclusion

This paper shows the application of ML and DL techniques for Side-Channel Analysis over the traditional methods. The ML techniques, such as SVM, Random Forests, Naive Bayes, MLP, and CNN have proved to be very effective in addressing SCA challenges. The research has shown the critical role of ML in increasing security measures against SCA attacks. In addition, an analysis of power consumption for extracting keys from AES implementation shows a practical example of ML in SCA. Moreover, the practicality of the ML and DL for SCA goes further than testing or designing new attacks (different from applying them as a trained classifier). We summarize some of different applications such as : using ML as a countermeasure for SCA, Deep Neural Network as a Side-Channel Distinguisher, and DL Against Public-Key Implementations. This leads to the takeaway messages of this paper: ML and specifically DL are powerful direction for various SCA applications when it properly used. Finally, the best practices and guidelines to enhance the effectiveness of ML and DL in SCA context were illustrated based on the recommendations from industry experts and academic researchers.

Acknowledgment

The experiment implementation conducted in this paper is freely available by Google research team under the links given in this paper [3]. The writers would like to thank Elie Bursztein (the leader of Google's anti-abuse research team) and his team for the fruitful and valuable practical guide.

References

[1] Jovic, A., Jap, D., Papachristodoulou, L., & Heuser, A. (2022). Security and artificial intelligence: A crossdisciplinary approach to solving (pp. 25–71). essay, SPRINGER NATURE.

[2] Shan, W., Zhang, S., & He, Y. (2017). Machine learning based side-channel-attack countermeasure with hamming-distance redistribution and its application on advanced encryption standard. Electronics Letters, 53(14), 926-928.

[3] Bursztein, E. (2019). SCAAML: Side Channel Attacks Assisted with Machine Learning. GitHub. https://github.com/google/scaaml

[4] Jin, S., Kim, S., Kim, H., & Hong, S. (2020). Recent advances in deep learning ‐ based side ‐ channel analysis. ETRI Journal, 42(2), 292-304.

[5] Picek, S., Heuser, A., Jovic, A., Ludwig, S. A., Guilley, S., Jakobovic, D., & Mentens, N. (2017, May). Side-channel analysis and machine learning: A practical perspective. In 2017 International Joint Conference on Neural Networks (IJCNN) (pp. 4095-4102). IEEE.

[6] Hettwer, B., Gehrer, S., & Güneysu, T. (2020). Applications of machine learning techniques in side-channel attacks: a survey. Journal of Cryptographic Engineering, 10, 135-162.

[7] Hospodar, G., Gierlichs, B., De Mulder, E., Verbauwhede, I., & Vandewalle, J. (2011). Machine learning in side-channel analysis: a first study. Journal of Cryptographic Engineering, 1(4), 293-302.

[8] Masure, L., Dumas, C., & Prouff, E. (2020). A comprehensive study of deep learning for side-channel analysis. IACR Transactions on Cryptographic Hardware and Embedded Systems, 348-375.

[9] Kerkhof, M., Wu, L., Perin, G.,Picek, S. (2023). No (good) loss no gain: systematic evaluation of loss functions in deep learning-based side-channel analysis. Journal of Cryptographic Engineering, 13,311–324

[10] Tehranipoor, S., Karimian, N., & Edmonds, J. (2023, July). Breaking AES-128: Machine Learning-Based SCA Under Different Scenarios and Devices. In 2023 IEEE International Conference on Cyber Security and Resilience (CSR) (pp. 564-571). IEEE.

[11] Maghrebi, H., Portigliatti, T., & Prouff, E. (2016). Breaking cryptographic implementations using deep learning techniques. In Security, Privacy, and Applied Cryptography Engineering: 6th International Conference, SPACE 2016, Hyderabad, India, December 14-18, 2016, Proceedings 6 (pp. 3-26). Springer International Publishing

[12] Lerman, L., Poussier, R., Markowitch, O., & Standaert, F. X. (2018). Template attacks versus machine learning revisited and the curse of dimensionality in side-channel analysis: extended version. Journal of Cryptographic Engineering, 8, 301-313.

[13] Saeedi, E., Hossain, M. S., & Kong, Y. (2015, July). Multi-class SVMs analysis of side-channel information of elliptic curve cryptosystem. In 2015 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS) (pp. 1-6). IEEE.

[14] Kim, J., Picek, S., Heuser, A., Bhasin, S., Hanjalic, A.: Make some noise: unleashing the power of convolutional neural networks for profiled side-channel analysis. Cryptology ePrint Archive, Report 2018/1023 (2018). https://eprint.iacr.org/2018/1023

[15] Langley, P.: Crafting papers on machine learning. In: Proceedings of the Seventeenth International Conference on Machine Learning (ICML), pp. 1207–1212 (2000)

[16] Picek, S., Heuser, A., Guilley, S.: Profiling side-channel analysis in the restricted attacker framework. CryptologyePrint Archive, Report 2019/168. https://eprint.iacr.org/2019/168 (2019). Accessed 19 Mar 2019

[17] Carbone, M., Conin, V., Cornelie, M. A., Dassance, F., Dufresne, G., Dumas, C., ... & Venelli, A. (2019). Deep learning to evaluate secure RSA implementations. IACR Transactions on Cryptographic Hardware and Embedded Systems, 132-161.

[18] Timon, B. (2019). Non-profiled deep learning-based side-channel attacks with sensitivity analysis. IACR Transactions on Cryptographic Hardware and Embedded Systems, 107-131.