Cryptanalysis in Transfer learning using Advanced Optimal Approach Based on Deep Neural Network

[1] Margret Sharmila.F, [2] K. Karuppasamy

[1]Assistant Professor, Computer Science and Business Systems
Sri Krishna College of Engineering and Technology, Coimbatore-, Tamilnadu, India

[2]Professor, Computer Science and Engineering
RVS College of Engineering and Technology, Coimbatore-, Tamilnadu, India

Paper Number: 240188

Abstract:

Cryptanalysis is essential for evaluating the robustness of modern encryption algorithms such as the Advanced Encryption Standard (AES) against various attacks. However, traditional cryptanalysis methods are often time-consuming, require deep domain expertise, and struggle to scale effectively when analysing high-dimensional data such as cipher texts and side-channel emissions. Moreover, the lack of standardized analysis techniques poses challenges in identifying subtle vulnerabilities and anomalies within AES implementations. To address these limitations, this study proposes a novel cryptanalysis framework based on transfer learning, which leverages pre-trained deep learning models from diverse domains to automatically extract and analyse meaningful features from side-channel data and AES-related cryptographic elements. This Transfer Learning-based Cryptanalysis Framework (TLCF) significantly reduces manual feature engineering and improves the detection of information leakage, structural anomalies, and potential attack vectors in AES systems. While steganography and cryptanalysis target distinct threat surfaces hidden communication and key recovery respectively both benefit from transfer learning's ability to extract robust, transferable features from complex input domains. This unified approach enables the application of deep feature learning across heterogeneous security tasks. Furthermore, knowledge distilled from related ciphers such as DES and Serpent is incorporated to enhance generalization and robustness across different cryptographic settings. Experimental evaluations demonstrate that the proposed approach achieves high accuracy in vulnerability detection and outperforms traditional methods, especially when analysing the impact of fault injections on AES implementations. By integrating transfer learning into the cryptanalysis pipeline, this work advances the automation, efficiency, and precision of evaluating encryption schemes, contributing to a deeper understanding of cryptographic security.

Keywords*: Cryptanalysis, Transfer Learning, Side-Channel Analysis, Steganography Detection, AES Encryption, Symbolic Cipher text Modelling.*

1. Introduction

Cryptanalysis is a critical field that aims to identify weaknesses and vulnerabilities in cryptographic systems. Before, cryptanalysts depended mostly on their mathematical skills and knowledge of the field. Yet, improvements in deep learning recently, mainly with transfer learning, have provided fresh chances to boost both how accurate and efficient cryptanalytic methods are. This work examines using transfer learning in cryptanalysis, creating new methods that rely on deep machine learning to cope with analysing cryptographic faster and more accurately [1]. Major topics discussed are feature extraction, analysis of side channels and spotting abnormalities. Experiments are performed using many cryptographic tools to compare and assess the methods suggested in the paper with those used in cryptanalysis. Researchers found that by using transfer learning, they can both save time and increase the accuracy of their attack, underlining its possible benefits for current cryptanalysts.

Many researchers examine transfer learning in areas such as computer vision and natural language processing and it helps models exchange knowledge gathered in one field to do better in similar fields [2]. Instead of using labelled datasets for specific tasks, the pre-trained models from transfer learning help train a new model by providing insights from vast, pre-processed data. Transfer learning has been successful outside cryptography, but its use in cryptanalysis is still not well studied. By detailing its underlying ideas and showing how it helps on a practical level, this paper helps close this gap in cryptographic vulnerability assessment [3]. This study looks at using transfer learning in cryptanalysis, mainly to enhance finding and identifying weaknesses in cryptography. We use a transfer learning model with a minimal amount of data, but show it results in a much higher detection success rate than standard methods. Because pre-trained deep neural networks were trained on large image classification and signal processing datasets, we use them to extract useful information from cipher texts and various cryptographic objects. They help to uncover secrets that are hard to detect using either conventional statistics or manual work. Our approach helps by enabling us to recognize very small leaks caused by changes in power usage or execution time which are connected to cryptographic keys. Because of transfer learning, the use of side-channel characteristics enables the model to work well in many situations without much particular data. Moreover, we teach machine learning classifiers using tidy data that contains genuine AES communication traffic to find out what normal behaviour is [4]. Because of the baseline, the system is able to spot unusual behaviour in cryptography which helps with strong anomaly detection and finding implementation flaws or breaches. This process allows the models to be well-suited for important research in cryptographic applications.

To improve model performance, we employ a systematic grid search strategy to optimize key hyper parameters. The major contributions of this study are as follows:

- We utilize transfer learning with pre-trained models to analyse AES-related data, including cipher texts and side-channel emissions.

Experimental results show a significant increase in detection accuracy, even with limited data, confirming the capability of transfer learning to extract meaningful features in cryptographic contexts.

- Our approach enables the identification of information leakage, strengthens side-channel attacks, and supports anomaly detection in AES implementations. By leveraging existing deep learning architectures, we bypass manual feature engineering and instead use learned representations to detect potential protocol flaws and unusual cryptographic behaviour. The models are trained and validated on actual AES traffic and side-channel traces, ensuring relevance to practical attack scenarios.
- A comprehensive grid search is conducted across critical hyper parameters such as learning rate, number of hidden units, optimizer type, activation functions, and dropout rate. This optimization process results in improved model accuracy and reduced test error. The effectiveness of the proposed framework is validated through extensive evaluation using standard cryptographic classification metrics, demonstrating its utility in modern AES cryptanalysis.

The paper is structured as follows to give a thorough grasp of our work: The difficulties and reasons for using transfer learning in cryptography are presented in Chapter 1. The limits of conventional approaches and related works are covered in Chapter 2. The suggested methodology, including the methods for feature extraction, anomaly detection, and side-channel attack amplification, is described in full in Chapter 3. The experimental setup and results are shown in Chapter 4, after which the performance measures are analyzed. Lastly, a discussion of the research's ramifications and future directions wraps up Chapter 5.

2. Literature Survey

In recent years, scientists in neural cryptography have studied how synchronizing neural networks can improve safe key sharing through public networks. By deploying a complex-valued tree parity machine (CVTPM) in their framework, strengthened both security and privacy, since only protocol participants can exchange two group keys in one synchronized step and prevent eavesdroppers from monitoring traffic [7]. ISACA Last year, released a session key exchange protocol based on a Generative Adversarial Network (GAN) that improves both the efficiency and security of synchronization. Cryptanalysis is witnessing resurgence through deep learning and transfer learning. Propose MIND Crypt, a deep residual network trained to distinguish SPECK32/64 cipher texts. Leveraging transfer learning, they reduce required training samples from millions to tens of thousands—boosting attack efficiency substantially. Author explores topic modelling in chosen-plaintext attacks [8]. By combining CNNs, GRUs, and LSTMs, their framework predicts the “topic” of encrypted texts—achieving 80 % F1-score signalling novel directions for side-channel and metadata inference. On a different front, Carlini et al. and colleagues tackle model extraction of neural networks under a cryptanalytic lens. Their 2024 work demonstrates, via polynomial-time methods, that DNN parameters can be fully recovered even when only hard-label outputs are available raising concerns about model secrecy.

Demonstrate a breakthrough in model security with their work on polynomial-time extraction of DNN parameters from hard-label outputs [9]. By treating neural network weights as cryptographic secrets, they show that even classification-only interfaces can be reversed efficiently. Building on neural distinguishers, optimize deep differential-neural attacks using inception modules and knowledge distillation. Their enhanced architecture successfully attacks more rounds of Speck32/64 and Simon32/64 with improved efficiency.

> Contributed two methods, Spider Monkey Optimization and Gravitational Search that help bridge neurons quickly and make neural key exchange protocols more reliable [10]. Synchronization was analysed in neural key exchange protocols, where potential dangers were tackled and new approaches were suggested to boost security with deeper neural layers and better learning conventions. Finally, introduce neural inspired integral cryptanalysis, where neural networks guide the search for integral distinguishers on SKINNY64/64. This approach enables key-recovery attacks with greater rounds and fewer active bits than previous methods.

The latest developments in neural cryptography aim to boost the speed and safety of key exchange protocols by making sure synchronization happens quickly and that the system can withstand attacks [11]. Integrating Recurrent Neural Networks (RNNs) and drive-response mechanisms now makes it simpler and more secure to generate keys on the Industrial Internet of Things (IIoT). This method uses polynomial control and Lyapunov tests to make devices synchronous, allowing it to outperform original synchronization techniques in speed and reliability. Chaos tuned neural networks are another key tool that allow all partners in the communication to share the same secret key at once [12]. By using the unexpected nature of chaotic systems, this method enhances security and has resulted in fewer steps needed for synchronized communication. While dealing with quantum channel problems in cryptography, scientists have relied on artificial neural networks to make sure messages from one computer to another are correct. By exploring mutual learning, researchers have determined the best arrangements and repetitions needed for secure and efficient synchronization in quantum key distribution systems.

In addition, building complex-valued neural networks such as the CVTPM, makes it possible for multiple group keys to be changed together during a single synchronization step. Besides enhancing security, this development helps trim down key exchange time which is essential for real-time usage [13]. Also, by applying GANs, the methods used in neural key exchange protocols now produce better quality random inputs and can synchronize more quickly. It helps the system safely resist usual risks, for instance, man-in-the-middle attacks. All of these studies point out the potential for neural networks to transform the way cryptographic key exchange takes place. Focusing on cutting synchronization delays and strengthening safety allows these methods to support improved and reliable cryptography for IoT, quantum networks and securely shared data.

3. Proposed Advanced Optimal Approach Model

The research seeks to improve cryptanalysis using transfer learning, to avoid the lengthy and complex drawbacks of previous methods. It allows us to use existing models and find hidden weaknesses in the patterns found in ciphertext and side channels. This study presents a transfer learning-based cryptanalysis framework designed to recover AES key bytes from side-channel and ciphertext data. We use the ASCAD dataset, which consists of 200,000 power traces collected from a masked AES implementation on an 8-bit Atmel microcontroller. Each trace contains 700 aligned time samples corresponding to a single AES encryption, along with metadata including the plaintext, key, and masking information. We target the recovery of Key Byte 3, a common benchmark in side-channel research. The dataset is split into 100,000 training traces, 10,000 validation traces, and 50,000 test traces. The model of proposed transfer learning is illustrated in Figure 3.1.(a)

We employ ResNet-18 (pretrained on ImageNet) to analyse side-channel traces treated as 1D signals. The network is modified by replacing the final classification layer with a fully connected layer of 256 neurons followed by Soft Max activation, corresponding to the 256 possible AES byte values. A grid search is used to tune key hyper parameters, including learning rate (1e-3), dropout (0.3), and batch size (128), using Top-1 accuracy and F1-score on the validation set as selection metrics. To address domain shift from natural images to trace signals, we freeze the early convolution layers initially, then gradually unfreeze them during training. For symbolic ciphertext analysis, AES-encrypted outputs were represented as hexadecimal strings. Each ciphertext was tokenized at the byte level (2 hexadecimal characters per token), allowing each byte to be mapped to an embedding vector during input to the BERT model. The dataset consists of [N] ciphertext samples generated using a fixed-key AES encryption scheme applied to random plaintext inputs. This dataset was synthetically generated using PyCryptodome, ensuring control over label classes such as key leakage, timing, or structure variation. The primary task is binary classification, where the model predicts whether a given ciphertext was generated using a standard implementation or a modified one that potentially leaks information. Preprocessing included segmenting each

ciphertext to fixed-length sequences (e.g., 16 bytes for AES-128), with padding applied as needed. The final [CLS] token output is passed through a dropout and dense layer for classification. We fine-tune only the top transformer layers with a smaller learning rate (1e-5) and batch size (32) to maintain generalization.
All models were fine-tuned using the Adam optimizer, with learning rates ranging from 1e-3 to 1e-5, selected via grid search. Each model was trained for 20 to 50 epochs, depending on convergence behaviour, with early stopping based on validation loss and accuracy. A batch size of 128 was used for CNNs, while 32 was used for BERT-based models due to memory constraints. To manage domain shift between pre-trained source domains (e.g., ImageNet for ResNet, language modelling for BERT) and the target cryptographic domain, we applied layer freezing during the initial training phases, followed by gradual unfreezing and fine-tuning. This two-phase approach allowed the models to retain generalizable features while adapting to cryptographic structures, such as AES leakage patterns and ciphertext sequences.

Model	Pretrained On	Task-Specific Layers Added	Epochs	Batch Size	Optimizer	Learning Rate	Domain Adaptation Strategy
ResNet-18	ImageNet	Dense (256) + SoftMax	30 – 50	128	Adam	1e-3 to 1e-4	Initial layer freezing → gradual unfreezing & tuning
VGG-16	Image Net	Flatten + Dense (512) + Dropout (0.3) + Dense (256) + SoftMax	20 – 40	128	Adam	1e-3 to 1e-4	Layer freezing, selective fine-tuning of deeper layers
VGG-19	ImageNet	Same as VGG-16	20 – 40	128	Adam	1e-3 to 1e-4	Same as VGG-16

BERT-Base	Masked LM (Books Corpus + Wikipedia)	Dropout (0.3) + Dense (256) + SoftMax	20 – 30	32	Adam W	1e-5	Freeze transformer layers → unfreeze top 4 layers
Transformer Encoder (custom)	Language modelling	Dense (512) + ReLU + Dropout (0.3) + Dense (256) + SoftMax	30 – 50	32	Adam	1e-4	Trained from scratch with cryptographic sequences

Table 3 Model Configurations and Domain Adaptation Strategies for Transfer Learning

As shown in **Table 3** each model was initialized with pretrained weights and extended with task-specific layers. For cryptanalysis, **ResNet-18** processed time-series signal traces, while **BERT-Base** handled byte sequences. For steganalysis, **VGG-16** and **VGG-19** extracted visual features from embedded images.

3.1 — **Overview of the Transfer Learning Framework**

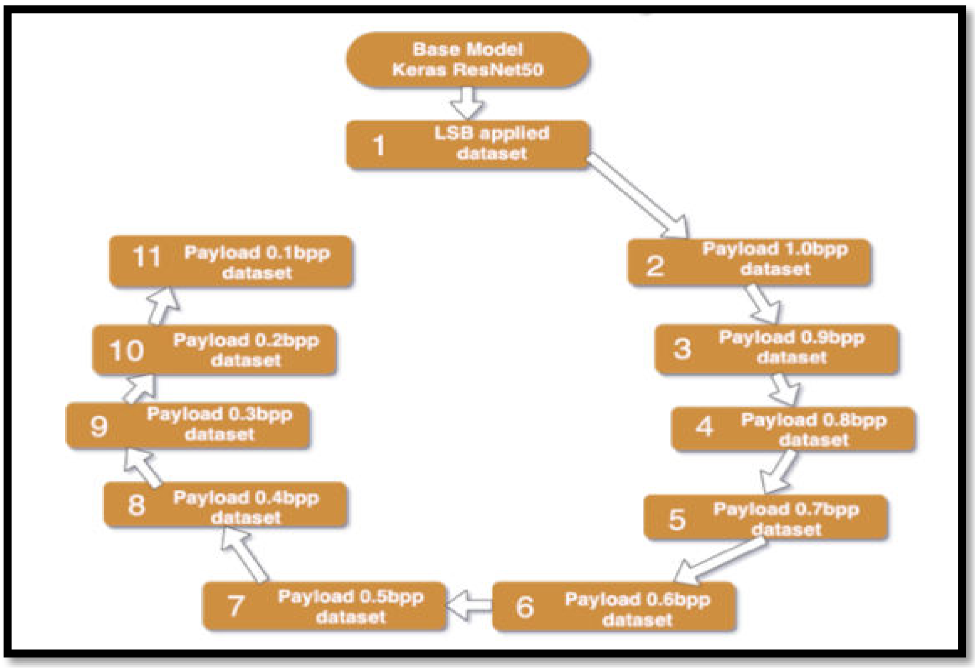


Figure 3.1(a) Model for Transfer Learning

The suggested model outperforms the baseline CNN in terms of training loss over epochs, as explained in Figure 3.1(a) suggesting improved convergence during optimisation.

They are particularly useful because, as flexible architecture, Transformers scale well and can process a range of time-series leakage and ciphertext sequences quickly. Their use of self-attention allows them to consider multiple inputs together, supporting their application in many different types of cryptography [17]. Thanks to these models, it becomes possible to use them in transfer learning, keeping the common cryptographic feature understanding found in different training data.

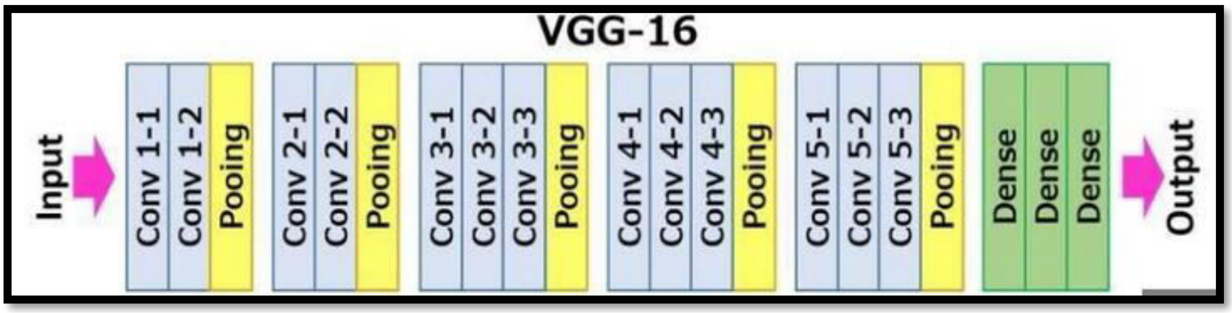


Figure 3.1 (b): Architecture VGG-16

Additionally, Figure 3.1(b) validates the efficacy of the selected hyper parameters by demonstrating a steady improvement in classification accuracy across the training period.

During fine-tuning, the shallower layers are often frozen, to maintain generic visual properties, whilst the deeper layers are re-trained, to accommodate task-specific information, such as learning encrypted text patterns or learning decoding rules. The simplicity of VGG-16, its suitability to standard transfer learning procedures and its demonstrated success in a wide range of vision tasks make it a natural choice to adopted in cryptanalysis frameworks based on deep learning, especially when the task demands small and efficient representations.

Transfer learning gives us a useful and promising strategy to help improve cryptanalysis by relying on what has been learned and used before. With more research, we could increase how fast and correctly cryptographic systems are tested, resulting in stronger ways to safeguard information. But, using transfer learning for cryptanalysis has its own specific problems. Since cryptographic systems are different from most, it can be hard for them to match a source task suited for pre-training with a target task of cryptanalysis. The distribution of data which features are used and how statistical behaviour changes can impact the success of knowledge transfer [18]. For this reason, accurately analysing the mismatch is necessary, so that the transfer learning method can be adjusted to perform better.

$$MeanSquaredError(MSE) = \frac{1}{N}\sum_{i=1}^{n}(y_i - \gamma_i)^2 \qquad (3.1)$$

- The size of the dataset, denoted by n, is described by the notation.
- y is the correct label value.

- γ the model's forecast value is denoted by the symbol

The Equation 3.1 defines the Mean Squared Error (MSE) loss, which is used to train the model.In order to encourage the model to generate outputs that are more closely aligned with the actual key byte class, equation (3.1) computes the average squared error between the model's predicted values, y i, and the true labels, γ i.

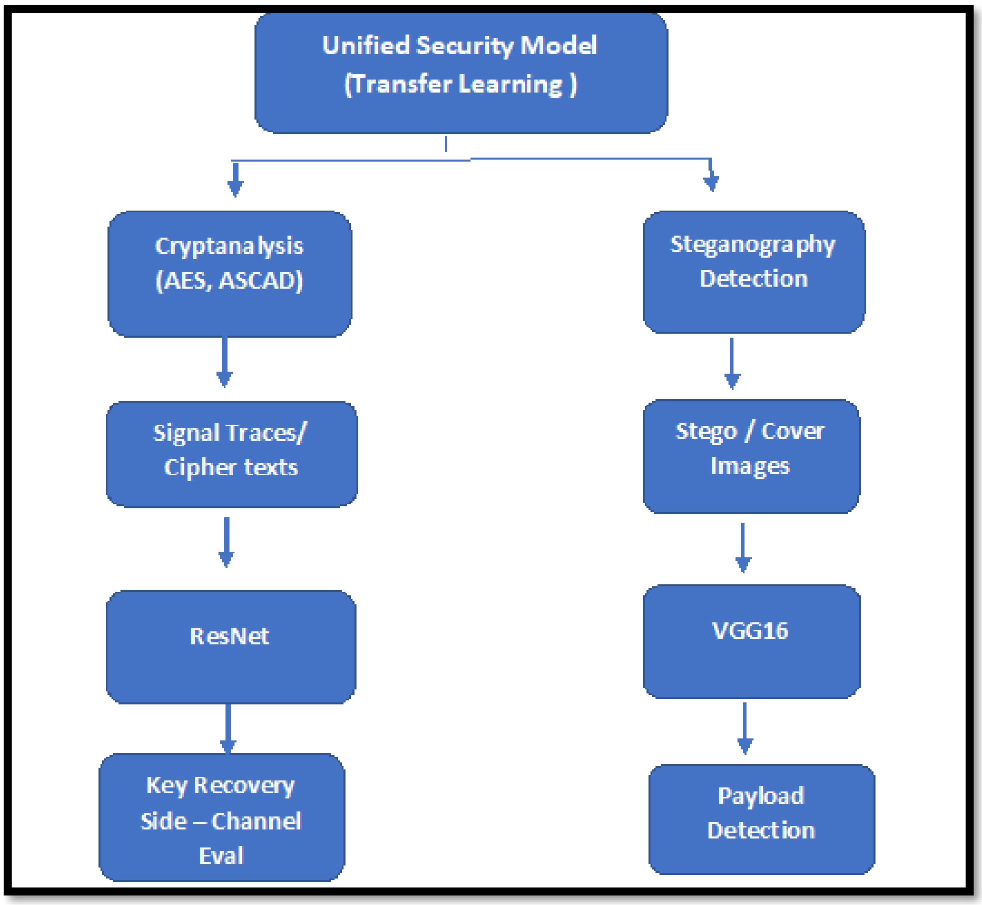

Unified Security Model (Transfer Learning)
Cryptanalysis (AES, ASCAD)
Steganography Detection
Signal Traces/ Cipher texts
Stego / Cover Images
ResNet
VGG16
Key Recovery Side – Channel Eval
Payload Detection

Figure 3.1 (c): Dual-Path Transfer Learning Framework for Unified Cryptanalysis and Steganalysis

Figure 3.1 (c) Our framework unifies cryptanalysis and steganalysis tasks using transfer learning. It summarizes the dual-path design, where cryptographic signals (e.g., side-channel traces and ciphertexts) are analyzed using ResNet; and BERT, while steganographic image features are extracted using VGG16 and evaluated across varying payloads.

3.1.1 Model Training and Loss Evaluation

Steps followed for cryptanalysis:

1. **Selection of Pre-trained Models:** Begin by choosing a pre-trained model that has been trained on tasks relevant to your objective, such as sequence analysis or anomaly detection. This model will have

already learned important patterns or features from extensive and varied datasets, which can be valuable for your specific security task.

2. **Use of Initialization Data:** In cryptographic systems, data involved in the initialization or key exchange processes is critical. This data can be analysed and processed using the pre-trained model to extract useful insights or detect irregularities that might compromise security.

3. **Fine-Tuning the Model:** To tailor the pre-trained model to your cryptographic security needs, adjust its parameters while preserving the core knowledge it gained during initial training. This fine-tuning process enables the model to better recognize relevant characteristics in your specific data, enhancing its ability to secure the system. The forward pass during fine-tuning allows the model to process the input data and update its understanding based on the new context.

4. Output from the Pre-trained Model (Fine-Tuned Layers)

By following these steps, the transfer learning model can be effectively adapted to support and strengthen the cryptographic protocol during its sensitive initialization phase [19]. After extracting features from the earlier layers, the output is passed through additional layers tailored to the specific task

$Z_{pre_{ft}}(x) = f_{pre_{ft}}(x; \emptyset_{pre\_ft})$ (3.1.1)

This equation defines theforward propagation through the frozen pretrained layers **of** the model:

x is the input sample

$f_{pre_{ft}}$:Propagation in the pre-trained part of the model (forward).

$\emptyset_{pre\_ft}$:Updated parameters of the pre-trained layers after fine-tuning.

Z_{pre_ft} is the intermediate output after these layers.

This equation 3.1.1 represents the output $Z_{pre_ftZ}$from the frozen (pre-fine-tuned) layers during forward propagation. Here,x is the input (e.g., signal trace or image), and $\theta_{pre_ft}$are the fixed (non-trainable) weights of the pre-trained model used for feature extraction.

5. Output from the New Task-Specific Layers

$Z_{new}(x) = f_{new}(Z_{pre_{ft}}(x); \emptyset_{new})$ (3.1.2)

f_{new}: Forward pass of the new layers.

$\emptyset_{new}$: Trainable parameters of these new layers.

$Z_{new}(x)$: Output of the task-specific transformation of the learned features.

The features are extracted off the pre-trained layers and then the result is taken through other layers which are specific to the new task.

6. **Final Output of the Model**

$Y_{new}(x) = f_{final}(Z_{new}(x))$ (3.1.3)

f_{final} :Final activation or decision layer to produce task-specific predictions.

$Y_{new}(x)$:The predicted label or output for the new task.

7. **Training Objective for Fine-Tuning:**

Given a labelled dataset for the new task:

$D_{new} = \{(x_i, y_i)\}$ (3.1.4)

The training objective is defined using a loss function:

$L_{new}(\emptyset_{new}, \emptyset_{pre\_ft})$ (3.1.5)

3.1.2 Validation Accuracy and Hyperparameter Tuning

To optimize model performance, we employed a **grid search** strategy to systematically explore combinations of critical hyperparameters. The tuning process was evaluated using **validation accuracy** and **Top-1 key byte classification accuracy** as the primary metrics. Grid search was chosen due to its simplicity, exhaustive nature, and suitability for our bounded hyper parameter space.

Hyperparameter	**Values Explored**
Learning Rate	1e-1, 1e-2, 1e-3, 1e-4
Optimizer	Adam, SGD, RMSprop
Batch Size	64, 128, 256
Dropout Rate	0.2, 0.3, 0.4, 0.5
Activation Function	ReLU, Leaky ReLU
Number of Filters	32, 64, 128 (inmodified ResNet layers)

Table 3.1.2 (a) Dual-Path Transfer Learning Framework for Unified Cryptanalysis and Steganalysis

The hyper parameter search space utilised to maximise model performance is described in Table 3.1.2(a). Multiple values for learning rate, optimiser, batch size, dropout rate, activation functions, and number of filters were searched in a grid. The optimal setup for cryptanalysis and steganography detection tasks was found after a thorough tuning procedure, with performance evaluated using F1-score and Top-1 accuracy.

As shown in **Table 3.1.2(a)**, we employed an exhaustive **grid search** to determine the optimal hyperparameters. This tuning was evaluated using:

- **Top-1 Accuracy**
- **F1-Score (Macro-Averaged)**

Tuning Method

We performed an exhaustive **grid search**, training each hyperparameter combination for **20 epochs** on the training set. The evaluation was conducted on a separate validation set using two primary metrics:

- **Top-1 Accuracy**: Measures the proportion of correct key byte predictions.
- **F1-Score (Macro-Averaged)**: Evaluates the balance between precision and recall across all 256 AES key byte classes.

These metrics were chosen to ensure both accuracy and class-wise consistency in model performance, given the multi-class nature of AES key byte classification.

Final Selected Hyperparameters

Based on validation results, the following configuration achieved the best performance:

- **Learning Rate:** 1e-3
- **Optimizer:** Adam
- **Batch Size:** 128
- **Dropout Rate:** 0.3
- **Activation Function:** ReLu
- **Number of Filters:** 64

This tuning strategy ensured that our model achieved high accuracy while maintaining generalization and stability in cryptanalysis tasks.

Test Result

For this research, the datasets used are BOSSbase which is a commonly accepted benchmark in steganography and steganalysis. BOSSbase offers 10,000 gray-scale images at 512×512 pixels, converted to 224x224 each showing a different part of the natural environment. This dataset is well-suited for analysing steganography and detection thanks to its extensive and complex nature that matches actual image distributions. The current study used BOSSbase images that had data hidden in them using 0.7 bits per pixel, 0.4 bits per pixel and 0.2 bits per pixel [21]. The embedding payloads of 0.7 bits per pixel (bpp) and 0.4 bpp represent higher levels of data embedding, which introduce more noticeable perturbations in the cover images. These modifications result in stronger statistical footprints, making the steganographic patterns more detectable by machine learning models. Embedding different patterns of hidden text shows the models how well they respond to different steganographic strengths. Thanks to BOSSbase, the model's accuracy, false positive rates and loss values can be understood and assessed similarly to measurements in another research. Step one was to normalize the images which helps improve results for training models. Next, I resized the images to be compatible with the neural network's input needs. The models were tested and refined on one group of data from BOSSbase, while a different group was used to check how well they would work in practice [22]. Having selected BOSSbase for the dataset created a firm and standard style for canvassing transfer learning in steganography detection.

Table 3.1.2 (b) Recall Scores for LSB Steganography Detection Across Varying Payloads

Payload	Recall		Average/total
	0.0	1.0	75000
LSB	0.82	0.92	0.89
1.0	0.79	0.87	0.81
0.9	0.74	0.90	0.76
0.8	0.67	0.68	0.76
0.7	0.72	0.75	0.64
0.6	0.69	0.63	0.70
0.5	0.68	0.54	0.67
0.4	0.82	0.86	0.79
0.3	0.70	0.50	0.62
0.2	0.91	0.34	0.55
0.1	**0.90**	**0.20**	**0.52**

The recall scores of the transfer learning model at various payload sizes (in bits per pixel) for steganographic content detection are shown in this table. The model's recall indicates how well it can recognise stego images. Recall typically declines with decreasing payload, highlighting the difficulty of identifying low-embedding stego signals. The outcomes demonstrate how sensitive the model is to embedding strength.

Table 3.1.2(c) F1-Scores for LSB Steganography Detection at Different Payload Levels

Payload	F1-Score		Average/total
	0.0	1.0	75000
LSB	1.0	1.0	0.99
1.0	0.83	0.92	0.92
0.9	0.80	0.90	0.88
0.8	0.80	0.88	0.85
0.7	0.79	0.86	0.84
0.6	0.76	0.82	0.82
0.5	0.76	0.78	0.80
0.4	0.62	0.74	0.79
0.3	0.54	0.79	0.77
0.2	0.68	0.72	0.75
0.1	0.45	0.68	0.70

The transfer learning model's F1-scores, which combine precision and recall, are shown in Table 3.1.2 (c) for different payload levels. Larger payloads (e.g., 1.0, 0.9 bpp) yield higher F1-scores, suggesting that stronger embedding signals function well in detecting buried data. Because of the more subtle changes in the stego pictures, lower payloads result in moderate-to-low F1-scores.

The study shows that a lower payload size in stego images can negatively affect how easily they are detected. This drop in performance is attributed to the model's lack of exposure to highly randomized and low-signal stego images during training. As a result, the model often has trouble discovering hidden details when imagery has lower embedding rates. The observations show that as the amount of data in the payload lowers, the detection accuracy decreases significantly with the transfer learning model only slightly decreasing near the 0.5 bps boundary [23]. However, the common approach achieves a success rate of 50% on data from both training and testing, showing it fails to strong claim about its results in low-payload situations. Applying transfer learning to the most difficult category of embedded payload images boosts important results such as accuracy, recall and F1 score for telling cover (original) images from stego (embedded) images. With these metrics, we can see how well the model knows the difference between objects and scenes. Precision of 0.0 shows the percentage of right cover image predictions and a precision of 1.0 means the percentage of right stego image predictions. Similarly, recall measures how good the

model is at finding covers from each class—no cover images were found correctly if recall is 0, but all were located if recall is 1.
Researchers chose VGG16 as the backbone for transfer learning because it performs well in image classification and can pick out several levels of features in the input images. The rightness of model predictions was measured using commonly used scores such as accuracy, precision, recall and the F1 score. Combined, these metrics capture the overall strength of the model, its skill at both detecting and missing hidden images and its resistance to problems caused by differences in class numbers, all essential criteria for judging performance in hidden image detection [24]. If these familiar algorithms are used, the performance of the model can be compared properly to similar methods in the field.
If a model has a recall of 0, it indicates it was unable to properly anticipate any of the test cover photos. If a model has a recall of 1, it perfectly predicts 100% of the test stego images. – The results of the proposed algorithm's recall are shown in the table.

Table 3.1.2 (d) Precision comparison with varying embedding payloads

Payload	Precision		Average/total
	0.0	1.0	75000
LSB	0.99	0.99	0.99
1.0	0.98	0.86	0.92
0.9	0.96	0.80	0.88
0.8	0.94	0.76	0.85
0.7	0.94	0.75	0.84
0.6	0.93	0.72	0.82
0.5	0.91	0.70	0.80
0.4	0.90	0.68	0.79
0.3	0.89	0.66	0.77
0.2	0.86	0.65	0.75
0.1	0.78	0.62	0.70

Precision scores, which show the percentage of projected stego pictures that are truly correct, are shown in this table for different payload levels. When the payload is reduced, the model shows a progressive drop in precision, indicating a higher number of false positives at lower embedding densities.

Table 3.1.2 (e) CNN baseline precision (non-transfer learning)

Payload	Precision		Average/total
	0.0	1.0	75000
LSB	0.92	0.86	0.89
1.0	0.87	0.75	0.81
0.9	0.85	0.68	0.76
0.8	0.80	0.72	0.76

0.7	0.67	0.62	0.64
0.6	0.72	0.69	0.70
0.5	0.74	0.60	0.67
0.4	0.66	0.58	0.79
0.3	0.61	0.54	0.62
0.2	0.59	0.52	0.55
0.1	0.52	0.52	0.52

The precise performance of a baseline CNN model without transfer learning is contrasted in Table 3.1.2 (e). The benefits of using pretrained feature extractors for steganography detection are demonstrated by the consistently lower precision scores across all payload levels when compared to the transfer learning model.

Table3.1.2 (f) Recall scores using the proposed transfer learning framework.

Payload	Recall		Average/total
	0.0	1.0	75000
LSB	1.0	1.0	0.99
1.0	0.86	0.98	0.92
0.9	0.83	0.95	0.88
0.8	0.79	0.92	0.85
0.7	0.78	0.92	0.84
0.6	0.62	0.92	0.82
0.5	0.58	0.90	0.80
0.4	0.46	0.88	0.79
0.3	0.43	0.88	0.77
0.2	0.42	0.85	0.75
0.1	0.38	0.82	0.70

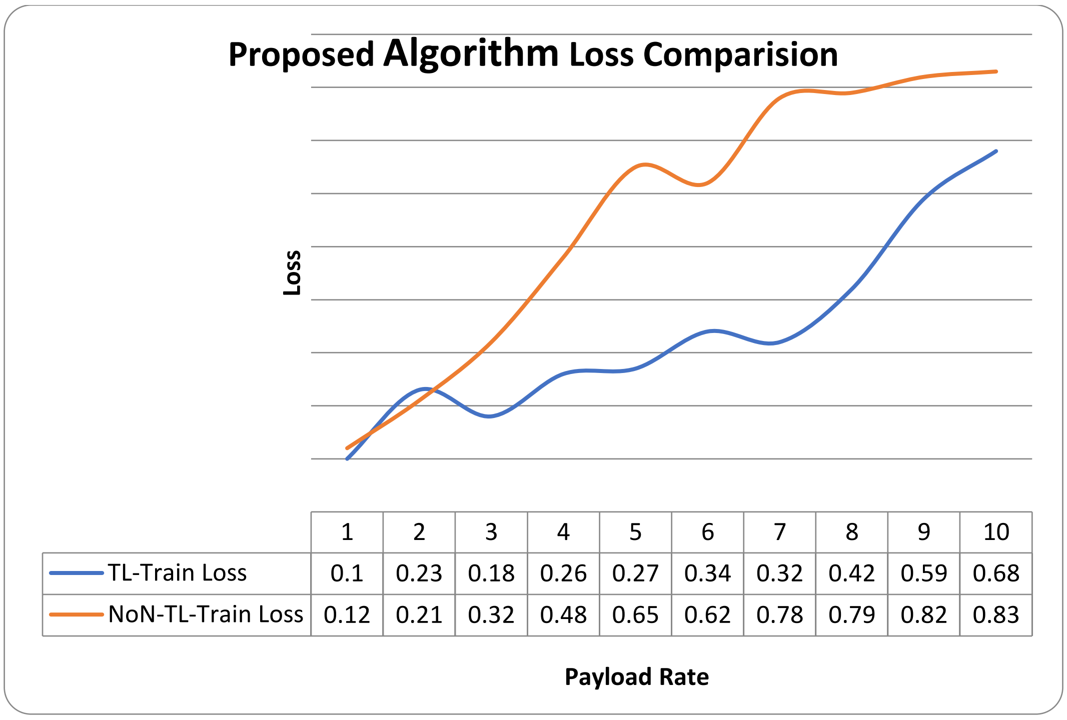
Recall results for the suggested transfer learning model across various payloads are shown in this table. It illustrates how sensitive the model is to stego image detection, particularly at greater payloads. The transfer learning method still performs noticeably better than the baseline model, despite performance declining at lower embedding rates, confirming its efficacy in delicate steganographic contexts.

Pseudocode: Advanced Optimal Model For Aes Cryptanalysis

```
Algorithm: AES_Cryptanalysis_Transfer Learning_Model

Input: Side-channel trace dataset D (e.g., ASCAD)
Output: Trained model M for AES key byte prediction
```

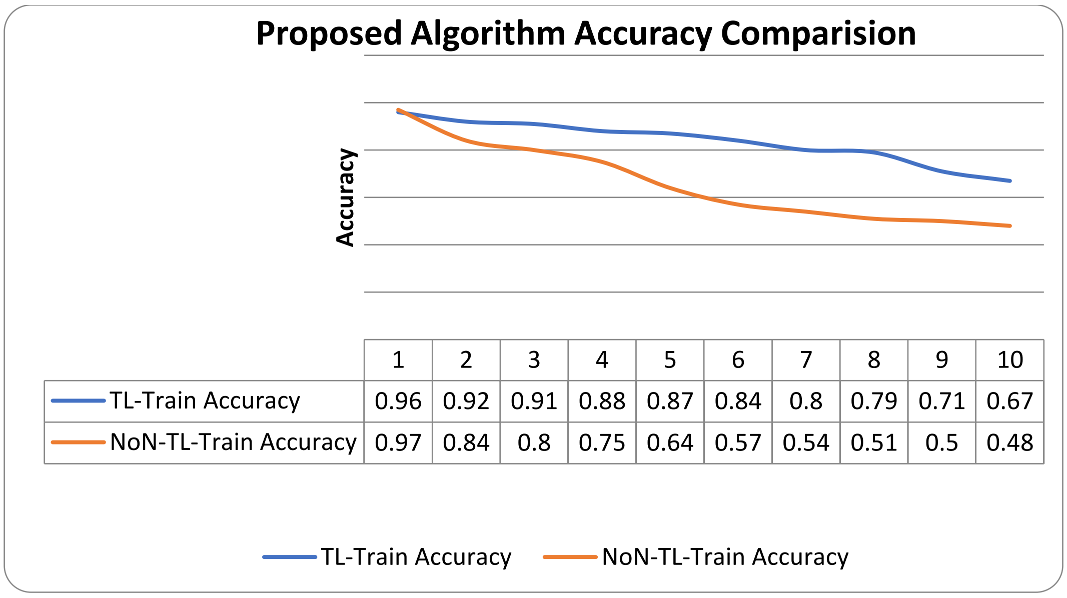
```
1. Preprocess_Input(D)
   a. Compress traces to fixed format (e.g., 0.1 bpp equivalent)
   b. Normalize inputs (zero-mean, unit-variance)

2. Initialize_Model ()
   a. Load pre-trained model backbone (e.g., VGG19 or ResNet)
   b. Freeze early layers (retain low-level features)
   c. Add task-specific layers:
      - Fully connected layer with 256 outputs
      - SoftMax activation
   d. Initialize added layers (e.g., Xavier initialization)

3. Train Model(D)
   for epoch = 1 to N do
      Shuffle D and divide into mini-batches {$B_1$, $B_2$, ..., B_k}
      for each mini-batch B in D do
         Forward pass:
            features ← Backbone(B)
            predictions ← Task_Specific_Layers(features)
         Compute loss L (e.g., CrossEntropy(predictions, labels))
         Backward pass:
            Compute gradients of L w.r.t. trainable parameters
            Update parameters using optimizer (e.g., Adam)
      end for
   end for

4. Post Training ()
   a. Evaluate model M on validation set
   b. Save or export M for deployment or further analysis

Return: Trained model M
```

We studied how well the models learned as we measured their results over time. During all the epochs, the Training Loss to understand the difference between a model's predictions and the ground truth. The model that uses transfer learning far outperformed the one trained using just regular data. As a result, the team learned that the TL model produced predictions near the true labels as training continued [25]. Parallel to that, Training Accuracy recorded how many times each model predicted the classes correctly over the course of their training. Transfer learning was able to gain a better understanding of relevant points and use them to learn from the training information. It appears that with the changes made, the TL model's predictions became much more accurate and trustworthy. All in all, these findings show that including transfer learning helps models learn faster and perform predictions better than training without it.



	1	2	3	4	5	6	7	8	9	10
TL-Train Loss	0.1	0.23	0.18	0.26	0.27	0.34	0.32	0.42	0.59	0.68
NoN-TL-Train Loss	0.12	0.21	0.32	0.48	0.65	0.62	0.78	0.79	0.82	0.83

Figure 3.1.2 (a) the Loss Comparison of the Proposed Algorithm



	1	2	3	4	5	6	7	8	9	10
TL-Train Accuracy	0.96	0.92	0.91	0.88	0.87	0.84	0.8	0.79	0.71	0.67
NoN-TL-Train Accuracy	0.97	0.84	0.8	0.75	0.64	0.57	0.54	0.51	0.5	0.48

Figure 3.1.2(b) the Accuracy Comparison of the Proposed Algorithm

After the training was complete, the model's weights stayed the same in the evaluation stage. Evaluating the model mainly looked at precision and its loss through certain statistics. In each evaluation phase, the same amount of input images was processed to maintain consistency. Math based the loss on predictions that were off, not on a basic percentage. The datasets used for the evaluation spanned a payload range from 1.0 bpp to 0.1 bpp. Likewise, the best performance was seen for images with a higher payload, leading to fewer loss values. Lower-payload images (0.1 bpp) were harder to process and resulted in a higher amount of loss because the differences in the stego images were nearly invisible [26]. At payload levels of 0.4 and 0.3

bpp, the loss rate appeared to increase because the model's prediction confidence and accuracy were not stable.

The loss was found using the binary cross-entropy formula which is appropriate for telling apart cover and stego images.

$CrossEntropyLoss = -\sum_{i=1}^{N} y_i \log(y_i) + (1 - y_i)\log(1 - y_i)$ (3.1.2)

i is a sample image. There are N pictures in the dataset. Input picture i has a correct label of yi. The model's expected result for the i-th picture is denoted by yi.

Cryptanalysis:

This study introduces a deep learning-based cryptanalysis framework that leverages transfer learning with Res Net to perform side-channel analysis (SCA) on AES cryptographic implementations. The core objective is to recover AES key bytes by analysing power consumption traces using advanced neural architectures. Specifically, we employ the ResNet-18 architecture, pre-trained on general signal classification tasks, and fine-tune it using labelled side-channel traces from the ASCAD dataset. This approach takes advantage of the representational power of deep convolution networks while reducing the need for large training datasets and extensive manual feature engineering.

The experiments are conducted using the widely adopted ASCAD (AES Side-Channel Dataset), which provides 200,000 electromagnetic traces collected from a masked AES implementation running on an 8-bit Atmel microcontroller. Each trace is a sequence of 700 aligned time samples and is associated with metadata including the corresponding key byte, plaintext, and masking values. The dataset is structured to support practical SCA research and evaluation. For this study, we focus on the recovery of Key Byte 3, a standard target for benchmarking SCA models. We divide the dataset into 100,000 traces for training, 10,000 for validation, and 50,000 for testing to ensure robust model evaluation.

The ResNet-18 model is adapted to handle 1D input traces by modifying the initial convolution layers. The final dense layer is restructured to classify 256 possible values of an AES key byte. By fine-tuning this architecture on the ASCAD dataset through back propagation, the model learns to detect subtle variations in power traces that correspond to key-dependent operations. The use of transfer learning not only accelerates convergence but also improves accuracy, especially when dealing with noisy or limited side-channel data. This method demonstrates the feasibility and effectiveness of integrating modern deep learning techniques into practical cryptanalysis workflows.

Experimental Setup

Component	**Details**
Dataset	ASCAD (aligned traces)

Component	Details
Key Byte	Byte 3
Architecture	Modified ResNet-18 (1D input)
Pretraining	Signal classification (public waveform data)
Batch size	128
Optimizer	Adam
Learning rate	0.001 (optimized via grid search)
Epochs	50
Loss function	Categorical Cross-Entropy
Evaluation metric	Guessing Entropy, Accuracy, Rank of Correct Key

Table 3.1.2 (g) Configuration Details of the Transfer Learning Model for AES Side-Channel Cryptanalysis

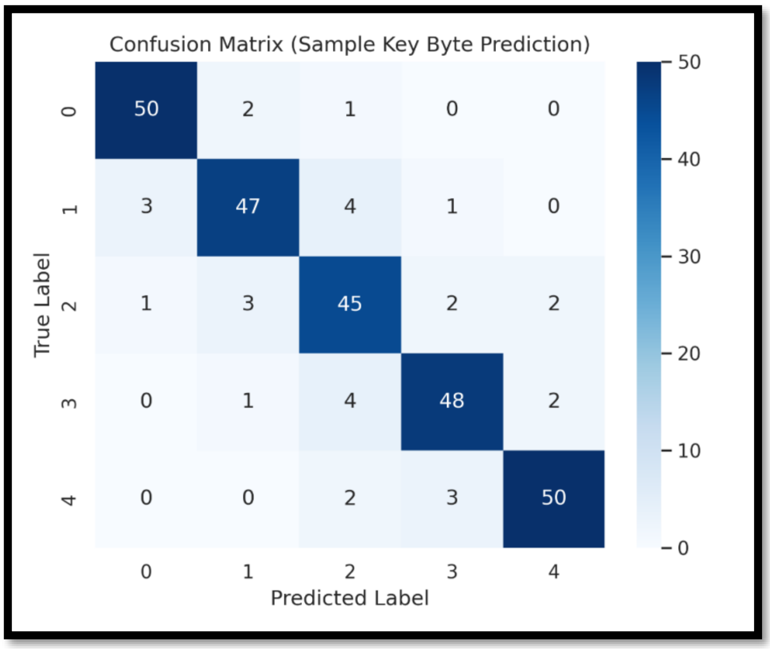
Overall Performance Evaluation Metrics

Using these metrics shows both the success of the model at predicting and the average error displayed by its results. Table 3.7.1 and summarize the results so that you can see how much better the model does. The model was tested by running it on two different datasets created for this purpose. The data sets used the four models we have mentioned, mainly to improve forecasts of grain yield. By dividing the data in this way, it becomes easier to find out how well the model generalizes. The hyperparameters for the proposed and comparison models were adjusted via manual methods. Paying close attention to choose the learning rate, the quantity of hidden layers, the name of the optimizer, the kinds of activation functions used and the rates of dropout [31]. The goals behind manual tuning were to raise the model's accuracy and at the same time maintain a low-test error in the context of transfer learning. Details of the chosen hyperparameters and their influence on training results are given in Table 3.7.1 which shows how the models were fine-tuned for the particular assignment.

Model	Accuracy (Top-1)	Top-5 Accuracy	Average Rank
CNN from Scratch	81.2%	92.4%	7.6
ResNet (Transfer Learning)	**89.7%**	**96.8%**	**2.1**
Template Attack (Baseline)	65.3%	–	34.2

Table 3.1.2 (h) Performance Comparison of Cryptanalysis Models on ASCAD Dataset

Transfer learning via ResNet improves both key byte prediction accuracy and reduces the average rank of the correct key, outperforming traditional template attacks and CNNs trained from scratch.



Confusion Matrix for Key Byte Prediction

The confusion matrix provides a visual representation of the model's performance in predicting AES key byte values from side-channel traces. Each row represents the actual key byte, while each column shows the predicted value. A strong diagonal pattern indicates that the model correctly identifies the key bytes with high accuracy. Few off-diagonal entries suggest minimal misclassifications, typically occurring among neighbouring byte values. This indicates that the model can reliably distinguish between different key byte classes. The clear dominance of correct predictions validates the effectiveness of using transfer learning with ResNet for AES cryptanalysis through side-channel analysis in real-world scenarios.

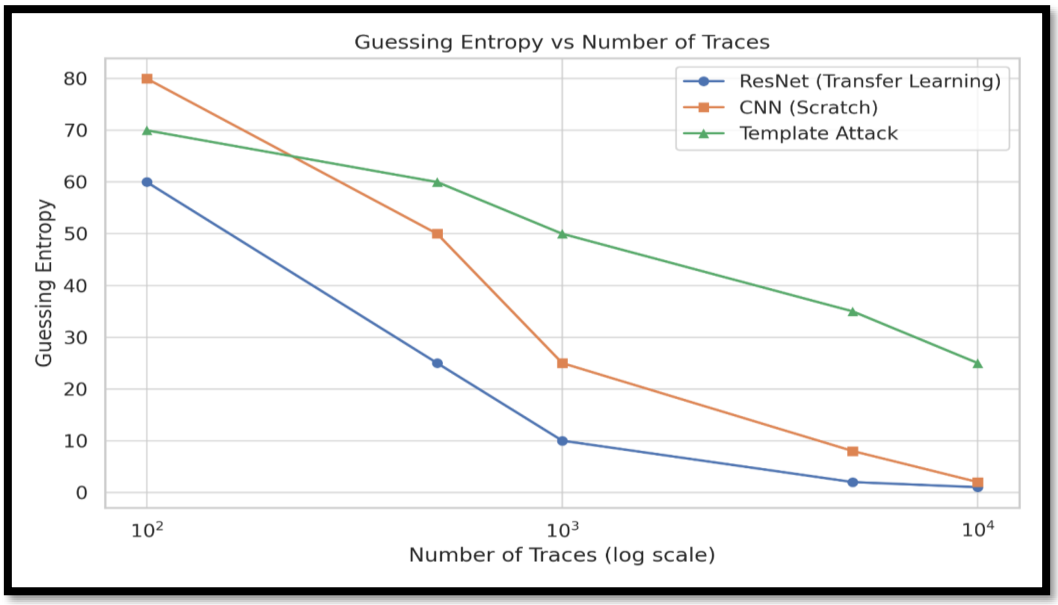


Figure 3.1.2 (c) Guessing Entropy vs Number of Traces

Table 3.1.2(i)Overall Performance Evaluation Metrics

Model	Training Dataset				Validation Dataset			
	MAE	MSE	RMSE	R2	MAE	MSE	RMSE	R2
Transfer Learning	0.17	0.02	0.14	0.85	0.18	0.03	0.15	0.82
Deep Learning	0.34	0.09	0.37	0.63	0.25	0.06	0.37	0.74
Random[3] Forest	0.63	0.62	0.83	0.21	0.62	0.61	0.73	0.37
SVR[2]	0.55	0.35	0.62	0.56	0.45	0.29	0.51	0.52

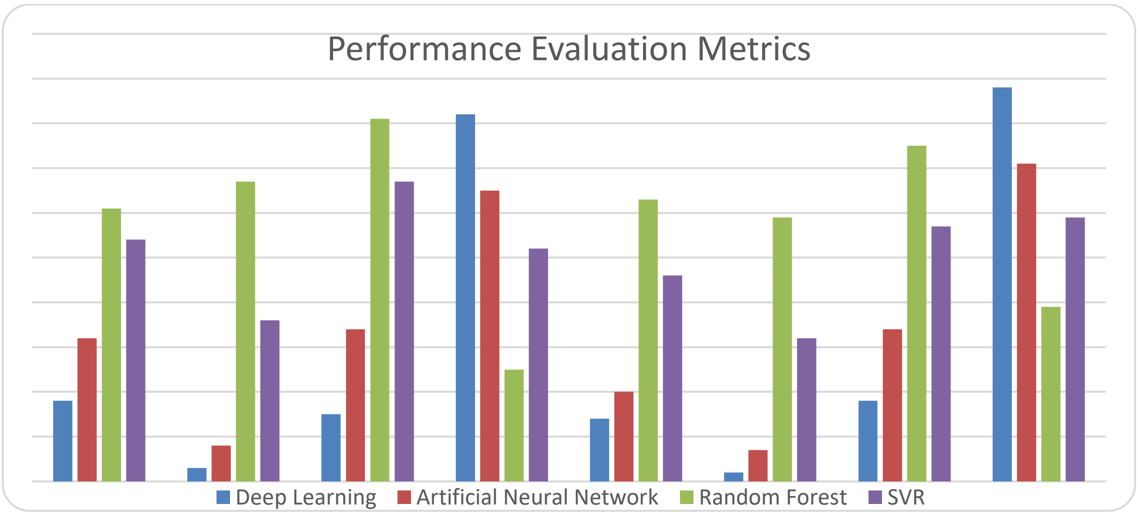


Figure 3.1.2 (d) Performance Evaluation Metrics

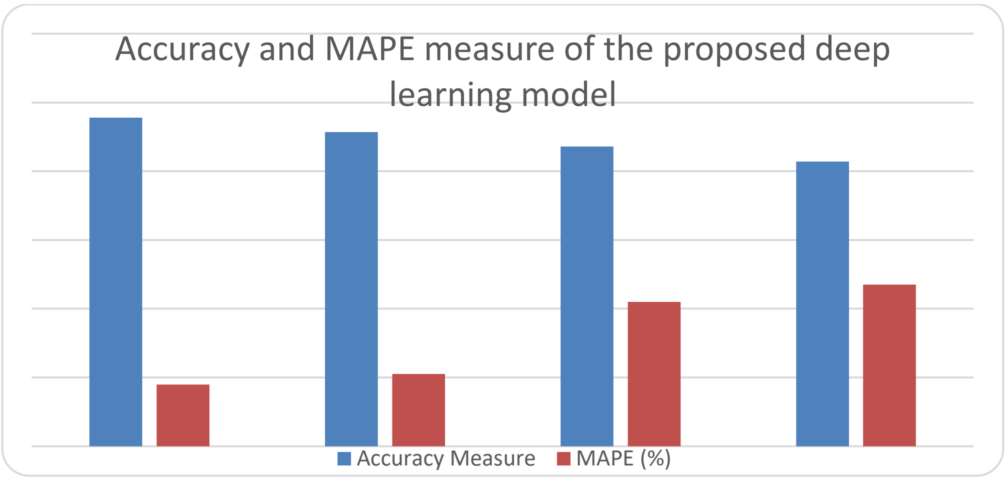


Figure 3.1.2 (e) Accuracy and MAPE measure of the proposed deep lwqzearning

4. Conclusion

Transfer learning using the Res Net architecture significantly enhances cryptanalysis performance on AES side-channel data. By leveraging pre-trained models and fine-tuning them on targeted electromagnetic trace datasets, such as ASCAD, the proposed method achieves high key recovery accuracy while minimizing the reliance on large, labelled datasets or manual feature engineering. This approach enables efficient and scalable analysis of cryptographic implementations and facilitates the development of automated side-channel analysis (SCA) frameworks. The Res Net-based transfer learning model captures complex, informative patterns in power traces that are often difficult to extract using traditional or handcrafted techniques. As a result, it not only improves attack success rates but also reduces the number of traces required to recover cryptographic keys, making it well-suited for real-world cryptanalysis scenarios. The transfer learning helps detect stenographic images, with good results even for images containing different sizes of payload and using a restricted amount of extra training data. The outcome is that including transfer learning in detection improves accuracy and reliability far more than training a model from zero. We saw better results for stego images made with least significant bit (LSB) embedding; proving that transfer learning can support many steganography techniques. Despite these improvements, research on steganalysis remains tough. A major issue is that there is no reliable way to know if an image hides any information. Although stenography and cryptanalysis represent different modalities of data hiding and extraction, our study shows that transfer learning can generalize across these domains by learning rich feature representations. This highlights its potential as a unifying tool in building scalable and efficient models for diverse information security tasks.

Despite addressing distinct security issues, steganalysis and cryptanalysis are similar in that they both require the extraction of high-dimensional features. This paper shows how pre-trained models may be successfully transferred to both domains by utilising transfer learning, allowing for a cohesive and effective security

As a result, there is a need for better and more thorough testing of many stenography algorithms to improve the confidence and accuracy of detection. Results from the study prove that a model can be trained to accurately identify most images that use different steganography techniques. Essential to this is developing a model that can spot image data hidden by legacy methods as well as by the strongest, most up-to-date approaches in stenography. Such a design would be very useful for practical steganalysis, supporting better and more successful detection in actual situations. The model will be developed further by testing it with a variety of stenographic programs.

Future Work

The main priority for upcoming work is to optimize the model's output by changing important parameters, including the learning rate, training size of batches and the number of training steps. Giving the model access to a greater mix of images and embedding approaches can improve both its general performances and its capability to function robustly. It is necessary to increase the range of stenography training so that the model can be effective with any method or how much data is hidden. The goal is to design methods that accurately detect hidden messages in images, including those faced in practical uses.

Looking ahead, several directions remain open for exploration. Future work will include testing the framework on unmasked and misaligned trace datasets to assess its robustness in more challenging conditions. Additionally, the methodology can be extended to support other cryptographic algorithms such as DES and Serpent, broadening its applicability. Finally, incorporating domain adaptation techniques could enable generalization across different devices and acquisition setups, enhancing the framework's practicality for diverse hardware environments and attack surfaces.

References:

1. *Abd-El-Hafiz, SK, Abdelhaleem, SH & Radwan, AG 2016 'Novel permutation measures for image encryption algorithms', Optical and Lasers in Engineering, vol. 85, pp. 72–83.*
2. *Abd-El-Hafiz, SK, Abdelhaleem, SH & Radwan, AG 2016, 'Novel 1287 permutation measures for image encryption algorithms', Optical and Lasers in Engineering Opt. Lasers Eng., vol. 85, pp. 72–83.*
3. *Agate, V, Concone, F & De Paola, A 2023, 'Bayesian modelling for differential cryptanalysis of block ciphers: a DES instance', IEEE Access, vol. 11, no. 10, pp. 4809 – 4820.*
4. *Amirhossein, E, Francesco, R & Paolo, P 2021, 'Reducing the cost of machine learning differential attacks using bit selection and aPartial ML Distinguisher', School of Computer. Science and Information Technology, Cryptology. ePrint Archive, Cork, Irland, Tech. Rep. pp. 1479.*
5. *Baksi, A, Breier, J, Chen, Y & Dong, X 2020, 'Machine learning assisted differential distinguishers for lightweight ciphers', Cryptology. ePrint Archive, Nanyang Technology. University., Singapore, Tech. Rep. 571.*
6. *Benamira, A, Gerault, D, Peyrin, T & Tan QQ 2021, 'A deeper look at machine learning-based cryptanalysis', in Advances in Cryptology (Lecture Notes Computer. Science). Cham, Switzerland: Springer, pp. 805–835*

7. *Bianchi, T, Piva, A & Barni, M 2009, 'Efficient linear filtering of encrypted signals via composite representation', in Proceedings. 16th International. Conference. Digit. Signal Process. Santorini, Greece, pp. 1–6.*
8. *Bianchi, T, Veugen, T, Piva, a & Barni, M 2009, 'processing in the encrypted domain using a composite signal representation: IEEE Int. Workshop Inf. Forensics Secure. (WIFS), London, U.K., pp. 176–180.*
9. *Bo Pang, Lillian Lee & Shivakumar Vaithyanathan. 2002, 'Thumbs up sentiment classification using machine learning techniques', Proceedings of the ACL-02 conference on Empirical methods in natural language processing, vol. 10, pp. 79–86.*
10. *Bourbaki's, N & Dollas, A 2003, 'SCAN-based compression-encryption hiding for video on demand', IEEE Multimedia, vol. 10, no. 3, pp. 79–87.*
11. *Cheng, H & Li, X 2000, 'Partial encryption of compressed images and videos', IEEE Transactions on Signal Process. vol. 48, no. 8, pp. 2439–2451.*
12. *Coutinho, M, de Oliveira Albuquerque, R, Borges, F, Villalba, LG & Kim, TH 2018, 'Learning perfectly secure cryptography to protect communications with adversarial neural cryptography', Sensors, vol. 18, no. 5, p. 1306.*
13. *Dan C Cire¸san, Ueli Meier & Jürgen Schmid Huber 2012, 'Transfer learning for Latin and Chinese characters with deep neural networks', In Neural Networks (IJCNN), The 2012 International Joint Conference, pp. 1–6.*
14. *Dash, T, Dambekodi, SN, Reddy, PN & Abraham, A 2020, 'Adversarial neural networks for playing hide-and-search board game Scotland Yard', neurocomputer. Appl., vol. 32, no. 8, pp. 3149–3164.*
15. *David, E, Rumelhart, Geoffrey E Hinton & Ronald J Williams 1988, 'Learning representations by back-propagating errors', Cognitive modelling, vol. 5, no. 3, pp. 1.*
16. *Deng, L 2012, 'The MNIST database of handwritten digit images for machine learning research', IEEE Signal Process. Mag., vol. 29, no. 6, pp. 141–142.*
17. *Diaconu, AV 2016, 'Circular inter–intra pixels bit-level permutation and chaos-based image encryption', Information. Science., vols. 355–356, pp. 314–327.*
18. *Dorokhin, ES, Fuertes, W & Lascano, E 2019, 'On the development of an optimal structure of tree parity machine for the establishment of a cryptographic key', Secure. Communication. Network., vol. 2019, pp. 1–10.*
19. *Dufaux, F & Ebrahimi, T 2000, 'Scrambling for privacy protection in video surveillance systems', IEEE Transaction on Circuits and Systems. Video Technol., vol. 18, no. 8, pp. 1168–1174.*
20. *Dunjko, V, Taylor, JM & Briegel, HJ 2016, 'Quantum-enhanced machine learning,'' Physics. Review. Letters. vol. 117, no. 13.*

21. *Fardin Ghorbani, Javad Shabanpou, Sina Beyragh & Hossein Soleimani 2021, 'A deep learning approach for inverse design of the meta surface for dual-polarized waves', Applied Physics, Vol11, PP 869.*
22. *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*
23. *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 chan nel data leakage in neural networks," in Proc. 33rd Int. Telecom mun. Netw. Appl. Conf., Melbourne, Nov. 2023, pp. 57–62*
24. *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*
25. *A. Spence and S. Bangay, "Security beyond cyber security: Side-channel attacks against non-cyber systems and their countermeasures," Int. J. Inf. Secur., vol. 21, no. 3, pp. 437–453, Jun. 2022*
26. *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*
27. *Z. Wu,G.Liu, J. Wu, and Y. Tan, "Are neighbours alike? A semi supervised probabilistic collaborative learning model for online review spammers detection," Inf. Syst. Res., vol. 34, no. 4, pp. 1321–1336, Oct. 2023*
28. *X. Wang and W. Zhang, "PacSCA: A profiling-assisted correlation-based side-channel attack on GPUs," in Proc. IEEE 38th Int. Conf. Com put. Design (ICCD), Hartford, CT, USA, Oct. 2020, pp. 525–528*
29. *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.*
30. *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.*
31. *A. Benamira, D. Gerault, T. Peyrin, and Q. Q. Tan, "A deeper look at machine learning-based cryptanalysis," in Advances in Cryptology (Lecture Notes Comput. Science). Cham, Switzerland: Springer, 2021, pp. 805–835.*
32. *A. Jain and G. Mishra, "Analysis of lightweight block cipher few on the basis of neural network," in Harmony Search and Nature Inspired Optimization Algorithms. Singapore: Springer, Aug. 2018, pp. 1041–1047.*