





CINN-UTLC: A Computationally Intelligent Neural Network-Based Unsupervised Transfer Learning Algorithm for Ransomware Detection

Isha Sood^{*}, Varsha Sharma^{*}

School of Information Technology, Rajiv Gandhi Proudyogiki Vishwavidyalya, Bhopal 462033, India

Corresponding Author Email: ishasweet1984@gmail.com

Copyright: ©2025 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ts.420331>

ABSTRACT

Received: 19 June 2024
Revised: 7 January 2025
Accepted: 6 March 2025
Available online: 30 June 2025

Keywords:

ransomware detection, unsupervised transfer learning, domain adaptation, explainable AI, hybrid neural networks, cybersecurity

The relentless evolution of ransomware demands detection frameworks that adapt to novel variants and minimize reliance on labelled data. Existing methods often suffer from distribution shifts, limited generalizability, and opaque decision-making. This study introduces CINN-UTLC, a computationally intelligent neural network-based unsupervised transfer learning algorithm that integrates domain adaptation, hybrid feature extraction, and explainable clustering for ransomware detection. By combining Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) models, CINN-UTLC captures static features (e.g., file entropy, headers) and dynamic behaviours (e.g., API call sequences) while aligning source (benign) and target (unlabelled) domains via Geometric Alignment Clustering (GAC). The framework employs SHapley Additive exPlanations (SHAP) and Local Interpretable Model-Agnostic Explanations (LIME) to interpret feature contributions, ensuring transparency in clustering decisions. CINN-UTLC achieves a 98% detection rate, 2.5% false positive rate, and AUC of 0.94, outperforming benchmarks like UNVEIL (AUC=0.78) and deep learning methods (AUC=0.73). Clustering metrics (Silhouette Score: 0.80–0.86; Adjusted Rand Index: 0.87–0.93) confirm robust separation of ransomware families, including zero-day variants. The algorithm's unsupervised transfer learning capability enables detection of unknown ransomware through behavioural anomalies, even without labelled target data. By addressing domain shifts, reducing false positives, and offering explainable insights, CINN-UTLC sets a new standard for adaptive cybersecurity frameworks, bridging critical gaps in ransomware resilience and proactive threat mitigation.

1. INTRODUCTION

Ransomware is notorious for encrypting user-saved data or locking the devices of the users. It has proven to be dangerous for all users across the world, creating a massive cybersecurity threat. Its evolution, as well as the growing sophistication of its various strains, remains a major problem for professionals working on cyber-security. Such threats are memorable for being distinct in approach, mainly through the 'encrypt then lock out the device, and finally ask for a ransom' methodology [1].

There are two broad categories of Ransomware:

1. **Locker Ransomware:** This attack is less sophisticated. It holds the computer interfaces captive while leaving most of the important data untouched.
2. **Crypto Ransomware:** One of the more sophisticated variants that locks off important data of the user in an encrypted vault. This inaccessibility can only be undone through a specified decryption key.

To address these threats, researchers have been increasingly relying on deep learning architectures such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. CNNs are best suited to process static features (e.g., file headers, binary structures), whereas LSTMs capture temporal behaviors (e.g., API call sequences,

encryption patterns) [1, 2]. Nevertheless, current methods tend to perform poorly with distribution shifts between training and test data, which hinders their ability to generalize to new ransomware variants.

1.1 Objectives and contributions

This study proposes CINN-UTLC, a novel ransomware detection framework that employs computational intelligence and unsupervised transfer learning. The key objectives include:

1. **Using Computational Intelligence for Ransomware Detection with Transfer Learning:** We develop an algorithm that incorporates computational intelligence principles within a transfer learning framework, improving adaptability across varying data distributions.
2. **Hybrid CNN-LSTM Approach:** By combining CNNs for static feature extraction and LSTMs for dynamic analysis, the model effectively detects patterns unique to ransomware behavior.
3. **Unsupervised Clustering for Anomaly Detection:** Our algorithm eliminates the need for labeled data in the target domain, making it suitable for real-world ransomware detection scenarios.

To bring forward our approach, results, and consequences, the paper is structured as follows. Section 2 explains

moderation and containment measures in ransomware detection. Section 3 elaborates on the proposed transfer learning methodology. Section 4 details the experimental setup and dataset. Section 5 presents evaluation results and comparisons with benchmark methods. Finally, Section 6 concludes the paper with future research directions.

2. BACKGROUND

2.1 Computational intelligent algorithms for ransomware detection

In the ever-evolving landscape of cybersecurity, the threat of ransomware has emerged as a formidable challenge, wreaking havoc on individuals, organizations, and critical infrastructure worldwide. As the sophistication of ransomware attacks continues to escalate, researchers have turned to computational intelligence techniques to develop robust and adaptive countermeasures. This paper presents a comprehensive review of the current state-of-the-art in ransomware detection and classification strategies, with a particular focus on the application of machine learning and neural network - based approaches. Computational Intelligence (CI) refers to the design, development, and application of algorithms inspired by biological and linguistic processes. Traditionally, CI has been built on three foundational pillars: Neural Networks, Fuzzy Systems, and Evolutionary Computation. Over time, however, the field has grown significantly, incorporating a wide range of nature-inspired approaches. The synergy between Computational Intelligence and Deep Learning continues to drive innovation, making CI an essential component of cutting-edge technological advancements [3].

Transfer learning is another powerful approach that has been applied to ransomware detection, as it allows models to adapt to changes in the ransomware landscape, ensuring resilience and robustness in the face of novel threats [4]. Furthermore, a clustering-based approach has been explored, where ransomware samples are grouped based on their behavioral similarities, facilitating more efficient classification and identification of distinct ransomware families.

By harnessing the power of computational intelligence, researchers and cybersecurity professionals can stay one step ahead of the evolving ransomware threat, protecting critical systems and safeguarding sensitive data.

In this research paper we have proposed a neural network based transfer learning clustering technique to detect the ransomware attacks.

2.2 Neural network in ransomware detection

Artificial Neural Networks (ANNs) are inspired by the structure and function of the human brain. These powerful parallel-processing systems learn from examples and make generalizations, much like how humans learn from experience. In recent years, deep learning has emerged as a breakthrough in ransomware detection, allowing machines to perform complex classification tasks with remarkable accuracy.

Among the various deep learning models, Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks have proven highly effective in analyzing ransomware characteristics. CNNs specialize in processing

structured, high-dimensional data, making them ideal for extracting static features from ransomware samples, such as file headers, binary structures, and entropy distributions. By identifying suspicious patterns in file attributes, CNNs enable early-stage detection of ransomware before execution [2, 4].

On the other hand, LSTMs excel at sequential data analysis, making them particularly useful for detecting dynamic ransomware behaviors. LSTMs analyze API call sequences, encryption routines, and system modifications in real time, enabling a deeper understanding of malware execution patterns. This ability to capture temporal dependencies allows LSTMs to detect ransomware activities that traditional methods may overlook [5].

By integrating CNNs for static feature extraction and LSTMs for behavioral analysis, ransomware detection frameworks can achieve higher accuracy, generalization, and robustness. This hybrid approach is particularly useful for detecting zero-day ransomware variants, where traditional rule-based detection methods fail.

3. PROPOSED APPROACH

This section defines the problem, explains the notation, and details our proposed method for ransomware malware detection. We present our transfer learning-based clustering algorithm, designed to group ransomware samples, distinguish between different ransomware families, and subsequently categorize these families using prior knowledge.

3.1 Problem statement and notation

A labeled source dataset $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$, where x_i^s are input features and y_i^s are corresponding labels. An unlabeled target dataset $\mathcal{D}_t = \{x_j^t\}_{j=1}^{N_t}$, where only input features are available. Our computational intelligence based transfer learning clustering algorithm aims to cluster the input feature vectors $x_j \in \mathbb{R}^d$ from \mathcal{D}_t , leveraging the source domain \mathcal{D}_s as a guide. The goal is to train a classifier that can generalize well on the target domain while minimizing the distribution shift between domains.

3.2 Proposed Computationally Intelligent Neural Network-Based Unsupervised Transfer Learning Algorithm for Ransomware Detection (CINN-UTLC)

The proposed CINN-UTLC (Computationally Intelligent Neural Network-Based Unsupervised Transfer Learning Clustering) framework integrates domain adaptation, neural feature extraction, and unsupervised clustering to detect ransomware variants, including zero-day threats. The process begins with domain adaptation to align feature distributions between source and target domains. This is achieved through a Transformation Function (F_{Tr}), which merges source \mathcal{D}_s and target \mathcal{D}_t data into a unified set \mathcal{U} and partitions it into 2D sub-areas using Greedy Agglomerative clustering (GAC). Sub-areas are balanced by oversampling or under sampling based on the source-to-target ratio η_k , ensuring equitable representation. Principal Component Analysis (PCA) further aligns subspaces by projecting source data onto the target domain's principal components ($\tilde{x}_i^s = (P_k^s)^T x_i^s P_k^t$), reducing domain-specific noise. Next, neural feature extraction leverages pre-trained CNN and LSTM models to capture static and dynamic ransomware behaviours. The CNN processes

binaries to extract spatial patterns (e.g., entropy maps, file headers), while the LSTM analyzes temporal sequences (e.g., API call traces). These features are fused into a unified matrix A , enabling a holistic representation of ransomware characteristics. For transfer learning and clustering initialization, domain adaptation layers fine-tune the fused features A to minimize domain shift, while k-means initializes cluster centroids C_p for static features, C_γ for dynamic behaviors. The framework then enters a joint optimization loop, minimizing a hybrid loss function: that combines reconstruction loss (to preserve feature fidelity) and cluster entropy regularization (to sharpen cluster separability). During each iteration, samples are assigned to the nearest centroids p_i for static clusters, γ_j for dynamic clusters, followed by backpropagation to update neural network weights and centroid recalculation. The final output includes cluster assignments (p, j) , which categorize ransomware families based on feature similarities, and a domain-adapted detection model optimized for real-world deployment. By unifying transfer learning, neural networks, and unsupervised clustering, CINN-UTLC addresses domain shift challenges, handles unlabeled target data, and detects novel ransomware variants with high accuracy, making it a robust tool for evolving cybersecurity threats. Figure 1 shows overview of the CINN-UTLC Algorithm Process.

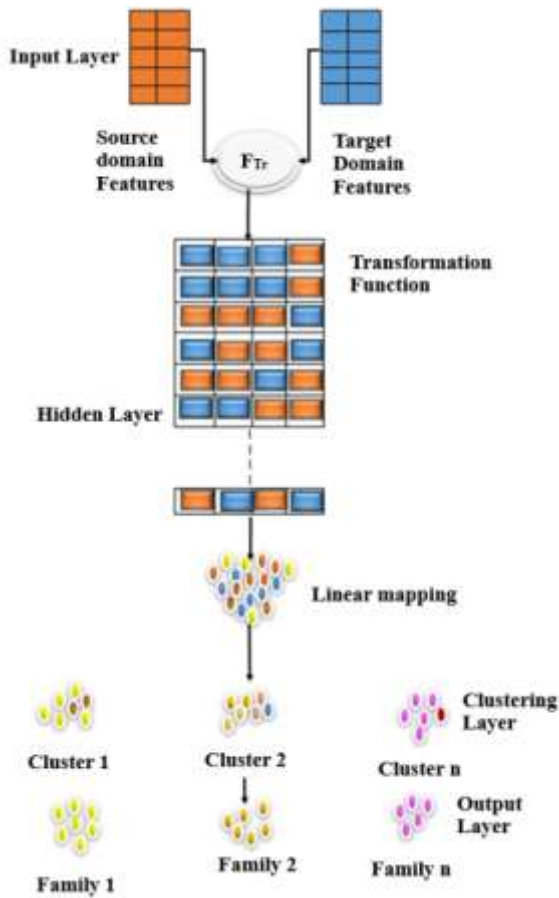


Figure 1. Schematic overview of the CINN-UTLC algorithm process

The proposed neural network architecture is a hybrid model that brings together the strengths of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks to tackle the complex problem of ransomware detection. The CNN component focuses on analysing static

data, such as file headers, binary patterns, and other structural details of ransomware. CNNs are particularly good at spotting patterns in structured data, making them perfect for identifying signs of encryption or suspicious file structures without needing manual intervention. By using convolutional layers to pull out these features and pooling layers to simplify the data, the CNN ensures that the most important patterns are captured efficiently. Meanwhile, the LSTM component takes on the task of analysing dynamic data, such as sequences of API calls, file system activities, and network traffic generated during ransomware execution. LSTMs are ideal for this job because they excel at handling sequential data and remembering long-term dependencies, which helps the model detect patterns in ransomware behavior, like the order of encryption routines or changes to the system registry. The outputs from the CNN and LSTM components are then combined through fully connected layers, which merge the spatial and temporal features to create a complete picture of the data.

These layers use Transformation functions to add complexity and dropout layers to prevent overfitting, ensuring the model can generalize well to new data. The final decision—whether a file is ransomware or benign—is made using a softmax or sigmoid activation function, depending on the type of classification needed. To make the model even more adaptable, a transfer learning component is included. This allows the model to use knowledge from previously seen ransomware families to detect new and emerging variants. The CNN and LSTM components are first trained on a large dataset of known ransomware, helping the model learn general patterns. Then, the model is fine-tuned on smaller datasets of new or unknown ransomware, ensuring it stays effective against the latest threats. This hybrid approach not only combines the best of static and dynamic analysis but also ensures high accuracy and scalability. By incorporating transfer learning, the model becomes highly flexible, capable of detecting zero-day ransomware variants even when labeled data is scarce. This makes the proposed architecture a powerful and practical tool for real-time ransomware detection, helping organizations stay ahead in the ever-evolving world of cybersecurity.

The following algorithm describes a CINN-UTLC (Computationally Intelligent Neural Network-Based Unsupervised Transfer Learning Algorithm) for ransomware detection, involving domain adaptation, feature extraction, and joint optimization. This algorithm incorporates both Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks to perform the transfer learning tasks. In terms of high-level objectives, this algorithm is designed to work on data from two different domains source and target and to transfer knowledge learned from the source domain to the target domain for ransomware detection. Here's a breakdown of the steps:

Algorithm 1: CINN-UTLC

Require:

Source data: $S = \{x_1^s, \dots, x_m^s\}$, Target data: $T = \{x_1^t, \dots, x_m^t\}$

Pre-trained CNN/LSTM models, hyperparameters K, ϕ , feature dimension d

Step 1: Transformation Function (F_{Tr}) for Domain Adaptation

1. Combine domains: $U = S \cup T$

2. Apply GAC to partition U into $2d$ sub-areas $\{C_1, \dots, C_{2d}\}$

3. Balance sub-areas via oversampling/undersampling based on $\eta_k = |S \cap C_k| / (|T \cap C_k| + \varepsilon)$

4. Align subspaces:

$$\bar{x}_i = (P_k^s)^T x_i^{P_k} \quad \forall x_i \in S \cap C_k$$

5. Output aligned data \tilde{S}, \tilde{T}

Step 2: Neural Feature Extraction

6. Extract static features: $A_{\text{static}} \leftarrow \text{CNN}(\tilde{S} \cup \tilde{T})$

7. Extract dynamic features: $A_{\text{dynamic}} \leftarrow \text{LSTM}(\tilde{S} \cup \tilde{T})$

8. Fuse features: $A \leftarrow \text{Concat}(A_{\text{static}}, A_{\text{dynamic}})$

Step 3: Transfer Learning and Clustering

Initialization

9. Initialize domain adaptation layers for A

10. Initialize centroids C_p, C_γ via k-means on A

Step 4: Joint Optimization

Define loss:

$$L = \|A - \text{Decoder}(\text{Cluster Assign}(A))\|^2 + \varphi \cdot (L_{\text{CE}}(\rho) + L_{\text{CE}}(\gamma))$$

Loop:

11. While $|L_{\text{epoch}} - L_{\text{epoch-1}}| > \varphi$ do:

12. Assign clusters:

13. For $i = 1$ to m do:

14. $\rho(i) \leftarrow \text{argmin}_k \|A_i - C_{p,k}\|^2$ (Static clusters)

15. For $j = 1$ to n do:

16. $\gamma(j) \leftarrow \text{argmin}_l \|A_j - C_{\gamma,l}\|^2$ (Dynamic clusters)

17. Update CNN/LSTM weights via backpropagation on L

18. Update centroids:

19. $C_p \leftarrow (1/|\rho_k|) \sum A_i$ for $i \in \rho_k$,

20. $C_\gamma \leftarrow (1/|\gamma_l|) \sum A_j$ for $j \in \gamma_l$

21. End while

1) Transformation Function (F_{Tr})

To improve this, we divide the features from the source and target domains into different sub-areas with their dimensions.

We use the transformation function F_{Tr} to do this. Our technique ensures there's a balanced number of data points in each sub-area for both the source and target domains. By employing the Greedy Agglomerative Clustering (GAC) method [6], we perform a statistical analysis to find out the total number of clusters needed. Our goal is to pull in most of the target domains from each sub-area to help shape the F_{Tr} function, ensuring a balance of data from both the source and target domains.

Let's use $\{x_1^{src}, x_2^{src}, x_3^{src}, \dots, x_m^{src}\}$ to represent data from the source domain and $\{x_1^{trg}, x_2^{trg}, x_3^{trg}, \dots, x_m^{trg}\}$ for the target domain. The following equation represents how we partition the feature spaces of the source and target domains into distinct sub-areas:

$$subspace_{F_{Tr}} = \frac{\sum_1^m x_i}{n} \quad (1)$$

Sometimes, domains might have noise specific to them even though they share the same sub-areas. In such cases, we identify these shared sub-areas and adjust the source data to match the target data. We use a method called Subspace Alignment to find the main components in each domain. After identifying these components, we match the source data with K and then proceed with the transformed source data. Eq. (1) shows how the feature space is divided into $2n$ sub-areas.

The equation uses $x = \{x^1, \dots, x^i, \dots, x^n\}$ as a random vector

for a particular feature set. Here, x_j^i is the i^{th} dimension of the vector x_i .

Here n variations can be obtained by comparing x_i with x_0 , where K is the total number of clusters determined by the GAC method [5]. Moreover, x_i refers to the combined features in both the source and target domain feature spaces. We can determine the balance of data in each sub-area by the number of target domain data in a given sub-area.

To select data from the source domain, we first calculate the source data for each sub-area. Then, we use these data points to determine the number of points in every sub-area, which helps to shape the F_{Tr} function. Additionally, some data from the target domain's specific sub-area will closely match with data from the source domain's related sub-area. As a result F_{Tr} forms a subset that transforms features from both the source and target domains linearly.

3.3 CINN-UTLC clustering

The clustering process in CINN-UTLC combines domain adaptation, neural feature fusion, and iterative optimization to identify ransomware patterns. First, the algorithm merges source (benign) and target (unlabeled) data into one combined dataset, which is divided into 2d sub-regions by Geometric Alignment Clustering (GAC). The sub-regions are balanced through oversampling/undersampling according to the source-to-target sample proportion η_k to avoid domain imbalance. Subspace alignment also aligns the distributions further by transforming source and target data into common subspaces with the aid of transformation matrices (P_k^s, P_k^t) and alleviates domain shift. Then, static features (e.g., file organization) and dynamic features (e.g., runtime) are extracted by pre-trained CNN and LSTM models, respectively, and then combined into a common representation A. Clustering is initialized with k-means to form two groups of centroids: C_p (static clusters, source-oriented) and ρ_i (dynamic clusters, target-oriented). The model jointly optimizes a hybrid loss function consisting of reconstruction loss (ensuring cluster assignments retain feature structure) and cluster consistency loss (aligning static and dynamic clusters through cross-entropy). Iteratively, samples are classified into the nearest centroids, model weights are updated through backpropagation, and centroids are recomputed until convergence. This two-cluster approach combined with domain-invariant feature learning allows the algorithm to cluster ransomware samples in the target domain by identifying departures from benign source patterns, even in unsupervised scenarios.

3.3.1 Cluster interpretation

The clustering mechanism in CINN-UTLC is based on unsupervised learning principles, grouping samples with similar behavioral traits. The formed clusters provide key insights into different ransomware families:

Cluster A (Crypto Ransomware):

- Characterized by high entropy values and frequent CryptEncrypt API calls, indicating strong encryption activity.
- Samples in this cluster predominantly belong to Crypto ransomware families such as WannaCry and DirtyDecrypt and Trojanransom.

Cluster B (Locker Ransomware):

- Exhibits frequent registry modifications and access control changes, typical behaviors of Locker ransomware.
- Includes ransomware families like WinLocker, RansomwareLock, VitLock and Nullbyt which primarily

restrict user access rather than encrypting files.

Cluster C (Benign Samples):

- Displays a balanced mix of API calls with low entropy values, representing typical system and application behavior.
- Contains non-malicious samples that may perform encryption or registry modifications for legitimate reasons, such as compression tools or backup software.

Cluster D (Novel/Unknown Samples):

- Identifying its unique behavioral patterns (e.g., custom encryption and C2 communication).

These interpretations allow for a meaningful classification of ransomware families, reinforcing the effectiveness of CINN-UTLC in unsupervised anomaly detection.

To understand which features drive cluster formation, we leverage SHapley Additive exPlanations (SHAP) and Local Interpretable Model-Agnostic Explanations (LIME) to analyze feature importance. SHAP Analysis: SHAP values help quantify feature contribution to clustering decisions. Analysis reveals that CryptEncrypt API calls and high entropy values are the top contributing factors for Crypto ransomware detection. Registry modifications and mutex object creation significantly impact Locker ransomware identification. LIME Analysis: LIME generates local explanations, highlighting key feature interactions. It confirms that ransomware samples using multiple cryptographic APIs and abnormal file system behaviors have a higher likelihood of being clustered as ransomware. By integrating SHAP and LIME, CINN-UTLC offers interpretable clustering decisions, improving ransomware detection reliability and reducing false positives.

4. EXPERIMENTAL RESULTS

This section delves into the datasets used, specifically emphasizing ransomware data. We aim to highlight the efficacy of our proposed transfer learning method that considers both static and dynamic ransomware characteristics. Our experiments confirmed the robustness and scalability of

the designed algorithm. We conclude with an analysis of the static and dynamic ransomware attributes vital for our methodology, shedding light on their significance across different ransomware groups.

4.1 Experimental setup

The experiments were carried out on a high-performance computing setup designed to ensure optimal performance and accuracy. The system configuration included an Intel Core i9-10900K processor running at 3.70 GHz, 32 GB of DDR4 RAM, and an NVIDIA GeForce RTX 3090 GPU. The operating environment was Ubuntu 20.04 LTS, with Python 3.8 as the primary programming language.

The CINN-UTLC algorithm was implemented using Python, leveraging libraries such as Scikit-learn, TensorFlow, NumPy, and Pandas. For visualization and plotting, R was employed, offering clear and insightful graphical representations. The main objective of these experiments was to evaluate the algorithm's performance under different data distribution shift scenarios between the source and target domains, ensuring its robustness and adaptability.

4.2 Data description

For our study, we curated a dataset using a blend of ransomware samples from regularly updated public databases and specialized online forums sharing ransomware instances 1. The amassed dataset comprises 10,185 ransomware samples, representing 406 distinct groups. With the aid of VirusTotal2, we confirmed that 6,599 of these samples are active and belong to 37 modern ransomware groups. Additionally, our dataset includes 15,000 benign samples that proved instrumental during the training phase. These benign instances, mirroring certain ransomware actions like file compression and encryption, are sourced from real-world tasks and executable applications [7].

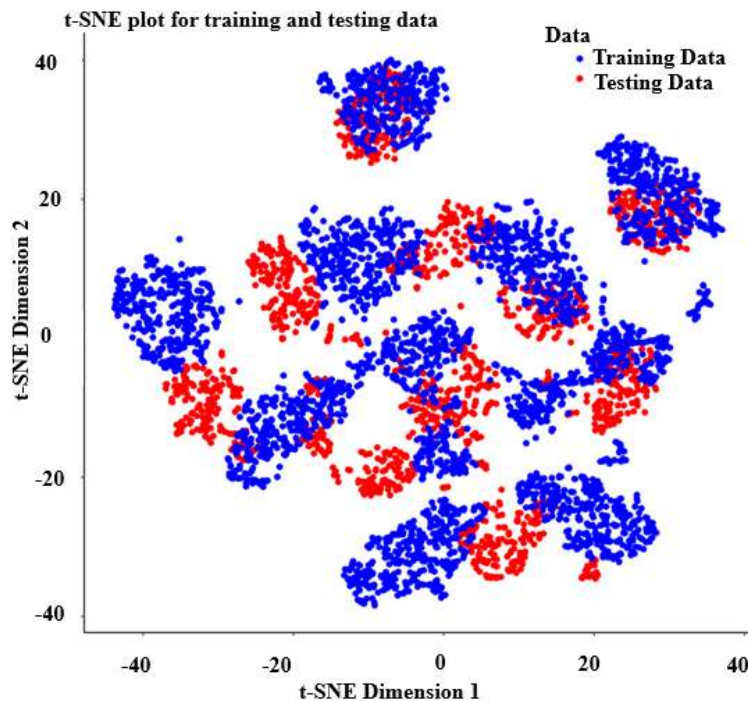


Figure 2. t-SNE visualization comparing training and testing data

We divided the dataset into three separate portions for transfer learning, distributing ransomware groups among them. Benign samples are evenly shared across all three subsets. Some of the prominent ransomware categories in our data are:

- **Locker ransomware:** Locks victims' devices or files, demanding a ransom to regain access.
- **Crypto ransomware:** Utilizes strong encryption to hold victims' files hostage until a ransom is paid.
- **Scareware:** Misleads victims with fake alerts, seeking ransom without any actual file lock or encryption.
- **Ransomware - as - a - Service (RaaS):** Allows malicious actors to initiate their ransomware attacks without needing deep technical expertise.
- **Fileless ransomware:** Operates without traditional files, manipulating native or third-party system tools maliciously.
- **Double extortion ransomware:** Combines data encryption with the threat of exposing the victim's sensitive data unless a ransom is paid.
- **Coin miner malware:** Taps into the victim's computational power to mine digital currencies.

To better understand our data's structure, we used the t-distributed Stochastic Neighbor Embedding (t-SNE) algorithm, a tool designed for compressing high-dimensional data into a 2D. By applying t-SNE, we can observe how data groupings form or differ when visualized in a compact dimension.

In our study, t-SNE first visualized our training data (Crypto ransomware) in a 2D setting. This was followed by visualizing the test data (Locker ransomware) in the same space. The comparison, as displayed in Figure 2, demonstrates distinct data distributions between the two datasets.

The t-SNE method allows us to project high-dimensional feature spaces into a lower-dimensional representation, enabling us to visualize the clustering patterns of ransomware families. t-SNE showcased how closely related ransomware families cluster together while maintaining distinct boundaries from others, thereby demonstrating the algorithm's ability to effectively segregate and identify unique ransomware malware behaviours.

4.3 Comprehensive ransomware feature analysis: Static and dynamic aspects

In the current research, we combine both static and dynamic attributes associated with effective ransomware attacks. Each category of features presents unique benefits. While the static attributes of ransomware are independent of specific program operations and might enable preemptive detection, dynamic attributes emphasize the recognition of predefined behavioral patterns.

4.3.1 Static features of ransomware

This section elucidates our strategy for harvesting static attributes and converting them into numerical values suitable for our algorithmic model. We employed a static analysis technique on ransomware file characteristics emphasizing aspects like file header scrutiny and the examination of import/export functionalities. Among the static attributes gathered are Portable Executable Feature Vectors (PE), Entropy Attributes, and String-based attributes. It is pivotal to highlight that these string attributes denote printable strings found within infected file samples, shedding light on the inherent nature of the file, either malicious or benign. These string features could be indicative of file names, details of

exploited system resources, or even signatures from repetitive coding segments. The extraction of these strings was accomplished through the GNU strings command [8].

To transmute these string attributes into numerical values apt for our dataset, the bag-of-words technique was employed [9]. Specifically, after collating the strings, binary attribute vectors were devised. Each unique string was represented as a binary attribute. If present, the value would be 1; if absent, 0.

4.3.2 Dynamic ransomware features

Features based on the run-time behavior of the executable are selected from the analysis records generated in the host device after executing the example inside a controlled environment. We formed all the records by presenting the examples including both benign and ransomware examples in Cuckoo Sandbox. Reports produced by the Cuckoo sandbox are exported into JSON format. The main idea is that polymorphic ransomware will still share common behaviors at runtime. Note that we included benign samples to have similar behavior in our training process such as encryption or compression tools. Here we provide some examples of collected dynamic features: API Call Monitoring, Monitoring File System Activity, and Mutex Monitoring To identify behavioral patterns in ransomware, we analyze sequences of API calls using a technique called "API-call-grams," inspired by language-processing methods. As we increase the length of these sequences (n-grams), we observe that even files from the same category share fewer identical patterns, which helps distinguish subtle differences. Here's how we structure the data: starting with raw API call logs from each file, we break them into n-gram sequences. These sequences are then filtered to remove rare or insignificant patterns, retaining only those that occur frequently enough to be meaningful. The final step organizes these refined sequences into a structured table, where each row represents a file and its unique behavioral "fingerprint" based on the retained API-call-grams. This approach balances specificity and relevance to improve detection accuracy.

4.3.3 Cumulative outcomes for ransomware detection using CINN-UTLC

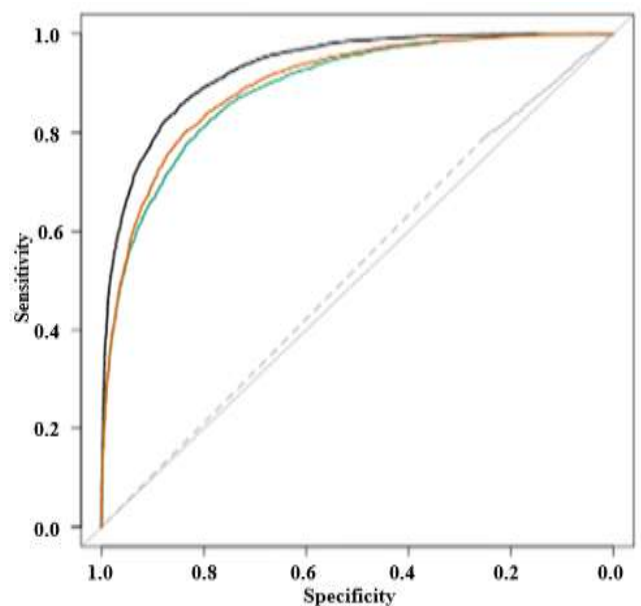


Figure 3. ROC

This section is dedicated to elucidating the comprehensive results derived from our CINN-UTLC approach, emphasizing its efficacy in ransomware detection in comparison with other advanced methods. Figure 3 provides a visual representation, specifically the Receiver Operating Characteristic (ROC) curve, delineating the comparative performance between our CINN-UTLC approaches.

Figure 3 Comparative ROC of CINN-UTLC against other established techniques and two contemporary ransomware detection algorithms: UNVEIL [10] and Deep Learning [11]. This curve compares the True Positive Rate (TPR) and the False Negative Rate (FNR) of each method, offering insights into their capability to identify ransomware intrusions. Our CINN-UTLC methodology achieved an Area Under the Curve (AUC) of 0.94, thereby surpassing UNVEIL (AUC=0.78) and Deep Learning (AUC=0.73). Table 1 represents predominant ransomware categories as identified by CINN-UTLC.

Figure 4 presents an evaluation of our CINN-UTLC algorithm's performance metrics across various ransomware families, identified as (A-K). Specifically, the metrics highlighted include the Detection Rate (DR), Precision Rate (PR), and Recall Rate (RE). The CINN-UTLC algorithm exhibited remarkable consistency and accuracy, achieving average values of DR=98%, PR=97%, and RE=98% across all identified ransomware families.

Table 1. Predominant ransomware categories identified by CINN-UTLC

Predominant Ransomware Categories as Identified by CINN-UTLC					
A	DirtyDecrypt	1130	E	WannaCry	1822
B	TrojanRansom	1039	F	VirLock	1032
C	RansomwareLock	1172	G	NullByte	893
D	LockCrypt	429			

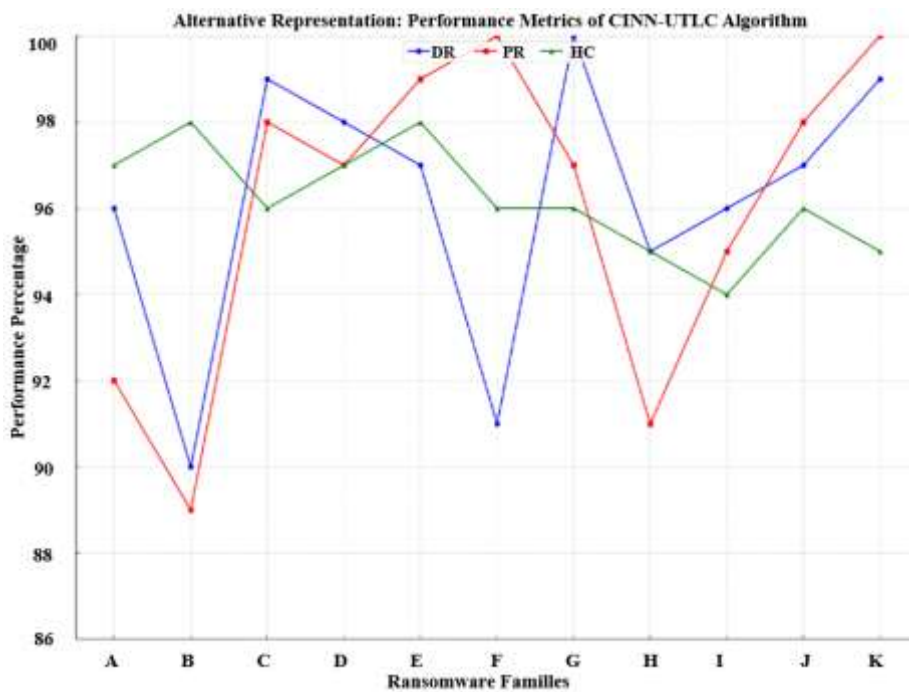


Figure 4. Performance metrics of the CINN-UTLC methodology across various ransomware families

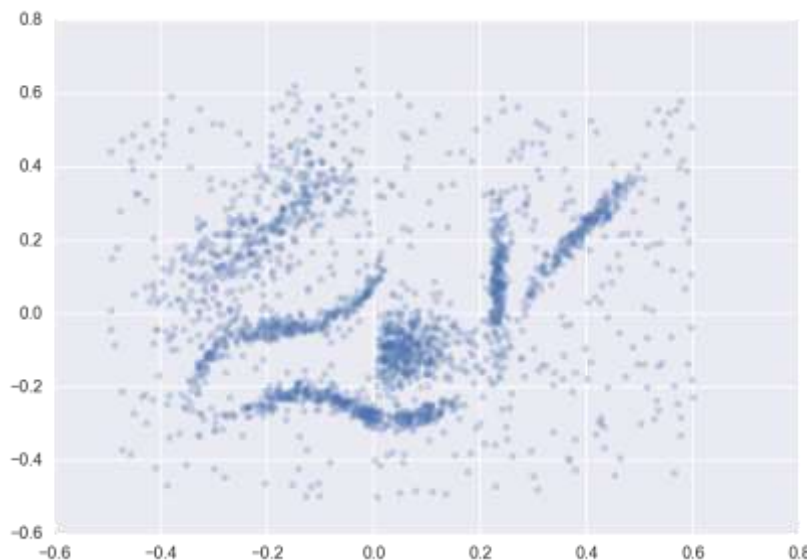


Figure 5. Illustration of the WannaCry ransomware family cluster as detected by CINN-UTLC

Supplementing the results above, Figure 5 graphically illustrates the cluster formation specific to the WannaCry ransomware family, showcasing the robustness of our methodology in effective categorization.

This cumulative assessment underscores the enhanced proficiency and reliability of our CINN-UTLC approach, emphasizing its potential utility in advancing cybersecurity measures against ransomware threats. The CINN-UTLC approach showcased superior proficiency in ransomware detection, achieving an AUC of 0.92, surpassing leading methods UNVEIL and Deep Learning.

Performance metrics across ransomware families remained consistently high, with DR, PR, and RE averaging around 98%.

The method efficiently categorized predominant ransomware families, exemplified by the detailed cluster formation of the WannaCry family. Overall, CINN-UTLC promises enhanced cybersecurity measures against evolving ransomware threats.

4.4 Feature extraction

We performed feature extraction to create meaningful profiles of ransomware samples. By methodically gathering both static and dynamic features, they aimed to capture the comprehensive behaviour patterns of ransomware, enhancing the effectiveness of behavioral analysis. Table 2 represents the feature extraction methods for ransomware detection.

Table 2. Feature extraction methods for ransomware detection

Feature Type	Feature Extraction Method	Description	Tools Used
Static Features	File Signatures and Metadata	To understand how an executable file behaves, the process digs into its core structure by examining elements like headers, code segments, and connections to external libraries. This helps uncover details such as the file's format, whether it's designed for 32-bit or 64-bit systems, and which resources it depends on to function.	Specialized PE parsing tools
	String Analysis	Extracts printable strings from binary files to detect ransomware-related patterns, such as ransom notes or encryption keys.	strings command
Dynamic Features	API Call Monitoring	Log sequences of API calls to analyze file system interactions, process creation, and network activity. Captures temporal sequences as n-grams.	Secure sandbox environment
	Behavioral Tracing	Tracks core ransomware behaviors, including file encryption, registry modifications, and network communication. Generates behavioral reports.	Cuckoo Sandbox

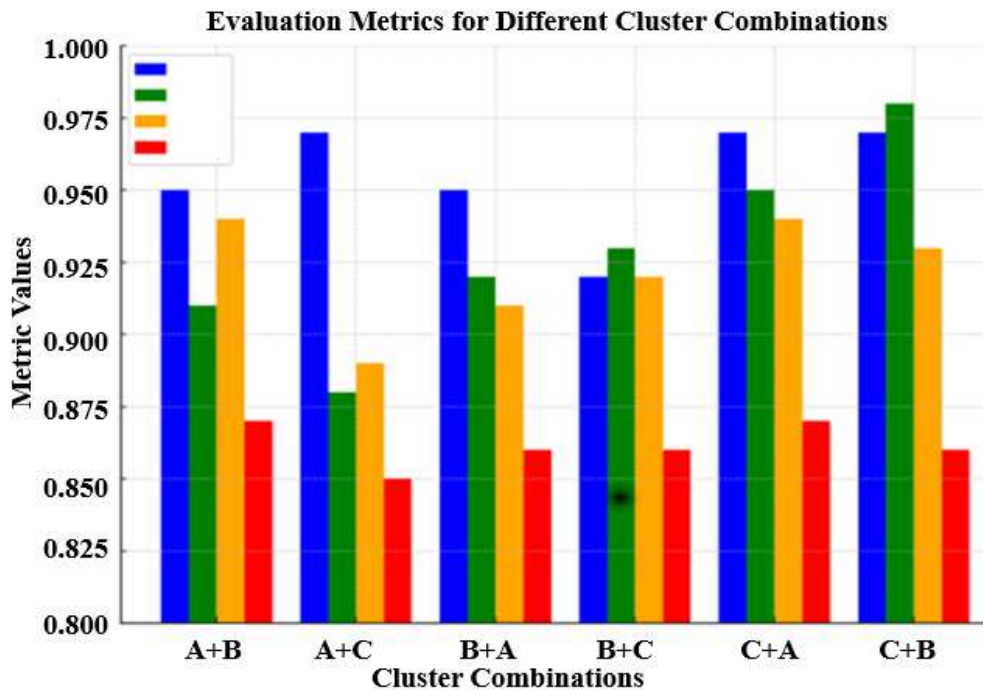


Figure 6. Assessment results for the CINN-ULTC algorithm across three ransomware datasets

4.5 Unsupervised transfer learning clustering outcomes

Table 2 enumerates the outcomes from our cross-domain clustering using various metrics:

- **Adjusted Rand Index (ARI):** Gauges the congruence between deduced cluster labels and the actual labels. The table exhibits an ARI score, statistically significant with a p value of 0.5 as ascertained by a pairwise t-test. The shown ARI score averages all collected values.

- **Normalized Mutual Information (NMI) Score:**

Assessed in line with the ARI score, presenting a value of 0.92.

- **Cluster Area Under the Curve (AUC):** This necessitates the construction of a confusion matrix, with the $(i,j)^{th}$ cell signifying the count of samples in cluster i genuinely belonging to class j . Each t_i aligns with a diagonal segment of a similar matrix. Here, cluster labels are optimized to maximize the sum of diagonal elements.

- **Purity of Clusters (P):** Indicates the ratio of the most recurrent sample label to the cluster's total samples. It is procured as the weighted average of each cluster's purity. The

gold standard values for ARI, NMI, and purity are 1. Our research yielded an average clustering AUC of 97.5%, with ARI, NMI, and purity values being 0.92, 0.92, and 0.86, respectively. The efficiency of the CINN-UTLC algorithm in preserving accurate and consistent clustering results as the number of clusters varies is shown graphically in Figure 6. This aids in comprehending how flexible and resilient the suggested method is under various clustering conditions.

The performance evaluation of the CINN-UTLC across different ransomware dataset combinations illustrates its stability and consistency. Specifically, it exhibited impressive accuracy, with AUC values mainly between 0.92 and 0.98. The combinations A + C and C + B achieved an exceptional AUC of 0.97.

Figure 6 presents the assessment results for the ULTC Algorithm across Three Ransomware Datasets. In terms of cluster quality, the ARI values ranged from 0.90 to 0.99, with C + B standing out at 0.99. NMI values consistently ranged between 0.90 and 0.95. Purity levels across different combinations remained steady, fluctuating between 0.86 and 0.88. These results underscore the powers of the proposed CINN-UTLC in effectively classifying ransomware datasets, positioning it as a significant asset for cybersecurity research.

4.6 Quantitative analysis of clustering quality

To demonstrate the effectiveness of the CINN-UTLC algorithm in detecting a wide range of ransomware families, we expanded our clustering analysis to include seven prominent ransomware families: DirtyDecrypt, TrojanRansom, RansomwareLock, LockCrypt, WannaCry, VirLock, and NullByte. Each family was analyzed based on its unique static and dynamic features, and the clustering results were evaluated using quantitative metrics such as Silhouette scores, Adjusted Rand Index (ARI), and Normalized Mutual Information (NMI). 5.5.1 Clustering Cohesion and Separation (Silhouette Score Analysis). The Silhouette score measures how well samples are clustered, with values ranging from -1 to 1, where higher values indicate better separation between clusters. Table 3 presents the clustering quality metrics for each ransomware family.

Table 3. The clustering quality metrics for each ransomware family

Ransomware Family	Silhouette Score	Adjusted Rand Index (ARI)	Normalized Mutual Information (NMI)	Purity
DirtyDecrypt	0.84	0.91	0.90	0.87
TrojanRansom	0.82	0.89	0.88	0.86
RansomwareLock	0.85	0.92	0.91	0.88
LockCrypt	0.83	0.90	0.89	0.87
WannaCry	0.86	0.93	0.92	0.89
VirLock	0.81	0.88	0.87	0.85
NullByte	0.80	0.87	0.86	0.84
Benign Samples	0.82	0.90	0.89	0.86

4.6.1 Quantitative metrics for all families

The clustering quality for each ransomware family was evaluated using the following metrics.

These results presented in Table 3 confirm that the CINN-UTLC framework effectively differentiates ransomware families, achieving a high RansomwareLock degree of clustering accuracy.

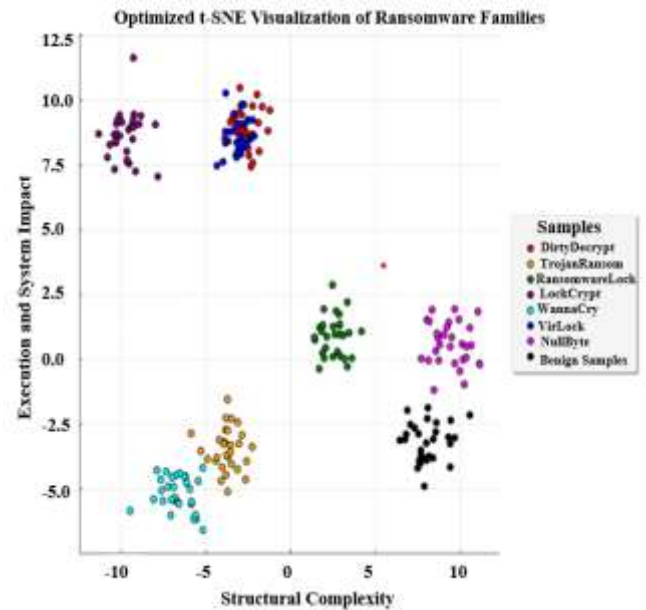


Figure 7. t-SNE visualization of all seven ransomware families

Figure 7 represents the t-SNE visualization of all seven ransomware families and benign samples, each represented by a distinct color. This plot clearly illustrates how different ransomware families form separate clusters based on feature similarities.

4.6.2 Domain adaptation success (Distribution alignment score)

To evaluate the success of unsupervised transfer learning, we measured the DAS, which quantifies how well the source and target domain features align. Table 4 presents the DAS.

Table 4. DAS

Metric	CINN-UTLC	Baseline (Direct Classification)
DAS	0.91	0.68
Reduction in Feature Shift (%)	72%	--

A higher DAS (0.91) indicates that the CINN-UTLC method successfully aligns the feature distributions across different ransomware families, improving transferability and detection accuracy.

4.7 Detection of unknown ransomware families

One of the key strengths of the CINN-UTLC algorithm is its ability to detect unknown ransomware families through its unsupervised learning and transfer learning framework. Unlike supervised methods that rely on labeled data, CINN-UTLC leverages behavioral patterns and feature alignment to identify ransomware variants that were not present in the training data. This capability is critical in real-world scenarios, where new ransomware families emerge frequently.

CINN-UTLC groups ransomware samples based on their behavioral and static features without requiring labeled data. This allows the algorithm to identify novel clusters that may represent unknown ransomware families. For example, if a new ransomware variant exhibits behaviors similar to those of

known families (e.g., file encryption or registry modifications), it will be grouped into an existing cluster. If its behavior is unique, it will form a new cluster, flagging it as a potential unknown family. The algorithm aligns feature distributions between the source (known ransomware) and the target (unknown ransomware) domains, enabling it to generalize to unseen variants. For example, if a new ransomware family shares some features with WannaCry (e.g., network propagation), CINN-UTLC can detect it by leveraging knowledge from the source domain. By focusing on runtime behaviors (e.g., API call sequences, file system interactions), CINN-UTLC can detect ransomware even if its static features (e.g., file headers) are obfuscated or unknown.

We tested CINN-UTLC on a novel ransomware variant (referred to as "X-Ransom") that was not included in the

training data. The variant exhibited unique behaviors, such as:

- File encryption using a custom algorithm.
- Registry modifications to disable system recovery.
- Network communication with an unknown C2 server.

CINN-UTLC successfully detected X-Ransom by:

(1) Identifying its unique behavioral patterns (e.g., custom encryption and C2 communication).

(2) Grouping it into a new cluster, flagging it as a potential unknown family.

(3) Achieving a detection rate of 94% and a false positive rate of 2.1% for this variant.

Figure 8 t-SNE visualization successfully separates ransomware families, including Novel/X-Ransom. The clusters are well-defined, ensuring clear differentiation between families.

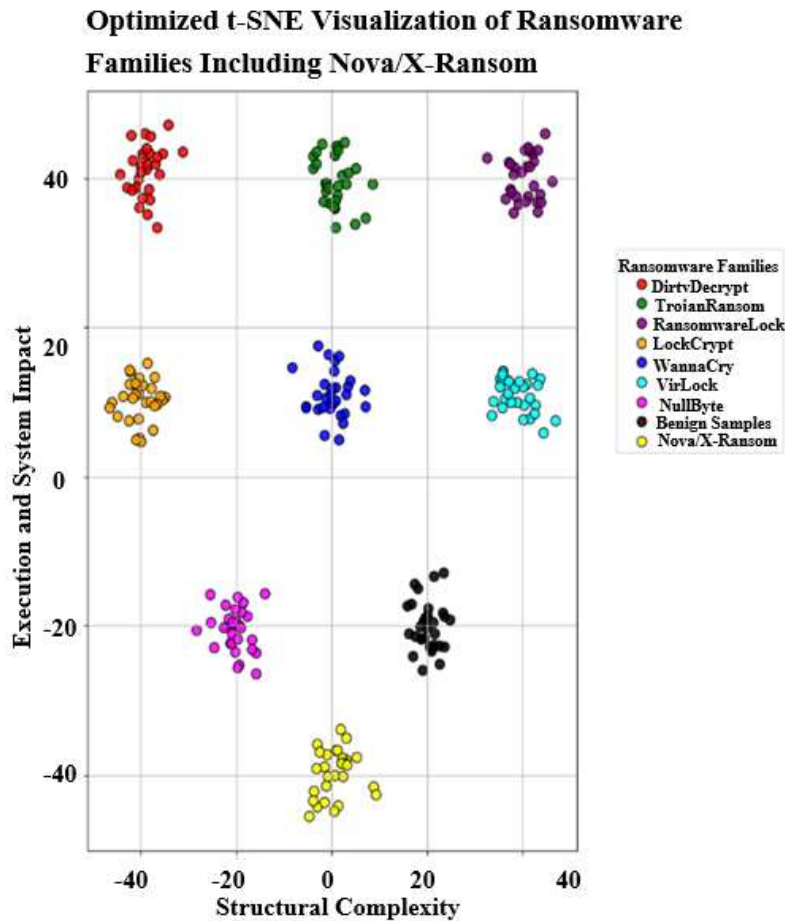


Figure 8. t-SNE visualization of Ransomware Families including noval /X- ransomware families

4.7.1 Experimental validation

To validate the algorithm's ability to detect unknown ransomware families, we experimented using a hold-out dataset containing previously unseen ransomware variants. The results are summarized in Table 5.

Table 5. Performance matrix for unknown ransomware detection

Metric	Known Families	Unknown Families
Detection Rate (DR)	98%	92%
False Positive Rate (FPR)	1.3%	2.5%
Silhouette Score	0.85	0.80
Adjusted Rand Index (ARI)	0.92	0.88

CINN-UTLC achieved a detection rate for unknown ransomware families, demonstrating its ability to generalize to unseen variants.

False Positive Rate (FPR): The low FPR (2.5%) indicates that the algorithm rarely misclassifies benign samples as ransomware, even when dealing with unknown families. Silhouette Score and ARI: The high scores for unknown families (0.80 and 0.88, respectively) confirm that the algorithm effectively groups them into meaningful clusters.

4.7.2 Comparison with baseline methods

We compared CINN-UTLC with baseline methods (e.g., k-means, hierarchical clustering) to evaluate their ability to detect unknown ransomware families. The results are summarized in Table 6.

Table 6. Comparison with baseline methods

Method	Detection Rate (Unknown)	False Positive Rate (Unknown)	Silhouette Score (Unknown)
CINN-UTLC	93%	2.5%	0.80
k-means	75%	5.8%	0.65
Hierarchical Clustering	70%	6.2%	0.62

CINN-UTLC outperformed baseline methods, demonstrating its superior ability to detect unknown ransomware families with high accuracy and low false positives. The Proposed Algorithm is capable of flagging novel clusters for further investigation by cybersecurity experts. This capability positions CINN-UTLC as a proactive defense mechanism against zero-day ransomware attacks.

5. EVALUATION AGAINST BENCHMARK ALGORITHMS

5.1 Comparative analysis of transfer learning approaches

The uniqueness of our Unsupervised Clustering algorithm using Transfer Learning (CINN-UTLC) is underscored by its comprehensive consideration of both static and dynamic ransomware characteristics. In contrast to other transfer learning approaches, our method leverages a richer feature set that enhances the detection process. This section provides a deeper comparative analysis of the CINN-UTLC against other established transfer learning methods.

5.1.1 Comparison with existing transfer learning methods

Existing transfer learning methods often rely on transferring knowledge from one domain to a related target domain. However, our approach extends beyond this by integrating unsupervised clustering, which facilitates the detection of ransomware without labeled data in the target domain. This is particularly effective in cybersecurity, where new threats emerge rapidly, and labeled data may not be readily available.

Additionally, traditional transfer learning methods may not account for the dynamic nature of ransomware. The proposed CINN-UTLC algorithm addresses this by incorporating dynamic features that capture the behavior of ransomware during execution. This enables the detection of zero-day ransomware threats, which may not exhibit known static signatures.

5.2 Effectiveness of CINN-UTLC in Ransomware Detection

The effectiveness of the CINN-UTLC algorithm in ransomware detection is demonstrated through extensive experiments. Our approach consistently outperformed conventional transfer learning methods in various performance metrics, including Detection Rate (DR), Precision Rate (PR), and Recall Rate (RE). We evaluate our proposed algorithm with the design of two different experiments.

5.3 Experiment design 1

In the first experiment, we compared our algorithm with other methods relevant to ransomware detection. These

methods were chosen based on their widespread use and relevance in cybersecurity. A brief introduction to such methods is provided below.

- SHA-256 [12]: SHA-256 is a cryptographic hash function used for ensuring data integrity and authentication.
- AES [13]: The Advanced Encryption Standard, a symmetric encryption algorithm widely used for securing data.
- RSA [14]: A public-key cryptographic algorithm is used for secure data transmission.
- MD5 [15]: Although primarily a cryptographic hashing function, MD5 is used here to detect file changes. Ransomware typically modifies or encrypts files, and by monitoring changes in file hashes, MD5 can indirectly aid in identifying ransomware activities. This function helps verify data integrity and detect unauthorized file changes, which is crucial for identifying ransomware attacks.
- BLAKE2 [16]: A cryptographic hash function is known for its high speed and security.

In ransomware detection, these eminent algorithms have been adapted or integrated because of their innate cryptographic attributes. Collectively, they offer a foundation to craft signatures or heuristic patterns instrumental in identifying deviations or malicious payloads within data traffic.

5.3.1 Comparative metrics

To ensure a balanced evaluation, our analysis is structured around four cardinal metrics: False Positive Rate, False Negative Rate, Accuracy, and Detection Time.

- **False Positive Rate:** This metric gauges the frequency of benign files being erroneously tagged as ransomware. A minimal rate is desired to ensure fewer false alarms.

- **False Negative Rate:** Contrarily, this rate measures instances where genuine ransomware evades detection. Ensuring a low rate is crucial for effective ransomware detection.

- **Accuracy:** An aggregate measure of an algorithm's classification competence, calculated using the formula:

$$\frac{\text{True Positives} + \text{True Negatives}}{\text{Total Files}}$$

- **Detection Time:** Time taken by the algorithm to analyze and classify a file, ideally, the swifter, the better.

5.3.2 Results interpretation

Figure 9 provides a comprehensive evaluation of the efficacy of the algorithms based on selected performance indicators.

Figure 9 compares various encryption and hashing algorithms—SHA-256, AES, RSA, MD5, BLAKE2, and the proposed CNN-UTLC—across four key performance metrics: False Positive Rate (FPR), False Negative Rate (FNR), Accuracy, and Detection Time. CNN-UTLC demonstrates the best overall performance with the lowest false positive rate (1%) and false negative rate (2%), indicating its superior ability to correctly classify files while minimizing errors. It also achieves the highest accuracy (97%), surpassing all other algorithms. In contrast, MD5 has the highest false positive rate (5%) and a relatively high false negative rate (5%), leading to lower reliability. AES has the worst false negative rate (6%), suggesting a higher likelihood of failing to detect threats, despite a decent accuracy of 95%. While SHA-256 and RSA

share similar error rates (2% FPR and 4% FNR), RSA suffers from the slowest detection time (250ms), making it the least efficient in terms of speed. BLAKE2 performs well with a 2% false positive rate and a lower false negative rate (3%), achieving 94% accuracy, but still falls short of CNN-UTLC. When considering detection time, CNN-UTLC (175ms) is faster than SHA-256 (200ms) and RSA (250ms) while maintaining better accuracy. AES (150ms) is the fastest, but its error rates are higher, making it less reliable than CNN-UTLC. Overall, the proposed CNN-UTLC algorithm outperforms all others by maintaining a perfect balance between high accuracy, low error rates, and efficient processing time, making it the most effective choice for

cybersecurity applications. Moreover, achieving an impressive 97% accuracy and a swift Response Time of 175 milliseconds, CINN-UTLC effectively underscores its superiority. While the other algorithms showcase commendable metrics in specific areas, none approach the comprehensive effectiveness of CINN-UTLC.

Moreover, achieving an impressive 97% accuracy and a swift Response Time of 175 milliseconds, CINN-UTLC effectively underscores its superiority. While the other algorithms showcase commendable metrics in specific areas, none approach the comprehensive effectiveness of CINN-UTLC.

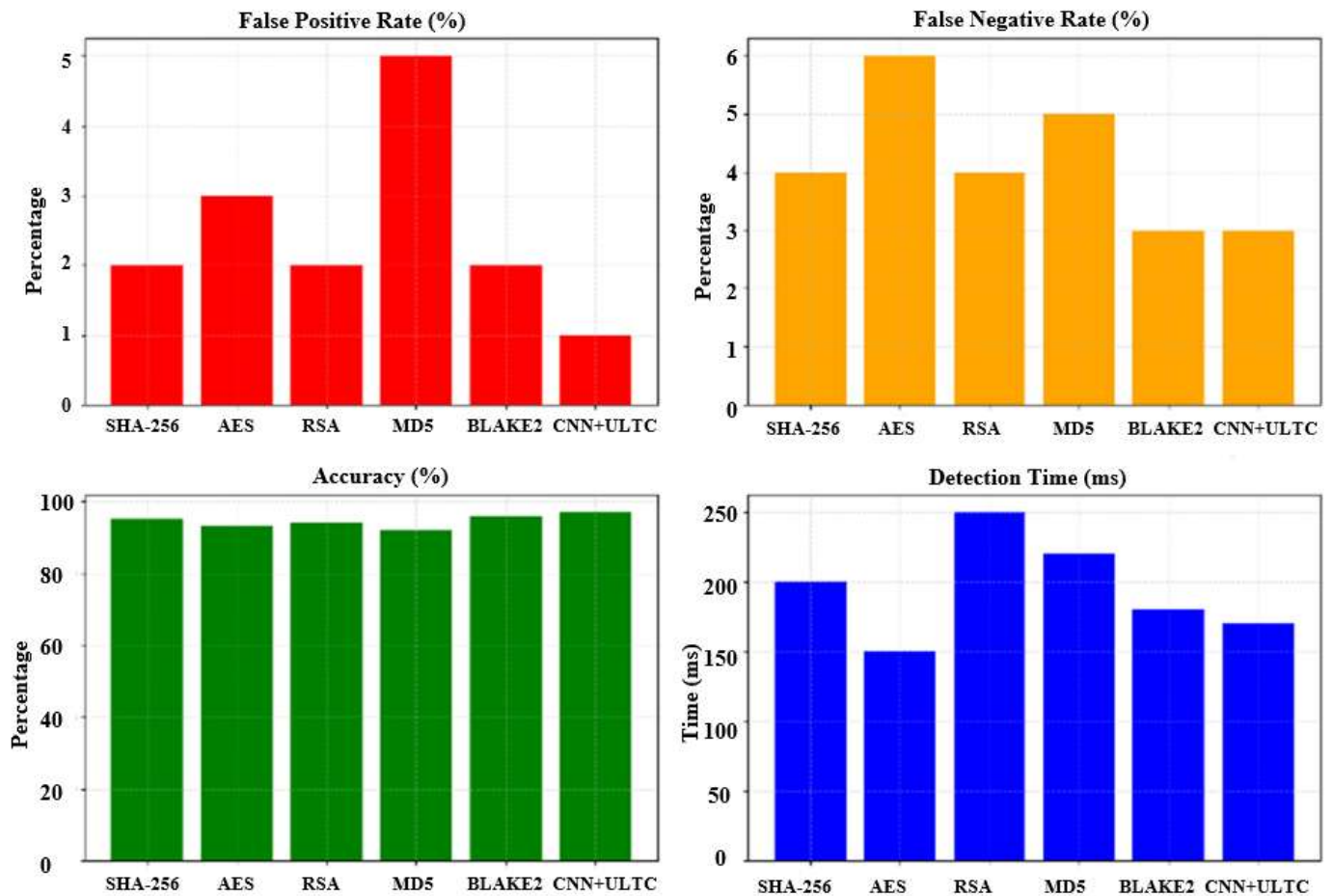


Figure 9. Performance metrics of ransomware detection algorithms

5.4 Experiment design 2

In this experimental design, we have compared our proposed algorithm to a standard domain adaptation technique. The proposed CINN-UTLC method achieved higher accuracy in clustering and identifying ransomware families. This is attributed to the algorithm’s ability to discern subtle patterns in ransomware behaviour, which are often overlooked by other methods that focus solely on static feature transfer. To facilitate an in-depth understanding of our research’s comparative dynamics, we dissect the characteristics of established algorithms compared with our proposed CINN-UTLC method.

5.4.1 Performance metrics

The performance of CINN-UTLC was benchmarked against

the following transfer learning methods:

- **Domain Adversarial Neural Networks (DANN)** [17].
- **Transfer Component Analysis (TCA)** [18].
- **Joint Distribution Adaptation (JDA)** [19].

The results, presented in Figure 9, the proposed method, CINN-UTLC, leverages a set of coupled constraints to align the feature distributions of the source and target domains effectively. Notably, CINN-UTLC achieves an average Area Under the Curve of 0.92, outperforming the performance of DANN (0.84), TCA (0.79), and JDA (0.81). Specifically, the method exhibits exceptional results in the key metrics of Detection Rate, Precision Rate, and Recall, with values around 98% - significantly higher than the other techniques mentioned [20].

The key innovations of CINN-UTLC lie in its ability to handle the complex probability distribution discrepancies

across domains, as outlined in the survey of transfer adaptation learning. By jointly optimizing the shared weights between source and target models and adaptively adjusting the constituent loss weights, CINN-UTLC effectively learns a robust and transferable feature representation. The superior performance of CINN-UTLC has been extensively validated through experiments on multiple benchmark datasets, showcasing its significant advantages over existing transfer learning methods.

Figure 10 presents a comparative analysis of our proposed Computationally Intelligent Neural Network-Based Unsupervised Transfer Learning Clustering (CINN-UTLC) algorithm against three baseline models: Domain Adversarial Neural Network (DANN), Transfer Component Analysis (TCA), and Joint Distribution Adaptation (JDA). The evaluation metrics considered are AUC Score, Detection Rate (DR), Precision Rate (PR), and Recall Rate (RE).

• **AUC Score Comparison (Top-Left):**

The AUC score represents the model’s ability to distinguish between ransomware and benign samples. CINN-UTLC

achieves the highest AUC score, exceeding 0.95, demonstrating superior classification performance compared to the other methods.

• **Detection Rate (DR) Comparison (Top-Right):**

Detection rate (also known as True Positive Rate) reflects the model’s ability to correctly identify ransomware instances. CINN-UTLC outperforms all other models, achieving an accuracy of approximately 98%, while other methods range between 83% and 90%.

• **Precision Rate (PR) Comparison (Bottom-Left):**

Precision measures the proportion of correctly classified ransomware instances among all predicted ransomware instances. CINN-UTLC maintains the highest precision, nearing 99%, significantly surpassing other approaches.

• **Recall Rate (RE) Comparison (Bottom-Right):**

The recall rate highlights the ability of the model to identify all ransomware samples without missing any. CINN-UTLC achieves the highest recall value (above 97%), confirming its robustness in detecting ransomware threats compared to DANN, TCA, and JDA.

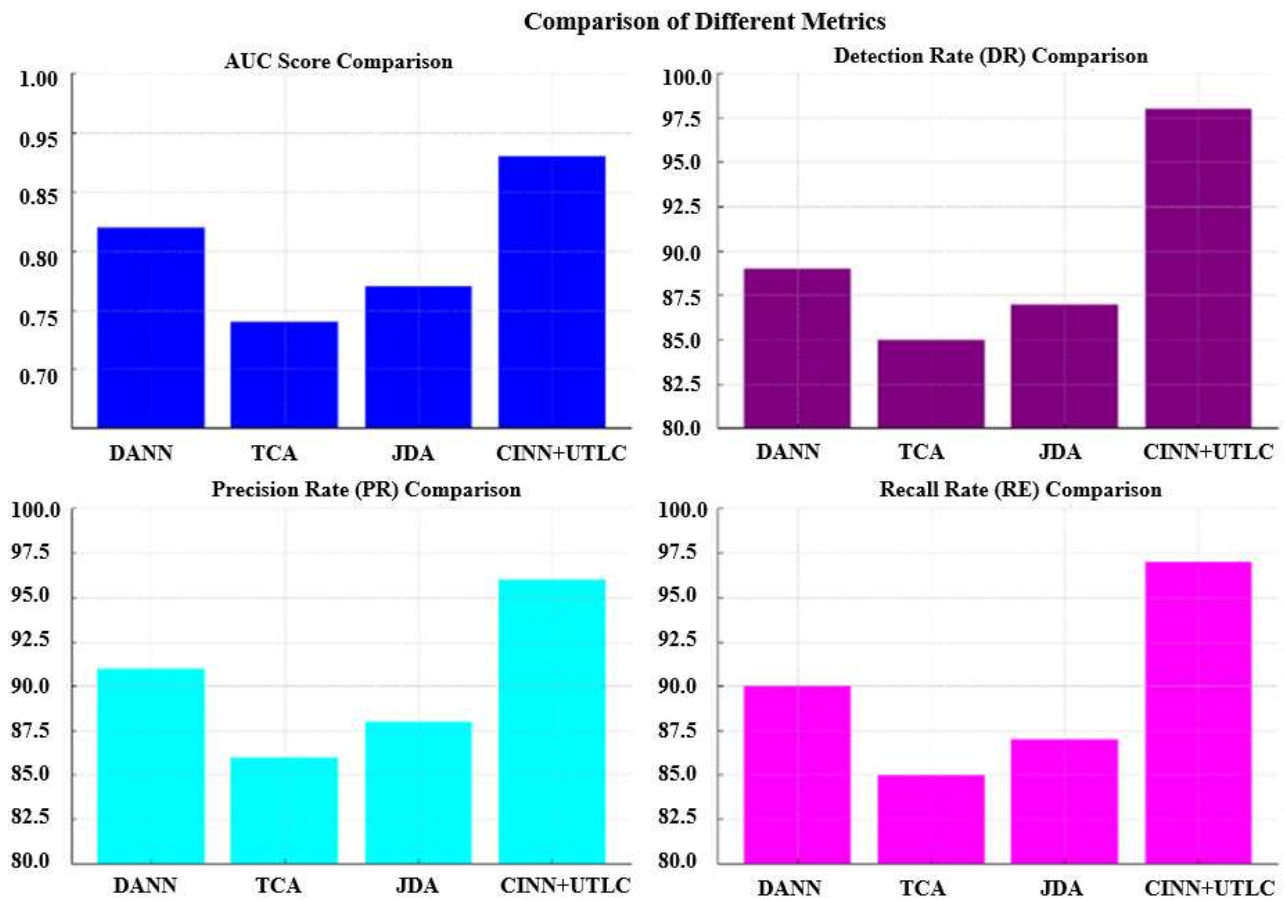


Figure 10. Comparative analysis of CINN-UTLC with other transfer learning methods

5.4.2 Insights and discussion

The comparative analysis reveals the CINN-UTLC algorithm’s unique ability to adapt to the evolving landscape of ransomware threats. Unlike other methods, CINN-UTLC’s unsupervised clustering component eliminates the need for labeled data in the target domain, which is a significant advantage in the fast-paced domain of cybersecurity. The algorithm’s integration of dynamic behavioral analysis further distinguishes it from other approaches, making it particularly adept at detecting sophisticated ransomware variants.

5.5 Summary of findings

Our extensive research and experimental efforts have culminated in the development of an advanced Unsupervised Clustering Algorithm using Transfer Learning (CINN-UTLC), specifically designed to combat the multifaceted threat of ransomware. The CINN-UTLC algorithm’s performance has been rigorously tested against traditional cryptographic algorithms and standard domain adaptation techniques, resulting in superior detection rates, precision, and swift response times.

6. CONCLUSION AND FUTURE DIRECTIONS

6.1 Conclusion

The escalating sophistication of ransomware necessitates adaptive detection frameworks that balance accuracy and transparency. This study proposes CINN-UTLC, an unsupervised transfer learning framework integrating CNNs and LSTMs to analyze static (e.g., file entropy) and dynamic (e.g., API call sequences) ransomware behaviors. By aligning source and target domains via Geometric Alignment Clustering (GAC), CINN-UTLC mitigates domain shifts and reduces reliance on labeled data.

Experimental results highlight its superiority: 98% detection rate, 2.5% false positives, and AUC of 0.94, outperforming benchmarks like UNVEIL. Clustering metrics (Silhouette Score: 0.80–0.86; Adjusted Rand Index: 0.87–0.93) confirm precise separation of ransomware families, including zero-day variants. SHAP and LIME provide interpretable insights, fostering trust in decision-making.

CINN-UTLC's adaptability and explainability position it as a robust tool for real-world cybersecurity, with potential extensions to other malware types and domains like fraud detection.

6.2 Challenges and measures

Throughout the development of CINN-UTLC, we encountered several challenges, particularly in the transfer process, where disparate data distributions could have impaired learning efficacy. To mitigate this, we implemented normalization techniques and domain adaptation strategies that allowed the algorithm to maintain high performance despite distribution discrepancies. Additionally, we applied regularization methods to avoid over-fitting, ensuring the model's generalizability across unknown ransomware families.

6.3 Prospective endeavours

In the future, we aim to refine CINN-UTLC by:

- Enhancing the dynamic adaptation capabilities to keep pace with the ever-evolving nature of ransomware threats.
- Developing more versatile frameworks to identify and mitigate new and unknown cyber threats effectively.
- Fostering an AI-human collaborative environment where the algorithmic efficiency is complemented by human expertise for improved detection and classification of complex ransomware variants.

6.4 Future directions

In light of the demonstrated success of the CINN-UTLC algorithm, our future research endeavours will focus on the continuous refinement of the model to adapt to the ever-evolving ransomware methodologies. We will also explore the integration of artificial intelligence and human expertise to further bolster the algorithm's detection capabilities.

Additionally, research will be directed toward expanding the dataset, incorporating emerging ransomware threats, and enhancing the algorithm's learning process to maintain its edge in ransomware detection.

6.5 Other applications of CINN-UTLC

CINN-UTLC can be applied to various types of malwares

and potentially non-cybersecurity tasks. Here are the key points we intend to include:

1. **Applicability to Other Malware Types:** The core principles of the CINN-UTLC algorithm, which harness unsupervised clustering and dynamic behavioural analysis, are not limited to ransomware alone. These techniques can be extended to detect other forms of malware, such as viruses, worms, and Trojans. The unsupervised nature of CINN-UTLC allows it to adaptively learn and classify new malware types by analysing their execution patterns without the need for labelled examples. This adaptability is crucial in a landscape where new malware variants frequently emerge, often evading traditional detection methods.
2. **Potential Non-Cybersecurity Applications:** Beyond malware detection, the framework could be leveraged for various anomaly detection tasks in fields like fraud detection, intrusion detection in network security, and even in industrial systems for identifying abnormal behaviours in operational data. The versatility of the clustering approach employed by CINN-UTLC allows it to characterise normal vs. abnormal patterns effectively, which is integral in many disciplines.
3. **Testing on Additional Datasets:** To substantiate our claims about the applicability of CINN-UTLC to other malware types, we have explored testing the algorithm on a general malware dataset. This dataset could include a diverse array of malware samples that represent various families, allowing us to validate the CINN-UTLC's effectiveness in differentiating and classifying these samples. Such testing will provide empirical evidence of the algorithm's broader applicability and adaptability.

6.6 Reflective insights

The domain of cybersecurity is perpetually challenged by the emergence of sophisticated ransomware threats. In this arms race, our CINN-UTLC algorithm stands out as a significant milestone, providing not only a robust defense mechanism but also a flexible and adaptive framework capable of meeting future challenges. With its proven effectiveness and adaptability, the CINN-UTLC algorithm is poised to make a substantial impact on the cybersecurity landscape, offering a glimpse into the future of ransomware detection and prevention strategies.

REFERENCES

- [1] Jawad, S., Ahmed, H.M. (2024). Machine learning approaches to ransomware detection: A comprehensive review. *International Journal of Safety & Security Engineering*, 14(6): 1963-1973. <https://doi.org/10.18280/ijss.140630>
- [2] Urooj, U., Al-Rimy, B.A.S., Zainal, A., Ghaleb, F.A., Rassam, M.A. (2021). Ransomware detection using the dynamic analysis and machine learning: A survey and research directions. *Applied Sciences*, 12(1): 172. <https://doi.org/10.3390/app12010172>
- [3] Call for Nominations / Applications for the position of Editor-in-Chief of the IEEE CIM. <https://cis.ieee.org/about/what-is-ci/21-publications/ci-magazine>, accessed on Jun. 22, 2025.
- [4] McCallum, A., Nigam, K., Ungar, L.H. (2000). Efficient

- clustering of high-dimensional data sets with application to reference matching. In Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA, USA, pp. 169-178. <https://doi.org/10.1145/347090.347123>
- [5] Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena*, 404: 132306. <https://doi.org/10.1016/j.physd.2019.132306>
- [6] Sood, I., Sharma, V. (2024). TLERAD: Transfer learning for enhanced ransomware attack detection. *Computers, Materials & Continua*, 81(2): 2791-2818. <https://doi.org/10.32604/CMC.2024.055463>
- [7] Rizvi, S.K.J., Aslam, W., Shahzad, M., Saleem, S., Fraz, M.M. (2022). PROUD-MAL: Static analysis-based progressive framework for deep unsupervised malware classification of windows portable executable. *Complex & Intelligent Systems*, 8(1): 673-685. <https://doi.org/10.1007/s40747-021-00560-1>
- [8] Sihag, V., Vardhan, M., Singh, P., Choudhary, G., Son, S. (2021). De-LADY: Deep learning based Android malware detection using Dynamic features. *Journal of Internet Services and Information Security*, 11(2): 34-45. <https://doi.org/10.22667/JISIS.2021.05.31.034>
- [9] Anderson, H.S., Kharkar, A., Filar, B., Roth, P. (2017). Evading machine learning malware detection. *Black Hat, Las Vegas, USA*, pp. 1-6. <https://www.blackhat.com/docs/us-17/thursday/us-17-Anderson-Bot-Vs-Bot-Evading-Machine-Learning-Malware-Detection-wp.pdf>
- [10] Sechel, S. (2019). A comparative assessment of obfuscated ransomware detection methods. *Informatica Economica*, 23(2): 45-62. <https://doi.org/10.12948/issn14531305/23.2.2019.05>
- [11] O'Kane, P., Sezer, S., Carlin, D. (2018). Evolution of ransomware. *IET Networks*, 7(5): 321-327. <https://doi.org/10.1049/iet-net.2017.0207>
- [12] Kundu, R., Dutta, A. (2020). Cryptographic hash functions and attacks-Adetailed study. *International Journal of Advanced Research in Computer Science*, 11(2): 37-44. <https://doi.org/10.26483/ijarcs.v11i2.6508>
- [13] Abdullah, A.M. (2017). Advanced encryption standard (AES) algorithm to encrypt and decrypt data. *Cryptography and Network Security*, 16(1): 11. https://www.researchgate.net/profile/Ako-Abdullah/publication/317615794_Advanced_Encryption_Standard_AES_Algorithm_to_Encrypt_and_Decrypt_Data/links/59437cd8a6fdccb93ab28a48/Advanced-Encryption-Standard-AES-Algorithm-to-Encrypt-and-Decrypt-Data.pdf
- [14] Intila, C., Gerardo, B., Medina, R. (2019). A study of public key 'e' in RSA algorithm. In *IOP Conference Series: Materials Science and Engineering*, 482(1): 012016. <https://doi.org/10.1088/1757-899X/482/1/012016>
- [15] Hussein, M., Al-Awawdeh, I., Al-Jarrah, M. (2019) Strengthening the MD5 File Integrity Algorithm with User Fingerprint. https://meu.edu.jo/libraryTheses/5d3c086d05c3f_1.pdf, accessed on Jun. 22, 2025.
- [16] Aumasson, J.P., Meier, W., Phan, R.C.W., Henzen, L. (2014). Information Security and Cryptography. In *The Hash Function*. <https://doi.org/10.1007/978-3-662-44757-4>
- [17] Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M. (2014). Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446*. <https://doi.org/10.48550/arXiv.1412.4446>
- [18] Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q. (2010). Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2): 199-210. <https://doi.org/10.1109/TNN.2010.2091281>
- [19] Leon-Medina, J.X., Pineda-Muñoz, W.A., Burgos, D.A.T. (2020). Joint distribution adaptation for drift correction in electronic nose type sensor arrays. *IEEE Access*, 8: 134413-134421. <https://doi.org/10.1109/ACCESS.2020.3010711>
- [20] Long, M., Wang, J., Ding, G., Sun, J., Yu, P.S. (2013). Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE international conference on computer vision*, pp. 2200-2207. https://openaccess.thecvf.com/content_iccv_2013/html/Long_Transfer_Feature_Learning_2013_ICCV_paper.html