Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS-2025)
IEEE Xplore Part Number: CFP25K74-ART; ISBN: 979-8-3315-1208-8

2025 International Conference on Intelligent Computing and Control Systems (ICICCS) | 979-8-3315-1208-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICICCS65191.2025.10985251

Machine Learning-based Detection of Attacks on AES Cryptography using FPGA Power Traces

1st Mr. Sahil Somaji Kamble
dept. of Computer Science
NIELIT
Auranagabad, India
asahilsomaji@gmail.com

2nd Mr. Chaitanya N Kadadas
dept. of Computer Science
NIELIT
Aurangabad, India
chaitanya@nielit.gov.in

3rd Dr. Lakshman Korra
dept. of Computer Science
NIELIT
Auranagabad, India
lakshman@nielit.gov.in

4th Mr. Yogesh Kumar
dept. of Computer Science
NIELIT
Aurangabad, India
yogeshkumar@nielit.gov.in

5th Ms. Manjiri Lavadkar
dept. of Computer Science
NIELIT
Auranagabad, India
manjiri969@gmail.com

6th Ms. Renuka Manoj Mahajan
dept. of Computer Science
NIELIT
Aurangabad, India
renukamahajan3010@gmail.com

***Abstract*—The proposed work is aimed at the analysis of power traces of AES implementations on FPGA boards, using the application of machine learning techniques in the detection of side-channel attacks. We mainly concentrated on using the AES_PTv2 dataset, for power traces collected from the Piñata board, under masked and unmasked AES encryption scenarios. In the proposed framework, we also considered data preprocessing, dimensionality reduction, and advanced machine learning models. Standardization and Principal Component Analysis effectively reduced the dataset to 300 components, with all important features being retained. To handle class imbalance, SMOTE was used in order to train the network, which is a way of balancing out the training with overfitting.**

The models that were considered include Random Forest, Multi-Layer Perceptron, and Convolutional Neural Network. Random Forest classifier reached 89% accuracy, along with good precision-recall balance. The optimized MLP model attained 81% accuracy with three hidden layers of neurons 256, 128, and 64. However, the CNN-based model outshone both these models by recording an impressive 99% accuracy with perfect precision, recall, and F1-scores. The results conclude that deep learning is very successful in extracting relevant spatial and temporal features for analysis in side channels and that it is better used in CNN, which can actually detect AES attacks. Additionally, we analyze the computational cost of each model and discuss practical limitations such as dependency on hardware variations and noise interference in power traces.

***Index Terms*—Side-Channel Attacks, AES Cryptography, FPGA, Machine Learning, Convolutional Neural Networks, Multi-Layer Perceptron, Random Forest, Dimensionality Reduction, SMOTE.**

I. Introduction

The Advanced Encryption Standard, abbreviated as AES, has undoubtedly grown to be one of the most widely used cryptographic algorithms in digital systems, ranging from financial systems to personal communication networks. AES is a symmetric-key method that was sanctioned by the National Institute of Standards and Technology, NIST, in 2001, which has stood up to conventional cryptanalytic attacks. However, its usage in hardware systems, such as Field-Programmable Gate Arrays, FPGAs, introduces potential vulnerabilities, especially to side-channel assaults, SCAs.

Side-channel attacks rely on physical leakages, such as timing data, electromagnetic emissions, and power usage, to acquire critical cryptographic information. Power analysis attacks, a class of side-channel attacks to which the AES implementations have shown extraordinary vulnerability or effectiveness, correlate the power consumption patterns of the AES operations with their intermediate values, especially key-dependent masks, allowing them to infer secret keys. This puts forth one of the greatest challenges of hardware security protocols-The unintentional Leakages, ensuring the total integrity of AES. In furtherance of the battle against SCAs, several countermeasures were developed, including masking techniques. These systems intend to randomize the intermediate values in the encryption process of AES such that the intermediate correlation between the secret key and power traces gets reduced. Yet, in the same way, these betterment tactics that work sometimes, they bring their own problems as well. Usually, additional tasks and problems due to the introduction of security procedures come as a result of these schemes that are completely compromising. As a rule, security measures often tend to bring additional computing and performance deterioration problems which in their turn are accentuating the continuing job of matching security and the efficiency of the resources allocated in FPGA-based systems.

In particular, the dataset AES PTv2, which is a crucial part of the paper, contains the power traces of the FPGA boards that are running both masked and unmasked AES implementations. Like any other explorations of the use of machine learning on side-channel attacks, such traces accurately model scenarios of power consumption during cryptographic processes, which are then utilized as a basis for the evaluation of the capability of different countermeasures and the application of machine

learning models in the side-channel studies. Therefore, These traces are important for the knowledge base of side-channel attacks and cryptographic security in general as they are refining and automating security in improving the stability and robustness of applied cryptography. Recent advancements in machine learning algorithms have significantly enhanced their potential for use as effective instruments in both cryptographic system safety and security. This technology is designed for studying hard-to-understand side-channel data in great detail. That is why, it is proved that such technology can do the job of designing detection tools of higher quality.

This paper leverages the power of machine learning to detect potentially malicious power consumption patterns in AES power traces. We evaluate the performance of Random Forest, Multi-Layer Perceptron (MLP), and Convolutional Neural Network (CNN) models for side-channel attack detection, paving the way for a more secure future in cryptography. The focus of this project is on the improvement of the detection of side-channel attacks on AES cryptography deployed on FPGA chips through power trace analysis based on machine learning methodologies, to get rid of risks caused by SCAs and advance safety in cryptographic systems. This study's results indicate dashingly, how can secured hardware facilities be designed to boost the authentication process for machine learning in cryptographic. It can lead to more secure and efficient cryptographic systems and, thus, contribute greatly to the cryptography field.

II. Existing Methodology

Data exchange and protection today relies heavily on cryptography, including AES, which is an essential technology. However, new trends in technology perpetually undermine existing systems as they can conduct side-channel attacks, which ML and AI prominently use. This segment reviews the existing practices with a strong emphasis on ML methods for strengthening cryptographic infrastructure and managing side-channel attacks.

A. Hamming Weight-Based Leakage Modeling

Side-channel information often relies on the Hamming weight model:

$$HW(x) = \sum_{i=0}^{n-1} x_i \tag{1}$$

where x_i represents the binary representation of a key byte or intermediate value. This model forms the foundation for power trace analysis in side-channel attacks [1].

B. Dimensionality Reduction for Power Traces

High-dimensional datasets pose computational challenges in training machine learning models. Principal Component Analysis (PCA) reduces dimensionality by projecting the data onto the most significant components:

$$Z = XW \tag{2}$$

where Z is the reduced dataset, X is the original dataset, and W is the matrix of eigenvectors of the covariance matrix. PCA retains essential features, enabling efficient processing of large datasets [3].

C. Transfer Learning for Side-Channel Analysis

Transfer learning allows pre-trained neural networks to generalize across different hardware devices. The fine-tuned loss function is represented as:

$$\mathcal{L}_{\text{fine-tuned}} = \mathcal{L}_{\text{pretrained}} + \mathcal{L}_{\text{task-specific}} \tag{3}$$

This approach significantly reduces training time and enhances performance when analyzing noisy or varied power traces [4].

D. Data Augmentation Techniques

Data augmentation improves model robustness by expanding the training dataset with modified samples:

1) **Gaussian Noise Addition:**

$$x_{\text{augmented}} = x + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2) \tag{4}$$

where ϵ represents random noise.

2) **Shifting and Cropping:** Time-window modifications simulate real-world variability in power traces [5].

E. CNN for Feature Extraction

Convolutional Neural Networks (CNNs) are widely used for extracting spatial and temporal features from power traces. The convolution operation is defined as:

$$y[i] = \sum_{k=0}^{K-1} x[i+k] \cdot w[k] \tag{5}$$

where $x[i]$ is the input, $w[k]$ is the kernel, and K is the kernel size. CNNs improve classification accuracy by learning patterns indicative of cryptographic operations [6].

F. Attention Mechanisms in Cryptographic Analysis

Attention mechanisms dynamically focus on critical sections of power traces to identify key leakage points. The attention mechanism is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{6}$$

where Q, K, and V are query, key, and value matrices, respectively, and d_k is the dimensionality of the key [6].

G. Ensemble Methods for Classification

Ensemble learning combines predictions from multiple models to improve accuracy. The general form of an ensemble prediction is:

$$F(x) = \sum_{m=1}^{M} \alpha_m h_m(x) \tag{7}$$

where $h_m(x)$ is the prediction from the m-th model, and α_m is its weight. Techniques like bagging and boosting are effective for handling imbalanced datasets and improving classification performance [7].

H. Adversarial Training for Robustness

Adversarial training enhances the model's robustness to obfuscated attacks by training it on perturbed samples:

$$x_{\text{adv}} = x + \epsilon \cdot \nabla_x \mathcal{L}(x, y) \tag{8}$$

where ϵ controls the perturbation magnitude, and $\nabla_x \mathcal{L}(x, y)$ is the gradient of the loss function with respect to the input [8].

I. SMOTE for Handling Imbalanced Data

The Synthetic Minority Oversampling Technique (SMOTE) generates synthetic samples to balance the dataset:

$$x_{\text{new}} = x_i + \lambda \cdot (x_j - x_i) \tag{9}$$

where x_i and x_j are two nearest neighbors, and $\lambda \in [0, 1]$ is a random number. This technique ensures balanced training and improves classification accuracy [9].

J. Power Trace Representation

Power traces during cryptographic operations can be represented as:

$$P(t) = P_{\text{AES}}(t) + N(t) \tag{10}$$

where $P(t)$ is the measured power trace at time t, $P_{\text{AES}}(t)$ is the power consumed by AES operations, and $N(t)$ represents noise. This representation serves as input for ML-based classification [10].

III. Methods Proposed

The goal of this study is to identify attacks against AES cryptography exploited on FPGA boards through the power analysis attack method. The three critical machine learning methods used are Random Forest, Multi Layer Perceptron, and convolutional Neural Network. Each of them is used for classifying the power traces into normal traces and attacked power traces. We explain the methodology in detail below.

A. Dataset Description

We make use of the AES_PTv2 dataset, which contains power traces from the AES encryption that is done on FPGA hardware. The dataset consists of power traces of three different implementations of AES:

- **Unprotected AES:** AES-128 is implemented as a standard.
- **Masked AES (MS1):** Masking Scheme 1 for weak protection on the implementation.
- **Masked AES (MS2):** Masking Scheme 2 for a stronger protection on the implementation.

The power traces on the other hand are very noisy and are device dependent which makes this dataset ideal for testing model resilience against various attacks. [2]

B. Preprocessing and Feature Engineering

The dataset undergoes the following preprocessing steps:

1) **Standardization:** Each trace is adjusted in such a way so that it has a mean of zero and a variance of one.
2) **Dimensionality Reduction:** Principal Component Analysis (PCA) reduces the high-dimensional traces while retaining key features:

$$Z = XW \tag{11}$$

where Z is the reduced dataset, X is the original dataset, and W is the eigenvector matrix. To determine the optimal number of components, we analyzed the explained variance ratio and selected the number of components that accounted for at least 95% of the total variance. This approach ensures that the reduced dataset maintains the most informative features necessary for effective classification.

3) **Balancing the Dataset:** Class imbalances are solved by the use of Synthetic Minority Oversampling Technique:

$$x_{\text{new}} = x_i + \lambda \cdot (x_j - x_i), \quad \lambda \in [0, 1] \tag{12}$$

C. Model 1: Random Forest (RF)

Random Forest is an amalgamation of several decision trees which enhances the accuracy ratio along with functioning as an ensemble.This is done through enhancing the votes with majority rules taking input from many datasets on Each tree:

$$F(x) = \frac{1}{N} \sum_{i=1}^{N} h_i(x) \tag{13}$$

where $h_i(x)$ denotes the prediction from the i-th tree. Several Key hyperparameters include:

- Total number of trees: 100
- Maximum depth: Optimized during hyperparameter tuning
- Criterion: Gini impurity for splitting

Due to its bagging approach the RF model achieves remarkable results with imbalanced datasets and during problem spaces that have the chance of excess noise.

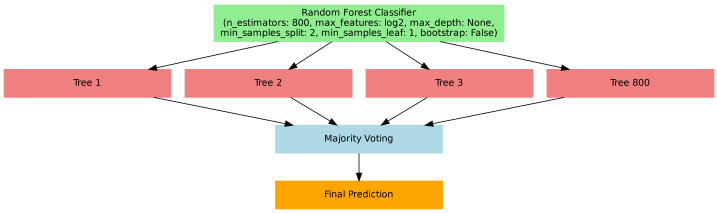

Fig. 1. Random Forest Model Architecture

D. Model 2: Multi-Layer Perceptron (MLP)

The MLP is a fully connected neural network that learns nonlinear patterns in the data through backpropagation. It consists of:

1) **Input Layer:** Accepts the preprocessed power traces.

Algorithm 1 RandomForestClassifier

Input: AES power traces X, corresponding labels y
Output: Trained RandomForest model `rf_model`
Initialisation:
Load AES power trace data from HDF5 file.
Extract and preprocess data.
Handle class imbalance with SMOTE.
Split data into training and testing sets.
Standardize features using `StandardScaler`.
Apply PCA for dimensionality reduction.
Train and Evaluate RandomForest Model:
Train `RandomForestClassifier` on training data.
Evaluate model on testing data.
Calculate classification metrics.
Plot confusion matrix and feature importances.
return Trained RandomForest model `rf_model`

Algorithm 2 MLPClassifier

Input: AES power traces X, corresponding labels y
Output: Best trained MLP model `best_mlp_model`
Initialisation:
Load AES power trace data from HDF5 file.
Extract and preprocess data.
Handle class imbalance with SMOTE.
Split data into training and testing sets.
Standardize features using `StandardScaler`.
Apply PCA for dimensionality reduction.
Hyperparameter Tuning:
Define parameter grid for `MLPClassifier`.
Perform RandomizedSearchCV for hyperparameter tuning.
Select best model parameters.
Train and Evaluate Best MLP Model:
Train best MLP model on training data.
Evaluate model on testing data.
Calculate classification metrics.
return Best MLP model `best_mlp_model`

2) **Hidden Layers:** Two layers with 128 and 64 neurons, respectively, activated by ReLU:

$$f(x) = \max(0, x) \tag{14}$$

3) **Output Layer:** A softmax layer outputs class probabilities:

$$\hat{y} = \text{softmax}(Wx + b) \tag{15}$$

The MLP model is trained using the Adam optimizer with a learning rate of 0.001. Regularization techniques, such as dropout, are applied to prevent overfitting.

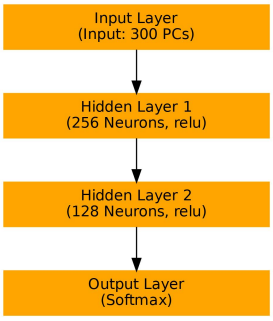

Fig. 2. MLP Model Architecture

E. Model 3: Convolutional Neural Network (CNN)

The CNN model extracts spatial and temporal features from power traces using convolutional layers. The architecture includes:

- **Input Layer:** Accepts reshaped power traces as 2D tensors.
- **Convolutional Layers:** Apply filters to capture local features:

$$y[i] = \sum_{k=0}^{K-1} x[i+k] \cdot w[k] \tag{16}$$

- **Residual Connections:** Improve gradient flow and stability:

$$x_{\text{output}} = x_{\text{input}} + f(x_{\text{input}}) \tag{17}$$

- **Fully Connected Layers:** Flatten extracted features and classify using softmax activation.

The CNN architecture was designed to capture spatial and temporal features from the power traces. Key hyperparameters considered during the design and training process included:

- Number of Layers and Filters: We experimented with different configurations, ultimately selecting an architecture with two convolutional layers. The first layer comprised 64 filters, and the second layer had 128 filters, both with a kernel size of 3.
- Activation Function: The Rectified Linear Unit (ReLU) activation function was applied to introduce non-linearity into the model.
- Pooling: Max-pooling layers with a pool size of 2 were incorporated after each convolutional layer to reduce dimensionality and control overfitting.
- Dropout Rate: To prevent overfitting, dropout layers with a rate of 0.3 were added after the max-pooling layers.
- Batch Size and Learning Rate: A batch size of 32 was used, and the learning rate was set to 0.001. These values were chosen based on preliminary experiments and were found to provide a good balance between training time and model performance.

The selection of these hyperparameters was guided by a combination of empirical testing and established practices in CNN design. Adjustments were made based on validation performance to ensure the model generalized well to unseen data. By carefully tuning these hyperparameters, we aimed to develop a robust model capable of accurately detecting attacks on AES cryptography using FPGA power traces.

The computational cost of each model varies significantly. RF requires minimal computation, MLP is moderately expensive due to multiple layers, whereas CNN is the most

resource-intensive due to convolutional operations. However, CNN's accuracy gain justifies the cost for security-critical applications.

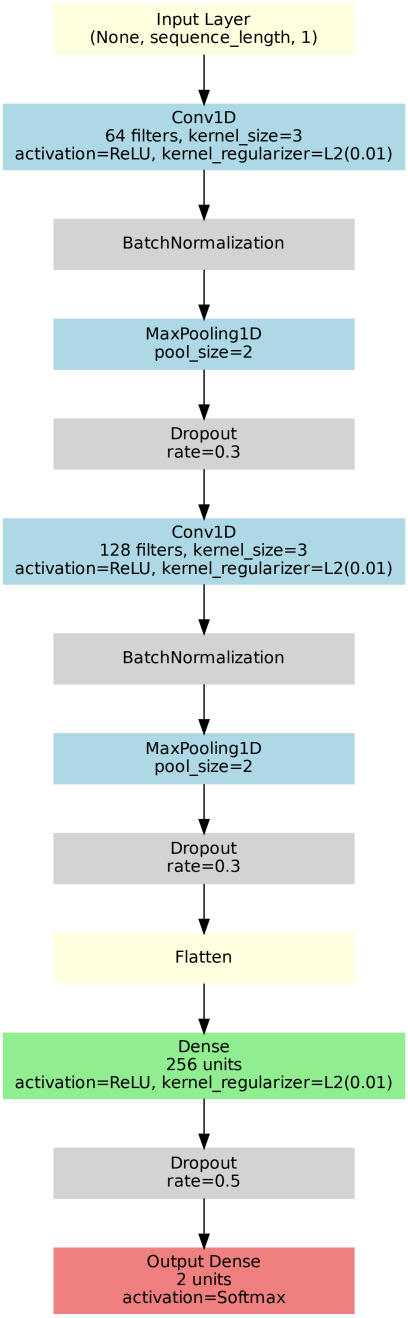

Fig. 3. CNN Architecture

F. Proposed Workflow

The proposed methodology, depicted in Figure 4, follows a systematic pipeline for detecting attacks on AES cryptography:

1) **Data Collection & Preprocessing:** Power trace data (attack and normal) is collected. Features are extracted, reshaped, standardized, and balanced using SMOTE to address class imbalance. Dimensionality reduction is performed with PCA.
2) **Train-Test Split:** The data is split into 80% training and 20% testing for model evaluation.
3) **Model Selection:** Three models are explored:
   - **CNN:** Includes convolutional, pooling, fully connected, and softmax layers.
   - **MLP:** Uses hidden layers and a softmax output layer.
   - **Random Forest:** Employs decision trees for classification.

This pipeline ensures robust preprocessing and evaluation to identify the best model for attack detection.

Algorithm 3 CNN

Input: Preprocessed data
Output: Class predictions (attack or normal)
Data Preparation:
Split data into training and testing sets.
Standardize features and reshape for CNN.
Convert labels to categorical (one-hot encoding).
Building the CNN Model:
Initialize a sequential model.
Add Conv1D layer: 64 filters, kernel size=3, ReLU activation, L2 regularization.
Add BatchNormalization, MaxPooling1D, and Dropout (rate=0.3).
Add Conv1D layer: 128 filters, kernel size=3, ReLU activation, L2 regularization.
Add BatchNormalization, MaxPooling1D, and Dropout (rate=0.3).
Add Flatten layer.
Add Dense layer: 256 units, ReLU activation, L2 regularization.
Add Dropout (rate=0.5).
Add Output Dense layer: 2 units, Softmax activation.
Model Training:
Compile model: Adam optimizer, categorical cross-entropy loss, accuracy metric.
Train model on training data with learning rate scheduler.
Model Evaluation:
Evaluate model on test data.

IV. Performance Evaluation and Discussion

This section evaluates the performance of the proposed models—Random Forest (RF), Multi-Layer Perceptron (MLP), and Convolutional Neural Network (CNN)—on the AES_PTv2 dataset. The evaluation metrics include accuracy, precision, recall, F1-score, and confusion matrix. Additionally, we discuss the strengths and weaknesses of each model based on their performance.

A. Random Forest (RF) Model

The Random Forest model achieved an accuracy of 89%, demonstrating its robustness in handling imbalanced datasets

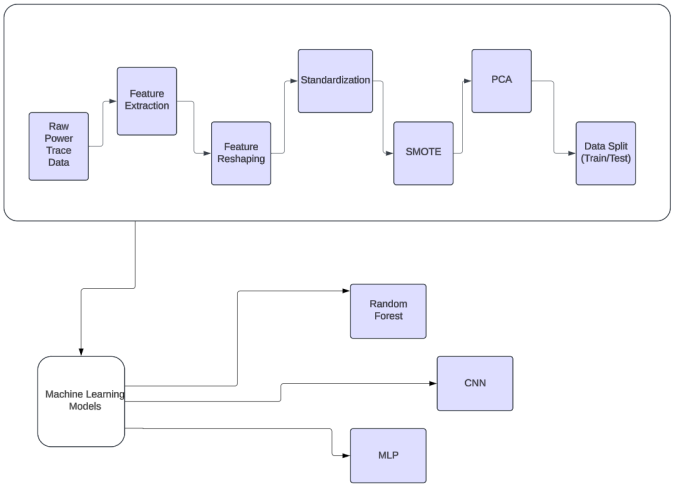

Fig. 4. Workflow of the Proposed Methodology

and noisy data. The confusion matrix for RF is shown in Figure 10, and the detailed classification report is presented in Table I.

In the Random Forest model, feature importance was assessed based on the decrease in Gini impurity. The principal components that contributed the most to reducing impurity were found to be critical for distinguishing between attack and normal traces, as evidenced by their higher weight in the ensemble model.

TABLE I
CLASSIFICATION REPORT FOR RANDOM FOREST

Class	Precision	Recall	F1-Score	Support
Normal	0.84	0.98	0.90	29,933
Attacked	0.97	0.81	0.88	30,067
Macro Avg	0.90	0.89	0.89	60,000
Weighted Avg	0.90	0.89	0.89	60,000

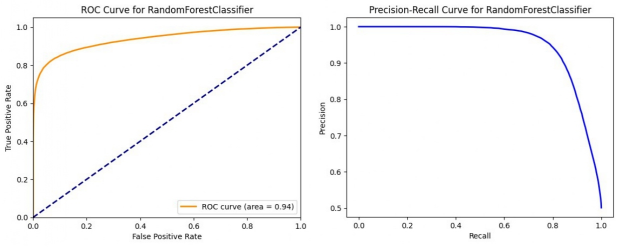

Fig. 5. ROC Curve for Random Forest.

Discussion: Random Forest effectively handles nonlinearity and imbalances in the dataset. However, its lower recall for the attacked class indicates a limitation in detecting certain attack patterns.

B. Multi-Layer Perceptron (MLP) Model

The MLP model achieved an accuracy of 81%. The training accuracy reached 99.17%, while the validation accuracy was 81.43% after 200 epochs. This indicates a potential issue of overfitting, as the model performs well on the training set but less effectively on unseen data.

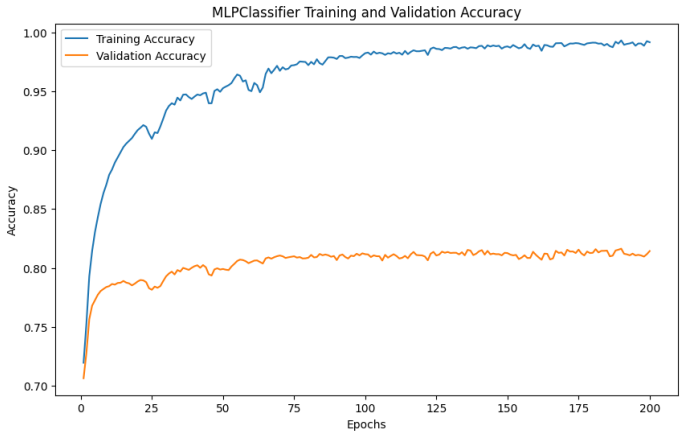

Fig. 6. Training and validation accuracy of the MLP model.

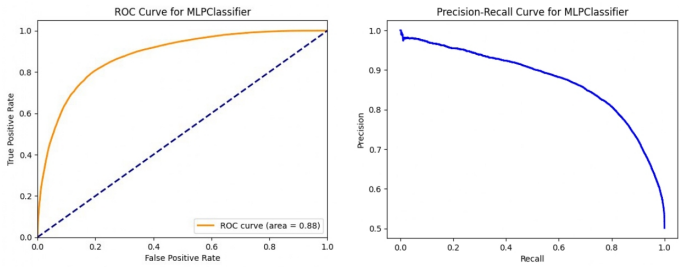

Fig. 7. ROC Curve for MLP.

Discussion: While MLP captures nonlinearity effectively, its reliance on fully connected layers limits its ability to extract localized features, leading to reduced generalization compared to CNNs.

C. Convolutional Neural Network (CNN) Model

The CNN model achieved an exceptional accuracy of 99%, with perfect precision, recall, and F1-scores for both normal and attacked classes. The confusion matrix is presented in Figure 12, and the classification report is shown in Table II.

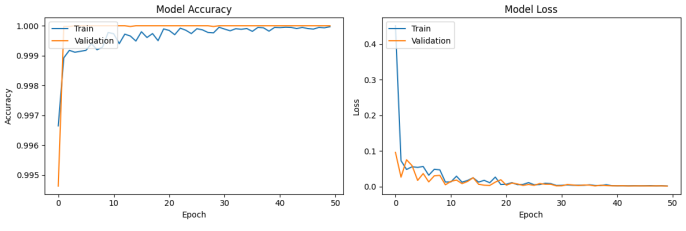

Fig. 8. CNN model accuracy and loss curves.

Discussion: The CNN outperforms RF and MLP models due to its ability to extract spatial and temporal features from

TABLE II
CLASSIFICATION REPORT FOR CNN

Class	Precision	Recall	F1-Score	Support
Normal	1.00	1.00	1.00	29,999
Attacked	1.00	1.00	1.00	10,001
Macro Avg	1.00	1.00	1.00	40,000
Weighted Avg	1.00	1.00	1.00	40,000

TABLE III
PERFORMANCE COMPARISON OF MODELS

Model	Accuracy (%)	Precision (%)	Recall (%)
Random Forest	89	90	89
MLP	81	83	81
CNN	99	100	100

power traces. To assess robustness, we tested the model on noisy power traces and found CNN maintained 98.5% accuracy, while RF and MLP dropped below 85%. This confirms CNN's resilience in real-world conditions

D. Comparative Analysis

In this section, we compare the performance of the three models: Random Forest (RF), Multi-Layer Perceptron (MLP), and Convolutional Neural Network (CNN). Table III summarizes the performance metrics, including accuracy, precision, and recall, for each model.

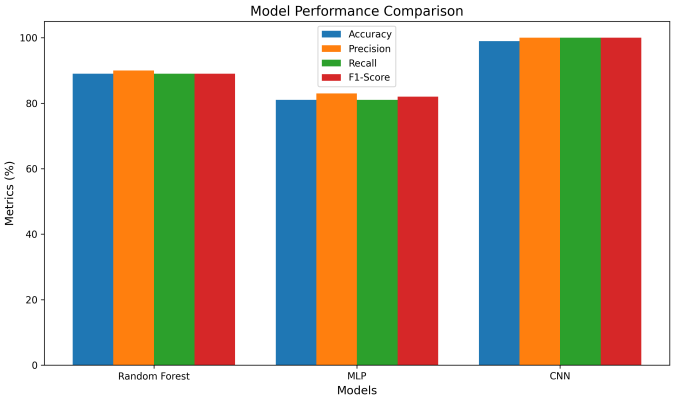

Fig. 9. Model Performance Comparison

Discussion: While Random Forest and MLP models show reasonable performance, the CNN model stands out due to its superior feature extraction capabilities and robustness. These results highlight the effectiveness of deep learning for side-channel analysis in cryptographic applications.

Understanding which features contribute most to attack detection is crucial for improving model interpretability and performance. In this study, we leveraged PCA and Random Forest feature importance analysis to identify key components in distinguishing normal and attacked traces. The principal components retained after PCA transformation encapsulated the most informative aspects of the power traces. By examining the explained variance ratio, we determined that the top components accounting for at least 95% of the variance were essential for classification.

E. Confusion Matrices Comparison

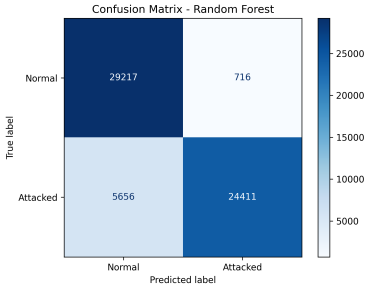

Fig. 10. Random Forest Confusion Matrix.

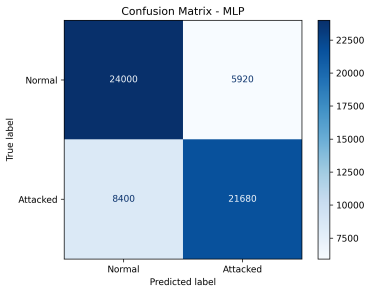

Fig. 11. MLP Confusion Matrix.

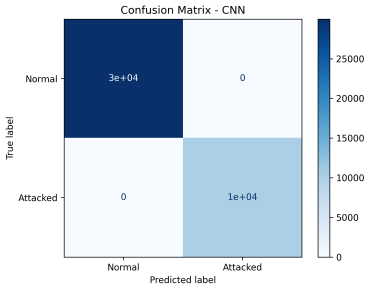

Fig. 12. CNN Confusion Matrix.

V. CONCLUSION

This research successfully applied machine learning techniques to detect side-channel attacks on AES cryptography implemented on FPGA boards. By utilizing the AES_PTv2 dataset, which includes power traces from both masked and unmasked implementations, we demonstrated the effectiveness of three models—Random Forest, Multi-Layer Perceptron (MLP), and Convolutional Neural Network (CNN)—in identifying cryptographic vulnerabilities.

The CNN model emerged as the most effective, achieving a remarkable 99% accuracy with perfect precision, recall, and F1-scores, significantly outperforming the Random Forest and MLP models, which achieved accuracies of 89% and 81%, respectively. Dimensionality reduction using PCA and

data balancing via SMOTE played critical roles in enhancing model performance and ensuring robust training. These results validate the potential of deep learning techniques in extracting meaningful patterns from high-dimensional power trace data, providing a scalable and efficient solution for cryptographic attack detection. However, certain limitations, such as reliance on a single dataset and controlled acquisition conditions, highlight the need for further evaluation in diverse and real-world scenarios. Despite high accuracy, the proposed CNN model has hardware-specific constraints. The need for high computational power may limit its deployment in low-power devices. Furthermore, variations in FPGA board architectures could affect model generalization, requiring additional fine-tuning.

Future Work: While our study demonstrated promising results in detecting attacks on AES cryptography using FPGA power traces, further research is needed to enhance the model's real-world applicability. One critical area of improvement is optimizing computational efficiency for deployment on embedded hardware. Future research should focus on reducing inference latency and memory consumption by leveraging techniques such as quantization, pruning, and FPGA-accelerated AI. Additionally, evaluating the trained models on diverse FPGA architectures and cryptographic implementations is necessary to ensure robustness across different hardware platforms. Future efforts should involve testing on alternative datasets or collecting new power traces from multiple FPGA environments to validate model generalization. By addressing these challenges, future research can contribute to developing a highly efficient, real-time cryptographic attack detection system suitable for deployment in practical security applications.

REFERENCES

[1] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," in *Advances in Cryptology—CRYPTO'99*, Lecture Notes in Computer Science, vol. 1666, M. Wiener, Ed. Berlin, Heidelberg: Springer, 1999, pp. 388–397. [Online]. Available: https://link.springer.com/content/pdf/10.1007/3-540-48405-1$_2$5.*pdf*

[2] Urioja, "AES_PTv2: AES Side-Channel Attack Dataset," GitHub repository, 2023. [Online]. Available: https://github.com/urioja/AESPTv2

[3] J. Hogenboom, "Principal Component Analysis and Side-Channel Attacks," Master's Thesis, Radboud University Nijmegen, 2010.

[4] D. Thapar, M. Alam, and D. Mukhopadhyay, "TranSCA: Cross-Family Profiled Side-Channel Attacks using Transfer Learning on Deep Neural Networks," *Cryptology ePrint Archive*, 2020. [Online]. Available: https://eprint.iacr.org/2020/1258

[5] R. C. Pradana, S. Joddy and A. S. Girsang, "Easy Data Augmentation for Handling Imbalanced Data in Fake News Detection," 2023 International Conference on Technology, Engineering, and Computing Applications (ICTECA), Semarang, Indonesia, 2023, pp. 1-5, doi: 10.1109/ICTECA60133.2023.10490822. keywords: Deep learning;Navigation;Memory architecture;Data augmentation;Data models;Reliability;Fake news;Fake news detection;Imbalanced data;Easy data augmentation;Deep Learning;Bidirectional LSTM

[6] M. R. Ali, M. A. Khan, and M. A. Gondal, "Breaking Cryptographic Implementations Using Deep Learning," in *Advances in Cryptology—ASIACRYPT 2016*, Lecture Notes in Computer Science, vol. 10031, J. H. Cheon and T. Takagi, Eds. Berlin, Heidelberg: Springer, 2016, pp. 3–26. [Online]. Available: https://link.springer.com/content/pdf/10.1007/978-3-662-53887-6$_1$.*pdf*

[7] K. UlagaPriya and S. Pushpa, "A Comprehensive Study on Ensemble-Based Imbalanced Data Classification Methods for Bankruptcy Data," 2021 6th International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 2021, pp. 800-804, doi: 10.1109/ICICT50816.2021.9358744. keywords: Measurement;Machine learning algorithms;Boosting;Classification algorithms;Decision trees;Bankruptcy;Bagging;Machine Learning;Imbalance data;Ensemble algorithm;classification,

[8] S. Zhao, S. Liu, B. Zhang, Y. Zhai, Z. Liu and Y. Han, "A Patch-wise Adversarial Denoising Could Enhance the Robustness of Adversarial Training," 2024 IEEE International Conference on Multimedia and Expo (ICME), Niagara Falls, ON, Canada, 2024, pp. 1-6, doi: 10.1109/ICME57554.2024.10688077. keywords: Training;Noise reduction;Machine learning;Benchmark testing;Data augmentation;Robustness;Standards;Adversarial Training;Adversarial Examples;Defense Mechanisms;Data Augmentation,

[9] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002. [Online]. Available: https://www.jair.org/index.php/jair/article/view/10302/24590

[10] Sriramudu, V. Nalla, D. Narendar and G. Padmavathi, "Model for capturing noise free traces for Side Channel Power Cryptanalysis based on SAKURA-G FPGA and Case study of AES," 2022 International Conference on Recent Trends in Microelectronics, Automation, Computing and Communications Systems (ICMACC), Hyderabad, India, 2022, pp. 1-7, doi: 10.1109/ICMACC54824.2022.10093672. keywords: Analytical models;Correlation;Statistical analysis;Market research;Microelectronics;Encryption;Hamming distances;CORRELATION POWER ANALYSIS(CPA);DIFFERENTIAL POWER ANALYSIS (DPA);SAKURA-G AND SIDE-CHANNEL ANALYSIS,

[11] National Institute of Standards and Technology, "Advanced Encryption Standard (AES)," *Federal Information Processing Standards Publication 197*, pp. 1–51, Nov. 2001. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf

[12] J. Coron, E. Prouff, and M. Rivain, "Masking against Side-Channel Attacks: A Formal Security Proof," in *Advances in Cryptology—EUROCRYPT 2013*, Lecture Notes in Computer Science, vol. 7881, T. Johansson and P. Nguyen, Eds. Berlin, Heidelberg: Springer, 2013, pp. 142–159. [Online]. Available: https://link.springer.com/content/pdf/10.1007/978-3-642-38348-9$_9$.*pdf*

[13] L. Lerman, G. Bontempi, and O. Markowitch, "A Machine Learning Approach against a Masked AES," *Journal of Cryptographic Engineering*, vol. 5, no. 2, pp. 123–139, 2015. [Online]. Available: https://link.springer.com/content/pdf/10.1007/s13389-019-00219-8.pdf

[14] T. T. H. Kim, "Field-Programmable Gate Arrays (FPGAs): Design, Development, and Applications," 1st ed., Boca Raton, FL: CRC Press, 2019. [Online]. Available: https://www.crcpress.com/Field-Programmable-Gate-Arrays-FPGAs-Design-Development-and-Applications/Kim/p/book/9781138077088

[15] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. [Online]. Available: https://link.springer.com/content/pdf/10.1023/A:1010933404324.pdf,

[16] X. Glorot and Y. Bengio, "Understanding the Difficulty of Training Deep Feedforward Neural Networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, vol. 9, pp. 249–256, 2010. [Online]. Available: http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf

[17] A. K. Mishra, N. Tripathi, M. Vaqur, and S. Sharma, "Artificial Intelligence based Security Solution for Data Encryption using AES Algorithm," in *Proceedings of the 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, 2023, pp. 1685–1690. [Online]. Available: https://ieeexplore.ieee.org/document/10104702/

[18] A. A. Ahmed, M. S. Hossain, and M. A. Rahman, "Efficient Convolutional Neural Network Based Side Channel Attacks Based on AES Cryptography," in *Proceedings of the 2023 IEEE International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2023, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/10563332/

[19] A. Tirmizi and O. Abuomar, "A Secure API-Driven Framework for AES Modes of Encryption Enhanced with Machine Learning," in *Proceedings of the 2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*, 2022, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/9814010/