IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 13, 2020, accepted September 24, 2020, date of publication October 6, 2020, date of current version October 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3029206

Improved Hybrid Approach for Side-Channel Analysis Using Efficient Convolutional Neural Network and Dimensionality Reduction

NAILA MUKHTAR[1], (Member, IEEE), APOSTOLOS P. FOURNARIS[2], (Member, IEEE), TARIQ M. KHAN[3], (Member, IEEE), CHARIS DIMOPOULOS[2,4], (Member, IEEE), AND YINAN KONG[1], (Member, IEEE)

[1]School of Engineering, Macquarie University, Sydney, NSW 2109, Australia
[2]Industrial Systems Institute, Research Center ATHENA, 26504 Marousi, Greece
[3]School of Information Technology, Deakin University at Geelong Waurn Ponds Campus, Geelong, VIC 3216, Australia
[4]Department of Electrical and Computer Engineering, University of Patras, 26500 Patras, Greece

Corresponding author: Naila Mukhtar (naila.mukhtar@students.mq.edu.au)

This work was supported in part by the Macquarie University Research Excellence Scholarship, and in part by the European Union's Horizon 2020 Research and Innovation Programme Cybersecurity Competence for Research and Innovation (CONCORDIA) under Grant 830927. The work of Apostolos Fournaris was supported by the European Union's Horizon 2020 Research and Innovation Programme Cross-Layer Cognitive Optimization Tools and Methods for the Lifecycle Support of Dependable CPSoS (CPSoSaware) under Grant 871738.

ABSTRACT Deep learning-based side channel attacks are burgeoning due to their better efficiency and performance, suppressing the traditional side-channel analysis. To launch the successful attack on a particular public key cryptographic (PKC) algorithm, a large number of samples per trace might need to be acquired to capture all the minor useful details from the leakage information, which increases the number of features per instance. The decreased instance-feature ratio increases the computational complexity of the deep learning-based attacks, limiting the attack efficiency. Moreover, data class imbalance can be a hindrance in accurate model training, leading to an accuracy paradox. We propose an efficient Convolutional Neural Network (CNN) based approach in which the dimensionality of the large leakage dataset is reduced, and then the data is processed using the proposed CNN based model. In the proposed model, the optimal number of convolutional blocks is used to build powerful features extractors within the cost limit. We have also analyzed and presented the impact of using the Synthetic Minority Over-sampling Technique (SMOTE) on the proposed model performance. We propose that a data-balancing step should be mandatory for analysis in the side channel attack scenario. We have also provided a performance-based comparative analysis between proposed and existing deep learning models for unprotected and protected Elliptic curve (ECC) Montgomery Power ladder implementations. The reduced network complexity, together with an improved attack efficiency, promote the proposed approach to be effectively used for side-channel attacks.

INDEX TERMS Side-channel attacks, machine learning analysis, elliptic curve security, embedded system security.

I. INTRODUCTION

Embedded device security in the internet of things (IoT) based systems is of paramount importance, and security measures should be integrated at the design level [1]. Public Key (asymmetric key) algorithms like Elliptic Curve Cryptography (ECC) are recommended for such resource-constraint environments [2]–[6]. These algorithms are theoretically and mathematically secure, but their weak implementations can lead to security breaches through side channel attacks. Side channel attacks can exploit the secure algorithm implementations by analyzing the side-channel leakages, including power signals, electromagnetic emanations, timing information, etc. [7]–[10]. Traditionally, profiled-based template attacks are considered one of the strongest side-channel practical attacks. In these attacks, the adversary has access to the open copy of the target device [11]. Successful practical template side-channel attack designs have been proposed over the past decade [12]. Machine learning (ML) analysis has been proposed as a mechanism to improve the side-channel attacks

The associate editor coordinating the review of this manuscript and approving it for publication was Mehul S. Raval.

due to the similarities between template attacks and machine learning-based data analysis [13]–[15].

Elliptic Curve Cryptography (ECC) based public-key algorithms are the preferred choice for authentication, digital signatures, certificates etc. in the resource-constraint environments due to their efficient processing and small key size [2]–[6]. ECC algorithms are mathematically secure but their weak implementations can introduce many exploitable side-channel attack vulnerabilities. Machine Learning attacks have been extensively performed and studied for side channel leakages from symmetric key algorithms (for example Advanced Encryption Algorithm, AES). However, very limited analysis exists for the asymmetric key-based public-key algorithms like RSA and Elliptic Curve Cryptography (ECC) [16], [17]. Some of these attacks use simple machine learning algorithms (including Random Forest or Support Vector Machine) [17]; however, deep learning (DL) techniques seem more promising for side channel analysis due to the noisy nature of the side channel leakage signals.

Additionally, in most of the scenarios side channel leakages are misaligned, and require pre-processing to exploit the leakages for recovering the secret information, which can be a tedious and possibly discouraging task for an attacker. Convolutional Neural Network (ConvNet) based deep learning technique constitutes an ideal candidate for eliminating the leakage traces' excessive noise. More specifically, the convolutional layer in ConvNet reduces the leakage trace samples by extracting and learning from only essential features by assigning weights and eliminating noise. Cagli *et al.* have proposed to use ConvNets for side-channel attacks and have shown successful results on data, for symmetric algorithm implementations, without requiring any pre-processing or alignment [18]. Kim *et al.* have shown the impact of adding noise to existing samples, which helps recover the secret information with reduced samples [19].

Furthermore, selecting important features or points of interest is crucial while launching side-channel attacks. Traditionally, various methods are proposed to select POIs [20]. Recently, Picek *et al.* and Mukhtar *et al.* have proposed feature engineering techniques to achieve optimal results by analyzing the impact of using the feature engineering techniques on side-channel leakages and processing them further by using machine learning classifiers [21], [22].

However, using ConvNets for side channel analysis still suffers from several problems. Firstly, ConvNets require a huge amount of traces/instances to extract sensitive information from the side channel leakages. This requirement becomes more exacting in complex cryptography algorithm implementations (like public-key cryptography algorithms), where the high sampling frequency is needed to ensure that enough leakage information is acquired. Hence, generating an enormous leakage dataset that is processed further to recover the secret information by utilizing the deep learning classifiers' pattern recognition capability, as proposed by various studies without applying any pre-processing or alignment on the data [19], [23]. This removes the need to use any pre-processing at a considerable computational complexity cost. The huge datasets lead to increased computational complexity and hardware resource usage of the deep learning based side channel attacks which, in turn, leads to substantial time to train the model and launch the attack. In several security scenarios where the secret information's life span is important, this delay in retrieving the secret might be unacceptable. One possible solution in such scenarios is to reduce the input dataset size by using feature extraction techniques. In non Machine Learning (classical) side channel analysis, principal component analysis (PCA) has been proposed as a pre-processing step to select the important features [24]. For machine learning-based side-channel attacks, Golder *et al.* have presented results for using PCA as a pre-processing step for classification using ML on symmetric ciphers [25]. However, all the existing machine learning-based side channel analysis with PCA pre-processing are applied to symmetric cipher datasets, and no substantial work has been done on public-key cryptosystems. Moreover, the number of samples (or features) per instance is generally small (ranging from 400 - 6000) in the existing studies [19], [23]. However, in the presented case of an asymmetric cipher, the number of samples per trace/instance is very large (33000 precisely).

Secondly, another aspect that can create problem while training side-channel leakages with the deep learning algorithms is the amount of data instances/traces per target class. If class data is highly imbalanced, it can hinder accurate modeling by giving rise to an accuracy paradox. Traditionally, there are data-level and algorithm-level data balancing techniques that can balance the target classes and improve the trained model performance [26], [27]. Picek *et al.* have recommended using the Synthetic Minority Over-sampling Technique (SMOTE) to balance data, based on the experimental findings for symmetric-key algorithm leakage information [28]. However, there is no analysis using SMOTE for side channel leakages of the public-key cryptography algorithm implementations.

Contributions: In this paper, we provide solutions to the above problems and offer a thorough study for performing ConvNet based deep learning side channel attacks on Elliptic Curve Cryptography scheme implementations, efficiently. We propose a hybrid deep learning-based attack methodology and an analysis framework to improve the side-channel attacks on imbalanced leakage datasets by using the combination of dimensionality reduction and class imbalance techniques along with the proposed simple Convnet model. The optimal number of convolutional blocks are used to build the powerful features extractor within the cost limit. The proposed efficient ConvNet-based approach has been evaluated for both protected and unprotected ECC scalar multiplication Montgomery Power Ladder (MPL) implementations. High sampling frequency was used during the data collection process to fully capture the side channel leakage of the public-key ECC implementations. Thus, 33000 samples per trace for analysis were collected, resulting in a massive dataset with a low instance-feature ratio. To handle the high

computational complexity of the attack due to this massive dataset, we proposed a time-efficient model for analyzing public-key cryptographic (PKC) schemes (ECC), based on the dimensionality reduction. Moreover, in line with the findings for symmetric ciphers, to solve the imbalance problem in traces per class, we have analyzed SMOTE's impact on the public-key ECC implementations using our proposed attack architecture. Based on the findings, it is determined that the data balancing should be included as a mandatory step for a reliable attack model for public-key cryptosystem. Our proposed method enables the network to train much faster with better performance than the existing state-of-the-art methods, as shown by our performed comparative analysis between the proposed and other traditional existing models.

The rest of the paper is organized as follows. Section II provides some background information and briefs the techniques and models used in this study. Section III explains the implementation approach and introduced countermeasures of the PKC algorithm under analysis. Section IV describes the proposed methodology and analysis framework. Section V, explains the experimental setup. Section VI presents the results and discussions on both (protected and unprotected) ECC dataset using proposed ConvNet architecture with PCA and SMOTE. Section VII concludes the paper.

II. BACKGROUND AND PRELIMINARIES

Assuming P is a point on the Elliptic Curve $E(F)$ defined over a finite field[1] F then this point is characterized by its coefficients x, y i.e. $P : (x, y)$ where $x, y \in F$. Scalar multiplication (SM) i.e., $e \cdot P$, where e is an integer, is the main operation used in Elliptic Curve Cryptography, and it has been widely studied for its side channel attack resistance. SM relies on the repetition of many point addition and point doubling operations that themselves are implemented using finite field arithmetic operation like modular addition, subtraction, inversion, and multiplication (in $GF(p)$ or $GF(2^k)$). Since modular inversion is a computationally complex operation, most designers exchange it with several modular multiplications and addition/subtractions by transforming the Elliptic Curve and its points from the affine coordinate domain to the projective coordinate domain [29]. In the projective coordinates domain, each point is represented by three coordinates i.e $P : (X : Y : Z)$ where $X, Y, Z \in F$.

A. SIDE CHANNEL ATTACKS

Traditional algorithms implementing SM (e.g., double-and-add algorithm) have serious imbalances associated with the value of each bit of the secret scalar (e); thus strong association of SM computation can be made with the secret scalar been processed. There is a broad range of SM focused SCA attacks both simple and advanced or horizontal and vertical [30], [31] and [32].

[1]In cryptography, prime finite fields, GF(p), and Binary extension fields, $GF(2^k)$, are used

Simple SCAs can be easily mounted in the double-and-add algorithmic approach followed in SM and are typically horizontal type of attacks i.e., they can be mounted using a single leakage trace that is processed in time. Such simple SCAs can be easily countered by using highly regular SM algorithms i.e. algorithms in which each round's operations are unrelated to the scalar bit that they are processing (e.g Double and always Add algorithm or Montgomery Power Ladder (MPL) [33]). However, there are a series of SCAs, known as comparative SCAs (focused initially on Power attacks (PAs) but also extended to Electromagnetic emission (EM) attack) that still manage to overcome the above regularity by manipulating the base point input of the SM (doubling attack (collision-based attack) [34] and its variants [35] or the chosen plain text attack in [36] (also known as 2-Torsion Attack (2-TorA) for ECC).

There are, however, more advanced attacks (advanced SCAs) on EC SM both of vertical and horizontal nature (where the attack needs many traces or a single trace, respectively). Differential Attacks (DSCA), originally proposed by Kocher in [7] for power consumption leakage, is the most widely known such attack. These attacks appear in many variations based on the used hypothesis distinguishers, leading to sophisticated DSCAs like Correlation SCA (requiring less traces to reveal the secret than DSCA) [37] and collision correlation attack [38]–[40]. These attacks are possible even when a single trace is available (horizontal attacks) [41] or the Horizontal Collision Correlation attack (HCCA) [31], [32].

Whitnall *et al.* [42] suggest that there are considerably more potent SCAs than DSCAs, known as profiling attacks. Such attacks rely on a profiling phase on the device under attack. In the profiling phase, an attacker identifies the leaking operation (Point of Interest, PoI) and produces all possible different states of this operation by feeding the device with all possible secret key value inputs (e.g., one byte or one bit). These states are statistically analyzed to create an identifiable profile for each secret key value. An attack phase then follows where the attacker targets a device with an unknown secret scalar and collects PoI leakage traces for various inputs using the same trace collection mechanism and parameters as in the previous phase. Using the profile and some discriminator, the attacker tries to identify the appropriate leakage trace from the profile that has a high probability of matching the unknown secret leakages and retrieves the secret. The most common type of such profiling attacks are template and online template attacks (TA) [12], [43], [44].

The concept of profiling a device to create a leakage model based on labeled leakage traces has been explored further by researchers using ML techniques for creating a profile. Using ML, the attacker does not need to create a perfect leakage model but rather lets an ML algorithm be trained with a non-exhaustive series of leakage traces (that can be associated/labeled to some, instead of all, secret block values). As the leakage noise increases (possibly also due to masking or hiding countermeasures), the ML profiling approach tents to provide better results as compared to traditional attacks [14].

B. MONTGOMERY LADDER ALGORITHM AND COUNTERMEASURES

Given the above analysis, an EC SM implementation should include appropriate countermeasures to be protected against a broad range of attacks. Profiling SCAs, as various researchers highlighted, [12], [44], [45], can overcome several existing countermeasures, indicating the need for a more sophisticated randomization throughout the whole computation flow of the scalar multiplication algorithm.

One of the most popular such variations of secure scalar multiplication algorithms is the Montgomery Power Ladder (MPL) algorithm. As seen in Algorithm 1, it has strong regularity in each round of operations (step 2 of Algorithm 1) and minimal interference from the secret scalar bit value. This MPL regularity is manifested by the constant number of identical point operations performed in each scalar round regardless of the corresponding secret scalar bit [33] and prohibits an attacker from performing simple horizontal and vertical SCAs. Apart from that, MPL favors parallelism since step 2a or step 2b operations (point addition ($R_0 + R_1$) and point doubling ($2R_0$ or $2R_1$)) can be performed in parallel. Apart from the performance benefit that such a feature offers, it can potentially scramble side channel signals to identify each one of those two operations that can become difficult for an attacker. However, the MPL algorithm still leaks some information about the scalar bit since the outcome of point addition and point doubling in each MPL round is stored in the different storage area (eg. registers) depending on the processed scalar bit value. For example point doubling result is stored in R_0 when $e_i = 0$ and in R_1 otherwise. This subtle irregularity can be identified and exploited using profiling SCAs (e.g., template SCAs and ML SCAs) to retrieve the secret scalar e [45].

Algorithm 1 Simple SCA Resistant MPL Algorithm

Input: P : EC base point $\in EC(F)$,
$e = (e_{t-1}, e_{t-2}, \ldots e_0) \in GF(2^k)$
Output: $e \cdot P$
1. $R_0 = \mathcal{O}, R_1 = P$
2. **For** $i = t - 1$ **to** 0
   If $(e_i = 0)$ then
    (a) $R_1 = R_0 + R_1, R_0 = 2 \cdot R_0$
   else
    (b) $R_0 = R_0 + R_1, R_1 = 2 \cdot R_1$
   end if
3. **Return** R_0

To remedy the MPL SCA problems, SCA countermeasures fitting into two different categories can be used, leakage hiding or leakage masking [10], [46]. In the hiding approach, appropriate measures are included in SM computation flow so that the leakage of point addition/doubling operation or storage area is made indistinguishable from random noise. To achieve that algorithmically within the EC SM we can introduce dummy operations in the computation flow or modify the algorithm, so traces of one MPL operation are very similar from traces of another operation (or their result storage process). Since MPL favors parallelism that enables a designer to merge operations in time (more than one point operation is processed in each time frame), hiding can be achieved by scrabbling the operation traces at the same time frame.

Masking aims at disassociating the sensitive information from the leakage trace. This approach relies on some form of randomization (additive or multiplicative) on the sensitive information associated with a leaky MPL point or storage operation. As originally proposed by Coron in [47] and later extended and adapted by various other researchers [46] EC SM masking techniques aim to randomize the EC multiplication secret scalar (e), the input point P (base point blinding), or the input point's projective coordinates (X, Y, Z). The most easily applicable are the first and the last (scalar blinding and projective coordinate blinding) since the point blinding technique requires the introduction and storage of a random point in each scalar multiplication [48].

C. DATA IMBALANCE TECHNIQUE-SMOTE

Data imbalance, meaning the number of instances of each class is not equal, leads to misclassification and can give rise to an accuracy paradox. In application domains of machine learning, there are various techniques to handle imbalance classes for accurate modeling. One of the techniques to address the class imbalance issue is to modify the input training data distribution to decrease the imbalance ratio of the target classes. There is no guarantee to have an equal amount of leakage bit information in side-channel leakage data, especially in multi-class classification problems. In this research, we have studied SMOTE's effect on improving the secret data recovery attack efficiency.

Generally, for imbalance datasets, under-sampling and over-sampling techniques are used. In under-sampling, majority class data instances are removed to bring it to the minority class level. In over-sampling, more samples are added for the minority class instances. Under-sampling discards data, which might contain important information required for accurate classification. On the other hand, over-sampling increases computation time and can cause over-fitting. To address these issues, numerous intelligent under-sampling and over-sampling techniques have been introduced to preserve sensitive information. Kubat *et al.* have presented a method for removing noise and redundant data from the majority class using one-sided selection [49]. Algorithms based on K-nearest neighbors (K-NN) classifiers are proposed to remove the majority samples based on their distance from minority samples [50]. Among all sampling techniques, the over-sampling technique of SMOTE is the most popular one. SMOTE generates artificial samples for the minority class synthetically, using minority samples and their minority neighbors [26]. Picek *et al.* presented results for SMOTE's performance on the side channel leakages from symmetric ciphers [28].

D. PRINCIPAL COMPONENT ANALYSIS

In the principal component analysis, the dimensionality of data is reduced to increase interpretability. It uses an orthogonal linear transformation to re-position the data onto a new coordinate system, and a new reduced smaller feature dataset is formed based on the existing feature space [51]. The feature that explains the maximum amount of variance is positioned at the new dataset's first location. PCA helps discard the features that capture similar information and thus aids in creating a more parsimonious model.

E. CONVOLUTIONAL NEURAL NETWORKS

CNN is a deep learning algorithm that takes input signals data and learns the differentiating aspects of the target class by assigning weights and importance. Generally, CNN consists of convolutional layers, a flatten layer, a pooling layer, and fully connected layers [52], [53]. Activation functions are used in each layer to deal with the non-linearity. The convolutional layer performs convolution on the input features, using filters/kernel to recognize the data's patterns. This filter hovers over the complete data trace from left to right, based on the set stride, and reduces the input features dimensionality by convolution. Generally, dimensionality can be reduced or stays the same depending upon the padding being used. For this layer, kernel and stride are the hyperparameters which can be tuned further to obtain a good performing model. The pooling layer is an approach to reduce the sample size by downsampling the features from the feature map by summarizing features in patched regions. The intuition of using a pooling layer is to select a dominating feature from a particular layer in a particular region. If the feature is not dominating, then the resulting value will be small and will wear out with further pooling in the next layer. Hence, it helps in reducing the computational complexity, combats over-fitting, and encourages translational invariance. It takes a filter, but instead of applying convolution, it either takes the maximum value from the feature map region or takes the average. Based on this, there are two main widely used pooling methods; max-pooling and average pooling. A combination of the convolutional layer and pooling layer forms a pair i-th layer in a neural network architecture. The number of such layers can be increased to capture the minor low-level details. Increased convolutional layers enhance the overall model's computational complexity, which takes a longer time in training. The existing proposed architectures for side-channel analysis are complex, consisting of numerous layers with a large number of filters [18], [19]. We have selected one such complex ConvNet architecture for comparison in this study. The existing architecture has been evaluated for AES leakages. However, we have tested the same network for ECC leakage data, and then we have presented results by evaluating with our proposed architecture.

III. HARDWARE DESIGN AND IMPLEMENTATIONS

As a target of the proposed deep learning side channel attacks, two EC SM hardware implementations of Binary Edwards curves (BEC) on $GF(2^k)$ has been chosen, based on the work of Fournaris *et al.* in [48]. Both implementations have the BEC intrinsic protection against SSCAs,[2] use the MPL algorithm described as Algorithm 1 and exploit the parallelism in step 2a or 2b of the algorithm in order to achieve efficiency and side channel attack resistance. In [48], point addition and point doubling operation are decomposed in their basic finite field operations, as shown in Table 1, and those operations are examined for their data dependability. Those operations that are data-independent (they do not rely on the result of some other finite field operation) are grouped in stages to be computed in parallel using some constrained number of parallel processing elements. In the architecture design of [48], three modular multiplier processing elements and three modular adder processing elements are used, operating in parallel, thus producing 11 parallel stages of grouped finite field operations (shown in Table 2).

As can be observed in Table 2, the parallel finite field operations provided in each stage are not associated with only

TABLE 1. Point operations partial results [48].

Point Addition $(X_3 : Y_3 : Z_3) = (X_1 : Y_1 : Z_1) + (X_2 : Y_2 : Z_2)$	Point Doubling $(X_{3D} : Y_{3D} : Z_{3D}) = 2(X_1 : Y_1 : Z_1)$
$A = X_1 \cdot X_2$	$DA = X_1 \cdot X_1$
$B = Y_1 \cdot Y_2$	$DC = Y_1 \cdot Y_1$
$C = Z_1 \cdot Z_2$	$DE = Z_1 \cdot Z_1$
$D = d_1 \cdot C$	$DB = DA \cdot DA$
$E = C \cdot C$	$DD = DC \cdot DC$
$F = d_1 d_1 \cdot E$	$DH = DA \cdot DE$
$G_1 = X_1 + Z_1$	$DI = DC \cdot DE$
$G_2 = X_2 + Z_2$	$DL = DE \cdot DE$
$G = G_1 \cdot G_2$	$DF = d_1 \cdot DL$
$H_1 = Y_1 + Z_1$	$DJ = DH + DI$
$H_2 = Y_2 + Z_2$	$DO = d_2 \cdot DJ$
$H = H_1 \cdot H_2$	$DM = DB + DD$
$I = A + G$	$DG = d_1 d_2 \cdot DM$
$J = B + H$	$DK = DG + DO$
$K_1 = X_1 + Y_1$	$DL_1 = DF + DJ$
$K_2 = X_2 + Y_2$	$DL_2 = DH + DD$
$K = K_1 \cdot K_2$	$DL_3 = DI + DB$
$L = d_1 \cdot K$	$DX_3 = DL_2 + DK$
$U_1 = K + I$	$DY_3 = DL_3 + DK$
$U_2 = J + C$	$DZ_3 = DL_1 + DG$
$U_3 = U_1 + U_2$	
$U_4 = L \cdot U_3$	
$U_5 = F + U_4$	
$U = C \cdot U_5$	
$V_1 = A \cdot B$	
$V_2 = G \cdot H$	
$V_3 = d_1 \cdot E$	
$V_4 = V_1 + V_2$	
$V_5 = V_3 + V_4$	
$V_6 = L \cdot V_5$	
$V_7 = D \cdot F$	
$V_8 = V_7 + V_6$	
$V = Z_3 + V_8$	
$M_1 = A + D$	
$N_1 = G + D$	
$O_1 = M_1 \cdot N_1$	
$M_2 = B + D$	
$N_2 = H + D$	
$O_2 = M_2 \cdot N_2$	
$P_1 = D \cdot O_1$	
$P_2 = D \cdot O_2$	
$X_3 = V + P_1$	
$Y_3 = V + P_2$	
$Z_3 = U$	

[2]They offer completeness and uniformity

TABLE 2. Paralleling BEC point addition and doubling $GF(2^k)$ operations.

Inputs		$(X_1 : Y_1 : Z_1)$		$(X_2 : Y_2 : Z_2)$		
Stage	**M1**	**M2**	**M3**	**Ad1**	**Ad2**	**Ad3**
1	A	B	DA	G1	G2	K1
2	G	DC	DB	H1	H2	K2
3	H	DD	DE	I	-	-
4	C	DH	DI	J	-	DM
5	V2	V1	K	DJ	DL2	DL3
6	DO	E	DG	U2	V4	U1
7	D	V3	L	DK	-	U3
8	DL	F	U4	M1	V5	N1
9	V7	V6	O1	M2	N2	U5
10	DF	O2	U	V8	DX3	DY3
11	-	P2	P1	DL1	-	V
12	rDY3	rZ3	rDX3	DZ3	Y3	X3
13	rY3	rX3	rDZ3	-	-	-
Outputs		$(X_3 : Y_3 : Z_3)$		$(X_{3D} : Y_{3D} : Z_{3D})$		
Rand Outputs		$(rX_3 : rY_3 : rZ_3)$		$(rX_{3D} : rY_{3D} : rZ_{3D})$		

- : idle r: random number

one point operation (point addition or point doubling) from Table 1 of an MPL round. This scrabbling mechanism can potentially prohibit identifying the performed point operation and/or storage since both point addition and doubling are performed in parallel using the same structural blocks (processing elements). The storage pattern (meaning, which intermediate results are stored in which register) is very similar in every round except some multiplexer units at the end of the computation, as described in [48]. This approach was designed to provide resistance against advanced SCAs and template attacks. However, by performing a Welch's t-test,[3] in [48], the authors discover that there is still non-trivial leakage of the secret scalar key during the SM computation. So, while the computation processing in each MPL round (all stages) and its storage pattern is fairly regular regardless of the secret scalar bit, there are still indications that some attack could potentially succeed in recovering the secret scalar key. In this paper, given the above remark, we use the proposed DL/ML attack methodology on an implementation (denoted as unprotected implementation) that is produced through the above-described process.

To solve the above-described leakage issue, in [48], a mechanism based on random operations is introduced in the parallel stages to mask the values stored in the implementations' registers. Using a random number generator integrated in the hardware implementation, a random value r is generated in each MPL round. This r is used in two extra stages of parallel operations that are introduced in Table 2 in order to multiplicatively mask the values of all performed finite field operations involved in an MPL round. The additional operations, colored in blue in Table 2, are following the random coordinates countermeasure approach, but instead of performing randomization once per SM, the countermeasure is expanded by re-randomizing the coordinates at every MPL round. This effectively masks/eliminates any processing leakage association between the scalar bits and the processed MPL round parallel operations. The Welch's t-test on such an implementation, denoted as protected implementation, applied in measurements of [48] indicates that indeed there is only trivial leakage of the secret key in all MPL rounds. However, although advanced SCAs may fail to retrive the secret scalar in the protected implementation (due to trivial leakage), profiling attacks (e.g., ML SCAs) may be successful since the storage leakage pattern (not assessed using TVLA) is mostly the same compared to the unprotected implementation of [48]. In this paper, we evaluate the proposed DL/ML attack methodology on this protected implementation to identify its efficiency, accuracy and to explore the limits of the TVLA test as a trusted SCA assessment approach in the presence of DL/ML attacks.

[3]The constant versus random scalar methodology was used following the Test Vector Leakage Assessment technique

IV. PROPOSED ATTACK METHODOLOGY AND EVALUATION FRAMEWORK

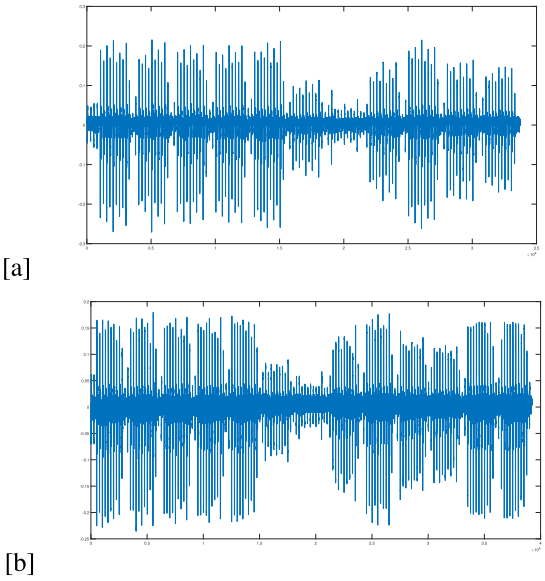
To launch a deep learning-based side-channel attack, assume the adversary is in possession of the open copy of the device and has computational and resource capacity to obtain and process the side channel leakage information. However, we assume that the adversary wants to recover the secret information in requisite attack time T_A, from the obtained leakage traces L_T. The proposed deep learning-based attack methodology is systematically divided into five steps. In step 1, the leakage data L_T is formatted and labeled to identify the target class for each trace. In step 2, the prepared dataset is processed to balance the target class instances synthetically. In step 3, the dimensionality is reduced using PCA, and in the last step, classification is performed using the proposed CNN model. Each step is further elaborated in Sec. IV-A - IV-D. However, Sec. IV-E describes the strategy followed in this study to evaluate the proposed approach.

One of the critical concerns in neural networks is over-fitting while dealing with the side-channel noisy leakages. In over-fitting, the model learns from the data so well, or we can say it learns from the noisy patterns as well, that it creates a model with high variance. The resulting model will fail to generalize on the unseen data. To handle the problem of over-fitting, we have taken specific measures at various stages of the analysis. Each measure is explained in the respective section.

A. STEP 1-DATASET PREPARATION

For analysis in this research work, both protected and unprotected implementations of the EC MPL algorithm are analyzed, as explained in II-B. For both implementations, to launch bit level machine learning based side channel attack, at first data traces T, of length S_T, are collected for each bit operation, and then each trace is labeled as target class '0' or '1', based on the processed bit during leakage collection. The formed datasets are then processed through a machine learning classifier to train the model. The trained model is finally tested on unseen data to predict the key bit used for the encryption. The resulting labeled signals are shown in Fig. 1. The trace where collected, following the approach in [54], using a PicoScope 5000D Series Oscilloscope with a sampling rate of 1GS/s that was connected through a pre-amplifier to a resistor onboard a SAKURA-X FPGA board. The description of both the datasets is given below:

- Leakage Dataset Unprotected LD_{UP} - This dataset consists of side-channel leakage traces for MPL implementation on FPGA and is not protected by any countermeasure. This dataset consists of $T = 5,000$ data traces (instances), and each instance consists of $S_T = 33750$ samples.
- Leakage Dataset Protected LD_P - This dataset consists of side-channel leakage traces for MPL implementation on FPGA, which are protected by the countermeasures described in Sec. III. This dataset consists of $T = 5,000$ data traces (instances) and each instance consist of $S_T = 39250$ samples.



[a]

[b]

FIGURE 1. Obtained raw data signals after labeling (bit 0) for (a) unprotected and (b) protected implementations.

To handle over-fitting, we have divided our datasets further into three subparts; training data, validation data, and testing data, in the ratio of 60:20:20 %, respectively. Training and validation data is used during training the model. However, test data is held back and is never shown to the model during training, which ensures that the test data's analysis produces reliable results during the testing phase.

B. STEP 2-HANDLING CLASS IMBALANCE

After obtaining the data and forming the datasets, the next step is to balance the target class instances. An imbalance dataset can lead to an accuracy paradox by misclassifying data due to the empowering majority class. For analysis of the side-channel leakage data, we propose to use the class balancing technique as a mandatory step to balance the classes before applying a machine learning classifier, for a better reliable trained model. Our presented case of bit-level attack is a binary classification problem where two class key bits '0' and '1' need to be classified. To analyze the impact of the class imbalance technique, we have generated the datasets with less number of 1's and more number of 0's. To be precise, there are 1500 and 2600, samples for 1's and 0's, respectively. After generating the dataset, SMOTE is applied. As explained in II-C, SMOTE is a synthetic oversampling technique; we have increased our samples for the under presented class, which is '1'. After applying SMOTE, both classes have an equal number of instances. The new generated samples have the same characteristics as those of the training dataset samples.

C. STEP 3-DIMENSIONALITY REDUCTION AND DATA VISUALIZATION

After handling class imbalance, we have reduced the number of features using the dimensionality reduction technique, Principal Component Analysis (PCA). PCA has been used for traditional analysis with regards to side-channel leakage data. PCA can capture and highlight the dataset's maximum variance in just a few principal components, hence, transforming the useful information by eliminating the redundant features.

In our presented case, as the number of instances (traces) is less than the number of features (no of samples per trace), so use of pre-processing or feature engineering, to reduce the number of samples/features, can aid in reducing the computational complexity and also will help in training a better-trained model. The extra features certainly contain redundant information and noise. Usually, deep learning is expected to pick up the data anomalies, but that might not always be true, especially if the ratio of instances to features is very low. In some instances, this can give rise to over-fitting, where the model learns from the noise instead of learning from the relationship between the secret information and leakage traces. Due to the noisy nature of the leakage information, for machine learning-based side-channel analysis, it is of crucial importance to select the most contributing features. Training the model with reduced feature dataset has various benefits, including reducing training complexity and accurate trained model.

PCA can achieve this goal because it tries to find a linear subspace that best fits our data. The aim is to minimize the sum of square of orthogonal distances or maximize our data's spread within low dimensional subspace. Let X be our mean subtracted data. To find subspace w such that our data have maximum spread in this subspace, we use,

$$w = \arg\max_{\|w\|=1} \left\| w^T X \right\|_2^2 \tag{1}$$

$$w = \arg\max_{\|w\|=1} w^T X X^T w \tag{2}$$

If we take the Lagrangian of w and then its derivative, we got

$$XX^T w = 4\lambda w \tag{3}$$

This ends it up with an eigenvalue problem. If we solve Eq.3 our solution w will give first principal component (largest eigenvalue). To get other eigenvalue, we need to subtract the largest value (already found) from X and find out the next largest value and so on.

Fig. 2 shows the proportion of variance due to PCA components for both protected and unprotected leakages. It can be seen that the variance of 79% and 87% is covered with 100 PCA components. The maximum variance will be covered if PCA principal components are selected beyond 100. To analyze the effect of the principal components' various sizes, we have performed analysis using the number of components from the group PCA_{CG} where $PCA_{CG} = 200, 400,$ 600, 800, 1000 and 1200 principal components. Analysis results are given in the results section.

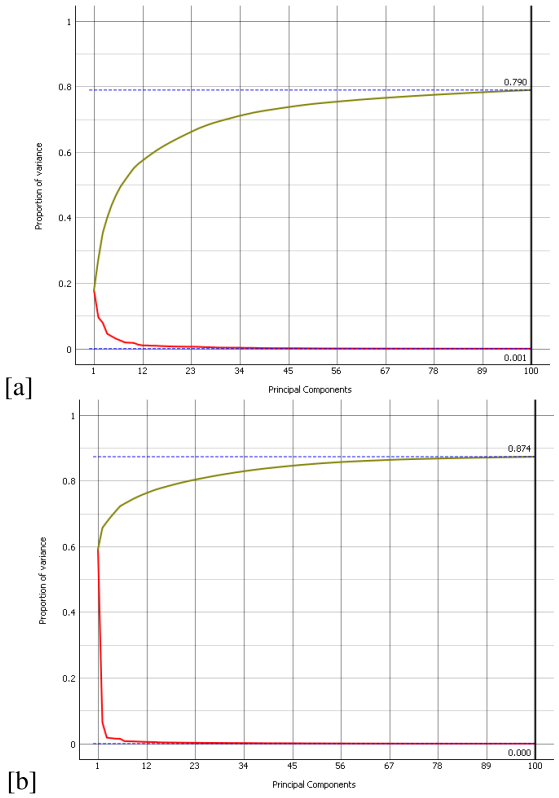


FIGURE 2. Proportion of variance for PCA components for (a) protected and (b) unprotected.

D. STEP 4-MODELING USING DEEP LEARNING CLASSIFIER

The instances of the target class '0' and '1' are balanced using the synthetic data balancing approach, SMOTE, and then the dimensionality of the data traces is reduced by applying PCA, based on the conclusions deduced from the observations in IV-C. In the last step, machine learning analysis is performed using the proposed Convolutional Neural Network (ConvNet/CNN) architecture. The proposed architecture is simple compared to the complex existing architectures and produces the same accuracy level in less time.

1) CNN PROPOSED ARCHITECTURE

The proposed simple CNN architecture is shown in Fig. 3.

There are three combinational layers in our proposed design, as shown in the summary table 3. Each combinational layer consists of a pair of convolutional layers and a pooling layer. However, in the last two combinational layers, an extra convolutional layer is added to extract more information before the pooling layer. There are five convolutional layers in the proposed architecture, consisting of 4,8,8,16, and 16 filters with specific kernel size and stride, which enables the model to distinguish the secret key bit. Kernel size and stride are varied between 4 and 8. In our proposed architecture,

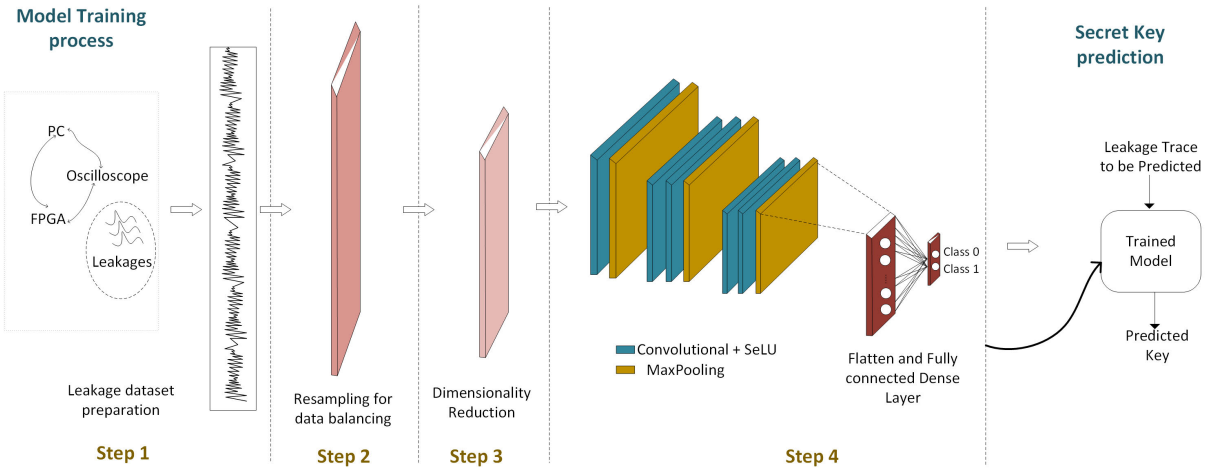


FIGURE 3. Proposed hybrid convolutional network-based system for the secret key recovery from the acquired side channel leakages. The figure shows the overall proposed system which consists of convolutional network layers, dimensionality reduction module and a class imbalance module.

TABLE 3. Proposed CNN model summary.

Layer (type)	Output Shape	Number of Parameters
$Conv1d\_1$	199x4	20
$conv1d\_2$	66x8	136
$conv1d\_3$	21x8	264
$conv1d\_4$	6x16	528
$conv1d\_5$	1x16	1040
$dense\_1$	2	34

we have used max-pooling, in which the maximum number is selected from a particular region. It has two primary hyperparameters, filter and stride. Once these hyperparameters are fixed, they do not change during the learning process. For our case, the value is set to 1-2.

Activation functions are used in convolutional layers to deal with the non-linearity of the data. We have tested various activation functions for our analysis, as listed in the table 4, and selected Scaled Exponential Linear Unit (SeLU) as it produced the best results. Because of its ability to self-normalise, SELU has shown improved performance in various classification tasks using feed-forward neural networks [55]. One of the advantages of SeLU is that its internal normalisation is faster than external normalisation, which means that the network converges faster. As the gradient problem of vanishing and exploding is impossible in SeLU, it can be a reason for its improved performance on our netwrok.

The previous max-pooling layer's output is flattened to form a column vector and is then connected to the fully connected layer. Fully Connected Layer is the final layer that takes the output of the previous flatten layer as input and then outputs N dimension vector where N is the number of the output target classes ($N = 2$ in this case), and then back-propagation is applied to each iteration of training during the epoch. After training over a few epochs, the model is able to learn from the provided features and classifies them using the softmax classification technique. We have trained our model for a longer time for 200 epochs, which provides enough batch training cycles to analyze the model performance. In our results, it is seen that the model performance becomes stable before 50 epochs.

2) NORMALIZATION AND OVER-FITTING

Having huge differences between the maximum and minimum value in the data might degrade the learning process. Normalization is performed to speed up the learning process, and the model converges quickly, which results in an accurate trained model. As mentioned before, over-fitting is one of the issues in noisy side-channel leakages. The model can learn data patterns along with the noise. Such a model performs well on the training data but fails to generalize on the test (unseen) data. Specific techniques can be used to avoid overfitting, including dropout and regularization. We have tried both and found better results with L2 regularization. It manages the weights and keeps them small in order to avoid overfitting. In addition to learning from the noise, the duplicate instances within the training dataset can also result in an overfitted biased model. To avoid this, duplicate rows are removed from the training dataset.

3) HYPERPARAMETER TUNING

There are certain hyperparameters related to each layer, which can be tuned to improve the CNN performance. Table 4 shows the lists of parameters that are tuned to select the best performing model. Grid search functionality, available in the Scikit library, is used. In grid search, exhaustive search is

TABLE 4. Parameter tuning CNN.

Parameter	Value Range
Learning Rate	[0.001,0.01,0.1, 0.5]
Epochs	[200]
Strides	4-6
Kernel Size	5-8
Pool Size	1-2
Pool Stride	1-2
Activation function	[relu,selu,elu,tanh,softplus]
Optimizer	[Adam,Nadam,RMSprop,Adamax,sgd]
Initialization Mode	[uniform,normal]
Batch Size	[32, 100]

performed, using all possible parameter combinations, and the best performing parameters are selected based on the model accuracy.

E. EVALUATION STRATEGY

In order to systematically analyze the affect of the proposed neural network based side channel attack on the leakage data, analysis is further divided into four sets, as given below.

- Analysis on unprotected implementation dataset LD_{UP} using existing model ($A1$)
- Analysis on unprotected implementation dataset LD_{UP} using proposed model using varying PCA Components sizes ($A2$)
- Analysis on protected implementation dataset LD_P using existing model ($A3$)
- Analysis on protected implementation dataset LD_P using proposed model using varying PCA Components sizes ($A4$)

For analysis set $A1$ and $A3$, the collected raw traces/ instances from FPGA implementations are processed through machine learning classifier CNN for both unprotected and protected implementations, respectively, as proposed in the existing literature. We have chosen the simplest existing CNN model for SCA. For analysis set $A2$, and $A4$, analysis is performed using our proposed model (explained in IV-D) for both unprotected and protected implementations. For these sets, data has been over-sampled using SMOTE, and then PCA is applied to change the dimensions of the data. For analysis in this study, we have tested the various number of principal components from PCA_{CG} group.

The accuracy and model training time is reported along with Receiver Operating Characteristic (ROC) curves. The outcome of the analysis will help in devising a time and resource-efficient mechanism for attacking FPGA implementations on PKC.

V. EXPERIMENTAL SETUP

For implementations of the proposed deep learning model, python platform is used along with Keras and scikit-learn libraries [56], [57]. The computation requirement of hyper parameter tuning for deep learning processing is high, so NCI (National Computational Infrastructure) Australia high-performance super-computing server has been used [58]. However, for comparative analysis stand alone system equipment with GPU GEFORCE GTX 1080 Ti, memory 32GB and CPU Intel Core i7 (@3.4GHz) processor is used.

VI. RESULTS AND DISCUSSIONS

Based on the proposed framework, results and analysis is presented in this section for both LD_{UP} and LD_P datasets. Results are presented for all four analysis sets.

A. RESULTS ON UNPROTECTED IMPLEMENTATIONS (A1 AND A2)

The results for unprotected implementations, using both existing and proposed models, are presented here. For analysis on full length raw traces using existing models ($A1$), accuracy of 100% is achieved for all analysis sets. It takes 3.6 hours on a GeForce GPU system, as shown in table 5. The results are obtained after fine-tuning the model with the hyper-parameters as mentioned in IV-D3. For some of the optimizers, in the initial few epochs, training and validation accuracy curves are flat because the high number of features slows down the training process. Best accuracy is achieved with Adamax, Selu, and 0.001, as an optimizer, activation, and learning rate, respectively.

TABLE 5. Timing for unprotected LD_{UP} using existing and proposed models.

	Analysis Set	Time (sec)	Accuracy
Existing Model	$A1$	13248.42	100
Proposed Model	$A2$	425.46	100

For analysis set $A2$, and $A4$, firstly SMOTE is applied to balance the data instance, then the dimensionality of the raw data is reduced by pre-processing with PCA. Out of $S_T = 33750$ samples or features, only 800 features are selected based on the presented visual representation in IV-C. It has been observed that the same resulting high accuracy is achieved in just 425.46 seconds, with Adamax, Relu, and 0.001, as an optimizer, activation function, and learning rate, respectively.

B. RESULTS ON PROTECTED IMPLEMENTATIONS (A3 AND A4)

For analysis on $A2$, the raw data trace leakages from the protected implementations are analyzed using the existing complex model. It has been observed that using existing model, 62.1 % accuracy is obtained in 3.46 hours. For analysis on the

TABLE 6. Timing for protected LD_P using existing and proposed models.

	Analysis Set	Time (sec)	Accuracy
Existing Model	$A4$	12473.28	62.1
Proposed Model	$A5$	321.61	67.91

set $A4$, the raw data trace leakages from the protected implementations are resampled using SMOTE, transformed using PCA, and then processed with the proposed CNN network. With PCA processing, features are reduced from $S_T = 39250$ to 800 only. The accuracy of 67.91% is achieved in only 321.61 seconds, as shown in table 6. To further tune the model performance, hyper-parameters are optimized. Both the best performance results are obtained using Adamax and Selu as optimizer and activation function, respectively. For most of the optimizers, including Adagrad and Adadelta, delayed learning is observed for analysis with existing models, which happens due to the large number of features per-instance, whereas the total number of instances is small.

Training and validation, accuracy and loss, for training with the existing model on protected design is shown in Fig. 4. It can be seen that the training loss is decreasing, but the validation loss starts increasing after 50 epochs, which shows that the model performs poorly and might cause over-fitting. The over-fitting phenomenon can be confirmed from the accuracy plot as around epoch 50; the validation accuracy slightly goes higher than the training accuracy.

Training and validation accuracy and loss, for training with 200 epochs with the proposed model with SMOTE is shown in Fig. 5. It can be seen that the training loss is decreasing throughout the learning process. However, the validation loss decreases in the initial 25 epochs only, and after that no variation is seen, which means that the model is not learning any further and might cause over-fitting. So the best possible results achieved, with the protected design implementations under consideration, are 67.91%. It is also observed that the overall training and validation loss is smaller in the proposed model than the existing model training on the ECC Datasets.

To compare the impact of integrating SMOTE and PCA with our presented attack model, we have also performed experiments without having SMOTE or PCA in the pipeline. The results are depicted in Fig. 6. It has been seen that the validation accuracy is fluctuating drastically, and the loss in certain cases goes beyond the training loss which shows a poor model performance. We have also seen that applying only PCA before the classifier does not return impressive results as well. This shows that the combination of all, PCA and over-sampling techniques for imbalance data and our proposed CNN model, provides better performance results than the existing complex models.

To further analyze the improvement produced by the proposed approach, Receiver operating characteristic (ROC) curves, obtained on test data evaluation, are plotted as shown in Fig. 7. ROC curves are the graphical plots, illustrating the classifier's diagnostic ability by displaying the True Positive Rate (TPR) and False Positive Rate (FPR). It can be clearly

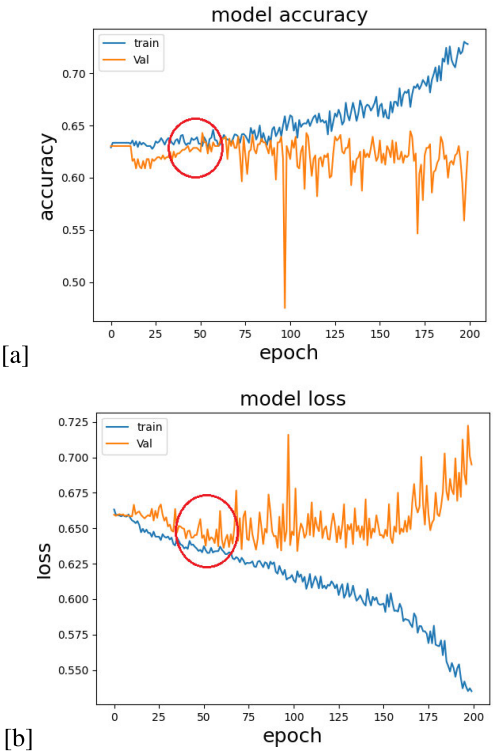


FIGURE 4. Training and validation. (a) Accuracy and (b) loss for the existing model on LD_P.

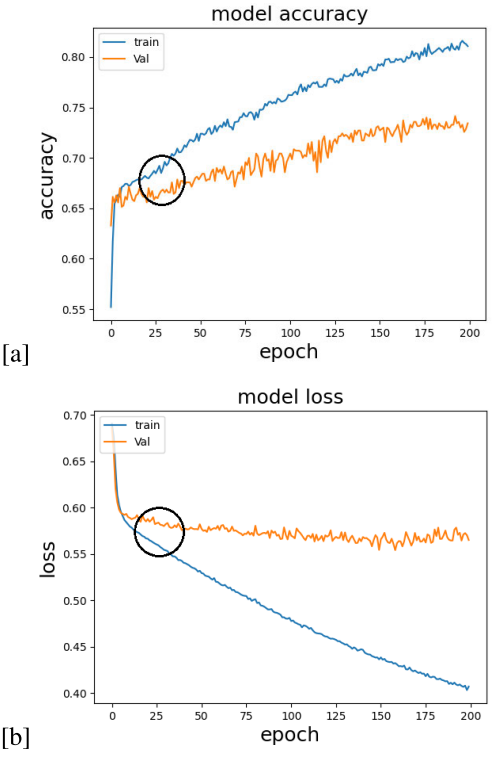


FIGURE 5. Training and validation. (a) Accuracy and (b) loss for proposed model on LD_P using SMOTE.

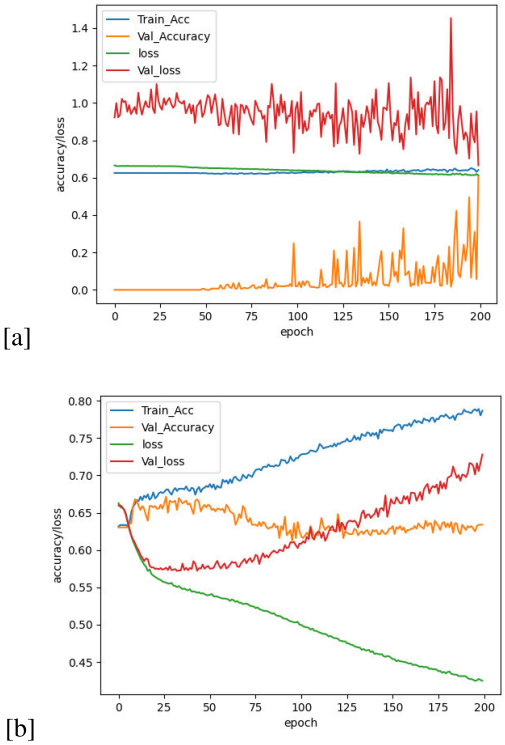


FIGURE 6. Accuracy vs loss plot for (a) SMOTE only and (b) PCA only.

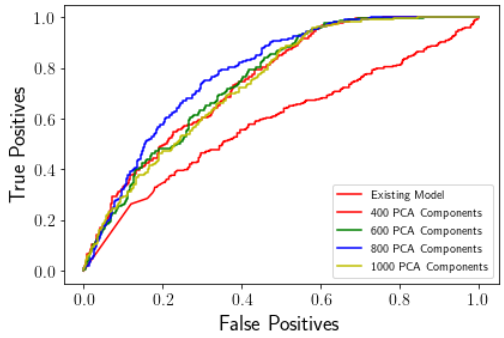


FIGURE 7. ROC of protected implementations for (a) existing model and (b) proposed model.

seen that the ratio of TPR to FPR, and the area under curve obtained using analysis performed on the proposed model is better as compared to the ratio of TPR to FPR, and the area under curve of the analysis of the existing model analysis. It has also been observed that the best performance has been achieved with 800 PCA components among all the test groups of PCA components, as explained in IV-C.

Based on the above results, it can be seen that the time efficiency of the attack has significantly improved using the proposed model. It is also observed that for both protected and unprotected implementations, the accuracy either stays the same or improves, in less training time as compared to the existing complex neural networks.

VII. CONCLUSION

This research work has proposed a hybrid deep learning-based side channel model based on CNN, PCA, and SMOTE, having an optimal number of convolutional layers. Our proposed model is computationally less complex than the existing deep learning-based models and performs better or the same in time-efficient manner. As a test case study, we have selected a variety of the ECC Montgomery Power Ladder Scalar Multiplication algorithm as minimal side-channel analysis exists on ECC from a machine learning perspective. We have used four analysis sets for our evaluation methodology, two for each protected and unprotected ECC implementations. Our experimental results have observed that accuracy improves by 6% using our proposed approach for protected implementations (which is 67%) and stays the same for the unprotected implementation that is 100%. We have also observed the effect of using SMOTE on the proposed model. It is also observed that the overall training and validation loss is less for the proposed model than the existing model training on the ECC datasets. Overall, it can be concluded that the proposed ConvNet enables the network to train much faster with better performance by consuming fewer hardware resources as compared to the existing state-of-the-art methods.

REFERENCES

[1] C. A. Lara-Nino, A. Diaz-Perez, and M. Morales-Sandoval, "Elliptic curve lightweight cryptography: A survey," *IEEE Access*, vol. 6, pp. 72514–72550, 2018.

[2] R. Azarderakhsh, K. U. Jarvinen, and M. Mozaffari-Kermani, "Efficient algorithm and architecture for elliptic curve cryptography for extremely constrained secure applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 4, pp. 1144–1155, Apr. 2014.

[3] Z. Liu, X. Huang, Z. Hu, M. K. Khan, H. Seo, and L. Zhou, "On emerging family of elliptic curves to secure Internet of Things: ECC comes of age," *IEEE Trans. Depend. Sec. Comput.*, vol. 14, no. 3, pp. 237–248, Jun. 2017.

[4] K. Mahmood, S. A. Chaudhry, H. Naqvi, S. Kumari, X. Li, and A. K. Sangaiah, "An elliptic curve cryptography based lightweight authentication scheme for smart grid communication," *Future Gener. Comput. Syst.*, vol. 81, pp. 557–565, Apr. 2018.

[5] S. Chaudhry, H. Naqvi, K. Mahmood, H. Ahmad, and K. Khan, "An improved remote user authentication scheme using elliptic curve cryptography," *Wireless Pers. Commun.*, vol. 96, pp. 5355–5373, Oct. 2016.

[6] A. Höller, N. Druml, C. Kreiner, C. Steger, and T. Felicijan, "Hardware/software co-design of elliptic-curve cryptography for resource-constrained applications," in *Proc. The 51st Annu. Design Autom. Conf. Design Autom. Conf. (DAC)*, 2014, pp. 1–6.

[7] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology*, M. Wiener, Ed. Berlin, Germany: Springer, 1999, pp. 388–397.

[8] E. De Mulder, S. B. Örs, B. Preneel, and I. Verbauwhede, "Differential power and electromagnetic attacks on a FPGA implementation of elliptic curve cryptosystems," *Comput. Elect. Eng.*, vol. 33, no. 5–6, pp. 367–382, Sep. 2007, doi: 10.1016/j.compeleceng.2007.05.009.

[9] D. Genkin, A. Shamir, and E. Tromer, "RSA key extraction via low-bandwidth acoustic cryptanalysis," in *Advances in Cryptology*, J. A. Garay and R. Gennaro, Eds. Berlin, Germany: Springer, 2014, pp. 444–461.

[10] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*, vol. 31. New York, NY, USA: Springer, 2008.

[11] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.*, (Lecture Notes in Computer Science), vol. 2523, B. S. Kaliski, Ç. K. Koç, and C. Paar, Eds. Berlin, Germany: Springer, 2002, pp. 13–28, doi: 10.1007/3-540-36400-5_3.

[12] M. Medwed and E. Oswald, "Template attacks on ECDSA," in *Proc. Int. Workshop Inf. Secur. Appl.*, Feb. 2009, pp. 14–27.

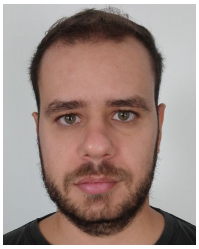
[13] L. Lerman, G. Bontempi, and O. Markowitch, "Side channel attack: An approach based on machine learning," in *Proc. 2nd Int. Workshop Constructive Side-Channel Anal. Secure Design*. Darmstadt, Germany: Center for Advanced Security Research Darmstadt, 2011, pp. 29–41.
[14] R. Gilmore, N. Hanley, and M. O'Neill, "Neural network based attack on a masked implementation of AES," in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, May 2015, pp. 11–106.
[15] H. Maghrebi, T. Portigliatti, and E. Prouff, "Breaking cryptographic implementations using deep learning techniques," in *Proc. Int. Conf. Secur., Privacy, Appl. Cryptogr. Eng.*, 2016, pp. 3–26, doi: 10.1007/978-3-319-49445-6_1.
[16] L. Weissbart, S. Picek, and L. Batina, "One trace is all it takes: Machine learning-based side-channel attack on EdDSA," in *Proc. Int. Conf. Secur. Privacy, Appl. Cryptogr. Eng. (SPACE)*, in Lecture Notes in Computer Science, Gandhinagar, India, vol. 11947, S. Bhasin, A. Mendelson, and M. Nandi, Ed. Cham, Switzerland: Springer, Dec. 2019, pp. 86–105.
[17] N. Mukhtar, M. Mehrabi, Y. Kong, and A. Anjum, "Machine-Learning-Based side-channel evaluation of elliptic-curve cryptographic FPGA processor," *Appl. Sci.*, vol. 9, no. 1, p. 64, Dec. 2018.
[18] 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.*, Aug. 2017, pp. 45–68.
[19] 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," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2019, no. 3, pp. 148–179, May 2019. [Online]. Available: https://tches.iacr.org/index.php/TCHES/article/view/8292
[20] G. Fan, Y. Zhou, H. Zhang, and D. Feng, "How to choose interesting points for template attacks more effectively?" in *Trusted Systems*, M. Yung, L. Zhu, and Y. Yang, Eds. Cham, Switzerland: Springer, 2015, pp. 168–183.
[21] S. Picek, A. Heuser, A. Jovic, and L. Batina, "A systematic evaluation of profiling through focused feature selection," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 12, pp. 2802–2815, Dec. 2019.
[22] N. Mukhtar and Y. Kong, "On features suitable for power analysis—Filtering the contributing features for symmetric key recovery," in *Proc. 6th Int. Symp. Digit. Forensic Secur. (ISDFS)*, A. Varol, M. Karabatak, and C. Varol, Eds. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers (IEEE), 2018, pp. 265–270.
[23] R. Benadjila, E. Prouff, R. Strullu, E. Cagli, and C. Dumas, "Deep learning for side-channel analysis and introduction to ASCAD database," *J. Cryptograph. Eng.*, vol. 10, no. 2, pp. 163–188, Jun. 2020.
[24] L. Batina, J. Hogenboom, and J. G. J. van Woudenberg, "Getting more from PCA: first results of using principal component analysis for extensive power analysis," in *Topics in Cryptology* (Lecture Notes in Computer Science), vol. 7178, O. Dunkelman, Ed. San Francisco, CA, USA: Springer, 2012, pp. 383–397, doi: 10.1007/978-3-642-27954-6_24.
[25] A. Golder, D. Das, J. Danial, S. Ghosh, S. Sen, and A. Raychowdhury, "Practical approaches toward Deep-Learning-Based cross-device power side-channel attack," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 12, pp. 2720–2733, Dec. 2019.
[26] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.
[27] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," 2017, *arXiv:1708.02002*. [Online]. Available: http://arxiv.org/abs/1708.02002
[28] 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," *IACR Trans. Cryptogr. Hardw. Embedded Syst.*, vol. 2019, no. 1, pp. 209–237, 2019. [Online]. Available: https://doi.org/10.13154/tches.v2019.i1.209-237
[29] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. New York, NY, USA: Springer-Verlag, 2003.
[30] A. P. Fournaris, *Fault and Power Analysis Attack Protection Techniques for Standardized Public Key Cryptosystems*. Cham, Switzerland: Springer, 2017, pp. 93–105, doi: 10.1007/978-3-319-44318-8_5.
[31] A. Bauer, E. Jaulmes, E. Prouff, and J. Wild, "Horizontal and vertical side-channel attacks against secure RSA implementations," in *Topics in Cryptology* (Lecture Notes in Computer Science), vol. 7779, E. Dawson, Ed. Berlin, Germany: Springer, 2013, pp. 1–17.
[32] A. Bauer, E. Jaulmes, E. Prouff, and J. R. Reinhard, "Horizontal collision correlation attack on elliptic curves," in *Selected Areas in Cryptography–(SAC)* (Lecture Notes in Computer Science), vol. 8282, T. Lange, K. Lauter, and P. Lisonek, Eds. Springer Berlin Heidelberg, 2014, pp. 553–570.
[33] M. Joye and S.-M. Yen, "The Montgomery powering ladder," in *Proc. 4th Int. Workshop Cryptograph. Hardw. Embedded Syst.* London, U.K.: Springer-Verlag, 2003, pp. 291–302.
[34] P.-A. Fouque and F. Valette, "The doubling attack–Why upwards Is better than downwards," in *Cryptographic Hardware and Embedded Systems* (Lecture Notes in Computer Science), vol. 2779, C. Walter, C. Koc, and C. Paar, Eds. Berlin, Germany: Springer, 2003, pp. 269–280.
[35] S. Yen, L. Ko, S. Moon, and J. Ha, "relative doubling attack against montgomery ladder," in *Information Security and Cryptology—ICISC* (Lecture Notes in Computer Science), vol. 3935, D. H. Won and S. Kim, Eds. Berlin, Germany: Springer, 2006, doi: 10.1007/11734727_11.
[36] S.-M. Yen, W.-C. Lien, S.-J. Moon, and J. Ha, "Power analysis by exploiting chosen message and internal collisions–vulnerability of checking mechanism for RSA-decryption," in *Proc. Mycrypt* (Lecture Notes in Computer Science), vol. 3715. Berlin, Germany: Springer, 2005, pp. 183–195.
[37] F. Amiel, B. Feix, and K. Villegas, "Power analysis for secret recovering and reverse engineering of public key algorithms," in *Selected Areas in Cryptography* (Lecture Notes in Computer Science), vol. 4876, C. Adams, A. Miri, and M. Wiener, Eds. Berlin, Germany: Springer, 2007, pp. 110–125.
[38] A. Bogdanov, I. Kizhvatov, and A. Pyshkin, "Algebraic methods in side-channel collision attacks and practical collision detection," in *Progress in Cryptology (INDOCRYPT)* (Lecture Notes in Computer Science), vol. 5365, D. Chowdhury, V. Rijmen, and A. Das, Eds. Berlin, Germany: Springer, 2008, pp. 251–265.
[39] A. Moradi, "Statistical tools flavor side-channel collision attacks," in *Advances in Cryptology (EUROCRYPT)*, ser. (Lecture Notes in Computer Science), vol. 7237, D. Pointcheval and T. Johansson, Eds. Berlin, Germany: Springer, 2012, pp. 428–445.
[40] B. Feix, M. Roussellet, and A. Venelli, "Side-channel analysis on blinded regular scalar multiplications," in *Progress in Cryptology (INDOCRYPT)* (Lecture Notes in Computer Science), vol. 8885, W. Meier and D. Mukhopadhyay, Eds. Cham, Switzerland: Springer, 2014, pp. 3–20, doi: 10.1007/978-3-319-13039-2_1.
[41] C. Clavier, B. Feix, G. Gagnerot, M. Roussellet, and V. Verneuil, "Horizontal correlation analysis on exponentiation," in *Information and Communications Security* (Lecture Notes in Computer Science), vol. 6476, M. Soriano, S. Qing, and J. López, Eds. Berlin, Germany: Springer, 2010, pp. 46–61.
[42] C. Whitnall, E. Oswald, and F. X. Standaert, "The myth of generic DPA... and the magic of learning," in *Topics in Cryptology—CT-RSA 20T-RSA* (Lecture Notes in Computer Science), vol. 8366, J. Benaloh, Ed. Cham, Switzerland: Springer, 2014.
[43] C. Archambeau, E. Peeters, F.-X. Standaert, and J.-J. Quisquater, "Template attacks in principal subspaces," in *Proc. 8th Int. Conf. Cryptogr. Hardw. Embedded Syst. (CHES)*, Yokohama, Japan. Berlin, Germany: Springer-Verlag, 2006, pp. 1–14, doi: 10.1007/11894063_1.
[44] L. Batina, L. Chmielewski, L. Papachristodoulou, P. Schwabe, and M. Tunstall, "Online template attacks," in *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8885. Cham, Switzerland: Springer, Dec. 2014, pp. 21–36.
[45] L. Papachristodoulou, A. P. Fournaris, K. Papagiannopoulos, and L. Batina, "Practical evaluation of protected residue number system scalar multiplication," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2019, no. 1, pp. 259–282, Nov. 2018. [Online]. Available: https://tches.iacr.org/index.php/TCHES/article/view/7341
[46] J. Fan and I. Verbauwhede, "An updated survey on secure ECC implementations: Attacks, countermeasures and cost," in *Cryptography and Security: From Theory to Applications* (Lecture Notes in Computer Science), vol. 6805, D. Naccache, Ed. Berlin, Germany: Springer, 2012, pp. 265–282.
[47] J.-S. Coron, "Resistance against differential power analysis for elliptic curve cryptosystems," in *Proc. CHES*. London, U.K.: Springer-Verlag, 1999, pp. 292–302.
[48] A. P. Fournaris, C. Dimopoulos, A. Moschos, and O. Koufopavlou, "Design and leakage assessment of side channel attack resistant binary edwards elliptic curve digital signature algorithm architectures," *Microprocessors Microsyst.*, vol. 64, pp. 73–87, Feb. 2019.
[49] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: One-sided selection," in *Proc. ICML*, 1997, pp. 1–8.
[50] I. M. J. Zhang, "KNN approach to unbalanced data distributions: A case study involving information extraction," in *Proc. ICML Workshop Learn. Imbalanced Datasets*, 2003, pp. 1–7.
[51] I. Jolliffe, *Principal Component Analysis*. Berlin, Germany: Springer, 2011, pp. 1094–1096, doi: 10.1007/978-3-642-04898-2_455.

[52] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and Cooperation in Neural Nets*, S.-I. Amari and M. A. Arbib, Eds. Berlin, Germany: Springer, 1982, pp. 267–285.
[53] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object recognition with gradient-based learning," in *Shape, Contour and Grouping in Computer Vision*. Berlin, Germany: Springer-Verlag, 1999, p. 319.
[54] A. Moschos, A. P. Fournaris, and O. Koufopavlou, "A flexible leakage trace collection setup for arbitrary cryptographic IP cores," in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, Apr. 2018, pp. 138–142.
[55] D. Kim, J. Kim, and J. Kim, "Elastic exponential linear units for convolutional neural networks," *Neurocomputing*, vol. 406, pp. 253–266, Sep. 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231220304240
[56] F. Chollet. (2015). *Keras*. [Online]. Available: https://keras.io
[57] F. Pedregosa, G. Varoquaux, and A. Gramfort, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.
[58] *National Computational Infrastructure Australia*. Accessed: Aug. 10, 2020. [Online]. Available: https://nci.org.au/our-services/supercomputing

NAILA MUKHTAR (Member, IEEE) received the B.S. degree in computer engineering from the CIIT, Pakistan, and the M.S. degree in computer engineering from the University of Engineering Technology (UET), Taxila, Pakistan. She is currently pursuing the Ph.D. degree with Macquarie University, Sydney. She has over five years of experience as a research and design engineer with different organizations with a primary focus on integrated security solutions. Her research interests include cryptography, network security, the IoT, data analysis, and embedded system security.

APOSTOLOS P. FOURNARIS (Member, IEEE) received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Patras, Greece, in 2008. He has worked with the Sophia Antipolis Hitachi Europe SAS Research and Development Centre for two years and as a Senior Research Fellow of the University of Patras, Greece, for several years. He is a Principal Researcher (research associate professor) with the Industrial Systems Institute, Research Center ATHENA, Greece. He is also a Sessional Lecturer with Monash University, Melbourne, Australia, where he lectures courses on cryptography, computers, software, and network security. He has authored more than 80 research articles and a member of the IACR, the IEEE Computer Society, and the IEEE Circuits and System Society. His research interests include asymmetric cryptography, side channel attacks and analysis, hardware attack resistance, WSN security, and trusted systems.

TARIQ M. KHAN (Member, IEEE) received the B.S. degree in computer engineering from the COMSATS Institute of Information Technology, Islamabad, Pakistan, the M.Sc. degree in computer engineering from the University of Engineering Technology, Taxila, Pakistan, and the Ph.D. degree in electronics engineering from Macquarie University, Sydney, Australia, in 2016. He is working as a Research Fellow of the Faculty of Science, Engineering, and Built Environment, School of Information Technology, Deakin University at Geelong Waurn Ponds Campus, Australia. His research interests include the most aspects of machine learning, pattern recognition, medical image analysis, scene understanding, and deep learning methods for image analysis.



CHARIS DIMOPOULOS (Member, IEEE) received the integrated master's degree from the Department of Electrical and Computer Engineering, University of Patras, Greece, where he is currently pursuing the Ph.D. degree with the Department of Electronics and Communication Engineering, researching in machine learning applications in hardware cryptography. He has been a part of the EU's Horizon 2020 Programmes CIPSEC under Grant 700378, while currently participating in the CONCORDIA Project under Grant 830927. His research interests include hardware design, cryptography, FPGA, and embedded systems development, and has three publications in international journals and conferences.

YINAN KONG (Member, IEEE) received the Ph.D. degree from the University of Adelaide, Adelaide, Australia. He has also been the Director of Higher Degree Research with the School of Engineering, Macquarie University, Sydney, Australia. He is the Deputy Director of the Graduate School and the Director of the VLSI Research Group, Macquarie University. He is a member of Engineers Australia and the Australasian Association Engineering Education. His current research interests include residue number systems (RNSs), cryptographic circuit immune to side-channel attacks (SCAs), low power embedded design, and visual and speech processing.