IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 May 2024, accepted 4 July 2024, date of publication 19 July 2024, date of current version 20 December 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3431218*

TOPICAL REVIEW

# Review on Hybrid Deep Learning Models for Enhancing Encryption Techniques Against Side Channel Attacks

**AMJED A. AHMED[1,2], (Senior Member, IEEE), MOHAMMAD KAMRUL HASAN[1], (Senior Member, IEEE), AZANA H. AMAN[1], (Member, IEEE), NURHIZAM SAFIE[3], (Associate Member, IEEE), SHAYLA ISLAM[4], (Senior Member, IEEE), FATIMA A. AHMED[5], THOWIBA E. AHMED[6], BISHWAJEET PANDEY[7], (Senior Member, IEEE), AND LEILA RZAYEVA[7]**

[1]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi 43600, Malaysia
[2]Department of Computer Science, Imam Alkadhim University College, Baghdad 10011, Iraq
[3]Research Center for Software Technology and Management (SOFTAM), Faculty of Information Science and Technology (FTSM), Universiti Kebangsaan Malaysia, Bangi 43600, Malaysia
[4]Institute of Computer Science and Digital Innovation, UCSI University, Kuala Lumpur 56000, Malaysia
[5]Computer Science Department, College of Computer Engineering and Science, Prince Sattam Bin Abdulaziz University, Al-Kharj 16273, Saudi Arabia
[6]Computer Science Department, College of Science and Humanities-Jubail, Imam Abdulrahman Bin Faisal University, Dammam 35811, Saudi Arabia
[7]Department of Intelligent System and Cyber Security, Astana IT University, 020000 Astana, Kazakhstan

Corresponding author: Mohammad Kamrul Hasan (mkhasan@ukm.edu.my)

This work was supported by the National University of Malaysia under Grant Scheme GUP 2023-010.

**ABSTRACT** During the years 2018-2024, considerable advancements have been achieved in the use of deep learning for side channel attacks. The security of cryptographic algorithm implementations is put at risk by this. The aim is to conceptually keep an eye out for specific types of information loss, like power usage, on a chip that is doing encryption. Next, one trains a model to identify the encryption key by using expertise of the underpinning encryption algorithm. The encryption key is then recovered by applying the model to traces that were obtained from a victim chip. Deep learning is being used in many different fields in the past several years. Convolutional neural networks and recurrent neural networks, for instance, have demonstrated efficacy in text generation and object detection in images, respectively. In this paper, we have presented a review on deep learning models for encryption techniques against side channel attacks with a comparison table. Also, we have detailed the necessity of hybrid deep learning models for enhancing encryption techniques against these side channel attacks.

**INDEX TERMS** Convolutional neural networks, deep learning, encryption, review, side channel attacks.

## I. INTRODUCTION

In computer and communications systems [1], security has long been a prominent problem, and much research has been done to solve it [2]. Cryptographic algorithms, which include the likes of symmetric ciphers, public-key ciphers, and hash functions, are a series of primitives that can be used to develop security mechanisms that are geared toward achieving certain objectives [3]. Network security protocols such as SSH and TLS, for example, incorporate these primitives in order to provide authentication between communicating entities, as well as to ensure the confidentiality and integrity of data that is communicated. In point of fact, all these security precautions do is identify the functions that are supposed to be taken, but they don't specify how those duties are supposed to be carried out. For example, the specification of a security protocol is typically unaffected by the implementation of encryption algorithms in software running on

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

a general processor or by using specialized hardware units. This is true regardless of whether the memory used to store intermediate data during these computations is connected to or independent from the computing unit.

Kocher [4] first proposed employing side-channel analysis as a way for regaining access to the key via the use of time series analysis back in 1996.The subsequent widespread use of algorithms like the RF (random forest) [5] and a support vector machine (SVM) [6] in the early machine learning algorithms sparked research on side-channel attacks founded on machine learning [7]. Machine learning-based side-channel attacks fall into two separate groups: supervised and unsupervised, depending on whether the attacker and the attack victim are using the same experimental equipment. The profile-based attack [8] along with the non-profiling attack [9] are corresponding to these two categories. Simple energy analysis for non-profiling attacks, such as differential power analysis (DPA) and differential energy analysis (SPA) [10], mutual information analysis, correlation energy analysis, and correlation power analysis (CPA). There have been discussions that Mutual Information Analysis (MIA) [11] can exist in multiple variants. Profiling attacks, as opposed to non-profiling attacks, consist of two stages: attack testing and training set analysis and learning. Random attacks [12] and template attacks [13] are two instances of profiling attacks. The only way to know whether a product is good is to try it out. Neural networks are the basis of deep learning, a subtype of machine learning. At the same time, it is also an innovative technology that is at the forefront of the third renaissance of artificial intelligence. Maghrebi and others [14] investigated the potential of deep learning models such as MLP and CNN by combining deep learning with side-channel analysis. This was the first of its kind. A lot of people are now looking at side-channel analysis that uses deep learning [15].

### A. MOTIVATION

Even if physical encryption techniques on both software and hardware are impervious to side-channel attacks, they are not impossible. An adversary can recover the encryption key and decipher encrypted data by taking advantage of information-dependent leakage sources such as power, time, noise, and radiation. It is necessary to build defenses in order to prevent these attacks on computer systems. Because of its ease of use and little influence on background noise, power analysis has proven to be the gold standard for side-channel inquiry. By analyzing trends in power consumption and data use, devices are able to decrypt encryption. Encryption systems are often tougher to attack because devices have defenses against power analysis methods at different abstraction levels. But these safeguards can make the cryptographic system bulkier, use more power, or take up more space, all of which reduce security. Protecting information in cyberspace requires striking a balance between price, power, area, and safety. To better understand the implementation's security and cost, it is helpful to explore possible power analysis attacks and responses at different abstraction levels. By combining deep learning methods with datasets from cryptographic devices, this research aims to provide light on differential power analysis attacks.

### B. CONTRIBUTION

In this section, our contributions are included as follows:

1) We have discussed the various types of side channel attacks, its taxonomy and the importance of encryption techniques against them.
2) We have explained a detailed review of the deep learning models for encryption techniques against side channel attacks and presented a comparison table of them.
3) Finally, we have detailed the necessity and importance of hybrid deep learning models for enhancing encryption techniques against side channel attacks.

Various challenges are encountered when employing deep learning models to enhance encryption techniques against side channel attacks. These challenges arise due to the intricate nature of both encryption and deep learning methods as well as individual characteristics of side channel attacks.

- In order to train, deep learning algorithms need large amounts of data. Side channel attack data is difficult to collect due to privacy concerns and the need for specialized equipment.
- The training and inference processes of deep learning models are computationally complex. Integrating these models into encryption systems while maintaining performance and resource economy can be challenging, especially in environments with limited resources.
- The nature of encryption and side channel attacks are evolving in response to new cryptographic techniques and defenses. In order to be effective, hybrid models need to adjust to these changes.
- When dealing with sensitive information, privacy concerns arise due to encryption and side channel attacks. Whether in training or inference, the hybrid model must keep sensitive data secure.

There are still certain gaps in the research on deep learning models for side channel attacks. Research is necessary in the following direction:

1) Strengthened encryption using deep learning models should be able to withstand attacks from adversaries.
2) It is possible for deep learning algorithms to perform well on certain datasets and environments, but they can not be able to adapt to new data or side channel attack situations.
3) In machine learning, deep learning models often require vast quantities of training data that are difficult to obtain for side-channel attack datasets due to privacy concerns and lack of access to confidential information.

4) In order to enhance encryption against side channel attacks, deep learning models have to be assessed with predefined criteria.

In section II, we have presented the literature review related to deep learning-based side channel analysis methods. Section III explains the description of datasets for side channel analysis. In section IV, we have shown the data pre-processing process and its results. Finally, section V presents the conclusion of our study on hybrid deep learning models for enhancing encryption techniques against side channel attacks.

## II. LITERATURE REVIEW

In this section, we have reviewed the state of art methodologies related to deep learning-based encryption techniques for side channel attacks. Here, this chapter starts with the brief explanation of side channel attacks (SCAs). Also, it explains the side channel taxonomy, analysis, power analysis attacks like simple power analysis [16], differential power analysis [17], machine learning and deep learning methods for SCA [18].

Lately, side-channel attacks have drawn the attention of researchers and security specialists worldwide [19]. They use a range of cipher-cracking techniques to first create countermeasures, and then they offer suggestions for enhancing the security of the cyphers. Some researchers employ deep learning models to perform side-channel attacks. Convolutional neural networks, or CNNs, were the main instrument they employed to illustrate the effectiveness of their attacks and provide instructions on how to execute them. There have been three main phases in the development of side-channel attacks all through history:

1) The identification and utilization of various side-channel information sources for the goal of critical analysis characterize this phase of the SCA, which ran from 1996 to 2000. It was discovered in 1996 [20] that RSA can potentially be broken by abusing the algorithm's execution time. In 1998, the issue of DES breakage was tackled using the power consumption leaking model. Electromagnetic radiation is also able to be exploited efficiently for side-channel attacks, according to research done in 2000 by Quisquater and Samyde [21].
2) The initial years of the SCA's establishment (2001-2010). The main characteristic that sets this stage apart is the growing attention that is paid to SCA evaluation, countermeasures, and applications, along with the identification of new leakage models. The DPA contest, a side-channel analysis competition, was established in 2008 [22]. The machine learning foundation for several later investigations was established by the traces gathered from this DPA competition. 2010 saw a significant increase in the usage of watermarks, fault sensitivity, and flash memory pumping as side-channel attack techniques.
3) The SCA's highest point of development (after 2011) [23]. The fundamental characteristic of this level is the increased utilization of cross-domain technologies for SCA. Specifically, convolutional neural networks (CNN) and multi-layer perceptrons (MLPs) are becoming more and more prominent deep learning techniques. Energy trace misalignment, camouflaged AES implementations, and jitter-based countermeasures are all evaded by CNNs.

### A. SIDE-CHANNEL ATTACKS

An attack that is known as ‘‘side channel attack'' [24] is one where attacker is successful in obtaining confidential data out of actual deployment of crypto-system. Therefore, analysis of side-channel was cryptography's subfield of that focuses on implementing an electronically systems rather comparatively of cryptographic algorithm's flaws that are utilized. It is in contrast to traditional approaches to cryptography, which examine the weaknesses of the algorithms themselves. In this kind of attack, the attackers can glean information from a number of sources, including as timing data, power utilization, the sound produced by gadgets and leakage of electromagnetic signals. Different kinds of side channel attacks, some of which include electromagnetic attacks, cache attacks, software implementation attacks, timing attacks and others. Fig. 1 illustrates most simple model about side-channel attack that can be used against a cryptosystem. In the majority of instances, the attacker is not aware of the cryptographic algorithms that are implemented in device or system; nonetheless, by knowing about deployment physically of cryptographic algorithms, attackers are able to discover essential information about the target.

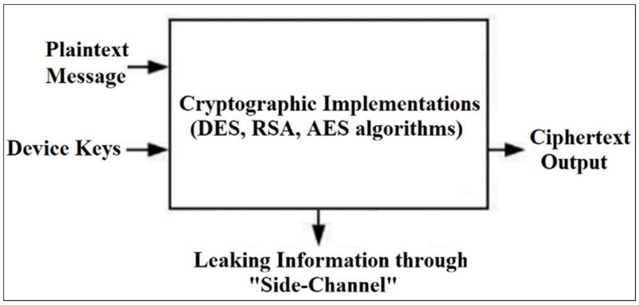

Plaintext Message
Device Keys
Cryptographic Implementations (DES, RSA, AES algorithms)
Ciphertext Output
Leaking Information through "Side-Channel"

**FIGURE 1. A block diagram of cryptography system with side-channel attacks.**

In this research work, we put a lot of emphasis over powered analysis attack that belongs to most unique categories of side-channel attack. The discussion of attack of power analysis will take place over consequent part.

#### 1) ANALYSIS OF SIDE-CHANNEL

Analysis of Side-channel, often known as SCA's, takes into consideration those attacks, which that are directed against the implementations of algorithms rather than the weaknesses in the methods themselves [25]. The main goal of side-channel analysis is to determine which data is likely to be processed by comparing the actual (measured) leakage with certain hidden, data-dependent assumptions about physical leakages. Such is accomplished by comparing certain hidden predictions of the physical leakages. It is important to be able to model leakages (to make data forecasting) having competent comparing tools in order to successfully take out confidential data. This is needed in order to extract the information efficiently (distinguisher). By simulating real-world conditions, a leakage model can assess the extent to which a device is physically defective [26]. Using either the Hamming weight of the hypothetical data or the Hamming distance between the registers that hold the data values at two distinct times, the two most popular and well-supported leakage models draw on these two concepts.

The divide-and-conquer [27] technique is one that is often used in side-channel analysis. This strategy aims to recovering important parameters of chunks (for instance, sub-key bytes that are found in AES), which makes strategy feasible computationally. Thus, indicating procedure outlined above, has to be applied to each sub-key until the original key k, which contains all of its bits, can be retrieved. In most situations, recovering only one sub-key is sufficient to show that there is a weakness in the implementation of the security measure.

### 2) SCA TAXONOMY

Two types of attacks that can take place in the territory controlled by the SCA are two-stage attacks and direct attacks [28]. The attack with directed attack, often referred to as a non-profiling attack, collects a considerable number of data from the device they are aiming their attack at and then uses statistical techniques to extrapolate sensitive information from those measurements. Commonly found instances of attacks like these are differential power analysis (DPA) and simple power analysis (SPA) [29]. Even if they assume that the attack is less capable, direct attacks can need millions of countermeasures to get crucial information.

The strong attacker [30] in a two-stage attack, also known as a profiling attack, owns a replica of the target device that is either identical to it or at least relatively equivalent to it. This is done in order to profile the target. Once the target device has been cloned, the attacker attacks it. This attack is comprised of two stages: the pre-attack phase and the attack phase.

A fictional leaky statistical distribution is the only thing that can come close to approximating the distribution of side-channel data, which is the foundation upon which profiling approaches are built. The template attack is the oldest and most well-known method for profiling attacks. In this kind of attack, an attack makes assumption that, leakage tracks a multiple-varying Gaussian distribution. Profiling process will continue with the subsequent step in evaluating parameters of statistical directed over Gaussian mixture prototype. As a result, model gets constructed pertaining to every potential hypothetic kind of leakages. During phases of attack, attacks will estimate chance that fresh side channel measurement (one that is now under the attacks) related to particular groups through making use of function of probability density function that was computed using the approximate statistics. Even while an attack of profiling requires additional powered attacks compared to non-profiling, this nevertheless manages to break the target with a great deal fewer traces than direct attacks do; in certain instances, only a single trace is sufficient.

Later on, techniques of machine learning were used in order to profile attacks, and in this particular instance, the profiling set is utilized in order to automatically learn the statistics of an unknown leaking distribution. To avoid making assumptions about the distribution of leaks, machine learning can construct the profile model independently of template attacks. Machine learning has several advantages, such as these.

Conventional machine learning, in which feature engineering is frequently done prior to an attack, can be further subdivided into two distinct categories: deep learning, in which raw features or an optimal trace time period are utilized; and deep learning, which is a subset of conventional machine learning [31]. We include a variety of resources for readers who can be interested in machine learning models, including [32], despite the fact that the first class of machine learning models is not the primary topic of this study. In addition to that, we provide a summary of the side-channel analysis that is based on machine learning [33]. The following is a list of the key differences that can be found between [34] and our work: Because [35] was documented within early periods of deep learning-dependent SCA (in early 2018), this merely covers small number of deep learning jobs. Furthermore, because [36] is surveyed tasks providing in-depth data regarding correlated work without systematizing knowledge, we only look into deep learning-based SCA. This is because [37] was documented within earlier periods of deep learning-based SCA.

### 3) COUNTERMEASURES

The examination of CNN behavior over side-channel information that was conducted by Samiotis and others [38] involves different categorization situations. They began by evaluating CNN's behavior on 4 separate side-channel data's datasets before moving on to compare their models to more age-old machine learning (ML) approaches [39] and CNN's prototype that was found within existing body of academic research. In our work, we have used CNN as well as models of RNN [40]. Differences amongst such 2 models were represented by us in terms of their precision and performance.

Maghrebi, Portigliatti, Prouff, and others [41] were the first people to apply the techniques regarding deep learning to problem of side-channel attack. They compared efficacy of their proposed attacks to that of the most common machine learning and template attacks in order to evaluate the usefulness of their ideas. They carried out its attacks over 3 distinct sets of information through estimating required amount of traces and then used those calculations. They were able to show that their attacks were more effective than the most recent and cutting-edge profiled side-channel attack. We looked at side-channel attacks by utilizing 2 separate information-sets. In upcoming time, we want to improve our study through doing it with the help of other datasets.

Picek and others [42] investigated the efficacy of a variety of machine learning techniques by taking into consideration a wide range of factors relevant to profiling Side-Channel Analysis (SCA). They also suggested employing convolutional neural networks, often known as CNNs, in order to get effective results while performing profiled SCA. However, we have shown that by making adjustments to the hyper-parameters, it is feasible to construct an RNN model that will perform better than model of CNN.

Picek, Kim, Hanjalic, Heuser and others [43] were able to get outstanding results with the CNN model that they developed by making use of the random delay countermeasure. This was made possible by the fact that the results were not dependent on the length of the delay. They suggested using techniques of deep learning as well as noise into the evaluation of the side-channel investigation. Since we found that recurrent neural networks (RNNs) give the best fit sequential data and for time series, we've decided to use them. As a result, evaluation of our results by us will be done and comparing them to conclusions of previous studies based on machine learning or CNN.

In addition, several investigators used machine learning (ML) methods [44] in order to extract data out of cryptographic systems that are either protected or not guarded. The vast majority of them center their attention on two methods that have garnered a lot of attention recently: Random Forest (RF) and Support Vector Machine (SVM). Research was done by Gierlichs, Verbauwhede, De Mulder, Hospodar, and others [45] on the use of machine learning to side-channel analysis. They aimed their efforts on analysis of power, which, when performed, discloses enormous amounts of data on the cryptographic key that is being processed through power traces. In addition to that, the classification approach that they utilized was called LS-SVM. In the course of our investigation on power-analysis attacks, we made use of two of the most well-known neural networks (CNN and RNN). Utilizing SCA defenses to protect implementations is considered to be common practice. The statistical connection between intermediate values and traces can be severed by the use of countermeasures (e.g., EM emanation or power consumption). Masking and concealment remain 2 fundamental subcategories that make up SCA countermeasures.

During the process of masking, each intermediate value is covered over with a randomly generated value (mask). Using random masks, the link that existed between the measurements and the concealed data is severed so that the results can be more accurate. Two types of masking that are often used are known as arithmetic masking and Boolean masking [46]. Contrarily, goal of hiding information was in providing the impression that measurements are random or continuous. When anything is hidden, the only thing that changes is signal-to-noise ratio (SNR). Concealment can occur in temporal domain (for example, via desynchronization, random delay interruptions, or jitter) as well as the amplitude domain (for example, by introducing noise to the signal). The power level of the attack directly influences the selection of appropriate countermeasures. An attack with infinite side channels and measuring capabilities may, in theory, avoid detection by concealing countermeasures.

In a similar vein, an attack who has access to mask shares during a profiling phase has the potential to, in the worst-case scenario, defeat a masking countermeasure that takes into consideration a first-order masking. Thus, highly protected targets can use high-order masking techniques or concealment and masking countermeasures.

### B. POWER ANALYSIS ATTACK

A kind of side-channel attack known as a "power analysis attack" involves attacking the device or system being targeted by utilizing the amount of power being used as the information that is being leaked. An electrical device is expected to carry out cryptographic technique execution such as RSA, AES, and DES and when this occurs, the device is supposed to make use of electricity in a certain manner. DES, AES, and RSA are examples of such methods. In order to successfully execute the cryptographic algorithms, a large quantity of power is necessary. During the course of the computation, the attack keeps an eye on how much power is being used by the gadget. These power consumptions are afterwards recorded in a computer as power traces in order to facilitate a further and more in-depth analysis. The attacker makes use of every strategy that is at his disposal in order to make an attempt to extract information about the algorithm's secret key from the power traces that have been captured. After doing painstaking investigation using a number of methods, the attack is successful in unearthing every vital piece of knowledge about the cryptosystem, including the device key.

Fig. 2 depicts one possible arrangement for a power analysis attack to illustrate the point. Power analysis attacks typically need the use of a digital oscilloscope, a personal computer, and the device that is being targeted.

The attacker makes a connection between the target device and an oscilloscope that is outfitted with probes. In essence, the equipment that is the focus of the investigation employs a cryptosystem of some type and is operating on one of the encryption algorithms. When monitoring and recording the power consumption of the target device, the attacker will utilize a sample rate that has been previously agreed

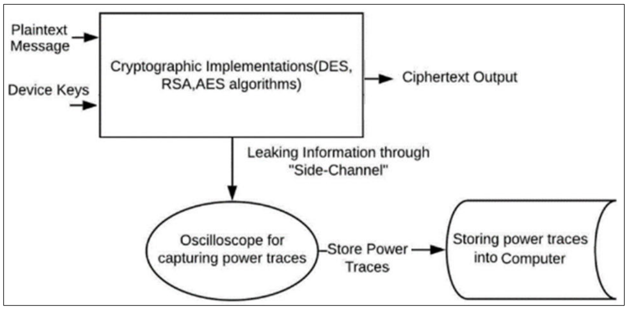

Plaintext Message
Device Keys
Cryptographic Implementations(DES, RSA,AES algorithms)
Ciphertext Output
Leaking Information through "Side-Channel"
Oscilloscope for capturing power traces
Store Power Traces
Storing power traces into Computer

**FIGURE 2.** A block diagram of a power analysis-based attack model.

upon. They are given the cluster of dots that contains the measurements of the voltage levels. Each of these safeguards is referred to by its formal term, which is power traces. This particular power traces accumulation is deposited on standalone computer system. The attack is able to analyze and alter the collected power traces with the help of common data processing tools with this capacity (Python, MATLAB, etc.). A positive aspect of the situation is that the attack does not comprehend the encryption method that the target device employs. To completely grasp the cryptosystem and be in a position to launch an attack all that is required is for the person to inspect the power traces in the appropriate manner and to make educated assumptions. Template-based analysis, Simple power analysis (SPA), horizontal power analysis and differential power analysis (DPA), and are only some of the many varieties of power analysis attacks (HPA). We will quickly discuss 2 significant kinds in order to familiarize readers with them.

### 1) SIMPLE POWER ANALYSIS

Most fundamental iteration of power analysis attack concerning side channel is referred to as an SPA. During this attack, one or more power traces will be watched and analyzed, and the attacker will make an effort to extract as much data and logic as possible from these traces. Additionally, the attacker will make an effort to get additional knowledge on the system or item that is the target of their attack. Afterward, by making use of the information that was obtained, he or she is able to determine all of the actions that are carried out by the device that is the object of the investigation. The individual may, at long last, be capable of determining the cryptographic key.

The single power trace of a cryptographic algorithm is shown in Fig. 3. The pattern of the dataset can be detected if one looks closely enough.

The Single-Phase Analysis (SPA) technique is a way for rapidly discovering the target system's knowledge as well as the secret key by making use of a single or a limited number of powered traces.

### 2) DIFFERENTIAL POWER ANALYSIS (DPA)

The term "non-profiled attacks" is also used to refer to DPA attacks [48] on occasion. When an attack lacks fully control to closed equipment, such the smart cards used by financial institutions, they are more likely to carry out an attack called attacks of non-profiled. Therefore, it is only permissible for him or her to consist of restricted amount of side-channel power traces concerning to operation of cryptograph that uses predetermined value of secret key that cannot be discovered. During these particular attacks, attack could attack to subsequent data obtained out of attacking gadgets [49]:

1) One of them is a static private key, denoted by the letter k, which is included into a key space.
2) Messages and inputs that are completely at random.
3) Outputs generated at random and a text cypher.

In order to obtain information and the key, attackers first accumulate side-channel traces and consequently mix entirely such data in order to perform analyzing

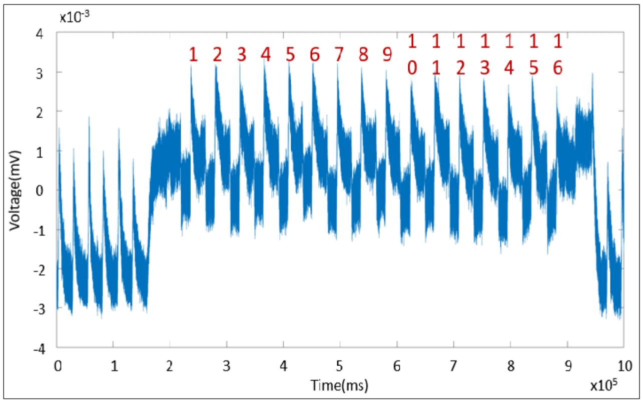


**FIGURE 3.** The power trace of a DES encryption [47].

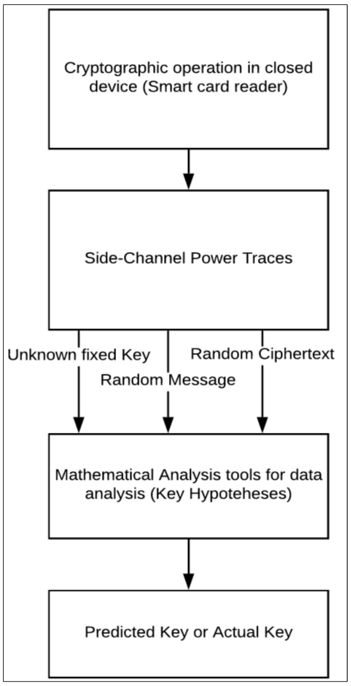

Cryptographic operation in closed device (Smart card reader)
Side-Channel Power Traces
Unknown fixed Key
Random Message
Random Ciphertext
Mathematical Analysis tools for data analysis (Key Hypoteheses)
Predicted Key or Actual Key

**FIGURE 4.** A differential power analysis of attack.

mathematically or important hypothesis utilizing several well-known algorithms or tools for data analysis. This allows the attacker to infer information and obtain the key.

The implementation of differential power analysis attacks is going to be a component of the research that we do for our thesis. We consist of two datasets: primarily consists of real information, and remaining consists of simulated information. The actual data are in the first dataset.

Fig. 4 depicts the most basic version of an attack that can be carried out against a cryptographic system by making use of a differential power analysis. Our dataset contains information on closed devices, which implies side-channel traces with known ciphertext, known messages and fixed unknown keys. This information can be found by searching for “closed devices” in the search bar. In order to produce an educated forecast about the procedure, we examine the trace data and, using the two most common neural network methodologies (CNN and RNN), anticipate information pattern. This allows us in making an informed prognosis. We mapped message-ciphertext couple based on the value of a certain key in its the space.

### C. TECHNIQUES OF DEEP LEARNING

Deep learning is a method that’s becoming more popular for use in algorithms of machine learning. With deep learning’s area, a prototype can simulate functioning of human brain by picking up knowledge from previous experiences and becoming better at carrying out certain tasks. Deep learning was found to be applied in different modern applications, including self-driving cars [50], pattern identification, face detection, digital image processing, and others. The selection of deep learning techniques over others is because deep learning ability to process large volume and high dimension data efficiently and model complex and hard-to-some patterns. When it comes to certain tasks such as identifying images, deep learning approaches including the Convolutional Neural Networks excels due to the best and efficient hierarchical features extraction from the raw inputs overpowering the traditional forms of approach. RNNs are other advanced models that perform well in applications involving sequential data since they understand the long-term dependency better than other models. The superiority is most visible when it comes to different evaluations and metrics of performance in different fields where deep learning models much of the time set a record, surpassing the performance of traditional machine learning models.

#### 1) CONVOLUTIONAL NEURAL NETWORK (CNN)

Image processing, data analysis, problem categorization, and image analysis are some of the most common applications of convolutional neural networks (CNN), which are also known as convNet in certain other settings. These networks are quite well known and are used often. Convolutional layers, as opposed to the more typical hidden layers, are what set

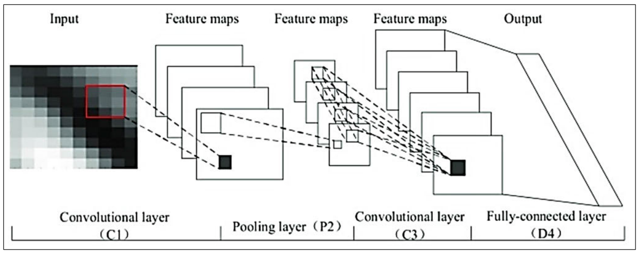


FIGURE 5. A top-level view of convolutional neural network model.

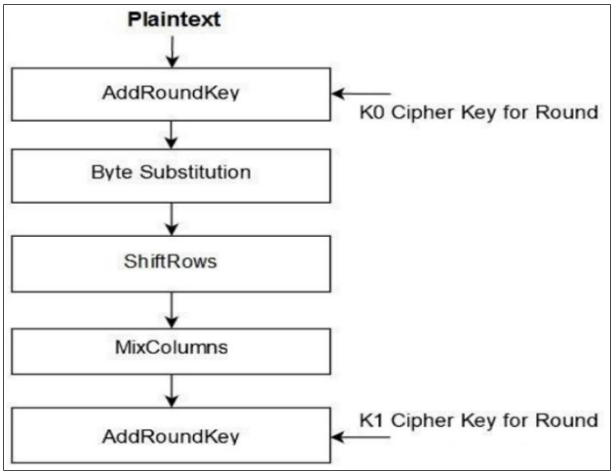


FIGURE 6. An illustration of AES algorithm.

CNN apart from other kinds of neural networks. Before continuing on to the subsequent convolutional layers, the data that was entered into the model is first altered by these convolutional layers. Convolutional operations are what are employed to make these kinds of changes to the data. A CNN model works on the assumption that the inputs are images, and that these images already have the properties that the model needs to encode.

Each convolutional layer makes use of a certain amount of filters or kernels in order to locate the specific characteristics in the images or patterns that we are searching for. In these filters, there are a total of n dimensions. All the way through convolution process, these kernels and filters are being used. In order for the output to be passed on to subsequent neurons present within neural networks, filters will first compute product of input and filter data. Filter settings are applied on top of the values of the input data. An n-dimensional array is used to store the results of the convolution operations, which can be referred to as either an activation map or a feature map. As we apply more filters, we will be better able to recognize and pull-out certain features from the incoming data. CNN uses convolutional layers, which function according to the basic concept outlined above, in order to recognize properties, present in images.

When working with a large data volume, such as when working with a large number of images, the operations of the convolution layer will take a significant amount of time to complete. If this is the case, the number of parameters utilized in the convolution technique can be reduced to allow for the usage of pooling layers. We have the option of using three distinct pooling layers, including the maximum, the average, and the sum.

The operation of a CNN network is seen in Fig. 5. reference [6], was retrieved from a piece of paper that was in the room.

### D. ADVANCED ENCRYPTION STANDARD (AES)

The Advanced Encryption Standard, or AES, is a different form of symmetric encryption mechanism than triple DES. AES, which is very much like DES, is included in the category of block cyphers. The functioning of the system requires keys of 128-256 bits and data that is 128 bits long.

The technique shown in Fig. 6 is an example of a typical kind of encryption.

AES uses an iterative method of operation, in contrast to the Feistel encryption method. In actuality, it does its computations with the help of bytes rather than bits. Plaintext communications with an AES key of 128 bits will take up a total of 16 bytes due to the fact that these were organized with 4 × 4 matrixes. Amount of AES rounds is completely random. As an instance, 128-bit key were necessary for about 10 rounds, 192-bit key are necessary for 12 rounds, 256-bit keys are necessary for 14 rounds.

The decryption procedure within AES method is just the encryption process performed in reverse order. AES is used in a significant amount of today's computer hardware and software.

### E. RELATED WORKS ON SCA

With their groundbreaking work, Maghrebi and others [51] first used CNNs for side-channel attacks. They use deep learning techniques such as MLP [6], CNN [7], and LSTM [8] to assess and contrast traditional machine learning methods. Among these methods are SVM [10] and random forest [9]. Their study's conclusions demonstrate the superiority of deep learning over traditional machine learning techniques and the favorable results that follow. They demonstrate this with two distinct datasets: one uses an implementation with no protection at all, and the other makes use of a countermeasures for masking. Furthermore, their findings demonstrate that CNNs typically outperform competitors on both datasets.

The ASCAD database, also known as the side-channel assessment dataset, is made available by the authors in [9]. The authors originally presented this database, which has been utilized in numerous investigations by other academics. Following the dataset introduction, they look into the impact of the hyperparameters to determine which CNN and MLP designs will work best. The results of Prouff and others study [52] indicate that when a CNN encounters mismatched traces, its kernel volume rises and its behavior improves. It is a little odd, though, that they fail to clarify why raising the kernel actually makes the attack more potent. We think this is a remarkable discovery that definitely warrants further discussion.

Since these two experiments demonstrate that CNN functions well in a variety of settings [10], [11], [12], more research on CNN's behavior was done. CNN's performance was contrasted with that of other ML techniques, for instance Random Forest [9], XGBoost [10], and Naive Bayes, by Picek and others [2]. They are mostly interested in finding out what conditions lead to CNNs outperforming the previously described methods. Their study's conclusions indicate that CNNs can only enhance performance overall. They claim that when traces are not pre-processed, noise levels are reduced, and information dimensions are increased (that is, when there are numerous features with numerous traces), CNNs perform at their best. On the other hand, performance from ML methods can be nearly equal to that of CNNs. An important finding is that ML techniques require far less processing power than CNNs. Consequently, the researchers are somewhat dubious about CNNs' use.

Subsequent investigation revealed that convoluted neural networks (CNNs) could potentially outperform advanced solution for the given datasets. The measurements for every single one of the datasets came from an implementation with a hidden countermeasure. In order to demonstrate that CNNs can synchronized misaligned traces with determining the attributes of more important traces, the authors of [3] conducted tests. As a result, we were able to categorize assets. In addition to these findings, the researchers detail the procedure of carrying out this attack using unprocessed raw trace data. This attack differs from a template attack in that it doesn't involve the attacker independently reorganizing the traces and choosing the points-of-interest. The results demonstrate that CNNs are useful even in cases when the paths are not aligned. However, due to the scale and intricacy of the design of CNN beneath the surface, overfitting is a possibility. They provide two approaches for augmenting training data using misaligned trace data. To demonstrate the effectiveness of the data supplementation choices for misaligned traces, demonstrations are conducted.

Hybrid deep learning is the key to preventing overfitting in the misaligned data in trace scenarios which are sophisticated and requires a combination of RNNs and CNNs. Consequently, this method succeeds in circumventing the designed complicatedness and subtleness of the CNN architecture. RNNs and CNNs are utilized together to increment generalization and minimize overfitting, hence the hybrid architectures. CNNs are good with spatial correlations and feature hierarchies, while RNNs are better at acquiring sequence relations, which gives them the ability to find useful temporal patterns out of misaligned trace data. In cases of complicated data structure as in case of misaligned traces, hybrid methods which combine RNN and CNN can be the answer and will help in improving scalability and performance of deep learning systems in challenging situations.

The findings of [4] are corroborated through Kim and others [5], showing their CNN framework performs at leading edge on data set of RD. This lends more credence to the findings of [4]. Notably, as compared to data set of DPAv4, that are considered to be fundamental information set, ideal network of their need less attacks traces in order to recovering RD dataset key [6]. During their study [6], Authors experimented with a wide range of information sets and topologies. According to the findings of these trials, no single design works well with every collection of data. Thus, it is still crucial to choose a structure that makes sense in relation to the problems that are at hand. Furthermore, they offer proof that adding distortion to a network's primary substrate improves performance by lowering the rate of overfitting. This is recommended to make greater noise levels beneficial when utilizing smaller datasets, whereas larger datasets require lower noise levels to get optimal performance.

According to these studies, there are two key characteristics of traditional news networks (CNNs) that make them appropriate for side-channel study. First of all, they are able to independently and independently identify the most significant elements. Consequently, obtaining higher behavior did not require doing previous processing on traces. We considered this to be a significant benefit over more traditional methods. The authors of [7] state that preparation processing is prone to errors and that poor choice of PoI decreases performance. CNNs can recognize features regardless of where they are located inside feature vectors due to their spatial invariance. This is the second advantage of CNN use. CNNs can exhibit good performance on datasets resulting from disguised countermeasure implementations because of this feature. The approaches employed in the research project that we have been talking about so far are common procedures in the deep learning community. Further investigation has led to the recommendation that novel and creative strategies be employed, especially created for side-channel attacks with the goal of exploiting a few characteristics.

Researchers proposed a CNN architecture using side-channel attack domain data [8]. The leaky model determines if the data supplied for the development of neural networks is ciphertext or plaintext. The part of a CNN architecture that receives the domain knowledge to be used as a new feature vector is the classification block. In their creative endeavors, they contrast and compare a number of architectural ideas presented in different literary works alongside and without the construction they themselves possess provided. They demonstrated how better performance can be achieved for both protected and non-protected information by utilizing domain knowledge in design. However, this approach is inappropriate to employ if profile trace is produced with a fixed key. Given that models are useless without properly tuned architecture and hyperparameters, Zaid and others [9] place a premium on these areas. They show how important it is to understand hyperparameters in order to effectively use an architecture. As a solution to this problem, the authors provide three visualization methods: weight visualization, gradient visualization, and heatmap. These methods simplify the process of reading and understanding hyperparameters. By giving an opponent, the ability to assess each hyperparameter's influence independently, these strategies facilitate the process of setting and fine-tuning the hyperparameters. They also suggest implementation solutions for both secured and unprotected contexts using these three visualization techniques. Adjust the CNN kernel measures to 50 percent of the maximum randomized delay for datasets that include a hidden countermeasure. Articles from deep learning forums propose, increasing the number of substrates relative to the number of neurons within each layer [10]. Through the development of designs and testing across all of the publicly accessible datasets using the approaches they provided, they enhanced state-of-the-art working with respect to complete information sets, resulting in an overall performance increase. While their method provides state-of-the-art performance for all publically available datasets, the selection of hyperparameters is often made arbitrarily. For instance, the authors don't clarify how or why particular learning rates were established for a limited number of particular datasets.

Pfeifer and Haddad [11] suggest using the spread layer, a deep learning layer. As the first of its type, this layer would be created with side-channel attacks in mind. Haddad and Pfeifer demonstrated in their research that this layer would not perform well without a substrate. Profiling expedites learning by reducing the amount of traces used. These results were exciting for the groups involved in side-channel analysis since they imply that there was a reason to design substrates specifically designed to exploit traces' side-channel capabilities. This is due to the fact that these results suggest that there is an incentive to design layers specifically designed to exploit the side-channel characteristics of traces. Details regarding the layer's hyperparameter setup or the reasons for the outcomes this layer can provide. We will look into the propagation layer in great depth and fix some of its flaws in chapter 4, which will address these problems.

Kim and others research [12] indicates that the deep CNN framework performs wonderfully for SCAs. Even yet, there are still certain problems with the deep neural network training process. The main problem is that gradients can either develop or vanish during DNN training. This makes the process challenging. We will discuss recent developments with the introduction of deep neural networks to address the previously listed problems in the areas that are relevant.

The problem of parameter initialization has received a fair amount of attention; variables are frequently chosen at arbitrary from a Gaussian spectrum. Glorot and Bengio [13] substantially revised this and concurrently established a completely new initialization technique known as Xavier's initialization. Parameter values are computed using this method, which takes inputs and outputs into consideration and uses a Gaussian distribution. Many large deep learning libraries employ this technique to initialize their parameters, and it is currently accepted as standard practice.

Scholars investigating deep neural network architectures discovered that a number of distinct studies had difficulties with their designs' convergence. For instance, the widely utilized VGG architecture, which at first is trained over the course of four phases, encountered convergence issues. After that, further layers are added to the network, and training is done at every turn to make sure it converges appropriately [14].

He and colleagues describe a novel technique for deep CNN initiation in their publication [15]. Their research indicates that the Xavier initial is not suitable for usage with the ReLU, even though it was intended to function with linear activations. Furthermore, they contend that it is more difficult for deeper networks to converge to a point. "He" initialization is supplied by them as a remedy for such problems, which, when compared to other initialization techniques, improves the degree of convergence of deep neural networks and was created especially for CNNs using ReLU. An alternate initialization technique called LSUV initialization is put forth in [16], [17]. Instead of being designed with ReLU as the function of activation in mind, this approach has more generic qualities and can be used to a variety of other architectural types. They carry out experiments to support their claims, offering proof of the approach's feasibility. The significance of precise initialization for network parameters for the convergence of deep neural networks has been demonstrated by both research sets. The published study has only recently started to investigate real-world SCA scenarios in which the profiling and attack traces came from the same devices. The fact that the same key was used for the attack path and the profile track wasn't unusual, though. Because of this, the findings of these investigations could directly lead to, give a false impression of the efficacy of certain therapy, such as DL, ML, and TA. This has led the SCA community to begin building a more realistic atmosphere where a range of devices are being utilized to gather attack and profiling evidence.

Recent research has shown that even identical gadgets can differ in certain ways. Because of this, there can be very little variations in these devices' measurements, resulting in it more challenging to identify true SCAs [18]. This is because devices could differ in features, even if they are similar. These two studies demonstrate how underestimated the value of SCAs in warfare scenarios are the findings of the existing available research. Therefore, conducting additional study within the context of mobility is imperative.

The authors of [52] provide a method that can be used to boost the efficacy of attacks in a circumstance where they are movable. Their technique solves the problem of measurements being different on identical devices by constructing a profile that makes use of measurements taken from a large number of duplicate devices. They accomplished this and demonstrated that they could get 99% accuracy on the test traces, which allowed them to pass the SCA and receive their certificate.

Around the same time, Bhasin and others [53] published a technique that was quite similar to address the variance that occurred amongst devices that were otherwise identical. In order to construct the training and validation set for their method, they make use of traces collected from a number of devices that are identical to one another. They show that their strategy is superior to use of a standalone gadgets pertaining to validation and training in portable environment by demonstrating its superior performance. In addition, they notice in their research that the efficacy of the attack decreases when additional traces were utilized in phase of profiling. This could be an indication that the networks are overspecializing in some areas.

In addition, the article [54] explains that distinct datasets react to diverse topologies in a variety of unique ways. This highlights how important is selecting an appropriate architecture and hyperparameters for issues that is currently being addressed. To continue, the authors point out that, according to the information presented in [55], improving size of kernel yields in rise of misaligned trace's performance. Nevertheless, kernel size influence is not primary contribution of their study, they fail in provide a great deal of detail on the topic of the kernel size's influence. Reference [56] shows that deep CNNs can perform effectively even with misaligned traces and passably even with masked traces. This is in addition to the kernel size. As a consequence of this, it is abundantly evident that all past efforts have concentrated only on the designs that provide the maximum performance and have not addressed the influence of certain hyperparameters. In the current investigation, the kernel and depth of CNNs are of particular importance for scenarios where counter-measure, such as masking or random delay, is used. In addition, we are interested in determining whether or not there lies correlation amongst size of kernel and network's depth. In addition, through differentiating between countermeasures, we plan to investigate whether or not generalizing is feasible to architecture depth and kernel size and for particular counter-measure.

Side-channel attacks have recently garnered a lot of attention from academics and security professionals all over the world. They devise countermeasures by employing a variety of cipher-cracking strategies, and afterward they offer recommendations regarding how to beef up the effectiveness of these countermeasures. Deep learning models are used by certain researchers in order to carry out side-channel attacks. The majority of the time, they relied on convolutional neural networks, often known as CNNs, to showcase their attacks and demonstrate how successful they were.

The examination of CNN's effect over side-channel information that was carried out by Samiotis et al. [57] included a variation of different grouping saturation. They began by evaluating CNN's performance on 4 separate side-channel information-sets data before moving on to compare their models to older machine learning (ML) approaches and a model of CNN that was found inside existing body of academic research. In our work, we have used CNN as well as RNN model [58], [59]. It shows that the differences between such models in terms of their precision and performance.

Kim, Picek, Heuser, Hanjalic, and others [60] were able to attain outstanding performance while using the random delay countermeasure because of the CNN model that they developed. They suggested using techniques of deep learning as well as noise into the evaluation of the side-channel investigation. Since we found that recurrent neural networks (RNNs) give the best fit for sequential and time series data [61], we've decided to use them. As a result, we aim at evaluating our outcomes and comparing them to conclusions of previous studies based on machine learning or CNN.

Table 1 shows the comparison various encryption techniques for side channel attack based on deep learning models.

**TABLE 1. Comparison various encryption techniques for side channel attack based on deep learning models.**

| Author(s) | Aim of the research work | Techniques used | Dataset | Measure. parameter | Limitations |
|---|---|---|---|---|---|
| Ngoc-Tuan, and others [62], 2022 | CNN and the MLP models for non-profiled attacks on the AES-128 encryption | CNN and MLP models | ASCAD dataset | Correlation Power Analysis (CPA) | Execution time is large (20h 28min 48s) |
| Yohaï-Eliel Berreby [63], 2022 | DL for SCAs on AES | DL using JAX framework | ASCAD dataset | Power Analysis | Further need investigation of technique, especially for high-resolution and raw traces |
| Huanyu Wang, and others [64], 2021 | Tandem DL SCA on FPGA Implementation AES | CNN | DPA_v2 and AES_HD dataset | Correlation Power Analysis (CPA) | Average Probability of Guessing Entropy (PGE) of CNN classifier is need to be improved |
| Zixin Liu, and others [65], 2021 | SCA using Long Short-Term Memory (LSTM) and Word Embedding | LSTM | DPA_v2 dataset | Correlation Power Analysis (CPA) | The total training time is high |
| Mani Karthikeyan and others [66], 2023 | DL Network evoked chaotic encryption to mitigate SCAs | DL based LSTM | NIST dataset | Correlation Power Analysis (CPA) | Efficiency of encryption model is only 95%, still needs to improve the efficiency |
| Lichao Wu, and others [67], 2023 | DL-based Non-profiling SCA using Plaintext/Ciphertext | Plaintext Labeling DL | AES_RD dataset | Comparison of Plaintext and Ciphertext models | Further improvement is needed in attack performance |
| Fanliang Hu, and others [68], 2021 | Multi-leak SCA investigation using DL | DL based multi-input model structure | AES_GPU dataset | Correlation Power Analysis (CPA) | Low accuracy of 82.5% in detection of SCA |
| Takaya Kubota, and others [69], 2020 | DL SCA against Hardware implementations of AES | CNN | AES_HD dataset | Correlation Power Analysis (CPA) | Further need to develop NNs and datasets for SCAs and high-order attacks |
| Guanlin Li, et. al., [70], 2020 | SCNet: A NN for Automated SCA | LSTM | ASCAD dataset | Power Analysis | Performance improvement is required |

In addition, several authors have used machine learning (ML) schemes [71] in order to extract data out of cryptographic systems that are either protected or not guarded. The vast majority of them center their attention on two methods that have garnered a lot of attention recently: Random Forest (RF) and Support Vector Machine (SVM). Verbauwhede, De Mulder, Gierlichs, Hospodar, and others [72] conducted research on uses machine learning with process of analyzing side-channel performances. They fixated their efforts over analysis of power, which, when performed, reveals significant quantities of data regarding the cryptographic key that is being processed via power traces. In addition to that, the classification technique that they used was called LS-SVM. We investigated power-analysis attacks by applying 2 of most famous neural networks available (CNN and RNN).

Using a pharmacogenetics model, the authors of the article [73] were able to successfully match the genotype of a patient to the appropriate dosage of a particular medicine. Additionally, they were able to retrieve user face images from the training stage of a face recognition system by using neural network models [74]. This was a really remarkable accomplishment.

To determine if a given data entry is an element of the training set, Shokri and others [75] created a membership inference technique. Black-box access to a model was necessary for this method. Tramer and others [76] presented a model inversion attack by exploiting the relationship between searches and levels of trust on a range of machine learning models, including logistic regressions, DNN, and so on. Despite the attacks [77] that exploited the training sets' privacy issues, Hua and others [78] introduced a novel technique to reverse engineering the underlying network data. They infer details of the network architecture, including the quantity of tiers and the size of the characteristic maps for every layer, by examining the sequences of memory access. Additionally, they demonstrated that even in cases when memory access was detected whilst in the "zero pruning" phase of the procedure, the weight values could still be recovered. The primary distinction between our strategy and other approaches previously documented is the attack's objective. An attempt is made to analyze the escape at the inference stage in the suggested methodology. The strength of the side channel allows us to reconstruct an image from runtime inputs without the need for network models or training set samples. This allows us to recreate the image.

The secret keys that are kept inside cryptographic equipment can be retrieved by exploiting any electrical side-channel leakage in the apparatus. In popular symmetric encryption systems like DES [2] and AES [3], attackers can find the secret key by comparing and analyzing the differences between several power traces with different inputs. In the channel on the power side, Eisenbarth and Msgna [79] demonstrated how to retrieve the type of instruction that was carried out by the CPU by making use of a hidden Markov model. This was accomplished in order to do so. Liu and others [80] were able to precisely locate each instruction instance while the program was being executed by using a modified version of the Viterbi algorithm. The creation of a ''power template'' is a technique that is frequently utilized in the process of breaking the secret keys of cryptographic systems. They begin by estimating a leaking model with the private keys and the collected power traces. Next, they put this model to use to predict the keys based on the online power traces that are collected during the runtime. This is very similar to the strategy that we recommended. Despite the fact that the overall process is very similar, one of the most difficult parts of creating ''power traces'' is finding an appropriate attacking surface.

To improve the ability of deep learning models to migrate learning and generalize across different devices, implementations, or operating conditions would require more study. It is therefore necessary to apply domain adaptation and transfer learning techniques that consider hardware designs, software settings and variables in order to facilitate smooth flow of information from source domain to target domain. Additionally, research should come up with robust methodologies that can assess model performance under diverse circumstances so as to ensure consistent and reliable behavior across platforms and deployment conditions. This will make deep learning models more adaptable and generalizable thereby making them useful in various real life applications across domains and environments. To resolve security issues in deep learning systems, an in-depth understanding of model predictions and vulnerabilities especially side channel attacks is necessary. Interpretability research tries to understand the decision making process of deep learning algorithms with a view to helping stakeholders analyze and reduce risks. Researchers can improve understanding of model predictions and malicious attacks by developing interpretable models and methodologies. Assurance of trustworthiness in essential applications demands that sensitive data be protected through fortification of deep learning models against security attacks.

## III. DATASET

In this part, we go even deeper into the datasets that were used for the study and discuss them in further detail [3]. Several different AES implementations were used in order to collect these traces, which range from unprotected to protected states. We go over the process of acquiring the traces for each dataset, as well as identifying which S-box is being targeted.

### A. ASCAD

This repository of ASCAD was shown through [4]. This database was structured in a manner that is similar to database of MNIST, it contains fifty thousand profile traces in addition to ten thousand attack traces. A masked version of AES-128 running on an AVR microcontroller using 8 bits of power for processing (ATmega85515) was used to get these traces. The traces were produced by electromagnetic radiation, and it was this emission that was recorded. The raw traces that constitute the database include the measurements that were taken throughout the whole encryption process. In addition, researchers have already pre-chosen a raw trace's window which corresponds to S-box execution of sub-key 3 having seven hundred different characters. In the course of our research, we make use of this portion of the dataset.

**TABLE 2.** Power traces values.

| Power Traces | Time |
|---|---|
| 0.015918 | 9.56E-07 |
| 0.015417 | 9.65E-07 |
| 0.01738 | 9.56E-07 |
| 0.01904 | 9.56E-07 |
| 0.019448 | 9.56E-07 |

### B. DPAV4

The DPA contest version 4 has a total of 100 000 traces, each of which possesses a total of 3000 attributes [5]. Due to the fact that the traces leak first-order data [6], the S-box output can only be utilized as an unprotected dataset. This is because it is impossible to use this dataset to store protected data.

### C. RANDOM DELAY

There are a total of 50,000 traces in this dataset, and each trace consist of 3,500 attributes. Random delay defense mechanism that is described within [7] was created with the help of an Atmel AVR microcontroller [8] that has 8 bits. In this dataset, our attention is focused on the very first key byte [9].

### D. PORTABILITY

Traces for such collection came out of a single gadget, and they were gathered with the use of a near-field electromagnetic probe [10]. An Arduino Uno [11] with an unprotected AES-128 implementation [12] loaded onto an AVR Atmega328p [13] serves as the measuring equipment that was utilized to capture the data. The name given to this dataset is Porta. This dataset has a total of 50,000 attack traces and profiling, every one of those traces includes a set of 500 attributes [14].

## IV. DATASET PROCESSING

### A. DATASET STRUCTURE

The pattern of our dataset, how we preprocess it, and how we put it up for our model are all covered in the first section.

Out of 50,000 traces, Table 2 displays a small number of power trace readings. In this case, we will simply discuss the values of the power traces and demonstrate how we preprocessed these numbers to make the model function.

Data sets have the following characteristics:

1) Time-series data make up the data values.

**TABLE 3.** DPA contest v1 dataset.

| File name | Trace Counts | File Sizes |
|---|---|---|
| secmatv120060408019.zip | 81.088 | 4Gbytes |
| secmatv3 20070924 des.zip | 81.569 | 1Gbytes |
| secmatv3 20071219 des.zip | 67.753 | 6Gbytes |

**TABLE 4.** Toggling count dataset.

| Filename | Total Traces Count | File Size |
|---|---|---|
| 0_key_message_ciphertext1.csv | 48 | 5KB |
| 1_key_message_ciphertext1.csv | 48 | 5KB |
| 0_key_message_ciphertext1.csv | 48 | 5KB |
| 1_key_message_ciphertext1.csv | 48 | 5KB |
| 0_key_message_ciphertext1.csv | 48 | 5KB |

2) The power used by the operation is represented by the power trace values.
3) Following the retrieval of DPA Contest v1 data, various message, key, and ciphertext combinations are used to produce hundreds of CSV files including power traces.
4) We can extract the CSV file by executing the C programming code that was supplied on the DPA contest website.
5) Two columns are included in every CSV file: one for time as well as another for the quantity of electricity used. In this example, the time is meaningless because we are merely interested about power examines and not the date the data was taken.

All of the datasets that we'll use in our research are time-series data. As a result, each data point in our series is successively sampled at equal time periods. The data points' chronological arrangement is determined by the time of collection.

Enclosed power trace measurements from an acquisition platform are included in the DPA Contest v1 Dataset. Table 3 shows the DPA Contest v1 Dataset features. Telecom Paris-Tech is the owner of these traces and has gathered and preserved them. Data in ".bin" format is tracked in terms of power consumption. A C-program file called "agilent bin reader.c" has also been given to convert ".bin" files into ".csv" files.

Voltage is displayed as power usage traces in the datasets. Nanovolts (1e-9 nV) are used to express the voltages.

Verilog simulated datasets will be utilized in our studies. These dataset files will include time and toggling counts. A dataset's values are derived from particular key-pairs, cipher-text, and plain text combinations.

In these datasets, the filename is made up of Plain-text, Cipher-text, Keys, and either encryption or decryption (0 or 1). The files contain information about time and toggle counts. The dataset's organizational structure is displayed in Table 4.

Experimental Dataset: These datasets are experimental and were gathered using the Artix-7 FPGA board. There are single and double differential I/O standards in the Artix-7 evaluation kit.

### B. DATASET CLEANING AND VISUALIZATION

Here, we describe the data cleaning process for the three datasets. It contains raw data, noisy data, and so on. Therefore, our objective is to clean these datasets before proceeding with the validation and training of them in neural networks.

To clean the datasets, we'll apply methods from digital signal processing. The Hanning and Hamming window routines will be used. The following illustrates the outcomes of using digital signal processing operations on our SCA DPA contest files.

Fig. 7 of the DPA contest datasets is created by using the Hanning and Hamming window functions. In the image, the y-axis represents time, while the x-axis represents the power traces in voltage. It's clear that the datasets were leaked, and most of the noise has subsided.

Since the toggling count datasets employ simulation data values from Verilog HDL, no digital signal processing techniques are needed. Fig. 8 shows DPA contest dataset after applying windowing functions. Here are the toggling count dataset patterns, as seen in Fig. 9. The figure shows how the detection of the data changes results in a big spike.

### C. DEEP LEARNING EXPERIMENTS

In this section, we will examine 3 distinct DL experiments using the toggle count and DPA contest datasets. We have utilized RNN and CNN as our DL technique.

First experiment: Using predictions DES sessions from the DPA contest and toggled count data sets, and then we compare CNN and RNN models. We compare the accuracy and loss of these 2 models. We compare the execution times of the 2 models, with similar data sets.

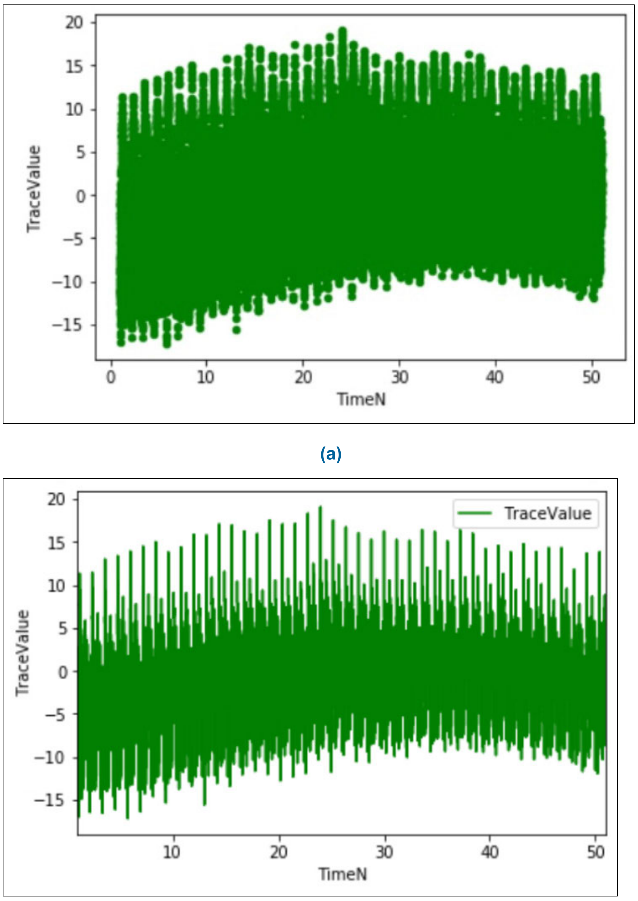
DES and nonDES are the two groups into which we split our datasets for the second experiment. We use CNN and RNN models for this. We compare and evaluate the performance of these two models.

Third Experiment: In this last experiment, we evaluate how well the hybrid model (CNN-RNN) maps a particular key for the combination of plaintext and the ciphertext according to whether side-band information is present or not.

#### 1) PREDICTION OF DES ROUNDS

DES algorithm uses 16 cycles for both encryption and decoding. We discuss deep learning techniques for DES round prediction in this section. The DES rounds have been predicted using CNN, RNN, and a hybrid model called CNN-RNN. We forecast DES rounds using toggled counts and DPA contest data sets. Lastly, the CNN, RNN, and CNN-RNN models are compared in terms of their accuracy and model.

We will state again that our primary aim was not only to use a CNN or RNN models, but rather a hybrid (CNN-RNN) model. Our goal in developing this hybrid model is to achieve an accuracy level comparable to that of RNN and CNN.



(a)

(b)

**FIGURE 7.** Digital Signal techniques (a) initial condition, (b) after cleaning).

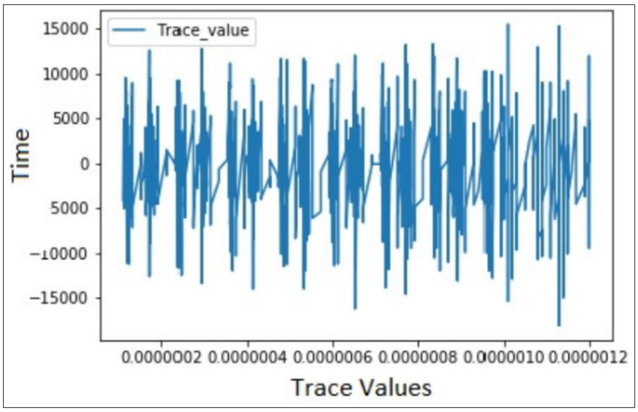


**FIGURE 8.** The results of windowing operations on DPA contest data.

A range of metrics, such as the starting point, the quantity of epochs, the simulation speed, precision, and the model loss, will be used to assess the models' performance. Since the datasets are not categorized in this case, the definition's arbitrary baseline is 100 percent. We have 50,000 unique values for keys, ciphertext, and plaintext, all generated at random using the baseline criteria. On the other side, without a constant key, ciphertext, or plaintext value, our baseline will most often be 0.

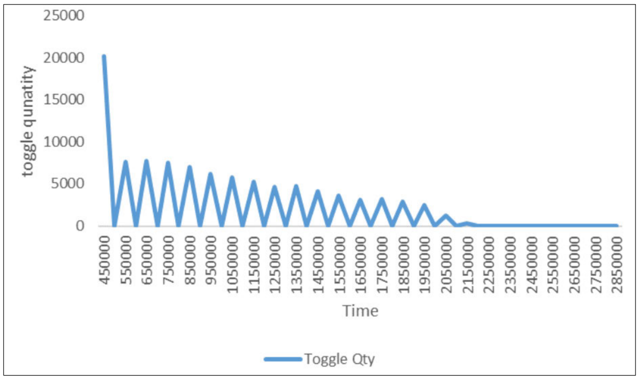


**FIGURE 9.** The toggle count dataset pattern.

### 2) CLASSIFICATION OF TWO CLASSES: DES AND NON-DES

The procedure for DES and non-DES class classification utilizing side-channel datasets is detailed in this part. Here, we assume that the attacker has sufficient ability to decipher the plaintext and ciphertext as well as the key values for certain permutations of the two. An attacker is interested in using DL techniques to separate their datasets into DES and non-DES classes now that they have access to the traces left by previous cryptography technologies. In order to categorize the datasets as DES or non-DES, we have used the RNN model and the LSTM network.

In this case, we have 68000 datasets that are DES and 55000 datasets that are non-DES (AES). Dividing the datasets into two groups will be the primary objective. Our goal is to train a model that will correctly identify AES datasets as non-DES with a classification of 0. Furthermore, DES datasets will have a label of 1 to denote that they are DES datasets.

The arbitrary and typical baselines in our dataset are considered in two baselines. Some of the factors we consider while assessing the performance of the 2 models are the execution duration, loss, and accuracy of each model for a predefined no. of epochs.

Since there are two categories here, DES and non-DES, the random baseline is 50%. Based on the model, it will be classified as DES if it is 1, and non-DES (AES) if it is 0.

The most common starting point: A total of 68000 DES datasets and 55000 non-DES (AES) datasets were used for the classification task. At 55.28 percent, it is the most typical baseline. Using the most popular baselines as a guide, we split 123000 DES records over 68000 datasets.

### 3) HYBRID MODEL (CNN-RNN) FOR MAPPING KEYS

The key values for a given plaintext/ciphertext combination can be derived using the hybrid model. We offer two separate pieces. Initially, we feed encryption/decryption bits (0/1) and plaintext and ciphertext to the neural network. The NN models' output for a specific set of plaintexts and ciphertext will match the key exactly.

The following set of inputs is given into the neural network: side-band data (toggled count quantity), ciphertext, plaintext, and the encryption/decryption bits (0/1). A set of keys mapping to a 56-bit key space is the anticipated result. Next, the effectiveness of key mapping is evaluated with and without the addition of a sideband.

#### a: KEY NAPPING WITHOUT SIDE-BAND INFORMATION

A hybrid (CNN-RNN) model will be created using 3 inputs: ciphertext, plaintext, and a 1-bit encryption or decryption key. The two text formats are in hexadecimal 16-byte format. The way the hybrid model works can be summarized as follows:

1) First, the hexadecimal data is converted to binary numbers. Consequently, the output is ciphertext with 64 bits of data, plus a one-bit key for encryption and decrypting. The network receives 129 bits of binary information as input.
2) Next, we just take into account a machine understanding multi-class logistical regression model, which converts inputs from two or more items to the matching 3rd element. Accordingly, we utilized softmax in the last layer. Softmax is known to generate the odds for a given input to a certain outcome.

#### b: KEY MAPPING WITH SIDE-BAND INFORMATION

The same approach is used to get the key value for a certain plaintext-ciphertext combination in this key mapping task. All of the input parameters are varied. In this instance, the model's inputs are side-band data (values from toggling count amount data), ciphertext, encryption/decryption keys, and plaintext. The purpose of using side-band information as an additional input parameter in the hybrid model is to see whether it can improve accuracy or outcomes compared to not using it. Here is how the process works:

1) As in the prior challenge, we will convert our ciphertext and plaintext from hexadecimal to binary numbers. Within the NN model, 129bits of binary input are used in conjunction to 1-bit encryption/decryption keys respectively. We have these inputs in addition to side-band data.
2) Softmax comes before the output layer in the same manner. Softmax provides us with the multi-class regression probability. A key value derived from the key-space for a certain pair of plaintexts and ciphertext-which also contains extra information-is mapped onto the model.

In this experiment, a number of factors are used to test the effectiveness of two RNN models, including execution time, accuracy and loss over a predefined number of epochs, utilization of memory for model storage spaces.

Cybersecurity is confronted with a number of challenges by side channel attacks using deep learning. These models are not easily generalizable to other devices or implementations because hardware variations and target system countermeasures. Unpredictability and side channel noise weaken the training and durability of the model. Neural network architectures that extract substantial information from side channel input need deep learning and side channel analytical abilities. Important issues also include addressing ethical concerns about their use and making sure these models are adversarial resilient.

Side channel attacks based on deep learning have several limitations, but they are promising nevertheless. For private or constrained systems, obtaining the massive amounts of labeled data needed for training could be a challenge. Because of differences in hardware, operating conditions, and countermeasures, these attacks could also struggle to generalize across devices or implementations. Side channel input presents extra challenges due to its inherent noise and unpredictability, perhaps.

It is necessary to develop hybrid encryption systems to bridge research gaps in deep learning-based encryption methods for side-channel attacks.

- Deep learning based encryption needs to be attack resilient. In order to understand these models' decision making process during a side channel attack, modeling interpretability is needed. To improve resilience assessment and security measures, researchers need to be able to make intelligence of how models operate and predict.
- There are certain factors that need consideration when designing scalable and efficient deep learning-based encryptions. Methods of such kind should handle huge amounts of data with fewer resources used and complex calculations involved. Knowledge is essential about training models that remain resistant despite hardware changes, software modifications or environment alterations.
- Encryption efficiency couldn't achieve the maximum due to conventional deep learning-based encryption thus increasing the cost of computation. These approaches should be optimized for both efficiency and security so as they can function in real world systems. One must understand whether adversarial attacks could target different deep learning based encryption systems.

Hybrid deep learning models could potentially offer more protection against side channel attacks. However, there are still unfulfilled research gaps concerning deeply integrated learning of hybrid methodologies for the encryption of side-channel attacks.

## V. CONCLUSION

We have demonstrated in this research work that the possible practical problems that could arise from applying DL-based SCA on power oriented traces with an excessive amount of samples. Here, our contribution shows the importance of hybrid deep learning models for enhancing encryption techniques against side channel attacks. Hybrid systems, which integrate the benefits of two distinct approaches, have long been recommended as best practices. Even in situations when there is a notable decrease in the amount of accessible data, the architecture can reliably identify the proper key because it was trained using more advanced features. Since the gadget can successfully extract the right key even in challenging real-world situations, more research is required to ascertain how robust the design is. The unique architecture is nevertheless able to show the proper key, for example, in a circumstance where the data quantity is one-fifth of what is generally there, or the data are downsampled to half. The efficient design's ability to successfully recover the right key from desynchronized traces further demonstrates the architecture's adaptability, as demonstrated by tests done on two different datasets. The traditional DL methods majorly suffers from the limitations such as low efficiency of encryption model (95%) and low accuracy in detection of SCA. Hence, further improvement of these models is necessary and this can be achieved through the Hybrid DL methods.

In our future research, we want to investigate a number of effective hybrid deep learning strategies, such as parallel architectures and the Early Stop, with the goal of maximum improving the performance of neural networks against non-profiled attacks. Future advancements in hybrid deep learning methods for the encryption of side-channel attacks are anticipated to include new algorithms, theoretical insights, and practical applications. At the times when threats are dynamic, the encrypted hybrid deep learning system makes more secure, effective, and user-friendly.

## ACKNOWLEDGMENT

The National University of Malaysia has supported this work Under the grant scheme GUP 2023-010. Also, the authors would like to thank the Network and Communication Research Laboratory, Center for Cyber Security, and Faculty of Information Science and Technology for supporting this research work.

## REFERENCES

[1] Z. Hongwei, K. Zhipeng, Z. Yuchen, W. Dangyang, and Y. Jinhui, "TSGX: Defeating SGX side channel attack with support of TPM," in *Proc. Asia–Pacific Conf. Commun. Technol. Comput. Sci. (ACCTCS)*, Jan. 2021, pp. 192–196, doi: 10.1109/ACCTCS52002.2021.00046.

[2] N. Musa, "A conceptual framework of IT security governance and internal controls," in *Proc. Cyber Resilience Conf.*, Nov. 2018, pp. 1–4, doi: 10.1109/CR.2018.8626831.

[3] J. Yu, L. Lu, Y. Chen, Y. Zhu, and L. Kong, "An indirect eavesdropping attack of keystrokes on touch screen through acoustic sensing," *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 337–351, Feb. 2021, doi: 10.1109/TMC.2019.2947468.

[4] P. C. Kocher, "Timing attacks on implementations of Diffie–Hellman, RSA, DSS, and other systems," in *Advances in Cryptology—CRYPTO '96*. Cham, Switzerland: Springer, 1996, pp. 104–113.

[5] A. Jamil and Z. Mohammad Yusof, "Information security governance framework of Malaysia public sector," *Asia–Pacific J. Inf. Technol. Multimedia*, vol. 7, no. 2, pp. 85–98, Dec. 2018.

[6] A.-T. Le, T.-T. Hoang, B.-A. Dao, A. Tsukamoto, K. Suzaki, and C.-K. Pham, "A real-time cache side-channel attack detection system on RISC-V out-of-order processor," *IEEE Access*, vol. 9, pp. 164597–164612, 2021, doi: 10.1109/ACCESS.2021.3134256.

[7] A. A. Ahmed, M. K. Hasan, N. S. Nafi, A. H. Aman, S. Islam, and S. A. Fadhil, "Design of lightweight cryptography based deep learning model for side channel attacks," in *Proc. 33rd Int. Telecommun. Netw. Appl. Conf.*, Nov. 2023, pp. 325–328, doi: 10.1109/itnac59571.2023.10368560.

[8] B. A. D. Kumar, S. C. Teja R, S. Mittal, B. Panda, and C. K. Mohan, "Inferring DNN layer-types through a hardware performance counters based side channel attack," in *Proc. 1st Int. Conf. AI-ML-Syst.*, Oct. 2021, pp. 1–7, doi: 10.1145/3486001.3486224.

[9] A. A. Ahmed, M. K. Hasan, N. S. M. Satar, N. S. Nafi, A. H. Aman, S. Islam, and S. A. Fadhil, "Detection of crucial power side channel data leakage in neural networks," in *Proc. 33rd Int. Telecommun. Netw. Appl. Conf.*, Melbourne, Nov. 2023, pp. 57–62, doi: 10.1109/itnac59571.2023.10368563.

[10] Q. Guo, A. Johansson, and T. Johansson, "A key-recovery side-channel attack on classic McEliece implementations," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 1, no. 4, pp. 800–827, Aug. 2022, doi: 10.46586/tches.v2022.i4.800-827.

[11] A. Spence and S. Bangay, "Security beyond cybersecurity: Side-channel attacks against non-cyber systems and their countermeasures," *Int. J. Inf. Secur.*, vol. 21, no. 3, pp. 437–453, Jun. 2022, doi: 10.1007/s10207-021-00563-6.

[12] Y. Liu, B. Zhao, Z. Zhao, J. Liu, X. Lin, Q. Wu, and W. Susilo, "SS-DID: A secure and scalable web3 decentralized identity utilizing multi-layer sharding blockchain," *IEEE Internet Things J.*, vol. 11, no. 15, pp. 25694–25705, Mar. 2024, doi: 10.1109/JIOT.2024.3380068.

[13] Z. Wu, G. Liu, J. Wu, and Y. Tan, "Are neighbors alike? A semisupervised probabilistic collaborative learning model for online review spammers detection," *Inf. Syst. Res.*, vol. 34, no. 4, pp. 1321–1336, Oct. 2023, doi: 10.1287/isre.2022.0047.

[14] S.-H. Cheng, M.-H. Lee, B.-C. Wu, and T.-T. Liu, "A lightweight power side-channel attack protection technique with minimized overheads using on-demand current equalizer," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 69, no. 10, pp. 4008–4012, Oct. 2022, doi: 10.1109/TCSII.2022.3185608.

[15] R. Abarzua, C. Valencia, and J. Lopez, "Survey for performance & security problems of passive side-channel attacks countermeasures in ECC," *Cryptol. ePrint Arch.*, vol. 1, pp. 1–43, Aug. 2019.

[16] Q. Lei, C. Li, K. Qiao, Z. Ma, and B. Yang, "VGG-based side channel attack on RSA implementation," in *Proc. IEEE 19th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Dec. 2020, pp. 1157–1161, doi: 10.1109/TrustCom50675.2020.00154.

[17] A. A. Ahmed, M. K. Hasan, I. Memon, A. H. M. Aman, S. Islam, T. R. Gadekallu, and S. A. Memon, "Secure AI for 6G mobile devices: Deep learning optimization against side-channel attacks," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 3951–3959, Feb. 2024, doi: 10.1109/tce.2024.3372018.

[18] X. Wang and W. Zhang, "PacSCA: A profiling-assisted correlation-based side-channel attack on GPUs," in *Proc. IEEE 38th Int. Conf. Comput. Design (ICCD)*, Hartford, CT, USA, Oct. 2020, pp. 525–528, doi: 10.1109/ICCD50377.2020.00094.

[19] A. Ihsan and E. Rainarli, "Optimization of K-nearest neighbour to categorize Indonesian's news articles," *Asia–Pacific J. Inf. Technol. Multimedia*, vol. 10, no. 1, pp. 43–51, Jun. 2021.

[20] L. X. Ying, A. H. Mohd Aman, M. S. Jalil, T. Mohd Omar, Z. S. Attarbashi, and M. A. Abuzaraida, "Malaysia cyber fraud prevention application : Features and functions," *Asia–Pacific J. Inf. Technol. Multimedia*, vol. 12, no. 2, pp. 312–327, Dec. 2023.

[21] S. Ghandali, S. Ghandali, and S. Tehranipoor, "Deep K-TSVM: A novel profiled power side-channel attack on AES-128," *IEEE Access*, vol. 9, pp. 136448–136458, 2021, doi: 10.1109/ACCESS.2021.3117761.

[22] M. Mohammed Kataa and W. Kaur, "Recognizing facial emotion in real-time using MuWNet a novel deep learning network," *Asia–Pacific J. Inf. Technol. Multimedia*, vol. 13, no. 1, pp. 1–20, Jun. 2024.

[23] A. Garg and N. Karimian, "Leveraging deep CNN and transfer learning for side-channel attack," in *Proc. 22nd Int. Symp. Quality Electron. Design (ISQED)*, Apr. 2021, pp. 91–96, doi: 10.1109/ISQED51717.2021.9424305.

[24] D. Wang, A. Neupane, Z. Qian, N. Abu-Ghazaleh, S. V. Krishnamurthy, E. J. M. Colbert, and P. Yu, "Unveiling your keystrokes: A cache-based side-channel attack on graphics libraries," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2019, pp. 1–50.

[25] D. Das, M. Nath, B. Chatterjee, S. Ghosh, and S. Sen, "STELLAR: A generic EM side-channel attack protection through ground-up root-cause analysis," in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust*, McLean, VA, USA, May 2019, pp. 11–20, doi: 10.1109/HST.2019.8740839.

[26] 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.*, 2019, pp. 86–105.

[27] Y. Xiang, Z. Chen, Z. Chen, Z. Fang, H. Hao, J. Chen, Y. Liu, Z. Wu, Q. Xuan, and X. Yang, "Open DNN box by power side-channel attack," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 11, pp. 2717–2721, Nov. 2020, doi: 10.1109/TCSII.2020.2973007.

[28] A. R. Javed, M. O. Beg, M. Asim, T. Baker, and A. H. Al-Bayatti, "AlphaLogger: Detecting motion-based side-channel attack using smartphone keystrokes," *J. Ambient Intell. Humanized Comput.*, vol. 14, no. 5, pp. 4869–4882, May 2023, doi: 10.1007/s12652-020-01770-0.

[29] M. Randolph and W. Diehl, "Power side-channel attack analysis: A review of 20 years of study for the layman," *Cryptography*, vol. 4, no. 2, p. 15, May 2020, doi: 10.3390/cryptography4020015.

[30] H. Wang and E. Dubrova, "Tandem deep learning side-channel attack against FPGA implementation of AES," in *Proc. IEEE Int. Symp. Smart Electron. Syst.*, Dec. 2020, pp. 147–150, doi: 10.1109/iSES50453.2020.00041.

[31] W. Shan, S. Zhang, J. Xu, M. Lu, L. Shi, and J. Yang, "Machine learning assisted side-channel-attack countermeasure and its application on a 28-nm AES circuit," *IEEE J. Solid-State Circuits*, vol. 55, no. 3, pp. 794–804, Mar. 2020, doi: 10.1109/JSSC.2019.2953855.

[32] R. Wang, H. Wang, E. Dubrova, and M. Brisfors, "Advanced far field EM side-channel attack on AES," in *Proc. 7th ACM Cyber-Phys. Syst. Secur. Workshop*, 2021, pp. 29–39.

[33] A. Kumar, P. Reddy, and J. Jhurani, "Design of a novel deep learning methodology for IoT botnet based attack detection," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 11, no. 2023, pp. 4922–4927, 2023.

[34] J. Cho, T. Kim, S. Kim, M. Im, T. Kim, and Y. Shin, "Real-time detection for cache side channel attack using performance counter monitor," *Appl. Sci.*, vol. 10, no. 3, p. 984, Feb. 2020, doi: 10.3390/app10030984.

[35] M. Arsath K F, V. Ganesan, R. Bodduna, and C. Rebeiro, "PARAM: A microprocessor hardened for power side-channel attack resistance," in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, Dec. 2020, pp. 23–34, doi: 10.1109/HOST45689.2020.9300263.

[36] C. Luo, Z. Wen, L. Gao, and X. Cui, "Side-channel attacks based on deep learning," in *Proc. Int. Conf. Adv. Artif. Intell. Appl.*, Wuhan, China, Nov. 2023, pp. 33–37, doi: 10.1145/3603273.3630505.

[37] 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, doi: 10.1109/TVLSI.2019.2926324.

[38] K. Ngo, E. Dubrova, Q. Guo, and T. Johansson, "A side-channel attack on a masked IND-CCA secure saber KEM implementation," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 1, pp. 676–707, Aug. 2021, doi: 10.46586/tches.v2021.i4.676-707.

[39] R. Wang, H. Wang, and E. Dubrova, "Far field EM side-channel attack on AES using deep learning," in *Proc. 4th ACM Workshop Attacks Solutions Hardw. Secur.*, Nov. 2020, pp. 35–44, doi: 10.1145/3411504.3421214.

[40] N. Gattu, M. N. Imtiaz Khan, A. De, and S. Ghosh, "Power side channel attack analysis and detection," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, Nov. 2020, pp. 1–7.

[41] D. Wang, Z. Qian, N. Abu-Ghazaleh, and S. V. Krishnamurthy, "PAPP: Prefetcher-aware prime and probe side-channel attack," in *Proc. 56th ACM/IEEE Design Autom. Conf. (DAC)*, Jun. 2019, pp. 1–6.

[42] Y.-S. Won, D.-G. Han, D. Jap, S. Bhasin, and J.-Y. Park, "Non-profiled side-channel attack based on deep learning using picture trace," *IEEE Access*, vol. 9, pp. 22480–22492, 2021, doi: 10.1109/ACCESS.2021.3055833.

[43] D. R. Gnad, J. Krautter, and M. B. Tahoori, "Leaky noise: New side-channel attack vectors in mixed-signal IoT devices," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2019, no. 3, pp. 305–339, 2019.

[44] J. Wan, Y. Bi, Z. Zhou, and Z. Li, "MeshUp: Stateless cache side-channel attack on CPU mesh," in *Proc. IEEE Symp. Secur. Privacy*, May 2022, pp. 1506–1524.

[45] T. Schamberger, J. Renner, G. Sigl, and A. Wachter-Zeh, "A power side-channel attack on the CCA2-secure HQC KEM," in *Proc. 19th Int. Conf.*, 2020, pp. 119–134.

[46] D. Chen, Z. Zhao, X. Qin, Y. Luo, M. Cao, H. Xu, and A. Liu, "MAGLeak: A learning-based side-channel attack for password recognition with multiple sensors in IIoT environment," *IEEE Trans. Ind. Informat.*, vol. 18, no. 1, pp. 467–476, Jan. 2022, doi: 10.1109/TII.2020.3045161.

[47] A. Baksi, "Classical and physical security of symmetric key cryptographic algorithms," in *Proc. IFIP/IEEE 29th Int. Conf. Very Large Scale Integr.*, Oct. 2021, pp. 1–2, doi: 10.1109/VLSI-SoC53125.2021.9606988.

[48] R. Kumar, X. Liu, V. Suresh, H. K. Krishnamurthy, S. Satpathy, M. A. Anders, H. Kaul, K. Ravichandran, V. De, and S. K. Mathew, "A time-frequency-domain side-channel attack resistant AES-128 and RSA-4K crypto-processor in 14-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 56, no. 4, pp. 1141–1151, Apr. 2021, doi: 10.1109/JSSC.2021.3052146.

[49] E. Karimi, Y. Fei, and D. Kaeli, "Hardware/software obfuscation against timing side-channel attack on a GPU," in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, Dec. 2020, pp. 122–131, doi: 10.1109/HOST45689.2020.9300259.

[50] A. Al Arafat, Z. Guo, and A. Awad, "Vr-spy: A side-channel attack on virtual key-logging in vr headsets," in *Proc. IEEE Virtual Reality 3D User Interfaces*, Mar. 2021, pp. 564–572, doi: 10.1109/VR50410.2021.00081.

[51] B. Colombier, V.-F. Dragoi, P.-L. Cayrel, and V. Grosso, "Message-recovery profiled side-channel attack on the classic McEliece cryptosystem," *IACR Cryptol. ePrint Arch.*, vol. 1, pp. 1–24, Nov. 2022.

[52] D. Das, J. Danial, A. Golder, S. Ghosh, A. R. Wdhury, and S. Sen, "Deep learning side-channel attack resilient AES-256 using current domain signature attenuation in 65nm CMOS," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Mar. 2020, pp. 1–4, doi: 10.1109/CICC48029.2020.9075889.

[53] K. E. Narayana and K. Jayashree, "Survey on cross virtual machine side channel attack detection and properties of cloud computing as sustainable material," *Mater. Today, Proc.*, vol. 45, pp. 6465–6470, Aug. 2021, doi: 10.1016/j.matpr.2020.11.283.

[54] S. Kumar, V. A. Dasu, A. Baksi, S. Sarkar, D. Jap, J. Breier, and S. Bhasin, "Side channel attack on stream ciphers: A three-step approach to state/key recovery," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2, pp. 166–191, Feb. 2022, doi: 10.46586/tches.v2022.i2.166-191.

[55] T. Huo, X. Meng, W. Wang, C. Hao, P. Zhao, J. Zhai, and M. Li, "Bluethunder: A 2-level directional predictor based side-channel attack against sgx," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2020, no. 1, pp. 321–347, 2020, doi: 10.13154/tches.v2020.i1.321-347.

[56] A. R. Javed, S. U. Rehman, M. U. Khan, M. Alazab, and H. U. Khan, "Betalogger: Smartphone sensor-based side-channel attack detection and text inference using language modeling and dense MultiLayer neural network," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 20, no. 5, pp. 1–17, Sep. 2021, doi: 10.1145/3460392.

[57] K.-S. Chong, J.-S. Ng, J. Chen, N. K. Z. Lwin, N. A. Kyaw, W.-G. Ho, J. Chang, and B.-H. Gwee, "Dual-hiding side-channel-attack resistant FPGA-based asynchronous-logic AES: Design, countermeasures and evaluation," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 11, no. 2, pp. 343–356, Jun. 2021, doi: 10.1109/JETCAS.2021.3077887.

[58] A. A. Ahmed, R. A. Salim, and M. K. Hasan, "Deep learning method for power side-channel analysis on chip leakages," *Elektronika Ir Elektrotechnika*, vol. 29, no. 6, pp. 50–57, Dec. 2023, doi: 10.5755/j02.eie.34650.

[59] F. Lemarchand, C. Marlin, F. Montreuil, E. Nogues, and M. Pelcat, "Electro-magnetic side-channel attack through learned denoising and classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2020, pp. 2882–2886, doi: 10.1109/ICASSP40776.2020.9053913.

[60] S. Liu and W. Yi, "Task parameters analysis in schedule-based timing side-channel attack," *IEEE Access*, vol. 8, pp. 157103–157115, 2020, doi: 10.1109/ACCESS.2020.3019323.

[61] X. Wang and W. Zhang, "Cracking randomized coalescing techniques with an efficient profiling-based side-channel attack to GPU," in *Proc. 8th Int. Workshop Hardw. Architectural Support Secur. Privacy*, Jun. 2019, pp. 1–8, doi: 10.1145/3337167.3337169.

[62] V.-P. Hoang, N.-T. Do, and V. S. Doan, "Performance analysis of deep learning based non-profiled side channel attacks using significant Hamming weight labeling," *Mobile Netw. Appl.*, vol. 28, no. 3, pp. 1187–1196, Jun. 2023, doi: 10.1007/s11036-023-02128-4.

[63] Y.-E. Berreby, "Prim: Deep learning for side-channel attacks on AES," *A Preprint*, vol. 1, pp. 1–12, Sep. 2022.

[64] H. Wang and E. Dubrova, "Tandem deep learning side-channel attack on FPGA implementation of AES," *Social Netw. Comput. Sci.*, vol. 2, no. 5, pp. 1–12, Sep. 2021, doi: 10.1007/s42979-021-00755-w.

[65] Z. Liu, Z. Wang, and M. Ling, "Side-channel attack using word embedding and long short term memories," *J. Web Eng.*, vol. 21, pp. 285–306, Jan. 2022, doi: 10.13052/jwe1540-9589.2127.

[66] M. Karthikeyan and V. Selvan, "FPGA centric attention based deep learning network evoked chaotic encryption to mitigate side channel attacks," *Proc. Bulgarian Acad. Sci.*, vol. 76, no. 6, pp. 936–945, Jun. 2023.

[67] L. Wu, G. Perin, and S. Picek, "Hiding in plain sight: Non-profiling deep learning-based side-channel analysis with plaintext/ciphertext," *Cryptol. ePrint Arch.*, vol. 1, pp. 1–28, Jul. 2023.

[68] F. Hu, H. Wang, and J. Wang, "Multi-leak deep-learning side-channel analysis," *IEEE Access*, vol. 10, pp. 22610–22621, 2022, doi: 10.1109/ACCESS.2022.3152831.

[69] T. Kubota, K. Yoshida, M. Shiozaki, and T. Fujino, "Deep learning side-channel attack against hardware implementations of AES," *Microprocessors Microsystems*, vol. 87, Nov. 2021, Art. no. 103383, doi: 10.1016/j.micpro.2020.103383.

[70] G. Li, C. Liu, H. Yu, Y. Fan, L. Zhang, Z. Wang, and M. Wang, "SCNet: A neural network for automated side-channel attack," 2020, *arXiv:2008.00476*.

[71] A. Zakaria and L. Qadri, "The effectiveness of URL features on phishing emails classification using machine learning approach," *Asia–Pacific J. Inf. Technol. Multimedia*, vol. 1, no. 1, pp. 49–58, 2022.

[72] J. Wan, Y. Bi, Z. Zhou, and Z. Li, "Volcano: Stateless cache side-channel attack by exploiting mesh interconnect," 2021, *arXiv:2103.04533*.

[73] T. Kamucheka, M. Fahr, T. Teague, A. Nelson, D. Andrews, and M. Huang, "Power-based side channel attack analysis on PQC algorithms," *Cryptol. ePrint Arch.*, vol. 1, p. 9, Oct. 2021.

[74] F. Durvaux and M. Durvaux, "SCA-pitaya: A practical and affordable side-channel attack setup for power leakage-based evaluations," *Digit. Threats, Res. Pract.*, vol. 1, no. 1, pp. 1–16, Mar. 2020, doi: 10.1145/3371393.

[75] J. Danial, D. Das, A. Golder, S. Ghosh, A. Raychowdhury, and S. Sen, "EM-X-DL: Efficient cross-device deep learning side-channel attack with noisy EM signatures," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 18, no. 1, pp. 1–17, Jan. 2022, doi: 10.1145/3465380.

[76] D.-H. Seo, M. Nath, D. Das, S. Ghosh, and S. Sen, "Enhanced detection range for EM side-channel attack probes utilizing co-planar capacitive asymmetry sensing," in *Proc. Design, Autom. Test Eur. Conf. Exhibition*, Feb. 2021, pp. 1016–1019, doi: 10.23919/DATE51398.2021.9474155.

[77] F. Dehkordi, I. Kalantari, and K. Aghazarian, "Internet of Things (IoT) intrusion detection by machine learning (ML): A review," *Asia–Pacific J. Inf. Technol. Multimedia*, vol. 12, no. 1, pp. 13–38, 2023.

[78] D. Das, S. Ghosh, A. Raychowdhury, and S. Sen, "EM/Power side-channel attack: White-box modeling and signature attenuation countermeasures," *IEEE Des. Test. IEEE Des. Test. Comput.*, vol. 38, no. 3, pp. 67–75, Jun. 2021, doi: 10.1109/MDAT.2021.3065189.

[79] H. Gupta, S. Mondal, R. Majumdar, N. S. Ghosh, S. S. Khan, N. E. Kwanyu, and V. P. Mishra, "Impact of side channel attack in information security," in *Proc. Int. Conf. Comput. Intell. Knowl. Economy*, Dec. 2019, pp. 291–295, doi: 10.1109/ICCIKE47802.2019.9004435.

[80] D. K. Shukla, V. K. R. Dwivedi, and M. C. Trivedi, "Encryption algorithm in cloud computing," *Mater. Today, Proc.*, vol. 37, pp. 1869–1875, Sep. 2021, doi: 10.1016/j.matpr.2020.07.452.

**AMJED A. AHMED** (Senior Member, IEEE) received the B.Sc. degree in computer science from the University of Baghdad and the M.Sc. degree in computer science from Binary University, Kuala Lumpur, Malaysia, in 2012. He is currently pursuing the Ph.D. degree with University Kebangsaan Malaysia, focusing on artificial intelligence. Previously, from July 2013 to July 2022, he was a Lecturer with the Imam Al-Kadhum College, Baghdad, Iraq.

**MOHAMMAD KAMRUL HASAN** (Senior Member, IEEE) received the Ph.D. degree in electrical and communication engineering from the Faculty of Engineering, International Islamic University, in Malaysia 2016. He is currently an Associate Professor. He heads the Network and Communication Technology Laboratory, Faculty of Information Science and Technology, Center for Cyber Security, Universiti Kebangsaan Malaysia (UKM). His expertise lies in cutting-edge information-centric networks, computer networks, data communication and security, mobile networks, privacy protection, cyber-physical systems, industrial IoT, transparent AI, and electric vehicle networks. With over 230 indexed papers published in reputable journals and conference proceedings, he is a member of the Institution of Engineering and Technology and the Internet Society. He is a certified professional technologist in Malaysia. His contributions extend to his role as the IEEE Student Branch Chair, from 2014 to 2016, and his active participation in various events, workshops, and trainings for IEEE and IEEE Humanity programs in Malaysia. He also serves as an editorial member in distinguished high-impact journals, such as IEEE, IET, and Elsevier and takes on roles like the general chair, the co-chair, and a speaker for conferences and workshops to promote knowledge sharing and learning within the academic and societal realms.

**AZANA H. AMAN** (Member, IEEE) holds the B.Eng., M.Sc., and Ph.D. degrees in computer and information engineering from International Islamic University Malaysia, Malaysia. Presently, she serves as a Senior Lecturer at the Research Center for Cyber Security within the Faculty of Information Science and Technology (FTSM), The National University of Malaysia, Malaysia. Her research focuses on computer systems, networking, and information and network computing.

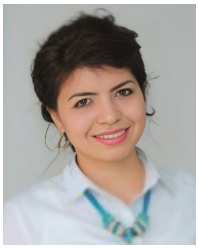
**NURHIZAM SAFIE** (Associate Member, IEEE) is currently the Dean (Industry and Community Partnership and Income Generation) with the Research Center for Software Technology and Management (SOFTAM), Faculty of Information Science and Technology (FTSM), Universiti Kebangsaan Malaysia. Specialization in e-learning technology, strategic information systems, information systems adoption, and diffusion. He is the Chairman of the Digital Transformation Strategic Organization (PSTD).

**SHAYLA ISLAM** (Senior Member, IEEE) received the B.Sc. degree in computer science and engineering from International Islamic University Chittagong, Bangladesh, the M.Sc. degree from the Department of Electrical and Computer Engineering (ECE), International Islamic University Malaysia (IIUM), in 2012, and the Ph.D. degree in engineering from the ECE Department, IIUM, in 2016, under Malaysian International Scholarship. She is currently an Associate Professor with UCSI University, Malaysia. She was awarded a Silver Medal for her research work with International Islamic University Malaysia. Consequently, she was also received the Young Scientist Award for contributing a research paper at the second International Conference on Green Computing and Engineering Technologies, in 2016 (ICGCET'16), organized by the Department of Energy Technology, Aalborg.

**FATIMA A. AHMED** received the B.S. degree in computer science from Sudan University of Science and Technology, in 2004, and the M.S. and Ph.D. degrees in computer science from Al Neelain University, Sudan, in 2007 and 2012, respectively. She joined the Information Systems Department, Prince Sattam Bin Abdulaziz University, Saudi Arabia, in 2013, as an Assistant Professor, from 2013 to 2016, and has been with the Computer Science Department, since 2017. In 2004, she joined the IT Department, Sudanese Company for Telecommunications, Sudan, as a Computer Programmer, where she analyzed, designed, and programmed a set of systems. Her research interests include artificial intelligence, systems and algorithms analysis and design, web applications, and e-learning.

**THOWIBA E. AHMED** received the B.S. degree in computer science and statistics from Al Neelain University, Sudan, in 2004, and the M.S. and Ph.D. degrees in information technology from Al Neelain University, Sudan, in 2007 and 2012, respectively. She has been with the College of Science and Humanities, Imam Abdulrahman Bin Faisal University, Saudi Arabia, since 2013. She was an Assistant Professor with the Computer Science Department and the Head of the Computer Science Department, from 2016 to 2019. In 2004, she joined the National Highway Authority, Sudan, as a Programmer and an Information Network Supervisor. She also joined Al Neelain University as a Collaborative Teaching Assistant, in 2004, and as a Collaborative Assistant Professor, in 2012. In 2012, she joined the Emirates College of Science and Technology, Sudan, as an Assistant Professor and the Head of the Information Technology Department. Her research interests include data science, HCI, systems and algorithms analysis and design, web applications, and e-learning.

**BISHWAJEET PANDEY** (Senior Member, IEEE) received the M.Tech. degree in computer science and engineering from Indian Institute of Information Technology (IIIT), Gwalior, India, and the Ph.D. degree in computer science and engineering from the Gran Sasso Science Institute, Italy. He is currently a Professor with the Department of Intelligent Systems and Cyber Security, Astana IT University, Kazakhstan. He is also a Visiting Professor with Eurasian National University, Astana, Kazakhstan (QS World Rank 355), and UCSI University, Kuala Lumpur, Malaysia (QS World Rank 300). He was the Research Head of the School of Computer Science and Engineering, Jain University (NIRF India Rank 68), Bengaluru, India. He has visited 49 countries, attended 101 conferences, and received the best paper awards in multiple countries. He has authored over 190 articles and has published seven books. He has more than 3300 citations and 28 H-index. He got the Professor of the Year 2023 Award at Lords Cricket Ground by the London Organization of Skills Development (LOSD), U.K.



**LEILA RZAYEVA** received the B.S., M.S., and Ph.D. degrees from L. N. Gumilyov Eurasian National University, Astana, Kazakhstan, in 2015 (QS World Rank 355). She is currently an Assistant Professor and the Head of the Department of Intelligent Systems and Cybersecurity, Astana IT University, Astana, Kazakhstan. She has published more than 40 national/international research articles. Her research interests include control systems and industrial automation, robust control systems, cybersecurity ML, DL and design of control information systems, and the design of neural networks and artificial intelligent systems. She authored or co-authored a significant number of research papers and was an active participant in many conferences. She is a Program Committee Member of IEEE SIST-2024 with Astana IT University and IEEE ICAIC-2024 with the University of Houston, TX, USA. She is the General chair of the RTCSE-2025 Conference with the University of Hawaii, USA. She delivered a Keynote Speech with the IEEE International Conference on AI in Cybersecurity (ICAIC), in February 2024, with the University of Houston, USA. She is invited to deliver a keynote speech at International Muti.