Journal of Information Systems Engineering and Management
2025, 10(43s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

RS-FEDRAD: Robust and Scalable Federated Ransomware Detection Using TTP-Enhanced Dataset

Chinonso E. Ali [1], Songfeng Lu [2*], Francis A. Ruambo [3], Francelle Tchamini [4]
[1, 2, 3, 4] *Hubei Engineering Research Center on Big Data Security, School of Cyber Science & Engineering, Huazhong University of Science and Technology, Wuhan 430074*
[1, 2] *Shenzhen Huazhong University of Science and Technology Research Institute, Shenzhen, China*
[3] *Department of Information Systems and Technology, College of Information and Communication, Technology, Mbeya University of Science and Technology, Mbeya 131, Tanzania*
**Corresponding Author Email: lusongfeng@hust.edu.cn*

ARTICLE INFO

Received: 15 Dec 2024

Revised: 19 Feb 2025

Accepted: 26 Feb 2025

ABSTRACT

Ransomware continues to pose a significant challenge to the cyberspace industry, with rising frequency and complexity threatening data integrity, availability, and confidentiality. Current detection methods often fail to effectively address modern ransomware due to inadequate feature sets and over reliance on centralized architectures, posing privacy and scalability challenges. We present RS-FEDRAD, a robust and scalable federated learning (FL)-based ransomware detection system that combines FL with deep dynamic analysis, using a novel Tactics, Techniques, and Procedures (TTP) enhanced dataset to overcome these limitations. This approach first captures critical ransom-ware behavioral attributes such as application programming interface (API) calls, dynamic link library (DLL) usage, and mutual exclusion (Mutex) operations, before mapping them to their corresponding ransomware-related TTPs using the MITRE ATT@CK framework. Extensive experimental evaluations highlight the effectiveness of the framework against unknown black-box and known white-box attacks, utilizing a hybrid convolutional neural network (CNN) and long short-term memory (LSTM) to achieve an impressive accuracy of 99.90% and an average federated accuracy of 99.50%. RS-FEDRAD offers a scalable, privacy-preserving solution that enhances ransomware detection and understanding of attacker strategies through its TTP-enhanced feature set., advancing ransomware mitigation with adaptive, decentralized, and robust security for today's rapidly evolving threat landscape.

Keywords: Deep learning, Dynamic analysis, Federated learning, MITRE ATT@CK framework, Ransomware detection, TTP enhanced dataset.

INTRODUCTION

Throughout recent history and presently, global communities have encountered numerous adversaries in the form of malicious software, commonly called ransomware. This program poses an enormous threat to organizations of all sizes, primarily by encrypting relevant data and making it out of reach to the owners until payment is made for the decryption key; the payment method usually uses cryptocurrency. The authors in [1] note that since the discovery of the first ransomware, Trojan Horse AIDS, in 1989, as noted in numerous high profile ransomware incidents have led to substantial financial extortion, amounting to billions of dollars [2], [3]. Similarly, the authors in [4] reported that in 2021, the US-based meat production company (JBS) paid $11 million in ransom following an attack that disrupted its operations.

Malware detection methodologies are generally grouped into static detection (signature-based) and dynamic detection(behavioural-based). The method of static detection entails examining a program's content without its execution[5], [6]. A dynamic detection technique is introduced to improve the shortcomings of the static detection method. In contrast to static approaches, dynamic detection evaluates a program's running-time behaviour after being executed in a hosted sandbox environment. Throughout this procedure, the system records memory utilization

Journal of Information Systems Engineering and Management
2025, 10(43s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

DLL, Mutex, API calls [7], [8]. However, recent studies [9], [10] suggest that depending exclusively on singular features may be inadequate for effective malware detection. This criterion has prompted scholars to investigate more extensive methodologies that can integrate multiple ransomware behavioral features at once.

More so, many researchers [11], [12], [13] emphasize that importance of acquiring and sharing Cyber Threat Intelligence (CTI) because its collaborative application for threat modelling is crucial for staying ahead of cyber attackers' constantly evolving tactics and techniques. The collaboration is fundamental to enhancing the overall resilience of the cybersecurity landscape. MITRE ATT&CK Framework [14] is generally known to be an efficient cybersecurity resource tool for accessing, sharing, analyzing, and deploying cybersecurity related models. For example, consider using the Windows API called "RegSetValueEx," which is commonly employed to modify registry settings. If a program calls this API to creatregistry,HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run\MLicious.exe, this action alone, whether detected by static or dynamic analysis, might not provide enough information to determine whether it is malicious definitively. Traditional detection systems may flag this as suspicious without a deeper understanding of context or historical adversarial behaviours; however, when mapped against the MITRE ATT&CK Framework, it can reveal a more complete picture by reviewing the strategy applied [15].

Notwithstanding the usefulness of these context-aware features, their utilization for ransomware detection has not been fully explored due to privacy related issues. For instance, enterprise managers may gather critical data, including API requests, DLL interaction, Mutex operations, and process injections but they face restrictions due to data protection policies thereby limiting their ability to share for effective and improved detection. In this regard, federated learning (FL) offers a compelling solution as it tackles the principal difficulties of centralized ransomware detection by ensuring scalability, privacy, and security by facilitating organizational collaboration in developing detection models while safeguarding sensitive data. FL encompasses both universal and context-specific ransomware attributes, facilitating the identification of established and novel threats. Distributing training across numerous nodes mitigates the chance of a single point of failure, hence augmenting overall security.

This study introduces RS-FEDRAD, a FL-based ransomware detection system that combines dynamic analysis with a TTP-enhanced dataset for a decentralized, privacy-preserving solution. Using a hybrid CNN and LSTM model, it first captures key ransomware behaviors like API calls, DLL usage, and Mutex operations before mapping them to their corresponding TTPs in the MITRE ATT&CK framework for comprehensive behavioral insights. The key contributions of this paper are as follows:

- The paper proposes RS-FEDRAD, a robust and scalable FL-based ransomware detection framework that significantly combines multiple facets of executable sequences to enrich the feature space.
- Introduces a novel TTP-enhanced ransomware dataset that integrates critical industry-based features, including API calls, DLL interactions, and Mutexes, with their corresponding TTPs based on the MITRE ATT@CK framework. By curating concatenated sequences that capture the ordered relationships between these features and their associated TTP attributes, RS-FEDRAD constructs a joint feature space that offers a more comprehensive overview of ransomware behavior.
- Evaluates the proposed RS-FEDRAD performance by conducting extensive experiments involving four distinct ransomware datasets. Comprehensive evaluation shows RS-FEDRAD's superior performance against benchmark models across all known measures of evaluation while maintaining decentralized, robust defense against unknown type black-box attacks as well as known type white-box attacks.

RELATED WORK

This section reviews contributing advancements in ransomware/malware detection techniques, focusing on static, dynamic, and hybrid approaches; we also discuss the FL concept.

Journal of Information Systems Engineering and Management
2025, 10(43s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Static Analysis:

Static analysis is considered a de facto malware detection method, focusing on identifying known signatures. This technique compares a program's digital fingerprint with a database of known malicious signatures, extracting key features like file hashes, API calls, executable interactions with DLLs, entropy, and opcode frequencies[7], [16]. API calls, particularly, are crucial in modern malware analysis, as they reveal how a program interacts with the operating system. Despite its successes, static analysis has significant limitations, particularly its vulnerability to code obfuscation techniques employed by modern ransomware. These techniques can hide malicious code, making detection more challenging and reducing the effectiveness of static analysis methods in identifying threats. This challenge highlights the need for more advanced detection strategies like dynamic or behavioural-based analysis that can adapt to evolving malware tactics.

Dynamic Analysis:

Dynamic analysis is structured to advance static analysis's shortcomings by thoroughly observing real-time software attributes. This method involves executing the said software in a container environment like the open sourced sandbox system in order to monitor its interactions, including memory usage, network activity, registry changes, and API calls to identify high level spatial correlations within these sequences thereby allowing for the recognition of malware behaviors that may not be immediately detectable using static or signature-based methods[17], [18]. Towards this research direction, several works have applied these concepts to enhance malware detection. For example, a hybrid classification model combining Multiscale CNNs (MSCNN) and ResNet was introduced by [19] the approach achieved a high accuracy of 99.20% in Android malware detection by extracting hierarchical features. While dynamic analysis has recorded impressive results in detecting malware behavior, especially against zero day attacks, it is not without challenges. The dynamic approach can be susceptible to evasion techniques, especially when relying on a single feature for detection [20]. This shortcoming highlights the need for a more hybrid method that involves robust feature integration and multi-dimensional analysis techniques to address the evolving sophistication of modern malware.

Hybrid Analysis:

Ransomware detection through the hybrid approach usually integrates the capabilities of static and dynamic analysis, providing a more comprehensive and effective solution. It combines signature analysis through code level inspection with real time behavioral observation to promote greater depth and accuracy in identifying ransomware threats [21], [22]. Hybrid analysis usually goes beyond static and dynamic approaches to incorporate the combination of advanced DL architectures, facilitating a more resilient and adaptable solution for malware and ransomware detection. Integrating static and dynamic analysis in hybrid methodologies has substantially improved ransomware detection; however, a critical challenge remains due to the lack of a comprehensive feature set. As noted by [20], relying solely on a single feature type often fails to capture sophisticated ransomware's complex and evolving behaviors, particularly modern variants with advanced capabilities. Leveraging this understanding, we have curated a comprehensive feature set that incorporates essential ransomware behavioral characteristics of API calls, DLL interactions, and Mutex operations—alongside their respective TTP attributes in accordance with the MITRE framework. This thorough strategy seeks to enhance detection accuracy by encompassing a wider array of ransomware behaviors, hence strengthening the detection system's resilience against progressively advanced threats.

Federated Learning Concept:

FL has emerged as a promising solution for modern ransomware detection due to its scalable and privacy-preserving architecture. Traditional approaches that involve transferring and storing sensitive data on a single server pose significant privacy and security risks, making systems vulnerable to data breaches. FL, on the other hand, enables organizations to collaboratively develop a unified ransomware detection model without sharing sensitive data. This decentralized approach allows each organization to maintain control over its data while still benefiting from the collective knowledge of others. Aggregating this collective knowledge is crucial for effectively addressing modern ransomware's evolving and diverse threats. Towards this research direction, several authors have applied FL in their

Journal of Information Systems Engineering and Management
2025, 10(43s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

methodology for malware detection and mitigation. The study by [23] applied FL in agricultural environments by implementing three distinct global models involving the CNN, In ransomware detection, FL application is heavily limited by the lack of comprehensive feature enriched; for example, the FLDetect framework proposed by [24] utilized API call logs from Windows systems to detect ransomware. Although it achieved a commendable accuracy of 93.1 %, the dataset comprised 600 ransomware samples, constraining its generalizability and scalability for practical applications. Moreover, the dependence on single API call logs may fail to identify more advanced ransomware behaviors that modify their execution patterns dynamically to avoid detection. We curated a comprehensive feature set that spans over 30000 samples integrating API calls, DLL interactions, mutex operations, and their corresponding MITRE TTP attributes to enhance detection accuracy and resilience against advanced ransomware threats.

MATERIALS AND METHODS

The Proposed RS-FEDRAD Model:

The RS-FEDRAD model introduces a robust distributive DL framework that employs a hybrid CNN-LSTM architecture for scalable ransomware detection. This framework is composed of four key modules. The first module involves dynamic analysis, where ransomware behavioral features are extracted using a sandboxing system. The second module focuses on the arrangement and concatenation of these extracted features, organizing them into sequences of API calls, DLL interactions, and Mutex operations. Next, the enhancement module enriches these ransomware attributes by mapping them to known TTPs. Finally, the enriched feature set is passed to the detection module, where it undergoes classification to identify ransomware. The CNN layer extracts key spatial features to identify patterns, even in obfuscated data. At the same time, the LSTM counterpart processes these features to detect crucial temporal dependencies, helping interpret ransomware execution sequences. FL enhances RS-FEDRAD's adaptability, enabling multiple organizations to train their models collaboratively without revealing private information. The RS-FEDRAD workflow is depicted in **Fig. 1**.

In RS-FEDRAD, the hyperparameters defined for the CNN-LSTM-FL model are chosen to have a technically balanced computation requirement alongside improved accuracy in a decentralized setting as shown in **Table 1**. The architecture ensured that it could handle various client datasets robustly, be representative of real-world federated deployments, and be adaptive as ransomware threats evolve. With the versatility and extensive library support of Python 3.12, particularly in ML and FL tasks, our hybrid CNN-LSTM architecture is built to utilize the benefits of CNN to capture relevant spatial features. At the same time, the LSTM pays attention to temporal features. The CNN Part has only one convolution layer of 64 filters and a kernel size of 3, which extracts localized spatial features from the ransomware behavior data like API call pattern, Mutex actions, TTP indicators, and DLL usage. This convoluted information then goes through a pooling layer where a pooling size of 2 is used to downsample the feature map while maintaining necessary information. Then, an LSTM layer with 100 units reads these spatial features and forms temporal associations, which helps to detect patterns in the order of operations during a ransomware execution process.

Algorithm 1 initializes the Hybrid CNN-LSTM model demonstrating sequential model architecture, wherein CNN layers initially extract spatial features using a Conv1D along 64 filters, three(3) size kernels using ReLU activation function, and MaxPooling with a pool size of two (2). Subsequently, it captures temporal dependencies in feature sequences through an LSTM layer that involves 100 units. The subsequent fully connected layers consist of 64 units of dense layers on ReLU, followed by a "0.5" dropout rate for regularization, culminating in an output layer before finally using sigmoid function for binary classification.

Journal of Information Systems Engineering and Management
2025, 10(43s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

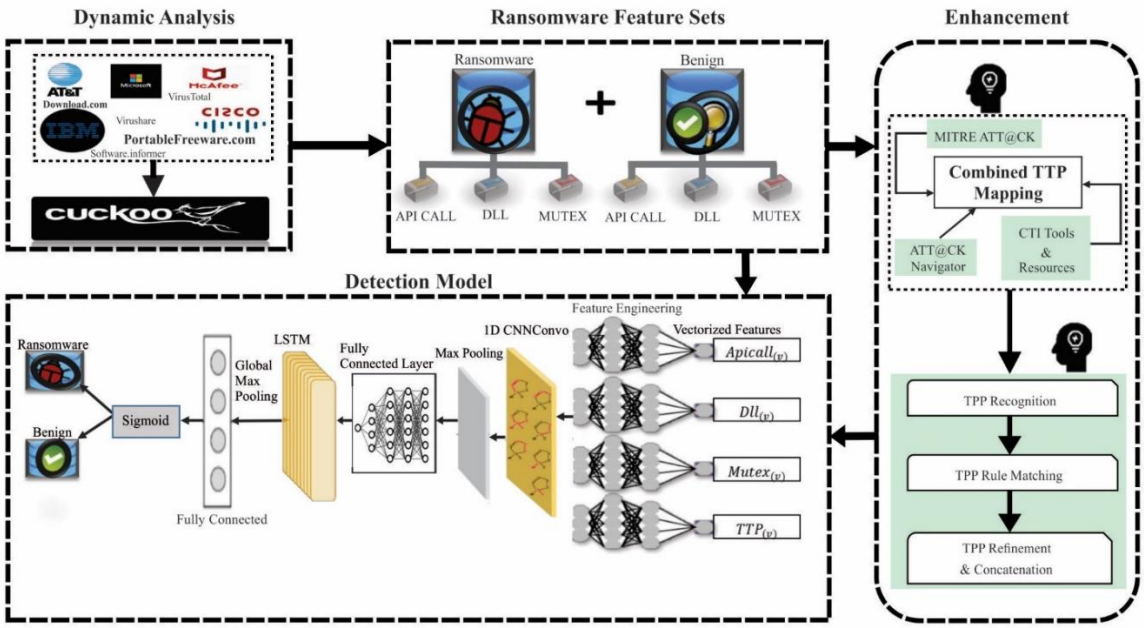

Figure 1. General Workflow of RS-FEDRAD

Table 1: Hyperparameter for CNN-LSTM Federated Model

Hyperparameter	Usable Values	Chosen Values
Programming Language	[Python, R, Julia]	Python 3.12
Framework	[TensorFlow, PyTorch, Keras, MXNet]	TensorFlow 2
Federated Server	[Flower, TensorFlow Federated, FedML]	Flower 1.2
Model Architecture	[CNN, LSTM, CNN-LSTM, RNN]	CNN-LSTM
Convolutional Layers	[1, 2, 3]	1
Convolutional Layers	[3, 5, 7]	3
Pooling Size	[2, 3, 4]	2
LSTM Units	[50, 100, 150]	100
Dense Layer Units	[32, 64, 128]	64
Dropout Rate	[0.1, 0.3, 0.5]	0.5
Output Activation	['sigmoid', 'softmax']	softmax
Optimizer	['SGD', 'Adam', 'RMSprop']	Adam
Batch Size	[32, 64, 128]	

Proposed Federated Learning Architecture:

After introducing the hybrid CNN-LSTM framework, the next focus is the FL architecture for orchestrating the training in a decentralized and privacy ensured paradigm. This federated architecture will allow us to train the DL model locally and in parallel without the need to share raw information. **Fig. 2** illustrates the proposed FL architecture, employing a Flower server and TensorFlow to create a decentralized, privacy preserving training paradigm. The server initializes the training process by orchestrating the global FL configuration (θ-global) and verifying whether clients have successfully joined the session. Once clients are connected, the server dispatches the

Journal of Information Systems Engineering and Management
2025, 10(43s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

training configuration to the available clients, enabling local training on their respective datasets (X_{train}, Y_{train}) for the current round or epoch.

Algorithm 1. Hybrid CNN-LSTM Model Initialization.

Input: Preprocessed feature set *X'*
Output: Initialized CNN-LSTM model
1: **Model Architecture Definition**:
2: Initialize Sequential model $\mathcal{M}$
3: **CNN Layers** for Spatial Feature Extraction:
4: Add Conv1D layer to $\mathcal{M}$ with:
5: 64 filters, kernel size = 3, ReLU activation, input shape = input_shape
6: Add MaxPooling1D layer to $\mathcal{M}$ with pool size = 2
7: **LSTM Layer** for Temporal Sequence Learning:
8: Add LSTM layer to $\mathcal{M}$ with 100
9: **Fully Connected Layers for Classification**:
10: Dense layer to $\mathcal{M}$ with 64 units and ReLU activation
11: Dropout layer with rate = 0.5 to prevent overfitting
13: **Model Compilation**:
14: Compile $\mathcal{M}$ with: Accuracy as the evaluation metric
16**: Return**: Compiled CNN-LSTM model $\mathcal{M}$

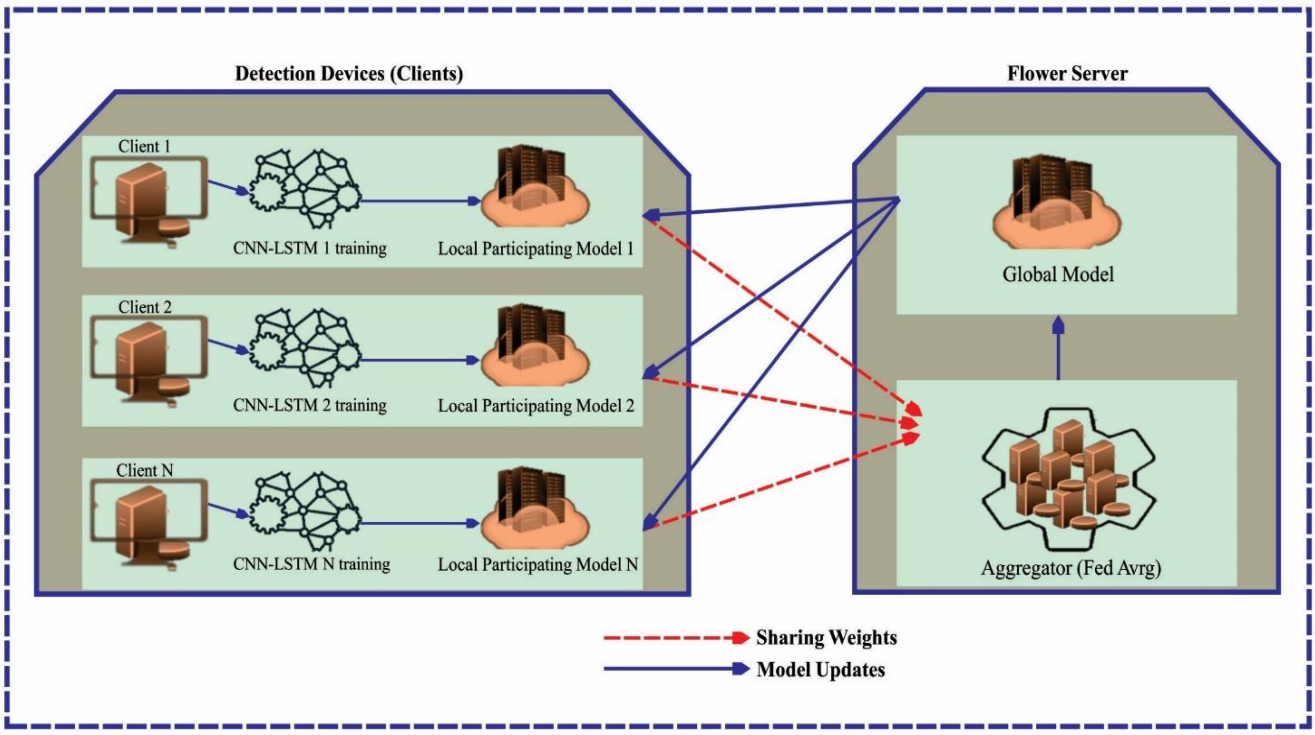

Figure 2: RS-FEDRAD Federated architecture

Journal of Information Systems Engineering and Management
2025, 10(43s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

These clients use their validation quota to evaluate local performance, measuring F1-scores, recall, precision, and accuracy. **Algorithm 2** gives further details on the training procedure. On the client side, each device preprocesses its local dataset by standardizing, normalizing, stratifying, and splitting it into validation and training percentages. Then, it trains on its local model using backpropagation for 10 epochs, fine tuning it to maximize local performance metrics. Early stopping is incorporated to halt the possibility of overfitting and enhance efficiency.

EXPERIMENTAL SETUP AND IMPLEMENTATION

Experimental Setup:

The experiments were conducted on a computer with 32 GB RAM, 11th generation Intel® Core™ i7-1165G7 processor at 2.80 GHz, and a Windows 10 operating system. The implementation was developed in Python 3.12, leveraging the TensorFlow and Keras libraries for model building and optimization. In each round of the RS-FEDRAD model, 10,15 and 20 clients participated, reflecting a distributed and scalable approach that closely simulates real-world federated scenarios.

Dataset Description:

The four datasets used in this study were collected from multiple sources, illustrating that the decentralized and varied characteristics of actual ransomware variants contribute unique characteristics, mimicking the diverse environments in which ransomware appears. such as enterprise networks, cloud infrastructures, and personal computing systems. A stratified data-splitting strategy was meticulously applied to generate 10 representative clients for the initial federated training, ensuring balanced and meaningful data distribution for each client.

The dataset sequence length was 530, comprising 500 API calls, 10 DLLs, 10 Mutex operations, and 10 TTP features derived from their concatenated attributes. **Table 2** details the sources of each dataset, including publication years, dataset source. The sample count shows a balanced distribution between benign (49 %) and ransomware (51 %) of the total samples.

Curating TTP Enhancement on Datasets:

A key component of our proposed methodology is the feature extraction pipeline, which led to the establishment of a novel TTP-enhanced dataset that represents industry recognized ransomware-specific attributes. The pipeline was explicitly designed to capture the behavioral traits of ransomware by mapping critical behavioral features like API calls, DLL, and Mutex to industry recognized threat techniques using the MITRE ATT&CK framework.

By leveraging this concept, this study tries to deliver a complete and structured understanding of the ransomware lifecycle, from initial access to impact. This ideology has motivated this study to create a novel, TTP-enhanced dataset that not only highlights the traditional indicators of compromise but also incorporates a broader range of ransomware behavioral attributes that are difficult to detect using conventional methods

Algorithm 2 Federated Training and Aggregation in RS-FEDRAD

Input: Initialized global θ_global, local datasets (X'_i, Y_i) for each client i
Output: Final global model parameters θ_global
1: **Server Initialization**: Initialize global CNN-LSTM
2**: Federated Training Procedure**:
3: **for** each communication round r = 1 to R
4: Server randomly selects a subset of N clients 5: **for** each client i = 1 to N (in parallel)
5: **Data Preprocessing**: Load local dataset (X'_i, Y_i), then perform:
6: initialize local model parameters θ_i ← θ_global.
7: **for** each epoch e = 1 to E Calculate local loss and update θ_i using backpr
8: **if** early stopping criteria met
9: break, **end if**
10: **end for**

Journal of Information Systems Engineering and Management
2025, 10(43s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

11: Calculate metrics (F1-scores, accuracy)
12: Server updates global model parameters θ_ global to all clients
13: **Return**: Final global model parameters θ_ global

Table 2: Dataset Information: a balanced distribution between benign (49 %) and ransomware (51 %)

Reference	Data Source	Year	Malware Type	Sample Count
RSDT 2	From Virusshare	2016	Ransomware	6264
RSDT 3	Extracted from SOREL-20M	2020	Ransomware	8444
RSDT 1	From ISOT	2020	Ransomware	668
BDT	Several sources, including systems files	2019 To 2020	Benign	14797

TTP Recognition:

Our approach to the TTP mapping process begins by recognizing or identifying key behavioral indicators, namely, sequences of API calls, DLL interactions, and Mutex instances within a structured feature set. Recognizing that API calls frequently serve as primary indicators of ransomware activity, we first and foremost examine the contextual links between API calls, the DLL interactions, and Mutex formations. These behavioral elements rarely function independently; instead, they exhibit coordinated patterns integral to ransomware evasion and propagation methods. For instance, a specific API call may signal the initiation of a malicious activity, while corresponding DLL calls and Mutex interactions further contribute to the ransomware's stealth and persistence mechanisms. Therefore, these principles have been applied following these steps.

Behaviour based Identification: Each observed API call, DLL interaction, and Mutex formation is analyzed for behavioral patterns typical of ransomware. By examining these interactions collectively rather than in isolation, this study captures ransomware's orchestrated tactics for evasion, execution, and propagation.

Sequence Alignment: The connected sequence structure reflects how individual elements within a behavior profile can collectively signify a TTP. For instance, an API call for file encryption may align with specific DLLs and Mutexes that support file access restrictions or disguise malicious actions, creating a behaviour signature corresponding to a ransomware TTP.

Cosine Similarity for Sequence Matching: Cosine similarity enables the assessment of the degree to which two sequences match. Higher similarity scores indicate a closer alignment with known ransomware TTPs, while lower scores suggest a lower likelihood of matching known ransomware profiles. In encapsulating these relationships mathematically, this work represents each behaviour type-API, DLL, and Mutex as feature vectors within a concatenated sequence, denoted by $x_{\text{sequence}} = [x_{\text{API}}, x_{\text{DLL}}, x_{\text{Mutex}}]$

Eq. (1) illustrates the cosine similarity metric, which yields a value ranging from -1 to 1, where 1 signifies identical vectors of perfect similarity, 0 denotes orthogonal vectors of no similarity, and -1 represents opposing vectors. Calculating this measure enables us to evaluate the alignment between observed behaviours and established ransomware patterns, thereby facilitating the detection of nuanced correlations among API, DLL, and Mutex features and identifying unique TTP patterns indicative of ransomware. Our TTP recognition process uses cosine similarity to evaluate the alignment between observed sequences. S_1 and S_2 with known TTP patterns from our dataset. The similarity is calculated as follows:

$$\text{Sim}(S_1, S_2) = \frac{x_{\text{sequence}_1} \cdot x_{\text{sequence}_2}}{\|x_{\text{sequence}_1}\| \|x_{\text{sequence}_2}\|} \quad (1)$$

Journal of Information Systems Engineering and Management
2025, 10(43s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Where: x_{sequence_1} and x_{sequence_2} are concatenated feature vectors, S_1 and S_2, representatives of the vectors. This similarity measure enables the detection of nuanced correlations within and across API, DLL, and Mutex features, which facilitates the identification of unique TTP patterns.

TTP Rule Matching:

TTP Rule Matching is designed to checkmate and correlate behavioral indicators, such as API calls, DLL interactions, and Mutex instances, more closely with predefined TTP patterns through a structured, technically rigorous analysis.

Two key mathematical measures are used, as shown in Eq. (2) and Eq. (3), to quantify behavioral similarity and relevance, they are:

(1)Jaccard Index**:** Used to assess the similarity between two TTP behaviour sets, we employ the Jaccard Index to calculate the ratio of intersecting behaviours to the union of behaviors within two sets , A and B.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \quad (2)$$

A and B represent sets of behaviors mapped to TTPs, allowing us to measure how closely the two behaviours align, particularly in the presence of specific API calls, DLL interactions, and Mutex patterns.

(2)Confidence Score Calculation: For each matched TTP, we compute a confidence score C(t) that reflects the weighted relevance of each behavioural feature. This score is calculated as follows: .

$$C(t) = \frac{\sum_{i=1}^{n} w_i \cdot \text{Sim}(S_i, T_i)}{n} \quad (3)$$

Where w_i represents the weight assigned to each behavior S_i to a known T_i. The confidence score aggregates similarity measures across multiple behaviour types, such as API calls, DLLs, and Mutexes, providing a comprehensive confidence metric that reflects the likelihood of the TTP match.

Security-Sensitive: Technical level labeling is used to refine TTP analysis by assigning sensitivity levels to each TTP based on its potential impact and criticality. High sensitivity levels are assigned to API calls involved in critical functions like file manipulation and encryption, central to ransomware activity. DLL usage that compromises system integrity or enables privilege escalation is also labeled with high sensitivity, as are mutex operations controlling access to crucial resources or processes, reflecting their role in synchronizing key stages of ransomware deployment. The sensitivity function for API calls, DLLs, and mutex operations is calculated using SL(t) for each TTP t as described in Eq. (4).

$$SL(t) = \alpha \cdot I(t) + \beta \cdot D(t) \quad (4)$$

Where: $I(t)$ is the impact score, $D(t)$ represents the damage potential and α and β are weighting factors that prioritize either impact or damage based on the ransomware's operational goals.

Refinement and Validation of TTP Mappings:

After the initial TTP mapping, a comprehensive refinement process was undertaken to maximize the precision and robustness of the dataset. This multi-stage approach aimed to elevate the dataset's fidelity to industry standards by employing advanced automation, domain expertise, and contextual analysis. Automated tools, including the MITRE ATT&CK Navigator and custom Python scripts, were pivotal in systematically cross-referencing API calls, DLL interactions, and Mutex behaviours against an extensive TTP database. Furthermore, expert analysts conducted meticulous manual reviews, verifying each mapping to prevent any benign behaviours from being misclassified as malicious.

This step included consultations with domain experts to resolve ambiguities—particularly in cases where benign and malicious behaviours shared similar features. For instance, the team distinguished between benign DLL loading operations and those indicative of malicious process injections, maintaining a clear distinction between regular operations and ransomware-specific tactics. Upon completing the refinement process, the enhanced TTP mappings

Journal of Information Systems Engineering and Management
2025, 10(43s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

were integrated into the existing dataset, rendering it both technically accurate and applicable to real-world scenarios. This integration phase involved automated data merging to harmonize the refined mappings with the original dataset while maintaining uniformity across data points. Automated consistency checks were performed to ensure that the merged dataset aligned with known TTP patterns, thereby preserving the dataset's integrity.

EVALUATION

Evaluations Metrics:

The performance of RS-FEDRAD is assessed using numerous standard criteria frequently employed in cutting edge research on malware detection and classification systems. These metrics evaluate the model's capacity to accurately distinguish between harmful and non malicious cases and its efficacy in a decentralized environment, hence ensuring a comprehensive assessment of its efficacy. These classification metrics include

(1) True Positive (TP): Best described as the number of malicious samples accurately classified.

(2) True Negative (TN): This metric is known as the non malicious samples accurately classified.

(3) False Positive (FP): Indicates the count of standard samples mistakenly identified as malicious.

(4) False Negative (FN): This shows the number of malicious instances incorrectly classified as good.

From these basic classification components, we derive the following key metrics to check the model's effectiveness.

Accuracy: This type of evaluation represents the proportion of correctly classified samples (malicious and normal) out of the total samples. Accuracy is described in Eq. (5)

$$\frac{\mathrm{TP} + TN}{\mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{5}$$

Precision: This criterion measures the proportion of true positives (malicious samples correctly identified) among all samples classified as malicious. Precision is described in Eq. (6).

$$\frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{6}$$

Recall: Also known as sensitivity or true positive rate, recall is the ratio of correctly identified malicious samples to the total number of malicious samples. It provides insight into the model's ability to detect malicious instances and is mathematically described in Eq. (7).

$$\frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{7}$$

F1-Score: This metric demonstrates the harmonic mean of precision and recall, as shown in Eq. (8).

$$2\,x\,\frac{\text{Precision x Recall}}{\text{Precision x Recall}} \tag{8}$$

RESULTS AND DISCUSSIONS

The F1-scores, Recall, Accuracy, and Precision Performance of RS-FEDRAD:

In ransomware detection, achieving high accuracy is crucial for distinguishing between malicious and benign cases. RS-FEDRAD demonstrated an impressive performance, achieving a final accuracy of 99.90% after training, highlighting its precision in identifying ransomware behaviors as shown in **Fig 3.**

Journal of Information Systems Engineering and Management
2025, 10(43s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

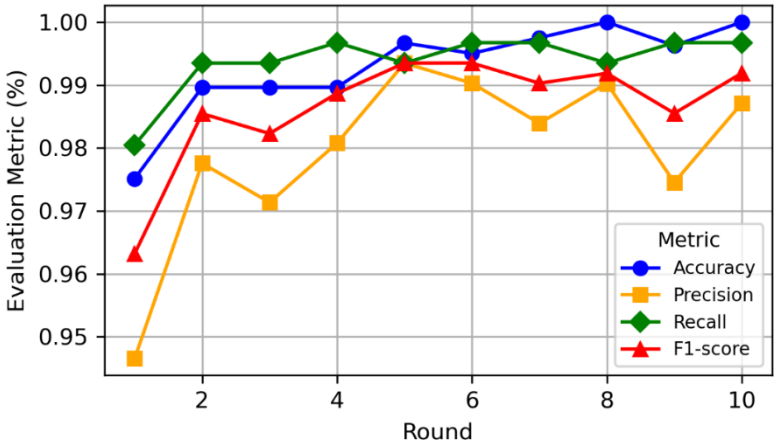

Figure 3: RS-FEDAD's general performance plot across all metrics

Additionally, RS-FEDRAD achieved a precision of 99.22% and recall of 99.12%, ensuring effective detection while minimizing false positives and negatives. The F1-score of 99.13 % by the final round substantiates RS-FEDRAD's efficacy in handling the often imbalanced nature of ransomware datasets, where missing even a single ransomware instance can have significant consequence.

General Performance Using Various Federated Aggregation Methods:

Inspired by the approach in [25], we evaluated the efficacy of RS-FEDRAD in enhancing ransomware detection accuracy in a decentralized environment by using various FL aggregation strategies across 10 clients over 10 training rounds with 10 epochs. The results, shown in **Fig. 4**, compare the performance of four federated averaging strategies—FedAvg, FedProx, FedAvgM, and FedMedian.

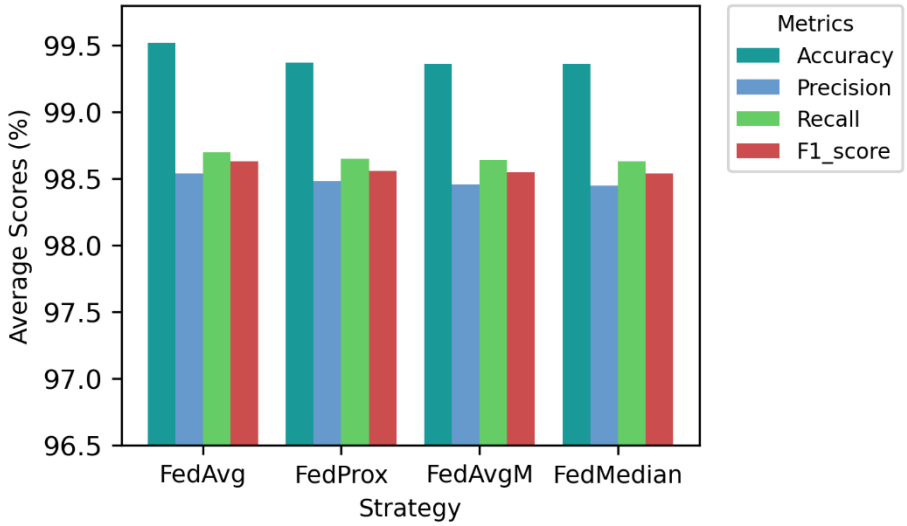

Figure 4: Comparison of various aggregation strategy performance

Each strategy demonstrated distinct characteristics, highlighting its suitability for different FL scenarios based on technical considerations. FedAvg demonstrated the highest accuracy at 99.52 %, alongside superior precision (98.54 %), recall (98.69 %), and F1-score (98.63 %). It's simple averaging mechanism effectively captures global trends when client data distributions are relatively homogeneous. FedProx achieved 99.32% accuracy with precision, recall, and F1-scores of 98.35%, 98.50%, and 98.43%, respectively. Addressing client heterogeneity with a proximal term ensures consistent updates across distributed nodes, making it suitable for non IID data, though at the cost of

Journal of Information Systems Engineering and Management
2025, 10(43s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

increased computational overhead. FedAvgM closely followed with 99.22% accuracy and demonstrated its ability to accelerate convergence by incorporating momentum, which is beneficial in environments requiring rapid iterations. However, its sensitivity to hyperparameters may affect performance consistency. FedMedian recorded 99.12% accuracy and prioritized robustness against adversarial clients or noisy updates, making it reliable in challenging FL environments.

Robustness Assessment: Federated Learning VS. Adversarial Training:

The robustness of RS-FEDRAD, a decentralized, FL model for ransomware detection, is rigorously evaluated against adversarial attacks to confirm its resilience and suitability in real-world distributed cybersecurity applications as shown in **Fig.5** and further illustrated in **Table 3**. Given the sensitive nature of ransomware detection, such a model should ensure data integrity, confidentiality, and availability, even under potential adversarial attacks. Adversarial testing was conducted for this experiment by introducing two standard methods of checking resilience to adversarial perturbations: The Basic Iterative Method (BIM), otherwise referred to as the unknown type black-box attacks, and the Fast Gradient Sign Method (FGSM), also commonly referred to as the known type white-box attack. These attacks simulate realistic adversarial conditions to evaluate the RS-FEDRAD response and resilience.

Assessing the Stability in Scalability of RS-REDRAD:

The decentralized structure of RS-FEDRAD enhances scalability, allowing additional clients to participate in federated training without overburdening the central server. Experiments with 10, 15, and 20 clients, demonstrate minimal performance variance, with accuracy ranging from 99.50% to 99.57%, confirming RS-FEDRAD's stability as client participation increases. The results visualized in **Fig. 6** highlight the model's consistently high performance across all metrics, confirming its reliability and scalability for ransomware detection in federated environments.

COMPARING WITH RELATED WORKS

RS-REDRAD Against DL-Based Approaches:

Fig. 7 presents a comprehensive comparison of the performance of RS-FEDRAD against recent DL methodologies for malware or ransomware detection. The comparison demonstrates the efficacy of RS-FEDRAD, particularly when benchmarked against recent dynamic DL models. The study by [18] introduced a malware detection model utilizing Markov Chains and word embeddings to examine API call sequences, attaining a detection precision of 99.0% and an early prediction accuracy of 99.7%. Nonetheless, their elevated malware to benign ratio probably exaggerated performance, and the lack of

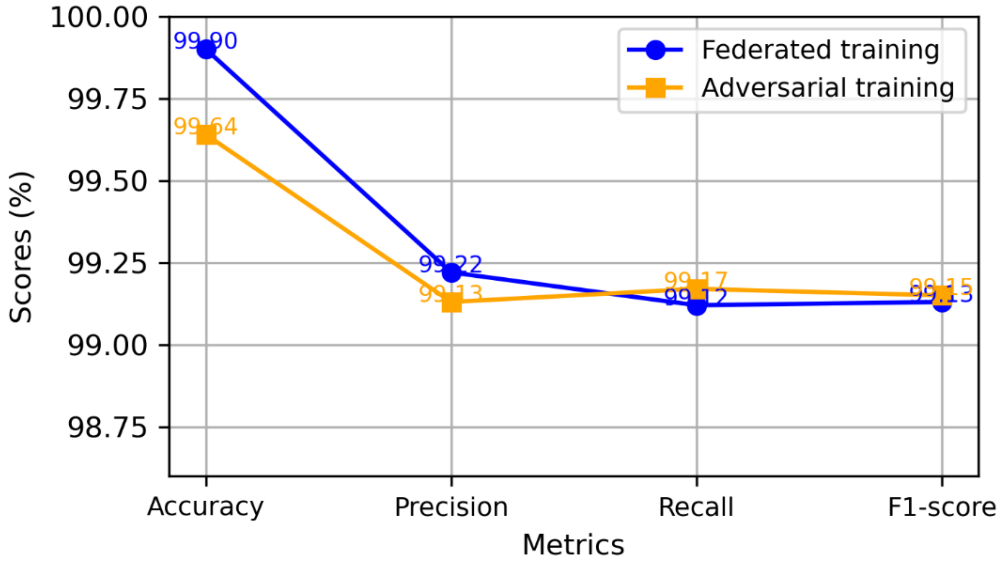

Figure 5: Model performance across rounds: federated learning vs. adversarial training

Journal of Information Systems Engineering and Management
2025, 10(43s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table 3: RS-FEDRAD's metric indicators with mean and standard deviation

Metric	Mean	Standard Deviation
Accuracy	99.507	0.035
Precision	98.539	0.044
Recall	98.695	0.050
F1-score	98.622	0.047
AUC-ROC	99.59	0.032

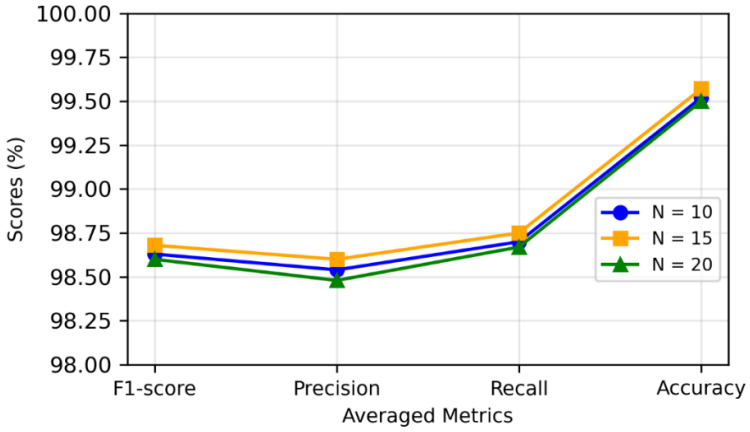

Figure 6: Scalability and stability analysis of RS-FEDRAD across varying client counts

reporting constrains the evaluation of model stability. Conversely, RS-FEDRAD's CNN-LSTM architecture derives spatial-temporal patterns from a feature-enriched dataset, hence providing superior generalization. It attains 99.9 % accuracy with an FPR of 0.20 %, markedly lower than its 1.0% FPR.

Furthermore, RS-FEDRAD's SD analysis substantiates stability, rendering it a more dependable tool for ransomware detection. In [26], they employed an RNN-based malware detection model that incorporated opcode sequences with one-hot, random, and Word2Vec feature vectors, attaining impressive performance , however, they used a relatively small sample size and reported substantial computational expense that could constrain scaling. Their methodology was deficient in API-level behavioural analysis, rendering it vulnerable to obfuscation tactics; likewise, RS-FEDRAD-attained an AUC-ROC of 98.7 %, surpassing the stability and scalability of their method.

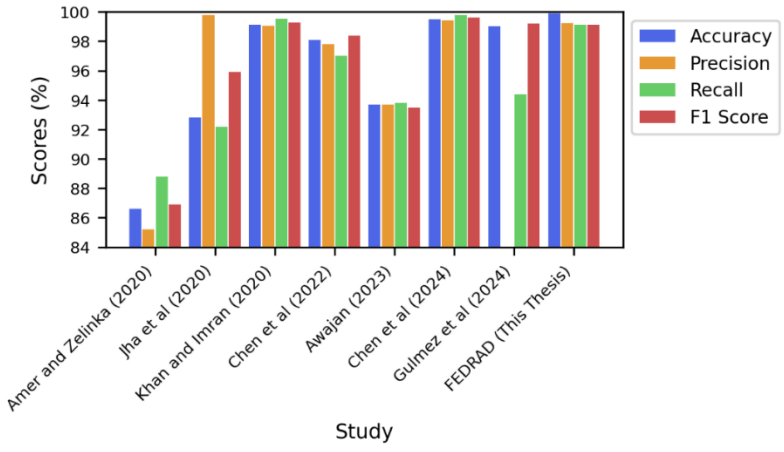

Figure 7: Comparison analysis of RS-FEDRAD VS Deep Dynamic approaches

Journal of Information Systems Engineering and Management
2025, 10(43s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

RS-REDRAD Against FL-Based Approaches:

When comparing RS-FEDRAD to state-of-the-art FL approaches, it consistently performs better in key metrics. The performance outcomes presented in **Fig. 8** highlight the considerable benefits of RS-FEDRAD, especially its capacity to detect the intricacies of ransomware behavior via its enhanced TTP-enhanced dataset alongside its hybrid CNN-LSTM architecture and FedAvg aggregation method. FedDICE [27] presents an FL framework for ransomware detection in decentralized hospital networks, attaining 99 % accuracy while safeguarding data privacy. Nevertheless, it was trained on logistic regression, resulting in a high computational burden and susceptibility to adversarial assaults. More so, RS-FEDRAD enhances FL-based ransomware detection by integrating a TTP-enriched dataset, unlike FedDICE, which primarily targets network layer mitigation through SDN.

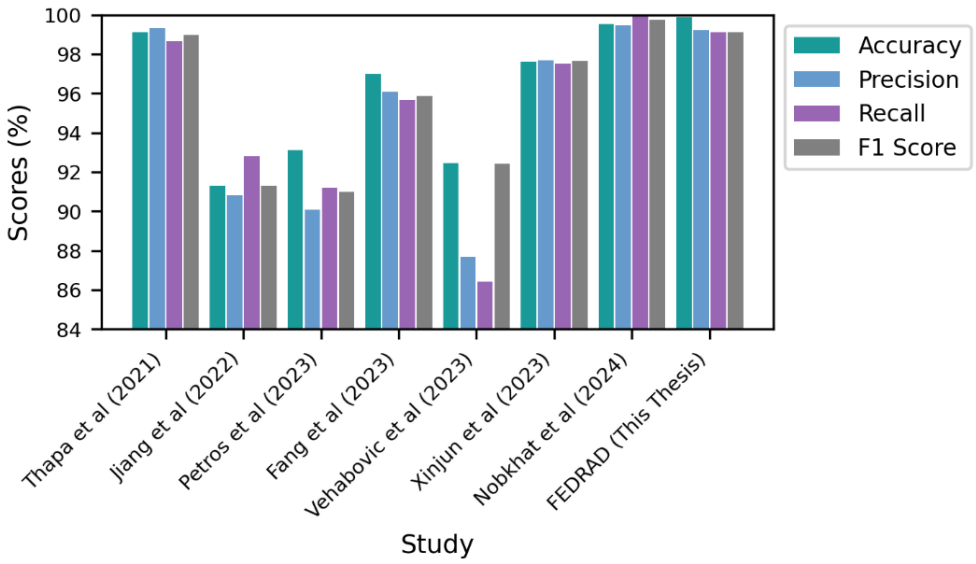

Figure 8: Comparison Analysis of RS-FEDRAD VS Federated Deep Learning Approaches

Similarly, FLDetect [24] utilizes API call-based detection to identify ransomware on Windows-based IoT devices, attaining an accuracy of 93.1% using the ransomwaredataset2016. They utilized a small sample dataset consisting of approximately 1500 samples, which is also quite outdated. RS-FEDRAD attains 99.90 % accuracy with a 0.9 % improvement in TPR on a 30000 data sample from multiple sources, significantly surpassing FLDetect, owing to our feature rich extraction pipeline. SIMFED [25] utilizes a lightweight one-dimensional CNN for the detection of IoT malware, employing FedAvg aggregation on the IoT-23 dataset. It attained a high accuracy of 99.52 %, surpassing some of the most centralized DL models while preserving privacy. Similar to RS-FEDRAD, SIMFED integrates hyperparameter optimization and adversarial robustness testing via FGSM and BIM attacks and minimizes computing costs. Nevertheless, it predominantly focused on network traffic analysis, thereby constraining its efficacy in detecting ransomware using behavior-based methods.

REFERENCES

[1] C. Beaman, A. Barkworth, T. D. Akande, S. Hakak, and M. K. Khan, "Ransomware: Recent advances, analysis, challenges and future research directions," *Comput Secur*, vol. 111, Dec. 2021, doi: 10.1016/j.cose.2021.102490.

[2] F. Javadnejad, A. M. Abdelmagid, C. A. Pinto, M. Mcshane, and R. Diaz, "An exploratory data analysis of malware/ransomware cyberattacks: insights from an extensive cyber loss dataset," *Enterp Inf Syst*, vol. 18, no. 9, 2024, doi: 10.1080/17517575.2024.2369952.

[3] A. Yuryna Connolly and H. Borrion, "Reducing Ransomware Crime: Analysis of Victims' Payment Decisions," *Comput Secur*, vol. 119, Aug. 2022, doi: 10.1016/j.cose.2022.102760.

[4] A. Kulkarni, Y. Wang, M. Gopinath, D. Sobien, A. Rahman, and F. A. Batarseh, "A Review of Cybersecurity Incidents in the Food and Agriculture Sector," Mar. 2024, [Online]. Available: http://arxiv.org/abs/2403.08036

[5] K. Gaur, N. Kumar, A. Handa, and S. K. Shukla, "Static Ransomware Analysis Using Machine Learning and Deep Learning Models," in *Communications in Computer and Information Science*, Springer Science and Business Media Deutschland GmbH, 2021, pp. 450–467. doi: 10.1007/978-981-33-6835-4_30.
[6] M. A. Ayub, A. Siraj, B. Filar, and M. Gupta, "RWArmor: a static-informed dynamic analysis approach for early detection of cryptographic windows ransomware," *Int J Inf Secur*, vol. 23, no. 1, pp. 533–556, Feb. 2024, doi: 10.1007/s10207-023-00758-z.
[7] J. A. Herrera-Silva and M. Hernández-Álvarez, "Dynamic Feature Dataset for Ransomware Detection Using Machine Learning Algorithms," *Sensors*, vol. 23, no. 3, Feb. 2023, doi: 10.3390/s23031053.
[8] R. Sihwail, K. Omar, K. A. Z. Ariffin, and S. Al Afghani, "Malware detection approach based on artifacts in memory image and dynamic analysis," *Applied Sciences (Switzerland)*, vol. 9, no. 18, Sep. 2019, doi: 10.3390/app9183680.
[9] A. Kapoor, A. Gupta, R. Gupta, S. Tanwar, G. Sharma, and I. E. Davidson, "Ransomware detection, avoidance, and mitigation scheme: A review and future directions," Jan. 01, 2022, *MDPI*. doi: 10.3390/su14010008.
[10] P. Maniriho, A. N. Mahmood, and M. J. M. Chowdhury, "A systematic literature review on Windows malware detection: Techniques, research issues, and future directions," *Journal of Systems and Software*, vol. 209, Mar. 2024, doi: 10.1016/j.jss.2023.111921.
[11] S. A. Wadho, A. Yichiet, M. L. Gan, L. C. Kang, R. Akbar, and R. Kumar, "Emerging Ransomware Attacks: Improvement and Remedies-A Systematic Literature Review," in *2023 4th International Conference on Artificial Intelligence and Data Sciences: Discovering Technological Advancement in Artificial Intelligence and Data Science, AiDAS 2023 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 148–153. doi: 10.1109/AiDAS60501.2023.10284647.
[12] B. S. Guendouzi, S. Ouchani, H. EL Assaad, and M. EL Zaher, "A systematic review of federated learning: Challenges, aggregation methods, and development tools," Nov. 01, 2023, *Academic Press*. doi: 10.1016/j.jnca.2023.103714.
[13] S. B. Guendouzi, S. Ouchani, and M. Malki, "Enhancing the Aggregation of the Federated Learning for the Industrial Cyber Physical Systems," in *Proceedings of the 2022 IEEE International Conference on Cyber Security and Resilience, CSR 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 197–202. doi: 10.1109/CSR54599.2022.9850301.
[14] "MITRE ATT&CK®." Accessed: Nov. 28, 2024. [Online]. Available: https://attack.mitre.org/
[15] Al-SadaBader, SadighianAlireza, and OligeriGabriele, "MITRE ATT&CK: State of the Art and Way Forward," *ACM Comput Surv*, Oct. 2024, doi: 10.1145/3687300.
[16] I. Kara and M. Aydos, "Static and Dynamic Analysis of Third Generation Cerber Ransomware," *International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism, IBIGDELFT 2018 - Proceedings*, pp. 12–17, Jan. 2019, doi: 10.1109/IBIGDELFT.2018.8625353.
[17] N. Hampton, Z. Baig, and S. Zeadally, "Ransomware behavioural analysis on windows platforms," *Journal of Information Security and Applications*, vol. 40, pp. 44–51, Jun. 2018, doi: 10.1016/J.JISA.2018.02.008.
[18] E. Amer and I. Zelinka, "A dynamic Windows malware detection and prediction method based on contextual understanding of API call sequence," *Comput Secur*, vol. 92, May 2020, doi: 10.1016/j.cose.2020.101760.
[19] X. Fu, C. Jiang, C. Li, J. Li, X. Zhu, and F. Li, "A hybrid approach for Android malware detection using improved multi-scale convolutional neural networks and residual networks," *Expert Syst Appl*, vol. 249, Sep. 2024, doi: 10.1016/j.eswa.2024.123675.
[20] T. Chen, H. Zeng, M. Lv, and T. Zhu, "CTIMD: Cyber threat intelligence enhanced malware detection using API call sequences with parameters," *Comput Secur*, vol. 136, no. October 2023, p. 103518, 2024, doi: 10.1016/j.cose.2023.103518.
[21] S. Aurangzeb, "A Machine Learning Based Hybrid Approach to Classify and Detect Windows Ransomware," 2018.
[22] K. P. Subedi, D. R. Budhathoki, and D. Dasgupta, "Forensic analysis of ransomware families using static and dynamic analysis," in *Proceedings - 2018 IEEE Symposium on Security and Privacy Workshops, SPW 2018*, Institute of Electrical and Electronics Engineers Inc., Aug. 2018, pp. 180–185. doi: 10.1109/SPW.2018.00033.

Research Article

[23] O. Friha, M. A. Ferrag, L. Shu, L. Maglaras, K. K. R. Choo, and M. Nafaa, "FELIDS: Federated learning-based intrusion detection system for agricultural Internet of Things," *J Parallel Distrib Comput*, vol. 165, pp. 17–31, Jul. 2022, doi: 10.1016/j.jpdc.2022.03.003.

[24] T. Petros, H. Ghirmay, S. Otoum, R. Salem, and M. Debbah, "FLDetect: An API-Based Ransomware Detection Using Federated Learning," in *Proceedings - IEEE Global Communications Conference, GLOBECOM*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 4449–4454. doi: 10.1109/GLOBECOM54140.2023.10437540.

[25] M. Nobakht, R. Javidan, and A. Pourebrahimi, "SIM-FED: Secure IoT malware detection model with federated learning," *Computers and Electrical Engineering*, vol. 116, May 2024, doi: 10.1016/j.compeleceng.2024.109139.

[26] S. Jha, D. Prashar, H. V. Long, and D. Taniar, "Recurrent neural network for detecting malware,"*Comput Secur*, vol. 99, Dec. 2020, doi: 10.1016/j.cose.2020.102037.

[27] C. Thapa, K. K. Karmakar, A. H. Celdran, S. Camtepe, V. Varadharajan, and S. Nepal, "FedDICE: A Ransomware Spread Detection in a Distributed Integrated Clinical Environment Using Federated Learning and SDN Based Mitigation," in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, Springer Science and Business Media Deutschland GmbH, 2021, pp. 3–24. doi: 10.1007/978-3-030-91424-0_1.