IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 31, 2021, accepted September 21, 2021, date of publication October 4, 2021, date of current version October 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3117761

Deep K-TSVM: A Novel Profiled Power Side-Channel Attack on AES-128

SOROOR GHANDALI[1], SAMANEH GHANDALI[2], AND SARA TEHRANIPOOR[1]
[1]Department of Electrical and Computer Engineering, Santa Clara University, Santa Clara, CA 95053, USA
[2]Google, Mountain View, CA 94043, USA

Corresponding author: Sara Tehranipoor (ftehranipoor@scu.edu)

ABSTRACT The appearance of deep neural networks for Side-Channel leads to strong power analysis techniques for detecting secret information of physical cryptography implementations. Generally, deep learning techniques do not suffer the difficulties of template attacks such as trace misalignment. However, the generalization of a trained deep neural network that can accurately predict Side-Channel leakages largely depends on its adjustable variables (parameters of a neural network). Although pre-training is no longer mandatory, it is needed for parameter selection of a deep neural network to improve the success rate and provide a better insight into the network's inner functionality. In this paper, we propose a novel model via Twin support vector with a deep kernel approach when targeting a hardware implementation of AES-128. The proposed model is pre-trained with the Restricted Boltzmann Machine method in a layer-wise manner and then fine-tuned via gradient descent. Further, we used the grid search technique for the selection of each hyperparameter which is used to compute class probabilities of every test trace in our deep model based side-channel attack. Based on our analysis and experiments, this model empirically shows its effectiveness by outperforming some of its competitors in profiling attack methods such as convolutional neural networks and multilayer perceptron models. We also evaluate our model on both masked and unmasked AES implementation. The results indicate that the proposed approach has achieved a success rate of greater than 99% even with a single trace using Keras library with Tensorflow. We investigate the correct "key rank" according to the number of traces; our model reaches the key $rank \leq 10$ when attacking the third AES SBox.

INDEX TERMS Hardware security, side-channel analysis, deep learning, machine learning, profiling attacks, twin SVM, RBM, kernel function.

I. INTRODUCTION

In the field of security and cryptography, Side-Channel Analysis (SCA) is a powerful attack that benefits from any leakage of a system such as power consumption and electromagnetic (EM) radiations, to retrieve the secret information [1]. Power-based side-channel attacks, according to how much they have access to the device under the attack (victim device), are categorized into non-profiling [2] or profiling attacks [3], [4]. Non-profiled side-channel attacks include statistical calculations on the power consumption of a device under the attack to retrieve the secret key. Profiled side-channel attacks assume the attacker has complete control on a copy version of the victim device to measure an infinite number of side-channel traces. Then the attacker classifies the leakages based on obtained data correlation and provides a predictive model to retrieve secret information of a new device.

One popular subset of profiling attacks is template attack (TAs) [5], in which an adversary is able to retrieve the secret information even with few traces of data. The attack phase is defined in two stages. The first stage creates a template for the operation of a copy version of the victim device by an infinite number of power traces and known different secret keys and public data. The second stage retrieves the key of a victim device using the template of the first stage with a small number of traces from the victim device. In high dimensional side-channel attacks, building a leakage model becomes difficult, hence, the points of interest are extracted from traces in order to have more informative kinds of traces. Note that each trace generally includes a large number of sample points which make the attack very costly, hence points of interest are selected which are samples with the most informative leakage from the device. Another powerful subset of profiling

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Liu.

attacks is a side-channel attack that uses Machine Learning (ML) approaches [3]. Exploring and studying machine learning techniques in unmasked and masked side-channel attacks implementation show that this approach enhances attack accuracy and is more efficient when compared to standard profiled attacks. However, the main drawback of this technique is the need for human processing to find the most informative data.

In order to extract the point of interest in machine learning algorithms, several dimension reduction techniques including Principal Component Analysis (PCA) can be employed and Linear Discriminant Analysis (LDA). PCA is also a common approach in SCA to reduce the amount of noise in the traces by keeping high variance data. The PCA generally transforms the dataset and fits with the training dataset.

With the emergence of big data, deep learning (DL) techniques have been assumed to outperform other profiling attack techniques as they learn patterns from the raw data themselves and identify features automatically without extensive statistical analysis. Cagli *et al.* [6] utilized deep neural networks to retrieve cryptographic secret information by investigating convolutional neural networks (CNNs). They attacked the AES masked with a jitter-based countermeasure and tried to deal with the trace misalignment by considering data augmentation in their convolutional neural network. Kim *et al.* [7] come up with a technique to overcome the misalignment problem. They solved the problem by inserting noise to the input traces or inserting random delays through dummy operation.

Although these publications provide a good insight on how to deal with side-channel attacks challenges using deep learning models, none of them have investigated the relation between initial parameters and the pre-training stage for generalization problems.

In most side-channel attacks when targeting AES-128, an attacker targets a subkey (one byte) and not the whole 128-bit (16 bytes) key. After finding one subkey, the attacker only needs to repeat his attack 15 more times to recover the remaining 15 bytes. Usually, in security papers [8], [9], authors only explained their methods and results on one subkey; since the rest can be recovered in the same manner/technique. Figure. 1 Show a diagram of a side-channel attack on one of the rounds of AES-128 cryptosystem using a neural network. AES encrypts 16 bytes of plaintext in several steps. The first step is SubBytes, where each byte of plaintext is replaced with a value from a lookup table, called SBox. Then the replaced value is passed through the second step named Shift Rows, in which each row is shifted except the first row which remains unchanged. Next, the Mix Columns step is applied to combine the four bytes in each column, and then each byte is combined with a byte of the key in the Add Round Key step.

In this work, we propose a novel and powerful deep learning model, "Deep Kernel learning Twin Support vector machine" (Deep K-TSVM) which is a great combination of deep neural network and Twin support vector machine (TSVM) classifier. We apply our 256-class deep model against both masked and unmasked AES-128 implementations. Our proposed model successfully breaks the cryptographic implementation using just one trace of the target device. Further, we study how generalization affects the performance of the attack. We pre-train a 256-class deep model with a deep Restricted Boltzmann Machine (RBM) for the initial weight of the neural network, In addition, we use grid search technique for proper choice of hyperparameters to prevent over or under-fitting problems.

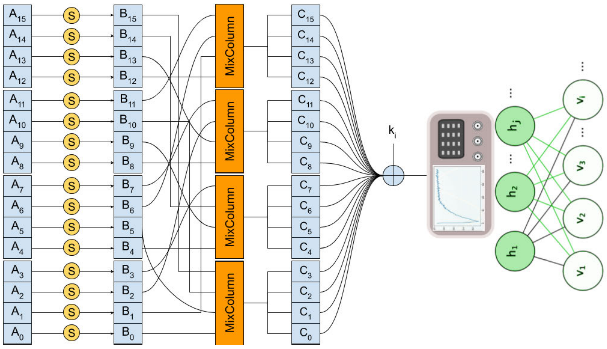


FIGURE 1. Diagram of SCAs on AES-128 using neural networks.

In summary, our main contributions are as follows:

- We propose a novel deep learning model which provides an efficient combination of deep neural networks and TSVM. The learning stage of the model carries out in two phases of pre-training with the RBM method and fine-tuning with the help of gradient descent.
- The proposed model is able to effectively break the cryptographic implementation even with a single trace. Our results indicate that the proposed model outperforms state-of-the-art techniques such as CNNs, NeuroSVM, and Multilayer perceptron (MLP) based attacks.
- We evaluate the impact of the hyperparameter selection in the generalization of our deep model. We also demonstrate how good is the effect of pre-training in the performance of the model.
- We investigate the correct key candidate from sorted probabilities of each possible value in the Identity (ID) leakage model and discuss that these probabilities. We also discuss how the information of these probabilities can be considered as a valid metric to evaluate deep model-based side-channel attacks.

The rest of the paper is organized as follows: Section II describes the Preliminaries and related works to conduct profiling side-channel attacks. Section III describes datasets that have been used to examine our method. Section IV explains the details of our proposed novel model "Deep K-TSVM". In Section V , we discuss numerical calculations and experiments to examine the performance of the proposed model. Finally, some concluding remarks are given in Section VI .

II. PRELIMINARIES AND RELATED WORK

Since our work aims to improve the generalization of deep neural network models based on side-channel attacks, in this section, we take a brief look at related publications on this topic in the recent few years. We also review some recent studies about supervised learning models including support vector machine (SVM) based side-channel attacks.

A. PROFILED SCA APPROACHES

Deep learning based side-channel attack models is a group of profiling attacks in which different factors easily affect their accuracy. Hence, there are alternative metrics for evaluating the success rate of the deep neural networks when attacking the victim device. In Table 1, the table summary of related works on SCAs using different learning methods, we provide a related survey on datasets, classifier, and kernel mapping of each method.

1) DEEP LEARNING BASED PROFILING SCAs

In every deep neural network, there is a learning step that requires a fixed-size training dataset for network training (in side-channel analysis, this step is known as the profiling phase) and fixed-size datasets for network testing (in side-channel analysis, this step is called the attack phase). In this paper, we use k-fold cross-validation method. In this method the whole data split randomly in k (the number of folds) disjunct subsets and iteratively use one of them as test set, while the rest of the data are used as the training set. The average error across all trails is the estimated generalization error.

Assume T_i as a single side-channel trace (which is captured using a random plaintext and a random secret key), and P_i as a vector of selected points of interest in the trace T_i (selected features). Then the class of each trace is defined based on the chosen leakage model because each label L_i of the input trace T_i is computed using a function (leakage model) of the secret key and P_i.

In profiled side-channel attack based deep learning, the adversary first collects a number of traces from a device which is a copy version of the device under the attack (on the best scenario, he has full controls on the copied device). He uses collected traces as the training set with known target labels L_i, to train his deep neural network. Then, in order to complete the attack, the adversary starts the attacking stage by applying the profiled model on a testing traces which have been collected from a different device called the victim device (device under the attack) with an unknown secret key value.

A demonstration of deep learning based profiled side-channel attacks is shown in Figure. 2. The device with the color gold in Figure. 2 is the copy version of the target device which is unlocked and the attacker can use this to measure an infinite number of traces for training his deep mode. Then he will use the trained network to unlock the device with the color gray in Figure. 2, as it is shown in the picture that he uses a very few traces (here a single trace) of the target device to retrieve the key.

In works [6], [7], some techniques have been proposed in profiled side-channel attacks to overcome the trace misalignment problems, they insert noise to input traces to solve this problem, however, this only helps some certain attacks by making the attack more difficult such as Gaussian template attack and does not affect deep learning based side-channel attacks and make no difference to their improvement. In 2017, Picek *et al.* studied the significance of how to improve the selected model with proper parameter tuning using tenfold cross-validation [19]. However, some of the classifiers in their works performed worse after the parameters tuning, and their proposed model, in the end, reaches an accuracy of 70% (not a high accuracy for learning methods) even with low noise in their traces. Also, different successful attacks against masking countermeasures on AES have been proposed. For example in 2014, Zeng *et al.* came up with an attack on the hardware implementation of AES which was masked with a Rotating SBox masking scheme which is a lightweight countermeasure. They used a support vector machine as the classifier of their model to find the secret mask. After that, they attacked each SBox with computing and use the hamming weight of the output byte of the mask SBox as the label of the model to retrieve the secret key [15]. However, the success rate in this model completely depends on the number of sample points. When there are a few sample points, the success rate is very low for this model.

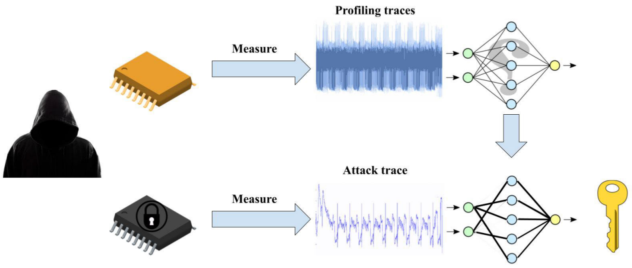


FIGURE 2. Deep learning based profiled SCA.

2) PERFORMANCE METRICS

Accuracy and loss function are common metrics to evaluate deep learning models. Accuracy is used to measure a specific learning algorithm performance by computing the rate of correct classifications. A loss function is usually applied to optimize a particular learning algorithm and shows how good the learning model is based on several iterations. However, these metrics are highly dependent on the small changes in parameters and hyperparameters which control the learning process. In other words, parameters (e.g. weights and bias) and hyperparameters (e.g. learning rate and layer-size) should be tuned to obtain a good performance model during the network training phase. Hence, training a deep neural network with high performance and well generalization is challenging for any new data. Generalization is a term used for this challenge to describe a model's ability to react to new data.

Generalization may lead to under-fitting or over-fitting problems. In under-fitting, a model has a poor performance on the training data and is unable to make accurate predictions, even for the training data. Over-fitting occurs when the

TABLE 1. Summary of related works on SCAs.

Datasets	REF./Year	Learning technique	Kernel mapping
Unmasked AES without pre-training	[10] /2020, [11] /2020, [12] /2015, [2]/2019, [13]/2015, [14]/2018	K-means, CNN, SVM, RF	PCA, Pooling, RBF
Masked AES without pre-training	[11]/2020, [15]/2014, [16]/2018, [17]/2019, [18]/2016, [10]/2020	SVM, MLP, CNN	RBF, Correlation, Pooling
Masked & Unmasked AES with pre-training and fine-tuning	**Our work**	**TSVM**	**Deep RBM**

learning model is highly accurate in fitting to training data, however, it is unable in the prediction of unseen data. In the profiled attack scenario, the over-fitting problem becomes more acute since the adversary has an unlimited number of traces from the copied version of the victim device in the training phase with known key but he should make a prediction with a very few numbers of real traces with an unknown key from the victim device.

Fortunately, there are alternative metrics that can be used in side-channel attack scenarios to overcome the generalization problems. Key rank and guessing entropy are examples of such alternative metrics which evaluate and analyze the ability of a deep model in revealing the secret key [20]. Cagli *et al.* [6] also showed that the accuracy achieved by the deep learning models cannot be a valuable alternative metric for Key rank which is a common performance measurement of side-channel attacks.

There are several publications that represent the performance of machine learning and deep learning techniques which have been applied for the purpose of side-channel attacks. Maghrebi *et al.* [18] have shown the improved performance of deep learning compared to the other side-channel attacks techniques, however, they also showed that on average, 200 traces are needed for recovering the key. Hettwer *et al.* [5] directly used the secret key as a label of the model; they also claimed that this would give more ability to the network to learn the most meaningful features of the leakage which is needed for classification. Additionally, there have been studies in supervised machine learning-based profiling side-channel attacks such as [13]–[15], [21] which use support vector machine, LS-SVM, KNN, and RF for the classification problem. In most cases still, a large number of traces is required to reach a stable key rank. In 2018, Benadjila *et al.* [16] showed the role of points of interest which is exactly the same as the features extraction module in deep learning models which have a significant effect on the performance of side-channel attacks. However, the selection of feature extraction module and the size of points of interest can be different depends on the victim device and deep learning structures. An overview of the related works is given in Table. 1.

Although all of these works give us a better view of the side-channel attacks challenges and solutions, pre-training the initial weights and proper choice of hyperparameter for generalization problem of the learning model has left for future works. Also, the deep structure of kernel mapping has not been investigated in side-channel attacks. In this work for the first time, we propose a deep kernel mapping with TSVM with pre-training of initial values and fine-tuning stage. We target hardware implementation of masked and unmasked AES-128, and we were able to retrieve the correct secret key even with a single trace.

B. SUPPORT VECTOR MACHINE & NEROSVM

Support vector machine is a supervised machine learning method which performed for linear binary classification and aims to provide a binary classifier with a high generalization ability using a specific hyperplane. In other words, consider the training set: $\{(x_i, y_i); i = 1, 2, \ldots, N\}$, where $x_i \in \mathbb{R}$ is the input and $y_i \in \{1, -1\}$ indicates the corresponding label of the two classes. A linear soft margin support vector machine model explores for a proper hyperplane $(Wx + b)$ which has the maximum margin from the two classes. This hyperplane could be obtained by calculating the following optimization problem:

$$\min_{W,b} \frac{1}{2}||W||^2 + C\sum_{i=1}^{N}\xi_i^2$$
$$s.t.\ y_i(Wx_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0,\ i = 1, \ldots, N \quad (1)$$

where $\xi_1, \ldots, \xi_N$ are the non-negative deviations from the margin. The parameter C is called the penalty factor which controls the trade-off between the amount of margin increase and the number of misclassified sample points. The matrix of weight represented by W and the vector of bias is shown by b.

We can show the equation (1) as the following unconstrained problem:

$$\min_{W,b} \frac{1}{2}||W||^2 + C\sum_{i=1}^{N}\left[\max(1 - y_i(Wx_i + b), 0)\right]^2 \quad (2)$$

Since most of the real-world problems are kind of nonlinear ones, and it is not possible to be separated by a linear classification hyperplane, the kernel function (kernel trick) is applied to support vector machine. Kernel function maps data into a higher dimensional space where the data can be separated linearly. The process of selecting an appropriate kernel function has a profound impact on the performance of a support vector machine since this function changes the dimensional space of the data to where the linear classification module is possible [22]. In the concept of side-channel attack, it is most common to use linear kernel function or the radial basis function [23]. There are some common approaches to extend SVM for multi-class classification problems, including one-versus-all and one-versus-one methods [24].

It is shown in [25] that support vector machine attack performs better than the template attack especially when the captured traces are noisy. In work [26], authors distinguished power traces of an unmasked AES implementation using Least Square support vector machine (LS-SVM) as a variant of SVM, and the number of classes is defined by computing the hamming weight of the output of the SBox. TSVM [27], as an extension of support vector machine, is learned by solving two smaller quadratic programming problems, therefore it is approximately four times faster than the classical support vector machine [28]. The two optimization problems for each hyperplane of TSVM are as follows:

$$\begin{aligned} &\min \frac{1}{2}||W_A + e_1 b_1||^2 + c_1 e_2^T \xi \\ &s.t. \; -(W_B + e_2 b_1) + \xi \geq e_2, \quad \xi \geq 0 \qquad (3) \\ &\min \frac{1}{2}||W_B + e_2 b_2||^2 + c_2 e_1^T \eta \\ &s.t. \; (W_A + e_1 b_2) + \eta \geq e_1, \quad \eta \geq 0, \qquad (4) \end{aligned}$$

where e_1 and e_2 are the column vectors of ones with a length equal to the row number of matrix A and B. The error vectors are shown by ξ and η [28].

According to the optimization problems of the TSVM model, it predicts new samples using perpendicular distance. Consider the new sample point x, it will be assigned to a specific class that has the minimum distance from its hyperplane. In other words, if we show the perpendicular distance of sample x from the each hyperplanes of TSVM with d_i, the new sample point x will be assign to the class i which has the minimum $d_i = |x^T w_i + b|/\sqrt{w_i^T w_i}$ for $i = 1, 2$.

Although TSVM is one of the strong classification models in machine learning techniques, its performance is very sensitive to the selection of the appropriate kernel function. However, choosing an appropriate kernel function for a problem is not an easy task. Saeedi and Kong [29] investigated the influence of kernel on the accuracy of the support vector machine-based side-channel attacks using RBF kernel function. Several approaches have been proposed to learn a suitable kernel function for support vector machine. For example, Cho and Saul [30] calculate the inner product of a mapping function Φ and propose a multilayer structure of the kernel function as follows: $k^{(\ell)}(x_i, x_j) = \langle \underbrace{\Phi(\Phi(\ldots(\Phi(x_i))))}_{\ell \text{ times}}, \underbrace{\Phi(\Phi(\ldots(\Phi(x_i))))}_{\ell \text{ times}} \rangle$.

However, their method came up with a limitation where the multi-layer kernel is limited to a single type of mapping function. Another model called Neural network support vector machine (NEROSVM), introduced by Ghanty *et al.* [31] proposed for selection of appropriate kernel function. This model consists of two distinct but related modules called the feature extraction module (SFM) and classification module. Figure. 3 represents the NEROSVM model which has m different trained MLPs labeled as $(MLP_1, \ldots, MLP_m)$. The output layer and its associated connections from each of the MLPs are removed after the networks are trained. Note that the combination of the two first layers is considered a feature extraction module. Feature extraction module of NEROSVM consists of several submodules $(SFM_1, \ldots, SFM_m)$, each of them takes the same p dimensional data $X = (x_1, \ldots, x_p)^T$ as input, and transfer it to a n dimensional hidden vector. Indeed, each submodule of the feature extraction module is obtained from a MLP network including just one hidden layer [31]. The classification module is driven from the collected outputs of the m submodules.

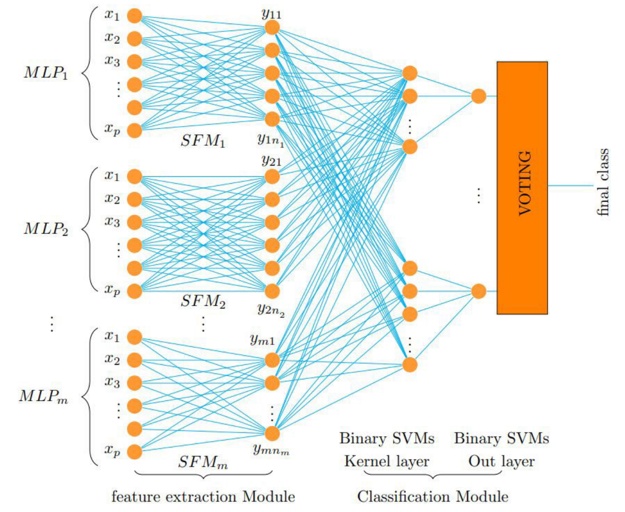


FIGURE 3. A graphical representation of the NEROSVM classifier [31].

To the best of our knowledge, there are only a few works have considered combining neural network and machine learning techniques which are supervised learning approaches. Therefore, rather than combining MLPs and SVMs, we develop a deep structure of RBM with TSVM as a strong learning model to provide a template particularly suitable for application in SCA.

III. LEAKAGE CLASSIFICATION

In deep learning based side-channel, attacks according to the selected leakage model, accuracy only computes the number of classes a deep neural network could correctly predict divide by the total number of predictions that have been made. Hence it is necessary to take into account an alternative metric to accuracy such as key rank that we talked about some of the associated publications with this method in II-A2. We also describe how to calculate this metric in the following subsection III-A

A. CLASS PROBABILITY

Deep learning models have great potential in side-channel attacks to map the problem into a supervised classification instead of computing the Gaussian probability density function used in regular template attacks.

An accuracy is a common approach in deep learning models to identify the rate of correct classification and just consider labels with great reliability. This approach is related to the side-channel analysis evaluation metric. In fact, neurons in the output layer show the number of classes in the deep

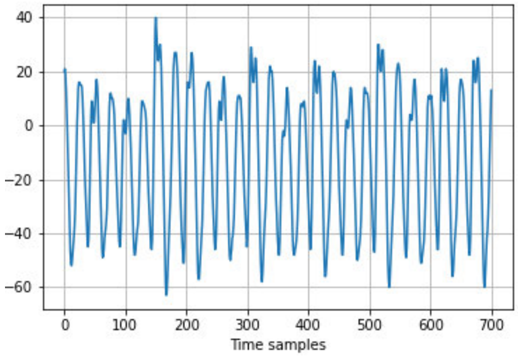


FIGURE 4. Captured original power traces from ASCAD dataset.

learning model. Probabilities of each neuron in the output layer could be obtained using the Softmax activation function, which is equal to the class probabilities for each trace of the leakage model. As we know accuracy is considered as an evaluation metric in side-channel analysis to distinguish class probabilities for every trace and its associated secret key in the selected model. However, any small changes in the structure of a deep model have an impact on these probabilities. Therefore alternative metric such as key rank is computed for each correct key candidate.

The number of classes based on the selection of the leakage model could be different. If we consider K is the number of classes and we indicate the number of traces that have been captured from the victim device with T, the output of the deep neural network is a $k \times T$ matrix. Then the log-likelihood for each key candidate is computed to distinguish classes. We computed the probabilities of each class of two different datasets and we were able to attack successfully on both masked and unmasked implementations.

B. DATASETS

For evaluating the performance of our model, we examined it on the public ASCAD dataset provided by Benadjila *et al.* [16] which has 50000 profiling traces, and 10000 attack traces. The traces are recovered from an 8-bit AVR microcontroller from a masked implementation of AES-128. The database consists of raw traces that contain the measurements from the entire encryption, however, the authors have preselected a window in the raw traces that correspond to the SBox operation and consists of 700 features. We consider the masked ASCAD with the leakage function of SBox$(p \oplus k)$ and we describe the unmasked ASCAD with the leakage function of SBox$(p \oplus k) \oplus M$, where M is the mask [32]. Figure. 4 shows the corresponding data trace for ASCAD which comprises 700 sample points.

Also, we use an open TeSCASE dataset to evaluate our model. This dataset has 1,400,000 power traces, and every trace includes 3125 samples [33]. The traces are captured from the Virtex-5 FPGA embedded on a SASEBO-GII board of masked and unmasked implementation of AES-128 and can be accessed on the TeSCASE website. Figure. 5 shows the corresponding data trace for TeSCASE which comprise 3000 sample points.

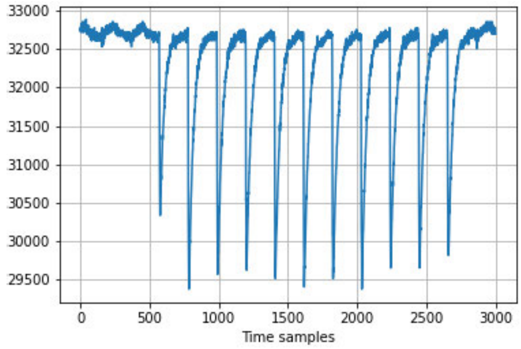


FIGURE 5. Captured original power traces from TeSCASE dataset.

For profiling and attacking phases in our proposed model, we apply 10-Folds Cross-Validation to split the data into 10 different folds. We consider 9 folds to train our model and keep the last fold as test data, we then average the model against each of the folds.

IV. PROPOSED DEEP LEARNING BASED SCAs

We propose a new deep learning architecture for profiled side-channel attacks which consist of two main modules. The first module is feature extraction module that maps the input data into a feature space to make the problem a linear separable classification (selection of point of interests). The second one is the classification module which identify label for each input trace.

For the first module to learn the pattern of inputs we use a deep architecture of RBM which is a stochastic neural network and learn a probability distribution from the raw input data. RBMs are shallow, two-layer undirected neural networks. The first layer in RBM is called the visible layer, and the second one is the hidden layer. Each layer of deep RBM can extract features in order to feed them as the input of the subsequent layer.

For the second module, we use the TSVM classifier which is an extension of the support vector machine and is fed by the last layer of the feature extraction module. The performance of TSVM mainly depends on the choice of its Kernel function which plays exactly the same role as the feature extraction module in the proposed model. This classifier follows the grid search optimization algorithm to select values of c_1 and c_2 for the two optimization problems in equations (3) and (4). The proposed model is trained in two steps, first, we pre-train the parameters with RBM, and then in the second step, we fine-tune all parameters using gradient descent.

A. DEEP K-TSVM

Our deep kernel learning TSVM consists of I/O and several hidden layers, in which the output of the last hidden layer is considered as the input layer of the TSVM to find two decision boundaries. It is assumed the adversary classified the leakages based on the label that is achieved by calculating

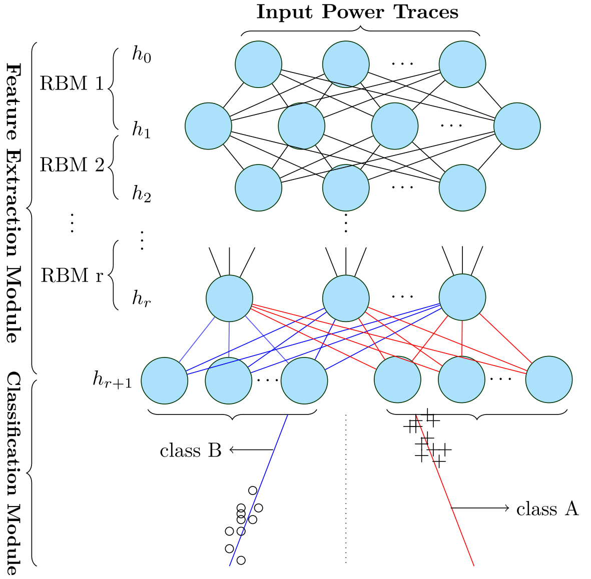


FIGURE 6. An architecture of deep K-TSVM.

the Hamming Weight(HW) of the output of the SBox values at the first round (9 classes) or achieved based on the identity (ID) leakage model (256 classes) of the AES-128.

The architecture of our proposed model is shown in Figure. 6 which has r different trained RBMs labeled as $(RBM1, \dots, RBMr)$. Hidden units of each RBM pass their values to visible units of the next layer. In other words, the hidden units of each layer can be considered as the visible units in the next layer. Note that the input layer h_0 and the hidden layers $(h_1, \dots, h_r)$ are considered as feature extraction module of deep K-TSVM model which maps raw data to a proper feature space. The classification module obtained by dividing the output of the $r+1^{th}$ layer into two vectors, and consider each vector as input of one of the hyperplanes of the TSVM classifier.

For each sample x_i and the relevance of hidden layers consider the following calculations:

$$\begin{cases} h_0^i = x_i;\ y_i = \pm 1 \\ h_j^i = \sigma(W^j h_{j-1}^i + b^j) \quad 1 \le j \le r, \\ \phi(x_i) = h_r^i;\ y_i = \pm 1 \end{cases} \tag{5}$$

where $i = 1, \dots, n$, $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid activation function and h_j^i denotes the i^{th} neuron in the j^{th} hidden layer. The optimization problems for deep K-TSVM can be written as follows:

$$\min_{W^j,\ \ b^j,\ \xi_i,\ 1 \le j \le r+1,\ 1 \le i \le n} \frac{1}{2}||\phi(A)W_1^{r+1} + e_1 b_1^{r+1}||^2 + C_1 e_2^T \sum_{i,y_i=-1} \xi_i^2$$

$$-(W_1^{r+1}\phi(x_i) + b_1^{r+1}) \ge 1 - \xi_i, \quad \xi_i \ge 0,\ y_i = -1 \tag{6}$$

and

$$\min_{W^j,\ \ b^j,\ \eta_i,\ 1 \le j \le r+1, 1 \le i \le n} \frac{1}{2}||\phi(B)W_2^{r+1} + e_2 b_2^{r+1}||^2 + C_2 e_1^T \sum_{i,y_i=+1} \eta_i^2$$

$$(W_2^{r+1}\phi(x_i) + b_2^{r+1}) \ge 1 - \eta_i, \quad \eta_i \ge 0,\ y_i = +1 \tag{7}$$

where $h_r^i = \phi(A)$ for $y_i = +1$ and $h_r^i = \phi(B)$ for $y_i = -1$. Line search would be a simple solution to the quadratic programming problems of above optimization problems to learn characteristics of the deep K-TSVM.

The initial value of the parameters W_j and b_j, are obtained using contrastive divergence (CD) learning [34]. This learning is a training algorithm that is commonly used when the direct evaluation of a function is not possible and needs to be approximated. In other words, a CD is a training algorithm that is applied to approximate the gradient of the model. RBM is considered as a probabilistic learning model which optimizes the log-likelihood values for its parameter selection by running a Markov chain on its dataset.

The unconstrained optimization problems of (6) and (7) can be written as the following quadratic programming problems:

$$\begin{aligned} L_{D_1} = \min_{W^j,\ \ b^j,\ 1 \le j \le r+1,\ 1 \le i \le n} & \frac{1}{2}||\phi(A)W_1^{r+1} + e_1 b_1^{r+1}||^2 \\ & + C_1 e_2^T \sum_{i,y_i=-1} \left[\max\left(1 + (W_1^{r+1}\phi(x_i) + b_1^{r+1}), 0\right)\right]^2 \end{aligned} \tag{8}$$

$$\begin{aligned} L_{D_2} = \min_{W^j,\ \ b^j,\ 1 \le j \le r+1,\ 1 \le i \le n} & \frac{1}{2}||\phi(B)W_2^{r+1} + e_2 b_2^{r+1}||^2 \\ & + C_2 e_1^T \sum_{i,y_i=+1} \left[\max\left(1 - (W_2^{r+1}\phi(x_i) + b_2^{r+1}), 0\right)\right]^2 \end{aligned} \tag{9}$$

Algorithm. 1 describes the learning process of Deep K-TSVM with respect to the weights of equations (8) and (9).

Algorithm 1 Learning Deep K-TSVM

```
Input: Datasets={$(x^i, y^i)$; $i = 1, \dots, n$}, maxepoch
Output: Parameters=$\theta$ : {$w_k^j, b_k^j$; $(1 \le j \le r+1), k = 1, 2$}
Pre-training Stage:
    initialize $w_k^j \approx 0, b_k^j \approx 0$;   $1 \le j \le r,\ k = 1, 2$
    learning $w_k^j, b_k^j$ withCD;
Fine-tuning Stage:
    initialize {$w_k^{r+1} \approx 0, b_k^{r+1} \approx 0; k = 1, 2$}
for epoch=1:maxepoch do
    for $i = 1 : n$ do
        Update weights of (8) & (9) by Gradient descent
    end for
end for
```

We apply the one-versus-all method which consists of multiple separate binary classifiers, one binary classifier for each possible outcome [24]. This method extends the binary

TABLE 2. Results of grid search hyperparameter optimization.

Model	Hyperparameter
MLP with 2 hidden layer	Epochs: [10,**30**,50]
	Optimizers: [**SGD**, RMSprop, Adam, Nadam]
	Activation: [ReLU, sigmoid, **Softmax**]
NeuroSVM	Epochs: [10,30,**50**]
	Optimizers: [SGD, **RMSprop**, Adam, Nadam]
	Learning rate: [**0.01**,0.001, 0.0001]
	P Dropout: [0.01,**0.1**,0.2]
	Activation: [ReLU, sigmoid, **Softmax**]
ConvNet	Epochs: [10,**30**,50]
	Optimizers: [**SGD**, RMSprop, Adam, Nadam]
	Learning rate: [**0.01**,0.001, 0.0001]
	Activation: [ReLU, sigmoid, **Softmax**]
Deep K-TSVM	Epochs: [10,**30**,50]
	Optimizers: [**SGD**, RMSprop, Adam, Nadam]
	Learning rate: [**0.01**,0.001, 0.0001]
	Activation: [ReLU, **sigmoid**, Softmax]

classification algorithm of TSVM to a multi-class classification algorithm where there are 256 binary classifiers. In this classification, the data from class i is considered as positive and data of all the other classes are considered as negative, where $i = 0, \ldots, 255$. All models predict a class by showing a score value, and the maximum of each prediction's scores is computed to unify the class label.

In our training procedure, we define 2^n classes for our model, where n is the number of bits in the ID leakage model. The value of learning rate, number of hidden neurons, and dropout, which have a serious impact on the accuracy and performance of the model, is set before the learning process begins by the grid-search method. Table. 2 is a summary of different values applied for different models in this work. Values that yield a model with the lowest key rank and high accuracy are selected for the attack model. Bold values in Table. 2 denote the selected values of each model. Note that for the CNN model, in addition to the CNN with Softmax activation function, we provide the CNN-SVM model, which has the same network parameters and details to ConvNet, except that the output of its final convolution layer which is a representation of the original data as the input for an SVM classifier. For NeuroSVM, as we mentioned in subsection II-B, we use trained MLPs that have similar network parameters and details to 2layer-MLP in the following table.

B. ATTACK SCENARIO

The proposed Deep K-TSVM based side-channel attacks are detailed in the Algorithm. 2. First, it captures the input power traces and the corresponding plaintexts from a copy version of the target device to create a template for attacking the target device. Finally, the model predicts the rank of the correct candidate key. The main idea of its algorithm is defined in two-step; The first step is to learn the deep K-TSVM with Algorithm. 1 and make predictions for each new attack trace. In the second step, we collect all probabilities of each possible value in the ID classification model as a list of outputs and sort the order of elements in the output list. The key rank is the index of the targeted key byte inside the list. We also define a certain threshold of $N^* \leq 10$ for all key guesses in computing key rank, such that the number of traces n and the size of hyperparameters θ are minimum:

$min\{n, \theta : N^* \leq 10\}$ where $n, \theta \geq 1$.

Algorithm 2 Deep K-TSVM SCA Attack

```
Input: Pl: Plaintext, key: 16-byte Key, tr: traces
Output: keyRank: Average "key rank" of one byte
Initialize maxepoch, SOP
for epoch= 1: maxepoch do
    Split tr and Pl into profile ($tr_p$ and $Pl_p$) and attack ($tr_a$ and $Pl_a$) subsets
    train Deep K-TSVM with Algorithm1
    for num=0: size($Pl_a$) do
        makes prediction for $tr_a[num]$
        for k=0: 255 do
            L=SBox($Pl_a[num] \oplus k$);
            add probability prediction of class L to $SOP[k]$
            store the key rank
        end for
    end for
compute average key rank of all epochs.
```

In Algorithm. 2, the SOP represents the sum of probabilities. The symbol $\oplus$ denotes xor. According to this, we try to recover the third byte of the AES key. However, the other remaining bytes can be attacked in the same way to retrieve 16 bytes of the key. We show in the experimental result (section V) that our model outperforms other state-of-the-art techniques with a key rate of $\geq 99\%$. The accuracy for the testing set is nearly similar to that of the training set, $\geq 99\%$ which proves no over-fitting in our model.

V. EXPERIMENTAL RESULTS

In the following, we present results from Deep K-TSVM against masked and unmasked hardware implementation of AES-128. We also test the effect of pre-training on the performance of the proposed model using RBM. We examine the architecture of our proposed model with different hyperparameters. However, the activation function, optimizer, and the number of epochs in the proposed model structure are obtained using the grid search optimization algorithm. Our model outperforms recent works mentioned in section II such as [17] which require 15,000 profiling traces in their MLP model and 20,000 profiling traces in their CNN model to reach a key rank lower than 20. We also compare our proposed model with four different models. The first model is an MLP with and without PCA, and the second model is a 5-layer convolutional neural network, NeuroSVM is the third model, and the fourth model is CNN-SVM. Table. 3 is giving

TABLE 3. Network settings.

Model	Network details
MLP	• Input layer: the number of samples in the processed trace • First hidden layer: 100 neurons • Second hidden layer: 50 neurons • Output layer: 256 neurons
ConvNet	• Input layer: the number of samples in the processed trace • Convolution layer: * number of filters = 8 * filter length = 32 • Max Pooling layer with a Pooling size: 2 • Dropout • Convolution layer: * number of filters= 8 * filter length = 16 • Max Pooling layer with a Pooling size: 2 • Dropout • Convolution layer * number of filters= 8 * filter length = 8 • Max Pooling layer with a Pooling size: 2 • Dropout • Convolution layer * number of filters= 8 * filter length = 8 • Max Pooling layer with a Pooling size: 2 • Dropout • Convolution layer * number of filters= 8 * filter length = 8 • Dropout • Input Dense layer: 512 neurons • Output Dense layer: 256 neurons
Deep K-TSVM	• Input layer: the the number of samples in the processed trace • DKTSVM-1: 100 neurons • DKTSVM-2: 64 neurons • Output layer: 256 neurons

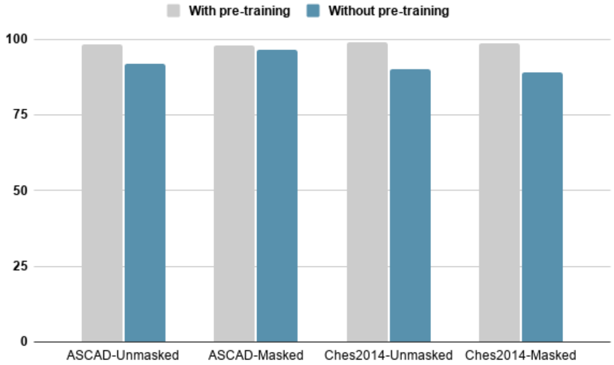


FIGURE 7. Effect of pre-training on the deep K-TSVM performance.

details of each network in our experiment. All models are trained and evaluated using two different datasets, the first one is ASCAD dataset traces, and the send one is TeSCASE dataset.

Although accuracy is not a common method for evaluating the performance of deep learning based side-channel attack techniques, in addition to the key rank average, we have examined this criterion for the proposed model to evaluate the robustness of the deep model with some previous work. Table. 4 reports the attack accuracy of our proposed method and its competitors, multilayer perceptron [18] and 5-layer CNN [35], NeuroSVM and CNN-SVM based side-channel attacks. The bold values show that our proposed model performs better than MLP, 5-layer CNN, NeuroSVM, and CNN-SVM.

TABLE 4. Accuracy (%) of our deep K-TSVM and its competitors on two different datasets.

Datasets	Accuracy		
	Model	**Unmasked**	**Masked**
ASCAD dataset	MLP with PCA	0.0045	0.0037
	5-Layer CNN	0.7136	0.5491
	MLP without PCA	0.0050	0.0061
	NeuroSVM	0.0047	0.0046
	CNN-SVM	0.6703	0.6741
	Deep K-TSVM	**0.9938**	**0.9914**
TeSCASE dataset	MLP with PCA	0.0034	0.0030
	5-layer CNN	0.9126	0.8994
	MLP without PCA	0.0041	0.0037
	NeuroSVM	0.0048	0.0057
	CNN-SVM	0.7215	0.7094
	Deep K-TSVM	**0.9983**	**0.9966**

From Table. 4, we can notice the following:

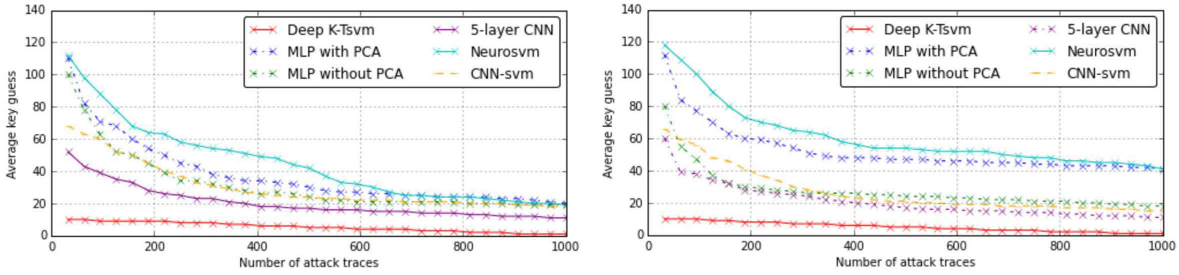
- Our model outperforms its competitors when applied on both datasets, however, the accuracy for 5-layer convolutional neural network on masked TeSCASE dataset has reached a high value of (91.26%) which is still less than the accuracy of our Deep K-TSVM (99.83%).
- MLP with PCA on masked TeSCASE dataset has the lowest accuracy among the other models.

We also investigated the effect of pre-training on the accuracy of our model, which is shown in Figure. 7. It can be concluded that the proposed model performs better when applied to two datasets using the pre-training method provided by RBM.

A. PERFORMANCE ANALYSIS OF UNMASKED AES-128

Figure. 8 shows the correct key rank according to the number of traces for unmasked TeSCASE and ASCAD datasets when attacking the third byte of the key with five different models. From these results, we can conclude the following:

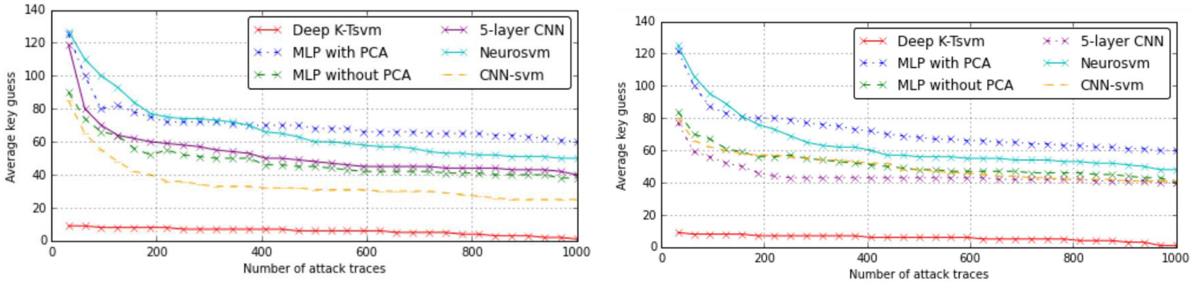
- The Deep K-TSVM even with a single trace outperforms the four other models since it has the lowest key rank (better performance). The 5-layer CNN performs better than the other ones except for our proposed model. 5-layer CNN requires about 200 power traces to reach a stable key guess while the CNN-SVM requires at least 400 attack traces to reach a stable key guess, however, it has a better performance than MLP with the PCA model and the NeuroSVM.



(a) Key guess for unmasked TeSCASE dataset

(b) Key guess for unmasked ASCAD dataset

FIGURE 8. **"Average key rank" for the third key byte of unmasked AES.**



(a) Key guess for masked TeSCASE dataset

(b) Key guess for masked ASCAD dataset

FIGURE 9. **"Average key rank" for the third key byte of masked AES.**

- The MLP with PCA performs worse compared to MLP without model on both datasets. This is because the PCA may remove some data components which are informative for linear classification representation, and negatively impact the accuracy of the non-linear profiling model of the MLP network.

B. PERFORMANCE ANALYSIS OF MASKED AES-128 DATASETS

We have also evaluated our model over protected hardware implementation, and apply the proposed model to the AES-128 dataset with masking countermeasures. The mask schemes for ASCAD and TeSCASE datasets are described in [16] and [36] respectively. Figure. 9 shows the average key rank for our proposed model and its four other competitors, multi-layer perceptron [18] and 5-layer CNN [35], NeuroSVM, and CNN-SVM based side-channel attacks according to the number of traces.

From Figure. 9(a), it can be concluded that our model performs better than other neural networks. The5-layer CNN performs worse at attacking protected datasets than attacking unmasked AES-128, which means the masking techniques make it hard to retrieve the key.

Figure. 9(b) shows the result of attacking to masked AES implementation on ASCAD dataset [16]. According to the results presented in this section, it is obviously illustrated that our Deep K-TSVM performs much better than its four different competitors.

VI. CONCLUSION

We proposed a new deep architecture for profiled side-channel attacks which includes two main modules, feature extraction, and classification modules. Our proposed Deep k-TSVM uses a deep architecture of RBM to map datasets (power traces) and learn the pattern of inputs, and it uses TSVM as the classifier. We trained our deep neural network in two different steps. First, we pre-trained the parameters with RBM by the captured power traces from the copy version of the victim device, and in the second step, we fine-tuned all parameters using gradient descent. Our Deep K-TSVM is a better attack compared to the state-of-the-art profiling attacks such as MLP and convolutional neural network-based side-channel attacks. Finally, we showed that our model with a single power trace of the AES-128 is able to recover the secret key with a success rate of $\geq$ 99% and a key rank of $\leq$ 10 on two different datasets, ASCAD, and TeSCASE.

REFERENCES

[1] L. Batina, S. Bhasin, D. Jap, and S. Picek, "Reverse engineering of neural network architectures through electromagnetic side channel," in *Proc. 28th USENIX Secur. Symp. (USENIX Secur.)*, 2019, pp. 515–532.

[2] B. Timon, "Non-profiled deep learning-based side-channel attacks with sensitivity analysis," in *IACR Transactions on Cryptographic Hardware and Embedded Systems*. 2019, pp. 107–131.

[3] S. Picek, A. Heuser, and S. Guilley, "Profiling side-channel analysis in the restricted attacker framework," IACR Cryptol. ePrint Arch., Tech. Rep. 168, 2019.

[4] S. Ghandali, S. Ghandali, and S. Tehranipoor, "Profiled power-analysis attacks by an efficient architectural extension of a CNN implementation," in *Proc. 22nd Int. Symp. Qual. Electron. Design (ISQED)*, Apr. 2021, pp. 395–400.

[5] B. Hettwer, S. Gehrer, and T. Güneysu, "Profiled power analysis attacks using convolutional neural networks with domain knowledge," in *Proc. Int. Conf. Sel. Areas Cryptogr.* Springer, 2018, pp. 479–498.

[6] E. Cagli, C. Dumas, and E. Prouff, "Convolutional neural networks with data augmentation against jitter-based countermeasures," in *Proc. Int. Conf. Cryptograph. Hardw. Embedded Syst.* Springer, 2017, pp. 45–68.

[7] J. Kim, S. Picek, A. Heuser, S. Bhasin, and A. Hanjalic, "Make some noise. unleashing the power of convolutional neural networks for profiled side-channel analysis," in *IACR Transactions on Cryptographic Hardware and Embedded Systems*. 2019, pp. 148–179.

[8] M. Neve, J.-P. Seifert, and Z. Wang, "A refined look at Bernstein's AES side-channel analysis," in *Proc. ACM Symp. Inf., Comput. Commun. Secur. (ASIACCS)*, 2006, p. 369.

[9] C. Luo, Y. Fei, P. Luo, S. Mukherjee, and D. Kaeli, "Side-channel power analysis of a GPU AES implementation," in *Proc. 33rd IEEE Int. Conf. Comput. Design (ICCD)*, Oct. 2015, pp. 281–288.

[10] A.-T. Hoang, N. Hanley, and M. O'Neill, "Plaintext: A missing feature for enhancing the power of deep learning in side-channel analysis?" in *IACR Transactions on Cryptographic Hardware and Embedded Systems*. 2020, pp. 49–85.

[11] G. Zaid, L. Bossuet, A. Habrard, and A. Venelli, "Methodology for efficient CNN architectures in profiling attacks," in *IACR Transactions on Cryptographic Hardware and Embedded Systems*. 2020, vol. 2020, no. 1, pp. 1–36.

[12] T. Güneysu and H. Handschuh, *Cryptographic Hardware and Embedded Systems–CHES 2015: 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, vol. 9293. Springer, Sep. 2015.

[13] V. Banciu, E. Oswald, and C. Whitnall, "Reliable information extraction for single trace attacks," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, 2015, pp. 133–138.

[14] L. Lerman, R. Poussier, O. Markowitch, and F.-X. Standaert, "Template attacks versus machine learning revisited and the curse of dimensionality in side-channel analysis: Extended version," *J. Cryptograph. Eng.*, vol. 8, no. 4, pp. 301–313, 2018.

[15] Z. Zeng, D. Gu, J. Liu, and Z. Guo, "An improved side-channel attack based on support vector machine," in *Proc. 10th Int. Conf. Comput. Intell. Secur.*, Nov. 2014, pp. 676–680.

[16] R. Benadjila, E. Prouff, R. Strullu, E. Cagli, and C. Dumas, "Study of deep learning techniques for side-channel analysis and introduction to ASCAD database," ANSSI, France & CEA, LETI, MINATEC Campus, France. Online verfügbar Unter, Tech. Rep. 22, 2018. [Online]. Available: https://eprint.iacr.org/2018/053.pdfZuletztgeprüftam

[17] S. Picek, A. Heuser, G. Perin, and S. Guilley, "Profiling side-channel analysis in the efficient attacker framework," Cryptology ePrint Archive, Tech. Rep. 2019/168, 2019. [Online]. Available: https://eprint.iacr.org

[18] H. Maghrebi, T. Portigliatti, and E. Prouff, "Breaking cryptographic implementations using deep learning techniques," in *Proc. Int. Conf. Secur., Privacy, Appl. Cryptogr. Eng.* Springer, 2016, pp. 3–26.

[19] S. Picek, A. Heuser, A. Jovic, S. A. Ludwig, S. Guilley, D. Jakobovic, and N. Mentens, "Side-channel analysis and machine learning: A practical perspective," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 4095–4102.

[20] L. Lerman, G. Bontempi, and O. Markowitch, "A machine learning approach against a masked AES," *J. Cryptograph. Eng.*, vol. 5, no. 2, pp. 123–139, Jun. 2015.

[21] G. Hospodar, B. Gierlichs, E. De Mulder, I. Verbauwhede, and J. Vandewalle, "Machine learning in side-channel analysis: A first study," *J. Cryptograph. Eng.*, vol. 1, no. 4, p. 293, 2011.

[22] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," *Emerg. Artif. Intell. Appl. Comput. Eng.*, vol. 160, no. 1, pp. 3–24, 2007.

[23] B. Schölkopf, A. J. Smola, and F. Bach, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2002.

[24] A. Rocha and S. K. Goldenstein, "Multiclass from binary: Expanding one-versus-all, one-versus-one and ecoc-based approaches," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 289–302, Feb. 2014.

[25] O. Choudary and M. G. Kuhn, "Efficient template attacks," in *Proc. Int. Conf. Smart Card Res. Adv. Appl.* Springer, 2013, pp. 253–270.

[26] J.-W. Chou, M.-H. Chu, Y.-L. Tsai, Y. Jin, C.-M. Cheng, and S.-D. Lin, "An unsupervised learning model to perform side channel attack," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*. Springer, 2013, pp. 414–425.

[27] S. Ding, J. Yu, B. Qi, and H. Huang, "An overview on twin support vector machines," *Artif. Intell. Rev.*, vol. 42, no. 2, pp. 245–252, Aug. 2014.

[28] S. Ding, X. Zhang, Y. An, and X. Yu, "Weighted linear loss multiple birth support vector machine based on information granulation for multi-class classification," *Pattern Recognit.*, vol. 67, pp. 32–46, Jul. 2017.

[29] E. Saeedi and Y. Kong, "Side channel information analysis based on machine learning," in *Proc. 8th Int. Conf. Signal Process. Commun. Syst. (ICSPCS)*, Dec. 2014, pp. 1–7.

[30] Y. Cho and L. K. Saul, "Large-margin classification in infinite neural networks," *Neural Comput.*, vol. 22, no. 10, pp. 2678–2697, Oct. 2010.

[31] P. Ghanty, S. Paul, and N. R. Pal, "NEUROSVM: An architecture to reduce the effect of the choice of kernel on the performance of SVM," *J. Mach. Learn. Res.*, vol. 10, no. 3, pp. 591–622, 2009.

[32] R. Tubbing, "An analysis of deep learning based profiled side-channel attacks: Custom deep learning layer, CNN hyperparameters for countermeasures, and portability settings," Tech. Rep., 2019.

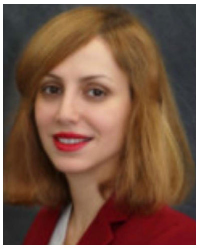
[33] A. A. Ding, L. Zhang, Y. Fei, and P. Luo, "A statistical model for higher order DPA on masked devices," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Springer, 2014, pp. 147–169.

[34] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, 2002.

[35] S. Picek, A. Heuser, A. Jovic, S. Bhasin, and F. Regazzoni, "The curse of class imbalance and conflicting metrics with machine learning for side-channel evaluations," in *IACR Transactions on Cryptographic Hardware and Embedded Systems*. 2019, vol. 2019, no. 1, pp. 1–29.

[36] M.-L. Akkar and C. Giraud, "An implementation of DES and AES, secure against some attacks," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Springer, 2001, pp. 309–318.

SOROOR GHANDALI received the M.Sc. degree in mathematical statistics from the School of Mathematics, Statistics and Computer Science, University of Tehran, in 2019. She is currently pursuing the Ph.D. degree in hardware security-based deep learning algorithms with the Department of Electrical and Computer Engineering, Santa Clara University, under the supervision of Prof. Sara Tehranipoor. Her research interests include machine learning, deep learning, side-channel analysis, and FPGA application.



SAMANEH GHANDALI received the Ph.D. degree in computer engineering from the University of Massachusetts, Amherst, in 2019, under the supervision of Prof. Christof Paar. She was a Graduate Research Assistant in computer engineering at the University of Tehran, Tehran, Iran. She is currently a Hardware Security Engineer with Google. Her research interests include cryptography, physical security of embedded systems, hardware Trojans, side-channel analysis attacks, fault injection attacks, and the corresponding countermeasures. She has served as the Technical Program Chair for the 2021 Design, Automation and Test in Europe Conference(DATE) and the Reviewer for the 2020 VLSI Symposia and the IET Computers and Digital Techniques.

SARA TEHRANIPOOR is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Santa Clara University, Santa Clara, CA, USA. Her research interests include hardware security and machine learning, the Internet-of-Things (IOT) security, and embedded systems. She received the Best Technical Paper Award at the 30th International Conference on VLSI Design (VLSID), in 2017. She is currently serving as a Section Editor for *Discover IoT* Journal and an Associate Editor for *IEEE Consumer Electronics Magazine*.