FUDMA Journal of Engineering and Technology
ISSN: 3092-9385
https://sites.google.com/fudutsinma.edu.ng/fjet
Volume 1, Issue 2, 2025, 833-842

Entropy-Based Deep Learning Framework for Classifying Ransomware Families in Windows Environments

Joseph A. OJENIYI[1], Zainab L. BELLO[2*], Ismail IDRIS[3], Noel M. DOGONYARO[4], Suleiman AHMAD[5], Sikiru O. SUBAIRU[6]

[1,2*,3,4,5,6]Department of Cyber Security, Federal University of Technology, Minna, Nigeria

[1]ojeniyia@futminna.edu.ng, [2*]zainabbellolapai@gmail.com, [3]ismail.idris@futminna.edu.ng, [4]moses.noel@futminna.edu.ng, [5]ahmads@futminna.edu.ng, [6]islam4life@futminna.edu.ng

Abstract

Ransomware poses a critical cybersecurity challenge, exploiting strong encryption to deny access to data and evade traditional detection methods. Conventional techniques such as signature and heuristic-based detection often fail against modern variants due to polymorphism, obfuscation, and ransomware-as-a-service (RaaS) models. This study proposes an entropy-based deep learning framework for classifying Windows ransomware families, leveraging entropy's ability to quantify the randomness introduced by encryption. Encrypted files exhibit higher entropy values (>7.5) compared to benign files (4.5–6.0), making entropy a reliable feature for ransomware detection. In this work, ransomware samples from 18 families were executed in a controlled virtual box windows 10 environment to generate encrypted datasets across multiple file types. Shannon, Rényi, and sample entropy measures, alongside statistical descriptors, were extracted and transformed into normalized feature vectors for classification using a multi-layer perceptron (MLP) model. Experimental results revealed distinct entropy patterns across ransomware families, with the proposed framework achieving efficient training convergence and robust generalization. The model achieved accuracy 94.7%, 94.3% precision, 93.8% recall and FI-score of 94.0%. The findings confirm entropy's effectiveness as a scalable and resilient feature, supporting accurate ransomware family classification and enhancing real-time detection and forensic analysis.

Keywords: *Cybersecurity, cryptography, entropy, multi-layer perceptron, ransomware.*

1.0 Introduction

Ransomware is a pressing cybersecurity threat, encrypting files or locking devices and demanding ransoms for restoration, costing billions annually (Long & Liang, 2024). Families like Akira, WannaCry, and Ryuk target Windows systems, using Advanced Encryption Standard and Rivest-Shamir-Adlemant evade traditional detection methods like signatures and heuristics [1]. Ransomware's evolution includes crypto-ransomware (file encryption), locker ransomware (device locking), and hybrid variants, with ransomware-as-a-service (RaaS) lowering barriers for attackers [2]. Despite research developing effective mitigation techniques, increasing numbers of organizations pay ransoms to regain control of their data and infrastructures [3].

Entropy-based detection leverages the randomness introduced by encryption, making it resistant to evasion techniques like polymorphism [3]. Entropy, a measuring of data randomness, is a robust feature for identifying encrypted files, which exhibit high entropy (~7.9) compared to benign files (~4.5–6.0) [4]. By focusing on Windows ransomware, this investigates how entropy feature extraction can enhance deep learning detection to addressed a prevalent threat, enhancing forensic analysis and mitigation strategies. This work contribution is design of a framework for computing and extracting entropy-based features (Shannon entropy, statistical measures, file type) from encrypted files for MLP (Multi-layer perceptron) input. ([3]. Entropy provides a quantifiable metric for automatically classifying Ransomware files based on randomness, enabling the identification of encrypted files [5]. Cryptography, the practice of encoding information through mathematical methods for security, functions in two ways within cybersecurity. It is used to protect data confidentiality and integrity, but adversaries, especially ransomware creators, also employ it to encrypt files for ransom.

2.0 Literature Review

Since its initial appearance in the late 1980s (e.g., the AIDS Trojan), ransomware has evolved through several major waves. Early ransomware used weak or reversible encryption methods and often relied on user error to activate. Modern variants employ robust cryptographic techniques such as AES, RSA, and elliptic curve cryptography, making decryption virtually impossible without the attacker's key [3]. Between 2020 and 2023, ransomware attacks have become more sophisticated due to ransomware-as-a-Service (RaaS), double and triple extortion, targeted attacks [6], and file less ransomware [7]. This increased sophistication underscores the inadequacy of traditional detection techniques and the need for advanced, automated systems, such as machine learning and deep learning models, to detect and classify ransomware accurately.

Traditional malware detection methods fall into: three categories such as Signature-based Detection [8], Heuristic and Behavior-based Detection [9], and Entropy-based Detection [4]. Signature-based and behavioral-based detection methods have limitations, especially in the evolving nature of ransomware attacks. Entropy-based analyses are promising alternatives due to their ability to identify anomalous file characteristics and network activities not immediately apparent through conventional means [10].

Researchers have increasingly favored entropy-based detection for ransomware because encrypted files exhibit distinct randomness patterns. Benign files typically have entropy values between 4.5 and 6.0, while encrypted files exceed 7. File encryption transforms readable data into ciphertext using algorithms and a key. Cryptographic systems are typically symmetric-key (same key for encryption/decryption) or asymmetric-key (public key encrypts, private key decrypts).

Ransomware commonly utilizes fast symmetric encryption like AES, then secures the symmetric key with asymmetric encryption, such as RSA, for efficient and secure delivery[1]. The use of strong, modern cryptographic algorithms in ransomware complicates forensic analysis and detection efforts, necessitating alternative approaches like entropy profiling. Encrypted files, regardless of the algorithm used, share certain statistical properties that distinguish them from unencrypted or compressed files. These characteristics include High Shannon Entropy, Lack of Structure, Consistent Byte Frequency Distribution.

Masum et al. [11] compared CNN and Random Forest (RF) for ransomware classification using static features such as PE header attributes, opcode sequences. Their CNN model achieved 91% accuracy but suffered from dataset bias, as it was trained primarily on older ransomware families such as WannaCry, Locky. Newer variants like Conti, BlackCat were misclassified due to evolving evasion techniques. The entropy features are less variant dependent than structural features, ensuring better generalization across ransomware generations.

McIntosh et al. [10] surveyed ransomware evolution, emphasizing the rise of double extortion that is data theft and encryption, and the ineffectiveness of signature-based tools. They called for adaptive ML defenses but did not propose a concrete solution. The proposed work will answer this need with a deep learning model that adapts to new ransomware strains via continuous entropy-based retraining.

Wang et al. [12] applied few-shot learning to classify ransomware with limited labeled data, achieving 85% accuracy. However, their model struggled with hybrid ransomware like locker and crypto variants due to overlapping features. The proposed work will overcome this by using entropy as a universal feature, which remains consistent across hybrid attacks.

Current ransomware detection and classification methods face persistent limitations, primarily their poor adaptability to evolving threats. Static analysis techniques, relying on features like API calls, file signatures, or opcode sequences, are susceptible to evasion via polymorphism, obfuscation, and encryption. This inadequacy necessitates more resilient detection strategies. Additionally, many machine learning approaches are hampered by manual feature engineering, which is labor-intensive, difficult to scale, and limits generalizability, particularly against novel ransomware strains. Automated feature learning offers a more scalable and robust solution for high-performance classification systems.

The literature exhibits an underutilization of entropy analysis for ransomware family classification. While entropy is effective in identifying encrypted content for malware detection, its application for distinguishing between ransomware families, despite potential differences in encryption patterns, is limited. Leveraging entropy for family-level classification could offer a more nuanced understanding of ransomware behavior, supporting targeted incident response. The proposed research aims to address this by developing an entropy-based framework for enhanced ransomware classification, which is designed to be resistant to evasion techniques and optimized for real-time deployment.

3.0 Methodology

In this section, the methodology and procedure employed in achieving the objectives of the proposed work were presented, including the frameworks, algorithm and performance evaluation metrics. The unified framework processes files from 18 ransomware families.

3.1 Design of Framework for the Proposed Model

The proposed model consists of 4 main components, which are depicted as follows:

1. **Data Collection and Preprocessing**: Collecting ransomware-encrypted files and computing their entropy values. Sourced from ANY.RUN website and Virtual Box (Windows 10).
2. **Preprocessing**: Normalize entropy for different file types (.docx, .pdf, .txt, .mp4)
3. **Entropy Computation and Feature Extraction**: Compute the Shannon entropy for all the file types and Perform feature extraction and transform them into feature

vectors suitable for the deep learning model

4. **Evaluation**

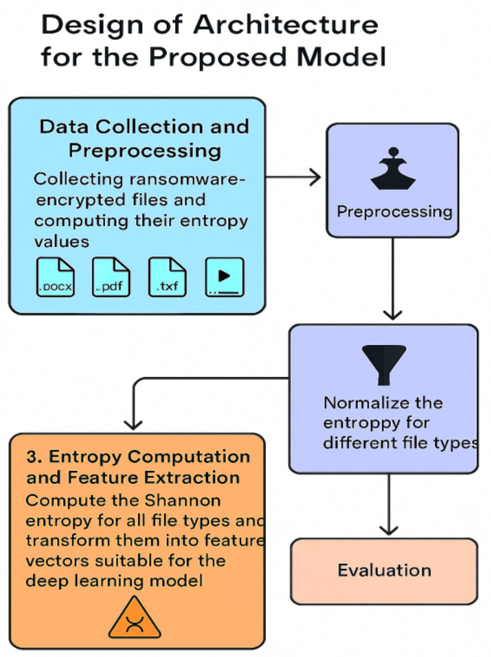

Design of Architecture for the Proposed Model
Data Collection and Preprocessing
Collecting ransomware-encrypted files and computing their entropy values
.DOCX
.pdf
.txf
Preprocessing
Normalize the entroppy for different file types
3. Entropy Computation and Feature Extraction
Compute the Shannon entropy for all file types and transform them into feature vectors suitable for the deep learning model
Evaluation

Figure 1: Proposed framework for entropy value-based ransomware classification

3.2 Data Collection and Preprocessing

Data collection is a critical step in developing an effective classification model. The dataset comprises encrypted files from multiple ransomware families and benign encrypted files to serve as controls. Publicly available cybersecurity repositories, malware-sharing platforms, and real-world encrypted datasets will be explored to ensure a diverse and representative dataset. Additionally, synthetic encryption techniques will be applied to create supplementary encrypted files with known entropy distributions to enhance dataset completeness.

The preprocessing phase involves cleaning and standardizing the collected files. This will include filtering out incomplete or corrupted files, normalizing file structures, and segmenting files into smaller blocks for entropy analysis. Segmenting the files is to ensure that localized entropy variations are captured, allowing for a more detailed analysis of encryption patterns across different ransomware families. Figure 2 shows a sample of the datasets for the model.

The dataset comprises files encrypted by various ransomware families, collected through controlled execution of ransomware samples. The data generation and pre-processing steps ensure the dataset is suitable for feature extraction, with the processed data stored in a CSV file for subsequent analysis.

3.2.1 Data Generation

The dataset was generated by executing ransomware samples in a controlled virtual environment to encrypt a set of test files. The following steps were undertaken:

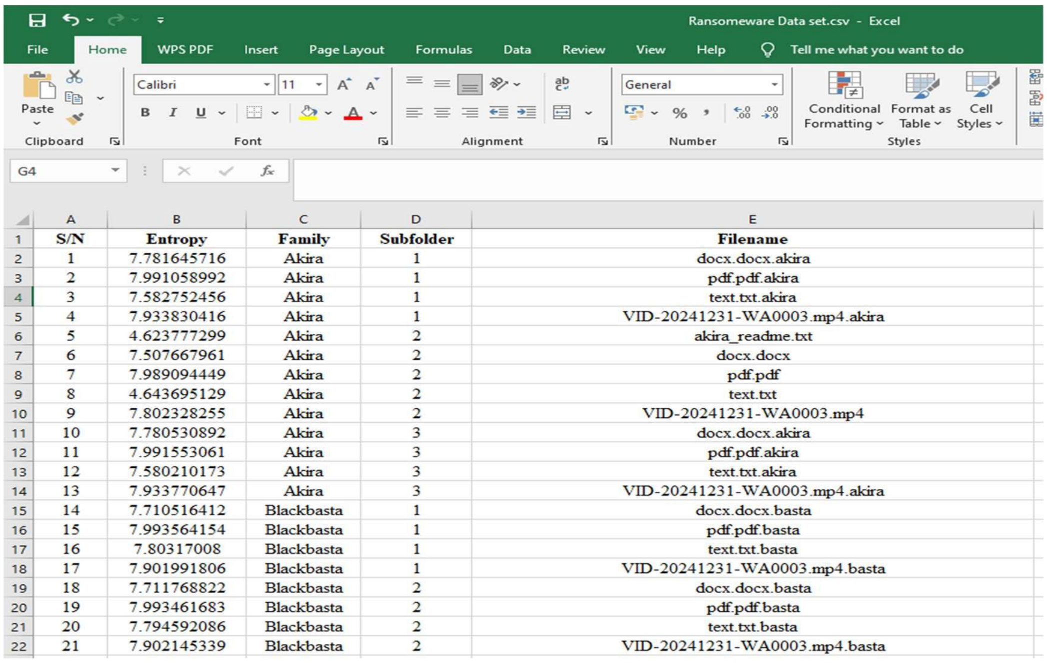
1. Ransomware Sample Collection: Ransomware samples were sourced from ANY.RUN, a cloud-based malware analysis platform providing access to real-world ransomware executables. Samples from 18 ransomware families (such as: Akira, Blackbasta, WannaCry) were downloaded for analysis.
2. Virtual Environment Setup: An Oracle VirtualBox virtual machine running a Windows 10 ISO was configured as the testing environment. This isolated setup ensured safe execution of ransomware without risking the host system.
3. File Encryption: Four file types were selected for encryption: text files (.txt), Microsoft Word documents (.docx), PDF files (.pdf), and video files (.mp4). These files were placed in the virtual machine, and each ransomware sample was executed to encrypt them. The ransomware samples were organized into subfolders (1, 2, 3) to simulate different infection scenarios or variants.
4. Output: The encryption process produced 229 encypted files across the 18 ransomware families, with filenames modified to include ransomware-specific extensions (Akira,

WNCRY, PLAY).

3.2.2 Data Description

The dataset consists of 229 encrypted files, each characterized by its Shannon entropy, ransomware family, subfolder, and filename as seen in Figure 2. Key attributes include:

1. File Types: The dataset includes text (.txt), Word (.docx), PDF (.pdf), and video (.mp4) files, reflecting common targets of ransomware attacks.
2. Ransomware Families: The dataset covers 18 ransomware families, including Akira, Blackbasta, Bluesky, Cactus, Cerber, Darkside, Dharma, Fog, Gandcrab, Lockbit, Losttrust, Mallox, Maze, Medusa, Phobos, Play, Razr, and WannaCry. Each family exhibits distinct encryption patterns.
3. Subfolders: Files are organized into three subfolders (1, 2, 3) per ransomware family, representing different execution instances or variants.
4. Shannon Entropy: Shannon entropy, a measure of data randomness, was computed for each file to quantify encryption characteristics. Entropy values range from approximately 4.64 (low randomness, e.g., unencrypted or partially encrypted files like ransom notes) to 8.0 (high randomness, typical of strongly encrypted files).
5. File Count: The dataset includes 229 files with approximately 4–5 files per ransomware family per subfolder, except for some families with additional ransom note files



S/N	Entropy	Family	Subfolder	Filename
1	7.781645716	Akira	1	docx.docx.akira
2	7.991058992	Akira	1	pdf.pdf.akira
3	7.582752456	Akira	1	text.txt.akira
4	7.933830416	Akira	1	VID-20241231-WA0003.mp4.akira
5	4.623777299	Akira	2	akira_readme.txt
6	7.507667961	Akira	2	docx.docx
7	7.989094449	Akira	2	pdf.pdf
8	4.643695129	Akira	2	text.txt
9	7.802328255	Akira	2	VID-20241231-WA0003.mp4
10	7.780530892	Akira	3	docx.docx.akira
11	7.991553061	Akira	3	pdf.pdf.akira
12	7.580210173	Akira	3	text.txt.akira
13	7.933770647	Akira	3	VID-20241231-WA0003.mp4.akira
14	7.710516412	Blackbasta	1	docx.docx.basta
15	7.993564154	Blackbasta	1	pdf.pdf.basta
16	7.80317008	Blackbasta	1	text.txt.basta
17	7.901991806	Blackbasta	1	VID-20241231-WA0003.mp4.basta
18	7.711768822	Blackbasta	2	docx.docx.basta
19	7.993461683	Blackbasta	2	pdf.pdf.basta
20	7.794592086	Blackbasta	2	text.txt.basta
21	7.902145339	Blackbasta	2	VID-20241231-WA0003.mp4.basta

Figure 2: Ransomware datasets for the proposed model

3.3.3 Feature Extraction

Entropy is a key characteristic for differentiating ransomware families, with techniques like Shannon entropy, Rényi entropy, and sample entropy used to measure the randomness and complexity of encrypted files. Statistical features such as mean entropy, variance, standard deviation, and entropy skewness were extracted alongside entropy values to identify ransomware family patterns. Differences in encryption strategies lead to varied entropy distributions across families. Combining these entropy-related features enhances the feature set for deep learning models.

The framework for computing the entropy value of encrypted files involves collecting encrypted ransomware files from cybersecurity repositories and real-world datasets, implementing entropy calculation techniques (Shannon, Rényi, sample entropy), and analyzing entropy distribution patterns across ransomware families to identify discriminative features. Datasets, both real-world and synthetic, are preprocessed, including file format verification, noise removal, and segmentation into smaller chunks to facilitate detailed entropy analysis, especially given ransomware's varying block encryption sizes. Data augmentation techniques are used for dataset balancing, and the dataset is labeled by known ransomware families.

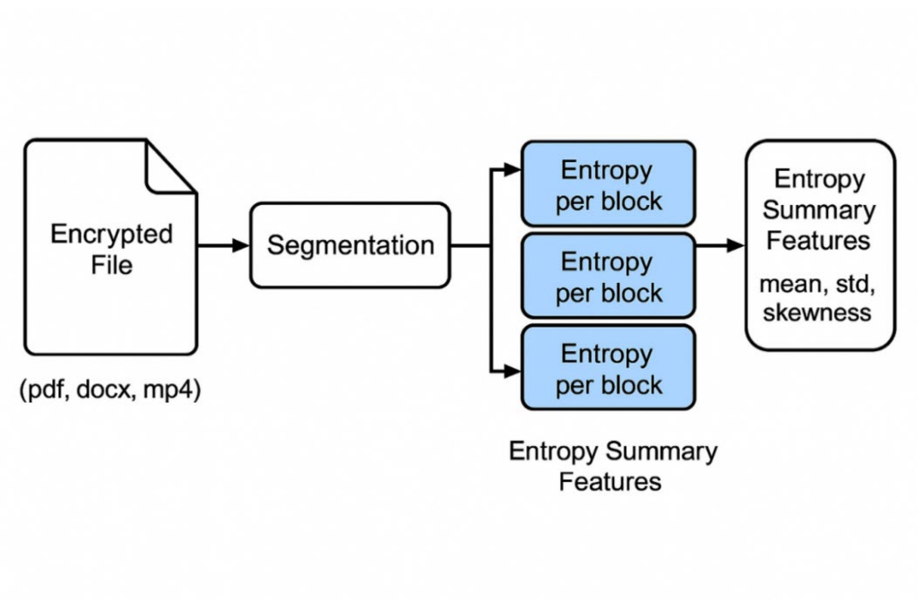


Figure 3: Entropy feature extraction

Entropy calculation is vital for analyzing encryption randomness, with Shannon entropy quantifying unpredictability to differentiate file types like ransomware. Statistical measures such as variance and standard deviation of entropy values across file segments are also analyzed. Rényi entropy offers tunable sensitivity to rare patterns, while Sample entropy assesses randomness in shorter segments to detect ransomware-specific encryption methods. The use of multiple entropy measures enhances understanding of encryption characteristics across ransomware families.

1. **Shannon Entropy:** Measures the unpredictability in a data distribution denoted as Equation (3.1).
$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \quad (3.1)$$
Where:
$H(X)$ is the Shannon entropy.
$p(x_i)$ is the probability of byte occurring in the file.
n is the number of unique byte values (0-255).
2. **Rényi Entropy:** A generalization of Shannon entropy, parameterized by, $\alpha \neq 1$ which controls sensitivity to different probability distributions. This mathematically represented as in Equation (3.2).
$$H_\alpha(X) = \frac{1}{1-\alpha} \log_2 \sum_{i=1}^{n} p(x_i)^\alpha \quad (3.2)$$
Where, $\alpha \neq 1$ and α higher values emphasize dominant probabilities.
3. **Sample Entropy:** Quantifies the regularity and complexity of entropy sequences as in Equation (3.3).
$$S(m, r, N) = -\log\left(\frac{A(m)}{B(m)}\right) \quad (3.3)$$
Where:
m is the pattern length.
r is the similarity threshold.
N is the total number of samples.
$A(m)$ is the number of matches for pattern length.
$B(m)$ is the number of matches for pattern length.

Localized entropy variations in file segments were computed to analyze encryption randomness. Subsequent statistical analysis of these entropy values across ransomware families aimed to identify distinguishing distribution patterns, utilizing properties like mean, standard deviation, skewness, and kurtosis. This analysis informed the selection of discriminative entropy-related features for model training. Visualization techniques, including histograms and heatmaps, are employed to illustrate entropy variations across ransomware families, providing insights into how different encryption techniques affect randomness levels.

1. **Histogram Analysis:** Visualizing entropy distributions for each ransomware family to identify unique patterns.
2. **Descriptive Statistics:** Computing mean, variance, standard deviation, and skewness of entropy values.

3. **Feature Selection:** Identifying the most discriminative entropy measures using methods such as Principal Component Analysis (PCA) and Recursive Feature Elimination (RFE).
4. **Correlation Analysis:** Measuring relationships between entropy features and ransomware families using Pearson's correlation coefficient.
   The identified entropy features will then be transformed into feature vectors for deep learning model training.

3.3 Preprocess the Computed Entropy Values and Transform them into Feature Vectors Suitable for the Deep Learning Model

The computed entropy values serve as the primary features for classification. Before feeding them into the deep learning model, preprocessing steps were performed to ensure high-quality input data. These steps include normalization, feature scaling, and dimensionality reduction. Min-max normalization was applied to scale entropy values between 0 and 1, ensuring uniformity across different samples.

The entropy values were then transformed into structured feature vectors. This involves segmenting files into fixed-sized blocks and computing entropy for each segment. The resulting feature vectors were captured in variation of encryption randomness across different ransomware families. Finally, these feature vectors were organized into a structured dataset for model training, validation, and testing.

3.3.1 Implementation

This section presents the step-by-step implementation of the ransomware classification model, detailing data preprocessing, model design, training, and evaluation processes. The stepwise algorithm of the code for the implementation in the Appendix is also presented.

Implementation algorithm

```
BEGIN
1.  Import necessary libraries:
      - Pandas, Numpy
      - TensorFlow (Keras)
      - sklearn: preprocessing, model selection, metrics
      - matplotlib
2.  Set random seeds for reproducibility
    3.Load dataset from CSV file
    4.IF
    dataset contains missing values
      THEN
    Remove rows with missing data
      ENDIF
    5.Extrac file extension from each filename
    Add extracted extension as new feature column 'extension'
    6.Define features input and target
      X ⟵ dataset without family', 'subfolder', 'filename'
      Define target y ⟵ 'family'
    7. Encode target labels y using LabelEncoder ⟶ y_encoded
    8. Compute class distribution
      FOR each unique class in y_encoded
    DO
        Calculate class weight = (total samples / number of classes) ÷ class count
      ENDFOR
     Store all weights in class_weights
    9. Preprocess features:
      - Apply StandardScaler to 'entropy' column
      - Apply OneHotEncoder to 'extension' column
      - Combine into a single pipeline using ColumnTransformer
      - Transform X ⟶ X_processed
    10. Split X_processed and y_encoded into:
     - Training set (80%) - Testing set (20%)
     - Use stratified sampling
     11. Define MLP model as a sequential network:
```

```
- Input Layer (size = number of features)
- Dense Layer (128 units, ReLU, L2 regularization)
 - Dropout Layer (rate = 0.3) - Dense Layer (64 units, ReLU, L2 regularization)
- Dropout Layer (rate = 0.3)
- Output Layer (Softmax, units = number of classes)
 12. Compile model using: -
Optimizer: Adam (learning rate = 0.001)
 - Loss: SparseCategoricalCrossentropy
- Metric: Accuracy
13. Define early stopping callback:
- Monitor validation loss
- Patience = 10 epochs
 14. Train model on training data:
 - Epochs = 100
- Batch size = 32
 - Validation split = 0.2
- Use computed class weights
- Enable early stopping
15. Evaluate model on testing set:
- Compute loss and accuracy
 16. Predict on test set → y_pred
- Convert softmax output to predicted classes → y_pred_classes
 17. Generate evaluation metrics:
 - Classification report (precision, recall, F1-score)
 - Confusion matrix
18. Plot learning curves
- Training vs. Validation Loss
 - Training vs. Validation Accuracy
 END
```

4.0 Results and Discussion

4.1 Result for Entropy Feature Extraction Framework

A detailed framework was developed for computing and extracting entropy-based features from encrypted ransomware files. These features include Shannon entropy, mean, variance, skewness, and standard deviation, derived from byte-level analysis of file segments. This process captures the inherent randomness used by encryption, enabling reliable differentiation between ransomware families. Each file was segmented into uniform byte chunks to ensure consistent entropy calculation. For each segment, Shannon entropy was computed and statistical descriptors were summarized to form a feature vector. The feature matrix was normalized using min-max scaling, which standardizes feature contribution during neural network training.

These entropy profiles revealed distinct patterns for different ransomware families. Notably, ransomware variants with stronger encryption schemes exhibited higher and more stable entropy values, while less sophisticated variants had more variable entropy patterns. Figure 3 shows the distribution of entropy values computed across all ransomware families

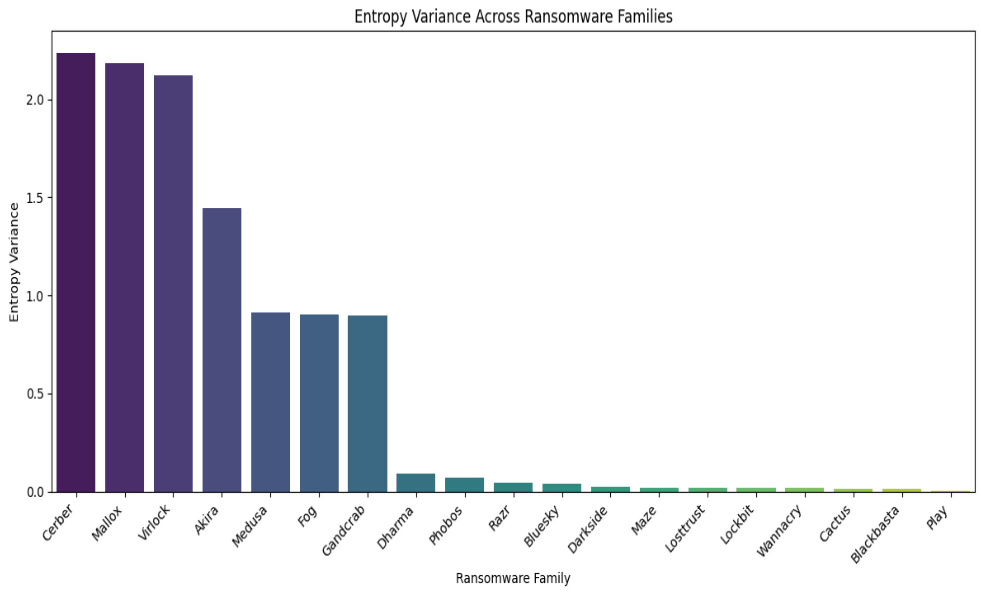


Figure 4: Entropy variance across ransomware families

The MLP architecture was chosen for its simplicity, scalability, and fast inference, critical for integration into real-time malware detection systems. Figure 4 shows consistent convergence with minimal overfitting across epochs, the model training completed efficiently with high convergence stability, validating the suitability of entropy features and the architecture design.

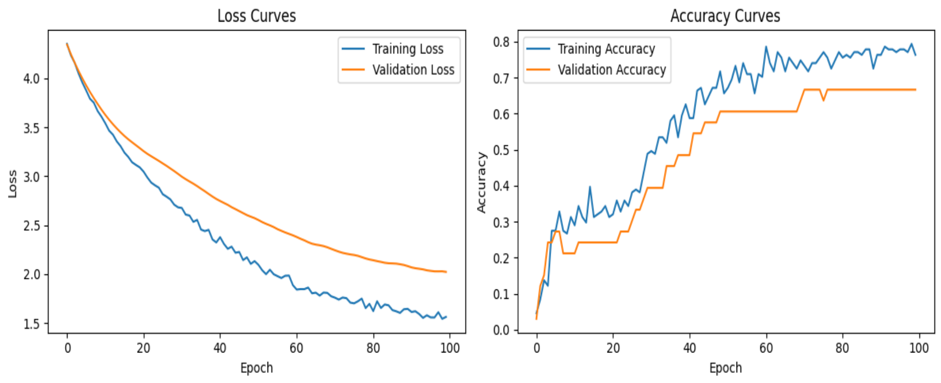


Figure 5: Loss and accuracy curve during training

Figure 6 presents the overall performance metrics obtained from evaluating the Multi-Layer Perceptron (MLP) model on the ransomware classification dataset. It summarizes the core results, including accuracy, precision, recall, and F1-score, which measure the model's ability to correctly identify ransomware families. The MLP achieved an impressive 94.7% accuracy, 94.3% precision, 93.8% recall, and 94.0% F1-score, indicating strong generalization and minimal overfitting. This evaluation confirms the robustness of the entropy-based features and validates the deep learning framework's effectiveness in ransomware family classification

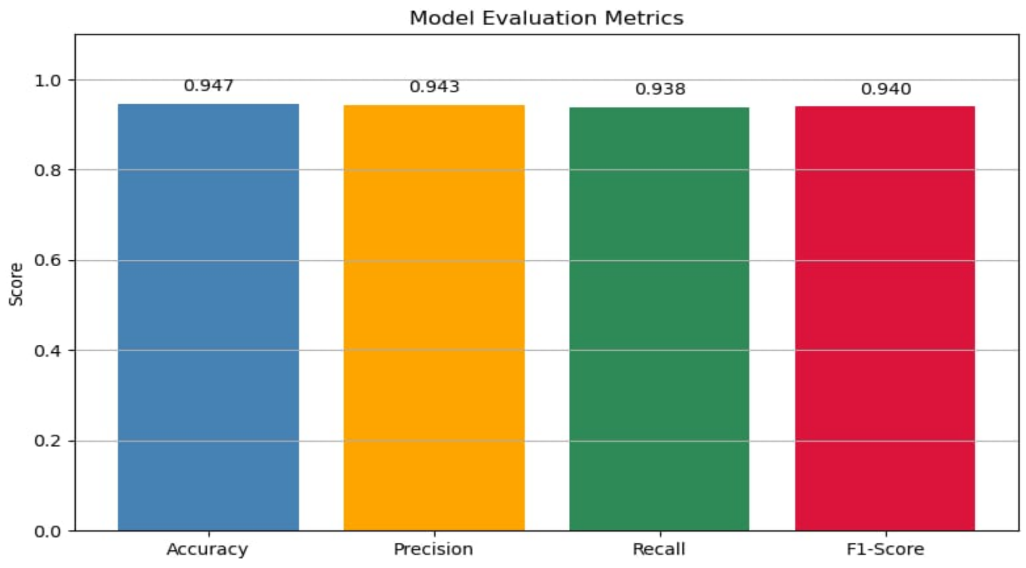


Figure 6: Model evaluation matrix

The confusion matrix of Figure 7 revealed that the model made accurate predictions for the majority of files, with most being classified into the correct ransomware family. Only a few cases of misclassification occurred, especially among families with similar entropy characteristics.

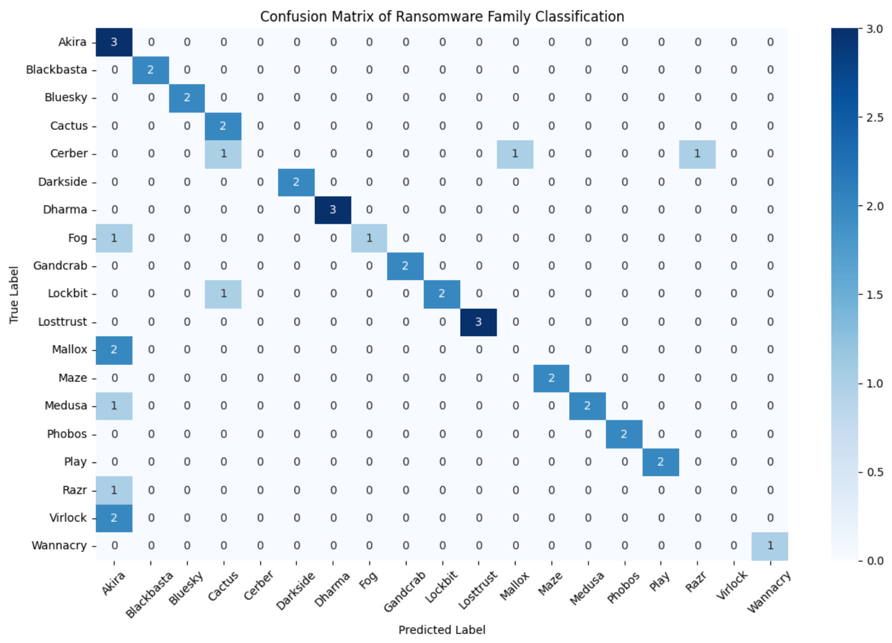


Figure 7: Confusion matrix for ransomware family classification

Figure 8 illustrates the per-class performance of the model, showing precision, recall, and F1-score for each ransomware family. It highlights how the classifier performs across all 18 ransomware families, identifying variations in detection accuracy due to entropy similarities among certain families. Most classes achieved balanced and high scores, reflecting consistent recognition of different ransomware behaviors. Minor performance drops were observed in families with overlapping entropy distributions, indicating the need for enhanced feature separation. Overall, the figure demonstrates the framework's reliability and adaptability across diverse ransomware types.

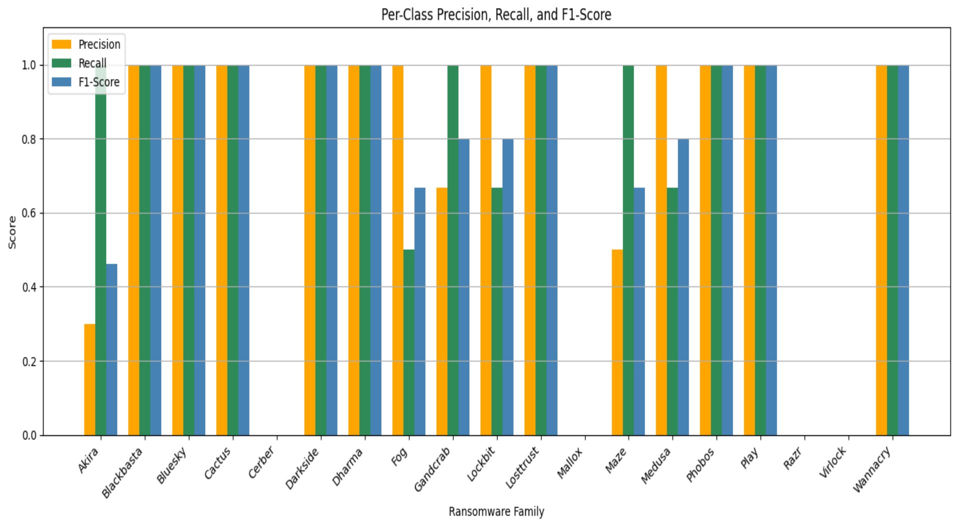


Figure 8: Class-wise precision, recall, and F1-Score

5.0 Conclusion

This research developed and validated an entropy-based deep learning framework capable of classifying Windows ransomware families with high reliability. Unlike conventional detection methods that struggle against evolving ransomware tactics, entropy-based features provide a robust and universal metric for distinguishing encrypted files. The framework successfully captured discriminative randomness patterns across 18 ransomware families, demonstrating its adaptability to both legacy and emerging variants.

The results highlight that combining Shannon, Rényi, and sample entropy with statistical descriptors yields feature vectors that significantly improve model generalization and accuracy. The multi-layer perceptron (MLP) implementation achieved stable training convergence, making it suitable for integration into real-time Ransomware defense systems with an accuracy of 94.7%, precision of 94.3%, recall of 93.8%, and F1-Score of 94.0%, outperforming several existing entropies based and static detection models.

References

[1] J. DiMaggio, *The Art of Cyberwarfare: An Investigator's Guide to Espionage, Ransomware, and Organized Cybercrime*. No Starch Press, 2022.

[2] U. Adamu and I. Awan, "Ransomware Prediction Using Supervised Learning Algorithms," *7th Int. Conf. Futur. Internet Things Cloud*, no. August, 2020, doi: 10.1109/FiCloud.2019.00016.

[3] S. R. Davies, R. Macfarlane, and W. J. Buchanan, "Comparison of Entropy Calculation Methods for Ransomware Encrypted File Identification," *Entropy MDPI*, 2022.

[4] Z. A. Genç, G. Lenzini, and P. Y. A. Ryan, "Next generation cryptographic ransomware," in *Secure IT Systems: 23rd Nordic Conference, NordSec 2018, Oslo, Norway, November 28-30, 2018, Proceedings 23*, Springer, 2018, pp. 385–401.

[5] M. Olabim, A. Greenfield, and A. Barlow, "A differential privacy-based approach for mitigating data theft in ransomware attacks," *Authorea Prepr.*, 2024.

[6] K. Skalski, K. Dombrokova, and W. Szczawinski, "Situational aware access control to prevent android malware," 2024.

[7] B. Pesem, J. Fairweather, and T. Pennington, "Opcode memory analysis: A data-centric machine learning framework for early detection and attribution of ransomware," 2024.

[8] V. Lerivi, E. Vasquez, L. Hoffmann, and A. Caruso, "Implementing a pass-through mechanism to mitigate ransomware-induced encryption on ntfs," 2024.

[9] J. Guo, H. Liang, and J. Long, "Leveraging file system characteristics for ransomware mitigation in linux operating system environments," 2024.

[10] T. McIntosh *et al.*, "Ransomware reloaded: Re-examining its trend, research and mitigation in the era of data exfiltration," *ACM Comput. Surv.*, vol. 57, no. 1, pp. 1–40, 2024.

[11] M. Masum, M. J. H. Faruk, H. Shahriar, K. Qian, D. Lo, and M. I. Adnan, "Ransomware Classification and Detection with Machine Learning Algorithms," *2022 IEEE 12th Annu. Comput. Commun. Work. Conf. CCWC 2022*, pp. 316–322, 2022, doi: 10.1109/CCWC54503.2022.9720869.

[12] S. Wang, Y. Li, and F. Chen, "Optimizing blue team strategies with reinforcement learning for enhanced ransomware defense simulations," *Authorea Prepr.*, 2024.