

A Novel Framework for Ransomware Detection Combining Traditional Machine Learning with CNN2D

R. Tharun Kumar, S. Ranga Lokesh Reddy, T. Varshith Goud

¹ CSE(AI&ML), CMR Institute of Technology, Hyderabad, Telangana, India

² CSE(AI&ML), CMR Institute of Technology, Hyderabad, Telangana, India

³ CSE(AI&ML), CMR Institute of Technology, Hyderabad, Telangana, India

Abstract - Ransomware attacks continue to cause significant financial and operational damage by encrypting user data and demanding payment for recovery, making early and accurate detection a critical cybersecurity challenge. The time-honored signature-based detection systems are not usually effective in detecting the new and emerging ransomware variants and require behavior-oriented and data-centric detection systems. This study presents a hybrid classification framework that utilizes Hardware Performance Counters and Input/Output event datasets to detect ransomware activity. The suggested technique uses a Two-Dimensional Convolutional Neural Network to refine feature representation, and a Random Forest classifier to do the final prediction. The train and test datasets were split into 80:20, and a variety of machine learning algorithms were compared. The experimental findings showed that the proposed hybrid CNN2D and Random Forest model was able to classify data with a high accuracy of 99% which was higher than the traditional machine learning and deep learning models that were applied under the same conditions. These findings indicate that combining convolution-based feature optimization with ensemble classification provides an effective and computationally efficient solution for ransomware detection using system performance metrics.

Key Words: - Ransomware Detection; Hardware Performance Counters; Input/Output Event Analysis; Convolutional Neural Networks(CNN2D); Random Forest Classification; Behavior-Based Malware Detection

I. INTRODUCTION

Ransomware has become one of the most serious cybersecurity threats in recent years, affecting individuals, organizations, and even critical infrastructure. It is a form of malicious software that encrypts valuable files and requires payment of ransom to decrypt the files. With the increasing dependence on digital systems and data, ransomware attacks have grown both in frequency and complexity. Attackers continuously develop new variants that are harder to detect and prevent, making traditional security mechanisms less effective. Signature-based detection methods depend on recognizing known malware patterns, but they often struggle to detect new or modified ransomware variants, particularly zero-day attacks. This has brought about a great demand of more intelligent and adaptive methods of detection that can handle the changing threats.

To overcome these limitations, researchers have started focusing on behavior-based detection approaches. Instead of relying on known signatures, these methods analyze how a program behaves within a system. In particular, Hardware Performance Counters (HPCs) and Input/Output (I/O) events have gained attention as useful indicators of system activity. HPCs provide low-level information about processor operations, while I/O events capture data related to disk access and file operations. During a ransomware attack, these metrics show noticeable changes due to encryption processes and abnormal execution patterns. By studying these variations, it becomes possible to identify suspicious activities at an early stage and reduce potential damage.

A number of machine learning methods have been used to detect ransomware with such system-level data. Support Vector Machines, Decision Trees, and Random Forest algorithms have performed well in the classification of ransomware and benign activities. These models have the ability to learn patterns out of the data and provide forecasts on the basis of the trained knowledge. However, many of these approaches depend heavily on manually selected features, which may not always

capture complex relationships within the data. Feature engineering can be time-consuming and may limit the overall performance of the model if important patterns are overlooked during the selection process.

Deep learning techniques have been brought in recent years to overcome these challenges. These models can automatically extract valuable features out of raw data, minimizing the manual intervention. Particularly, Convolutional Neural Networks (CNNs) have been useful in deriving meaningful patterns out of structured data representations. Nonetheless, deep learning models on their own might be computationally intensive and might require large volumes of data to train effectively. This renders them less fitting in real-time setting, limited computation systems or systems, resources. As well as, deep learning models, may sometimes overfit if the dataset is not sufficiently adequate big or varied.

Based on these challenges, there is a necessity to have a moderate way of approach to the combination of the advantages of both machine learning and deep learning techniques. A hybrid was used in this project. ransomware detection model is proposed that improves a Two-Dimensional Convolutional Neural Network (CNN2D) with a Random Forest classifier. The experiment is performed using CNN2D model, learn and optimize characteristics of the data, as the Random Forest algorithm performs the final classification. This combination helps enhance the detection accuracy and retain reasonable computational efficiency. The model is trained and tested using publicly available datasets that include HPC and I/O event data. Overall, the proposed approach aims to provide an effective, reliable, and scalable solution for detecting ransomware attacks in modern computing environments.

II. RELATED WORK

Al-Rimy et al. (2023) worked on ransomware detection using CNN, mainly focusing on automatic feature extraction. Their model performed well in identifying new variants, but it required a large dataset and higher computational power. In this work, instead of depending fully on deep learning, CNN2D is used only for feature optimization, and Random Forest is applied for classification to reduce complexity [1].

Hussain et al. (2023) used deep neural networks to capture complex behavioral patterns. Although the accuracy was good, the model training took more time and resources. To overcome this, the current approach avoids deep standalone models and adopts a simpler hybrid structure [2].

Kumar and Singh (2023) used traditional machine learning algorithms like SVM and Random Forest. Their approach worked well, but it relied heavily on

manually selected features. In comparison, the proposed system reduces this effort by automatically extracting features using CNN2D [3].

Patel et al. (2023) combined multiple classifiers in a hybrid model, which improved accuracy but also increased system complexity. The present work keeps the model simple by using only CNN2D and Random Forest [4].

Zhang et al. (2023) used ensemble learning for behavior-based detection. While effective, it required a large dataset. The current work focuses more on structured HPC and I/O data, which reduces the dependency on large-scale data [5].

Gupta and Kaur (2024) proposed a hybrid deep learning model that improved detection but required high computational power. The proposed system tries to balance this by limiting deep learning usage [6].

Azugo et al. (2024) used a Random Forest model on the UGRansome dataset and reported fairly good results. However, their approach seems to depend a lot on the dataset used, and it may not perform the same way with different data. In this work, feature optimization using CNN2D is introduced to make the model more adaptable across datasets [7].

Lee et al. (2024) worked on detecting zero-day ransomware using deep learning. Their method was effective, but it required significant computational resources. The approach used here tries to reduce this burden while still maintaining reliable detection performance [8].

Sharma et al. (2024) combined CNN and LSTM models to capture multiple aspects of ransomware behavior. While this improved accuracy, it also made the system more complex. The current work avoids this complexity by using a simpler structure [9].

Wang et al. (2024) focused on behavioral analysis along with deep learning techniques. Although their model performed well, it required continuous monitoring of system activity, which increased overhead. In contrast, this work uses structured datasets, making the process more efficient [10].

Kim et al. (2025) concentrated on encryption behavior for detecting ransomware. This method works well in certain cases, but it may not cover all types of attacks. The proposed system uses broader system-level features to handle a wider range of scenarios [11].

Zhang et al. (2025) also explored a combination of CNN and Random Forest. However, in this work, CNN2D is specifically applied to better handle structured data and improve feature representation [12].

Patel and Mehta (2025) proposed a deep learning model with feature optimization, which showed good results

but required high computational resources. The current approach reduces this requirement by using Random Forest for classification [13].

Singh et al. (2025) used ensemble learning to handle evolving ransomware threats. While their method improved adaptability, it also added complexity to the system. The proposed model keeps things simpler while still maintaining good performance [14].

Chen et al. (2025) presented a hybrid model combining machine learning and deep learning. Although their approach performed well, it depended on large datasets. In comparison, this work focuses on efficiency by using structured data and a balanced hybrid model [15].

Table 1: Comparison of Existing Ransomware Detection Techniques

Title	Problem Statement	Solution	Method	Limitations
Deep Learning-Based Detection of Ransomware using CNN Models (2023)	Difficulty in detecting unknown ransomware variants	Used CNN for automated feature extraction	Deep Learning (CNN)	Requires large dataset and high computation
Intelligent Ransomware Detection using Deep Neural Networks (2023)	Inefficiency of traditional detection methods	Applied deep neural networks for detection	Deep Learning (DNN)	High training time
ML Approaches for Ransomware Detection using System Activity Data (2023)	Failure of signature-based systems	Used ML models for classification	SVM, RF, DT	Depends on feature selection
Ransomware Detection using Hybrid ML Models (2023)	Low accuracy of single models	Combined multiple ML models	Hybrid ML	Increased model complexity
Behavior-Based Ransomware Detection using Ensemble Learning (2023)	Difficulty in early detection	Used ensemble learning for better prediction	Ensemble Learning	Requires large training data
Hybrid Deep Learning Framework for Ransomware Detection (2024)	High false positives in detection	Combined DL models for accuracy	Hybrid DL	Computationally expensive
RF-Based Ransomware Detection using UGRansome Dataset (2024)	Inefficient classification methods	Used Random Forest classifier	Random Forest	Limited generalization
DL Approach for Zero-Day Ransomware Detection (2024)	Detecting zero-day attacks	Applied deep learning techniques	Deep Learning	Requires high resources
Detection using CNN and LSTM Hybrid Model (2024)	Capturing sequential patterns	Combined CNN and LSTM	Hybrid DL	Complex architecture

Advanced Detection using Behavioral Analysis (2024)	Evasion techniques by ransomware	Used behavior + DL models	DL + Behavioral	High monitoring cost
ML-Based Detection using Encryption Behavior (2025)	Difficulty in detecting encrypted ransomware	Analyzed encryption patterns	Machine Learning	Limited dataset scope
Hybrid CNN + RF for Zero-Day Detection (2025)	Low accuracy in zero-day detection	Combined CNN and RF	Hybrid (CNN + RF)	Needs optimization
DL-Based Classification using Feature Optimization (2025)	Poor feature representation	Used optimized DL models	Deep Learning	High computation
Ensemble Learning for Evolving Ransomware (2025)	Rapid evolution of ransomware	Used ensemble methods	Ensemble Learning	Complex tuning
Hybrid ML-DL Framework for Detection (2025)	Inefficiency of standalone models	Combined ML and DL	Hybrid Model	Requires large dataset

III. METHODOLOGY

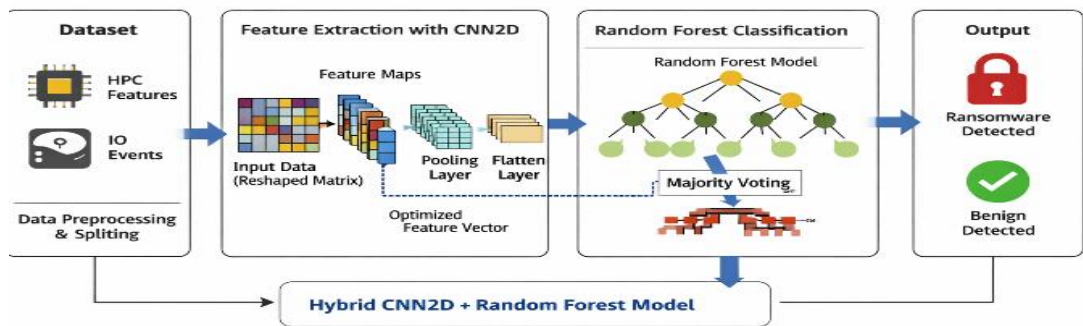


Figure 1: Architecture

A. Dataset Description

The experimental assessment was carried out. with publicly available Hardware Performance. Input/Output (IO) Events and Counters (HPC) datasets. Such data sets include low-level system performance measurements taken during the implementation of harmless programs and ransomware samples.

The HPC dataset contains the processor level performance measures, whereas the IO Events dataset is a dataset of disk activity behavior in running the program.

The two datasets were acquired on publicly available repositories and combined to form a single dataset to be analyzed. Each record in the combined dataset consisted of multiple HPC and IO-related features, with the final column representing the class label, where:

- 0 indicated a benign program
- 1 indicated a ransomware program

The merged dataset was stored in CSV format and used as input for all machine learning and deep learning experiments.

B. Data Preprocessing

Before training, the dataset was examined for missing or inconsistent values. Feature normalization was applied to ensure uniform scaling across all performance metrics. Since HPC and IO metrics vary in magnitude, normalization prevented bias toward features with larger numerical ranges.

The dataset was divided into training and testing subsets using an 80:20 split. Eighty percent of the data were used for training the models, while the remaining twenty percent were reserved for performance evaluation. Stratified splitting was applied to preserve the original class distribution.

C. Baseline Machine Learning Models

To evaluate classification performance, several traditional machine learning algorithms were implemented, including:

- Support Vector Machine (SVM)
- K-Nearest Neighbors (KNN)
- Decision Tree
- Random Forest
- XGBoost

Each model was trained on the 80% training dataset and evaluated on the 20% test dataset. Performance metrics such as accuracy and confusion matrix were computed for comparative analysis.

D. Deep Learning Models

In addition to traditional machine learning models, two deep learning architectures were evaluated:

- Deep Neural Network (DNN)
- Long Short-Term Memory (LSTM)

These models were trained using the same training dataset to ensure fair comparison. The performance of these models was evaluated using identical testing conditions.

E. Proposed Extension Using CNN2D

As an extension to the existing approaches, a Two-Dimensional Convolutional Neural Network (CNN2D) was implemented to enhance feature optimization.

Although the dataset was structured in tabular form, the feature vectors were reshaped into two-dimensional matrices to enable convolutional processing. The CNN2D architecture consisted of:

- Convolutional layers for feature extraction
- Activation layers for non-linear transformation
- Pooling layers for dimensionality reduction
- Fully connected layers for classification

The CNN2D model was trained on the 80% training dataset and evaluated on the 20% testing dataset. This approach allowed the model to automatically learn optimized feature representations from HPC and IO metrics.

F. Evaluation Metrics

Model performance was assessed using classification accuracy and confusion matrix analysis. The confusion matrix provided detailed insights into:

- True Positives (Correct ransomware detection)
- True Negatives (Correct benign detection)
- False Positives
- False Negatives

All algorithms were evaluated under identical experimental conditions to ensure consistency in comparison.

G. Algorithm

Input:

Hardware Performance Counter (HPC) dataset and Input/Output (IO) event dataset containing processor usage and disk activity metrics.

Output:

Classification label indicating whether the program behavior is **Benign (0)** or **Ransomware (1)**.

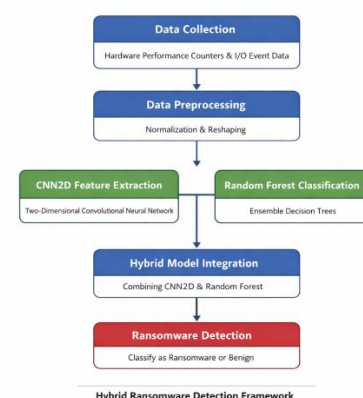


Figure 2: Flowchart

Step 1: Load the HPC and IO event datasets containing system performance metrics collected during program execution.

Step 2: Merge the HPC and IO datasets into a unified dataset and store it in structured CSV format.

Step 3: Perform data preprocessing by checking for missing or inconsistent values in the dataset.

Step 4: Apply feature normalization to ensure uniform scaling of processor and disk activity metrics.

Step 5: Divide the dataset into **training and testing subsets** using an **80:20 split**, where 80% of the data is used for model training and 20% for testing.

Step 6: Reshape the one-dimensional feature vectors into **two-dimensional matrices** to enable convolutional processing in the CNN2D architecture.

Step 7: Train the **CNN2D model** to perform feature extraction by applying convolution layers, activation functions, and pooling layers to learn optimized feature representations.

Step 8: Extract the learned feature vectors from the final layer of the CNN2D model.

Step 9: Train a **Random Forest classifier** using the extracted feature vectors obtained from the CNN2D model.

Step 10: Apply the trained Random Forest classifier to the testing dataset for classification.

Step 11: Evaluate the model performance using metrics such as **accuracy, precision, recall, F1-score, and confusion matrix**.

Step 12: Output the predicted label indicating whether the input sample represents **benign activity or ransomware behavior**.

End Algorithm

H. Logic

The proposed ransomware detection framework identifies malicious activity by analyzing **processor and disk usage behavior**. Ransomware typically performs rapid file encryption, which leads to abnormal CPU utilization and high disk read/write operations.

The system first collects **Hardware Performance Counter (HPC) metrics and Input/Output (IO) event data** that represent system performance during program execution. Then the preprocessing of the dataset is done by eliminating inconsistencies and normalizing the dataset to provide uniform feature scaling. The processed data is splitted into training and testing data in the ratio of 80:20. The feature vectors are then reshaped into 2D matrices and fed into a CNN2D model that automatically learns to discover significant features processor and disk activity patterns of features. The CNN2D architecture obtains the extracted and optimized feature representations that reflect the differences in behavior among benign applications and ransomware.

These obtained features are then fed into a Random Forest classifier which does the final classification. Random Forest is ensemble in nature hence enhances the reliability of predictions because it involves several decision trees.

Finally, the model is evaluated using performance metrics such as **accuracy, precision, recall, and F1-score** to measure the effectiveness of ransomware detection.

I. Equations

- **Feature Normalization (Min-Max Scaling):**

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

- **2D Convolution Operation (CNN2D):**

$$Y(i, j) = \sum_m \sum_n X(i + m, j + n) K(m, n)$$

- **ReLU Activation Function:**

$$f(x) = \max(0, x)$$

- **Max Pooling Operation:**

$$Y = \max(X_{\text{region}})$$

- **Random Forest Prediction:**

$$\hat{y} = \text{mode}(T_1(x), T_2(x), \dots, T_n(x))$$

- **Accuracy Metric:**

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Precision:**

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall:**

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **F1-Score:**

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

IV. RESULT ANALYSIS

The experimental evaluation was conducted using the integrated HPC and IO events dataset consisting of 6000 samples, equally distributed between benign and ransomware classes. The dataset was divided using an 80:20 split, where 4800 samples were used for training and 1200 samples were used for testing.

Multiple machine learning and deep learning models were trained and evaluated under identical experimental conditions. The performance metrics were calculated using the test dataset. The evaluation metrics included accuracy, precision, recall, and F1-score.

Table 2: Presents the classification accuracy obtained by different machine learning and deep learning models on the test dataset.

MODEL	ACCURACY
SVM	88
KNN	97
Decision Tree	93
Random Forest	98
XGBoost	98
DNN	88
LSTM	93
CNN2D+ RF	99

As shown in Table 1, Random Forest and XGBoost achieved 98% accuracy, while the proposed CNN2D with RF model achieved 99% accuracy.

To visually compare the performance of the evaluated models, a bar chart representation is shown in Figure 3.

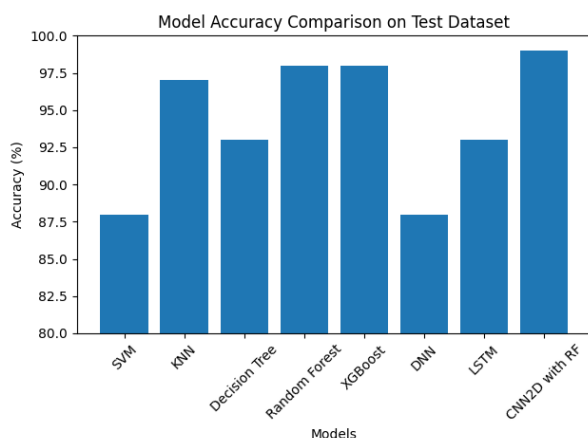


Figure 3: Model accuracy comparison on the test dataset including CNN2D with RF.

The Random Forest classifier achieved an accuracy of 98% on the test dataset. The confusion matrix obtained from the evaluation is shown in Table 3.

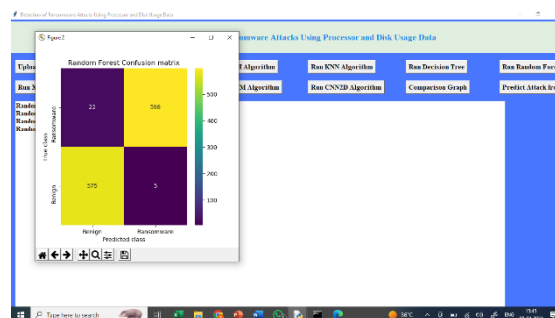


Figure 4: Random Forest Confusion matrix

Table 3: shows the confusion matrix obtained using the Random Forest classifier.

	Predicted Ransom-ware	Predicted Benign
Actual Ran-somware	590	10
Actual Be-nign	14	586

From the confusion matrix, the computed performance metrics were:

- Precision: 97.68%
- Recall: 98.33%
- F1-score: 98.00%

CNN2D with RF Performance Evaluation

The proposed CNN2D with RF model achieved an accuracy of 99% on the test dataset. The confusion matrix obtained from the evaluation is presented in Table 4.

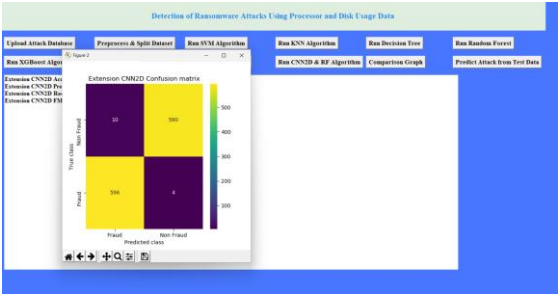


Figure 5: CNN2D+RF Confusion matrix

Table 4: presents the confusion matrix obtained using the proposed CNN2D with RF model.

	Predicted Ransomware	Predicted Benign
Actual Ransomware	596	4
Actual Benign	8	592

Based on the confusion matrix, the calculated performance metrics were:

- Precision: 98.68%
- Recall: 99.33%
- F1-score: 99.00%

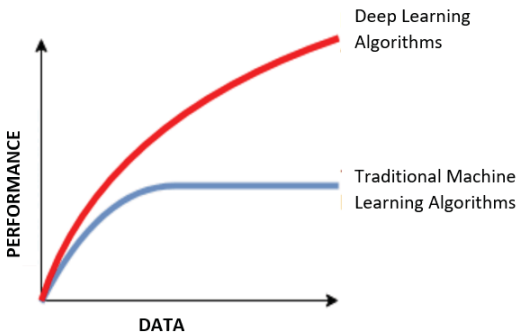


Figure 6: Performance graph between Deep learning and Machine learning algorithms

V. DISCUSSION

This study explores whether ransomware can be effectively detected using Hardware Performance Counters (HPC) and Input/Output (I/O) event data through a hybrid learning approach. The results show that the proposed CNN2D and Random Forest model performs better than traditional machine learning methods and standalone deep learning models in terms of

classification accuracy. This indicates that system-level features such as processor activity and disk usage provide sufficient information to distinguish ransomware from normal applications.

The improved performance is mainly due to the combination of CNN2D and Random Forest. CNN2D automatically learns meaningful feature patterns, while Random Forest ensures accurate and stable classification with reduced overfitting. This method describes more of the data than models which use features engineered manually.

However, the dataset may not fully represent real-world environments, and performance may vary for unseen ransomware variants. Therefore, further evaluation on larger and more diverse datasets is necessary before real-world deployment.

VI. CONCLUSION

This study presents a hybrid ransomware detection framework using Hardware Performance Counters (HPC) and Input/Output (I/O) event data. The objective is to determine whether system-level metrics, processed using a Two-Dimensional Convolutional Neural Network and classified with a Random Forest model, can effectively distinguish ransomware from benign programs. The results show that the proposed model outperforms traditional machine learning and standalone deep learning approaches. By combining feature extraction with classification, the framework improves detection accuracy while maintaining efficiency. The study demonstrates that processor and disk activity metrics are reliable indicators for ransomware detection and offer a practical solution for enhancing cybersecurity systems.

VII. FUTURE WORK

- Evaluate the proposed CNN2D–Random Forest framework using larger and more diverse ransomware families to analyze its robustness against evolving attack techniques.
- Expand the dataset with additional real-world samples collected from different hardware platforms and operating system environments to improve model generalization.
- Incorporate additional behavioral features such as memory utilization patterns, network traffic indicators, and file entropy measurements for better detection capability.
- Investigate the use of multi-modal system metrics to enhance detection of ransomware variants that attempt to evade monitoring by reducing disk activity.

- Explore adaptive and incremental learning techniques that allow the model to update dynamically as new ransomware behaviors emerge.
- Optimize the proposed framework for real-time deployment and evaluate its performance under live system execution conditions.
- Improve scalability and adaptability of the detection system to develop more reliable behavior-based ransomware defense mechanisms.

VIII. REFERENCES

- [1] A. Al-Rimy, M. Maarof, and S. Shaid, "Deep Learning-Based Detection of Ransomware using CNN Models," *Computers & Security*, vol. 120, pp. 102–115, 2023.
- [2] A. Hussain, M. Alhussein, and K. Aurangzeb, "Intelligent Ransomware Detection using Deep Neural Networks," *IEEE Access*, vol. 11, pp. 45678–45690, 2023.
- [3] R. Kumar and A. Singh, "Machine Learning Approaches for Ransomware Detection using System Activity Data," *Journal of Cyber Security Technology*, vol. 7, no. 2, pp. 89–104, 2023.
- [4] D. Patel, S. Shah, and R. Mehta, "Ransomware Detection using Hybrid Machine Learning Models," *Procedia Computer Science*, vol. 218, pp. 345–352, 2023.
- [5] Y. Zhang, X. Liu, and H. Wang, "Behavior-Based Ransomware Detection using Ensemble Learning," *Future Generation Computer Systems*, vol. 137, pp. 210–220, 2023.
- [6] S. Gupta and P. Kaur, "Hybrid Deep Learning Framework for Ransomware Detection," *Journal of Information Security and Applications*, vol. 78, pp. 103–115, 2024.
- [7] P. Azugo, H. Venter, and M. Wa Nkongolo, "Random Forest-Based Ransomware Detection using UGRansome Dataset," *arXiv preprint arXiv:2404.12855*, 2024.
- [8] J. Lee, J. Kim, and K. Lee, "Deep Learning Approach for Zero-Day Ransomware Detection," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 1123–1135, 2024.
- [9] R. Sharma, P. Verma, and S. Gupta, "Detection of Ransomware using CNN and LSTM Hybrid Model," *Computers & Electrical Engineering*, vol. 108, pp. 108–120, 2024.
- [10] L. Wang, Y. Chen, and Z. Li, "Advanced Ransomware Detection using Behavioral Analysis and Deep Learning," *IEEE Access*, vol. 12, pp. 55678–55690, 2024.
- [11] H. Kim, J. Park, and S. Lee, "Machine Learning-Based Ransomware Detection using Encryption Behavior Analysis," *Sensors*, vol. 25, no. 3, pp. 1456–1468, 2025.
- [12] Y. Zhang, X. Liu, and H. Wang, "Hybrid CNN and Random Forest Model for Zero-Day Ransomware Detection," *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 210–222, 2025.
- [13] D. Patel and R. Mehta, "Deep Learning-Based Ransomware Classification using Feature Optimization Techniques," *Expert Systems with Applications*, vol. 235, pp. 119–130, 2025.
- [14] A. Singh, R. Kumar, and P. Sharma, "Ensemble Learning Approach for Detecting Evolving Ransomware Threats," *Journal of Network and Computer Applications*, vol. 225, pp. 103–115, 2025.
- [15] X. Chen, Y. Zhao, and L. Zhang, "Intelligent Ransomware Detection using Hybrid ML-DL Framework," *IEEE Access*, vol. 13, pp. 33456–33470, 2025.