Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.055469

ARTICLE

Privacy Preservation in IoT Devices by Detecting Obfuscated Malware Using Wide Residual Network

Deema Alsekait[1], Mohammed Zakariah[2], Syed Umar Amin[3,*], Zafar Iqbal Khan[3] and Jehad Saad Alqurni[4]

[1]Applied College, Department of Computer Science and Information Technology, Princess Nourah bint Abdulrahman University, Riyadh, 11671, Saudi Arabia

[2]College of Computer and Information Sciences, King Saud University, Riyadh, 11362, Saudi Arabia

[3]College of Computer and Information Sciences, Prince Sultan University, Riyadh, 11586, Saudi Arabia

[4]Department of Educational Technologies, College of Education, Imam Abdulrahman Bin Faisal University, Dammam, 31441, Saudi Arabia

*Corresponding Author: Syed Umar Amin. Email: samin@psu.edu.sa

Received: 28 June 2024 Accepted: 20 August 2024 Published: 18 November 2024

ABSTRACT

The widespread adoption of Internet of Things (IoT) devices has resulted in notable progress in different fields, improving operational effectiveness while also raising concerns about privacy due to their vulnerability to virus attacks. Further, the study suggests using an advanced approach that utilizes machine learning, specifically the Wide Residual Network (WRN), to identify hidden malware in IoT systems. The research intends to improve privacy protection by accurately identifying malicious software that undermines the security of IoT devices, using the MalMemAnalysis dataset. Moreover, thorough experimentation provides evidence for the effectiveness of the WRN-based strategy, resulting in exceptional performance measures such as accuracy, precision, F1-score, and recall. The study of the test data demonstrates highly impressive results, with a multiclass accuracy surpassing 99.97% and a binary class accuracy beyond 99.98%. The results emphasize the strength and dependability of using advanced deep learning methods such as WRN for identifying hidden malware risks in IoT environments. Furthermore, a comparison examination with the current body of literature emphasizes the originality and efficacy of the suggested methodology. This research builds upon previous studies that have investigated several machine learning methods for detecting malware on IoT devices. However, it distinguishes itself by showcasing exceptional performance metrics and validating its findings through thorough experimentation with real-world datasets. Utilizing WRN offers benefits in managing the intricacies of malware detection, emphasizing its capacity to enhance the security of IoT ecosystems. To summarize, this work proposes an effective way to address privacy concerns on IoT devices by utilizing advanced machine learning methods. The research provides useful insights into the changing landscape of IoT cybersecurity by emphasizing methodological rigor and conducting comparative performance analysis. Future research could focus on enhancing the recommended approach by adding more datasets and leveraging real-time monitoring capabilities to strengthen IoT devices' defenses against new cybersecurity threats.

KEYWORDS

Obfuscated malware detection; IoT devices; Wide Residual Network (WRN); malware detection; machine learning

Terms	Abbreviations
IoT	Internet of Things
ML	Machine Learning
BiLSTM	Bidirectional Long Short-Term Memory Networks
GNB	Gaussian Naïve Bayes
DL	Deep Learning
EL	Ensemble Learning
GBT	Gradient Boosted Tree
ResNet	Residual Network
ReLU	Rectified Linear Unit
TP/TN	True Positive/True Negative
ROC	Receiver Operating Characteristic
TPR	True Positive Rate
WRN	Wide Residual Network
CNN	Convolutional Neural Network
LR	Logistic Regression
KNN	K-Nearest Neighbors
NB	Naïve Bayes
MLP	Multilayer Perceptron
RFE	Recursive Feature Elimination
OLS	Ordinary Least Squares
AML	Advancing Machine Learning
FP/FN	False Positive/False Negative
FPR	False Positive Rate

1 Introduction

The protection of privacy on Internet of Things (IoT) devices is becoming increasingly important as linked devices proliferate quickly across many industries [1]. The IoT improves comfort and efficiency by enabling smooth data flow and connectivity between devices [2,3]. But vulnerability is also introduced by this interconnection, especially in terms of security and privacy. Malware aimed at IoT devices has grown more complex, frequently hiding its existence to avoid detection and jeopardize user privacy. Because such malware can mask its operations and avoid detection by conventional approaches, it presents a considerable difficulty to identify [4,5]. Wide Residual Networks (WRNs) are a promising method in computer vision and machine learning because of its state-of-the-art performance and capacity to handle complex data. By using their deep learning capabilities to examine complex patterns and anomalies in device activity, WRNs can be applied to the detection of obfuscated malware in IoT devices, improving detection accuracy and reliability [6].

Further, a paradigm shift in the strategies used by malicious actors to penetrate and compromise IoT networks has occurred with the introduction of disguised malware [7,8]. In contrast to conventional malware, which frequently uses static signatures for detection, obfuscated malware uses advanced evasion tactics to hide its presence from security measures and obfuscate its code [9,10]. Because of this dynamic nature, it is very challenging for traditional detection systems to recognize and successfully neutralize these elusive threats [11,12]. As a result, protecting the confidentiality and integrity of IoT devices requires the creation of sophisticated detection systems that can distinguish obfuscated malware from the large volume of network traffic [13,14]. Moreover, it is impossible to overestimate the importance of tackling the problem of obfuscated malware detection

within the IoT ecosystem [15,16]. Given how many linked devices are in use in every aspect of contemporary life, a successful malware assault might have far-reaching effects that go well beyond simple data breaches. IoT technologies are everywhere, from wearables and smart homes to vital infrastructure and healthcare systems, making them easy targets for criminal exploitation [17,18]. In order to protect user privacy and lessen the potentially disastrous effects of cyberattacks, it is crucial to strengthen IoT security measures against the constantly changing threat landscape of obfuscated malware.

In addition to our research uses the CIC-Malmem-2022 OMM dataset, an extensive collection of network traffic data that has been specially selected for the analysis of obfuscated malware, to meet this difficult issue [19,20]. We provide a unique method based on the use of Wide Residual Network architecture for the detection of obfuscated malware in IoT environments, utilizing the power of ML and deep neural networks. In contrast to conventional detection techniques that depend on pre-established signatures or heuristics, our method provides a proactive and flexible solution that can recognize malware that has been hidden by looking for underlying patterns and irregularities in network traffic data [21,22].

The use of Wide Residual Network architecture is justified by its innate ability to manage the complex, high-dimensional data that is characteristic of IoT network traffic. WRN models can identify minute variations that point to disguised malware activity by utilizing deep learning techniques to extract complex features and correlations within the data [22,23]. Furthermore, WRN architecture's inherent scalability and resilience make it a good choice for implementation in IoT scenarios with limited resources, where computational efficiency is crucial.

Further, for advanced malware assaults, especially obfuscated ones, have made IoT devices more vulnerable. To address this urgent issue, substantial research has been done on methods to detect and mitigate such attacks. To identify malicious behavior, these efforts have mostly used machine learning models like CNNs. Due to malware evolution, some approaches have shown encouraging results in controlled conditions but are uncertain in real-world scenarios. In this paper, we suggest using WRNs to improve IoT device virus detection [24,25]. Our solution uses WRN depth and width to capture nuanced malware behavior details, unlike static feature extraction and predetermined criteria. Our technique trains on a diverse dataset of benign and disguised malware samples to obtain strong detection performance across IoT contexts [26–28]. This study improves on machine learning-based malware detection research and addresses obfuscation issues.

Essentially, our work aims to close the gap between the increasing prevalence of IoT devices and the need to protect user privacy from the ubiquitous threat of malware that has been obfuscated. Our goal is to provide IoT stakeholders with a proactive security mechanism that can defeat even the most complex malware invasions by utilizing the capabilities of WRN technology. We want to show the effectiveness and feasibility of our suggested method in strengthening the security posture of IoT ecosystems and protecting user privacy in an increasingly connected world through empirical evaluation and comparative research.

Above figure illustrates a thorough strategy for enhancing the security of networked IoT devices using new approaches for detecting malware. The framework is specifically built to efficiently manage the complexities of IoT data, while also guaranteeing strong detection capabilities through the utilization of wide residual networks (WRNs). Moreover, the CIC-MALMEM-2022 dataset, which is an essential element of the system, undergoes meticulous preprocessing procedures. Firstly, the dataset undergoes meticulous data cleansing to eliminate inconsistencies and inaccuracies. Normalization is a process that brings data to a common scale, making it easier to compare [29–31]. On the other

hand, encoding is a method used to convert categorical variables into a format that can be analyzed effectively. Imputation is a process that replaces missing values in order to preserve the integrity of a dataset, while feature selection is used to identify the most important qualities for detecting malware. Techniques for balancing address the problem of class imbalance in order to mitigate bias towards dominant classes, which is essential for preserving the effectiveness of the detection model. After undergoing preprocessing, the dataset is divided into separate training and testing sets in order to construct and evaluate the model. Validation is crucial to guarantee that the model's performance can effectively apply to new data, which is vital in real-world situations where IoT surroundings are always changing. The utilization of WRNs as the principal model architecture is essential due to its capacity to effectively manage the intricate data structures inherent in IoT networks. These networks utilize residual learning to efficiently train deeper models, catching subtle patterns that are indicative of virus actions. In addition, the evaluation measures used consist of accuracy, precision, recall, F1-score, and examination of the confusion matrix. These metrics offer a thorough evaluation of the model's capacity to precisely identify instances of malware in various IoT device scenarios. The framework greatly improves the security of IoT ecosystems by including advanced methodology and rigorous evaluation criteria. It enhances the security of networked devices by protecting them from developing malware threats and also helps to strengthen the overall resilience of IoT systems. Fig. 1 depicts a complex structure that utilizes WRNs and rigorous data processing approaches to enhance IoT security by effectively detecting malware.

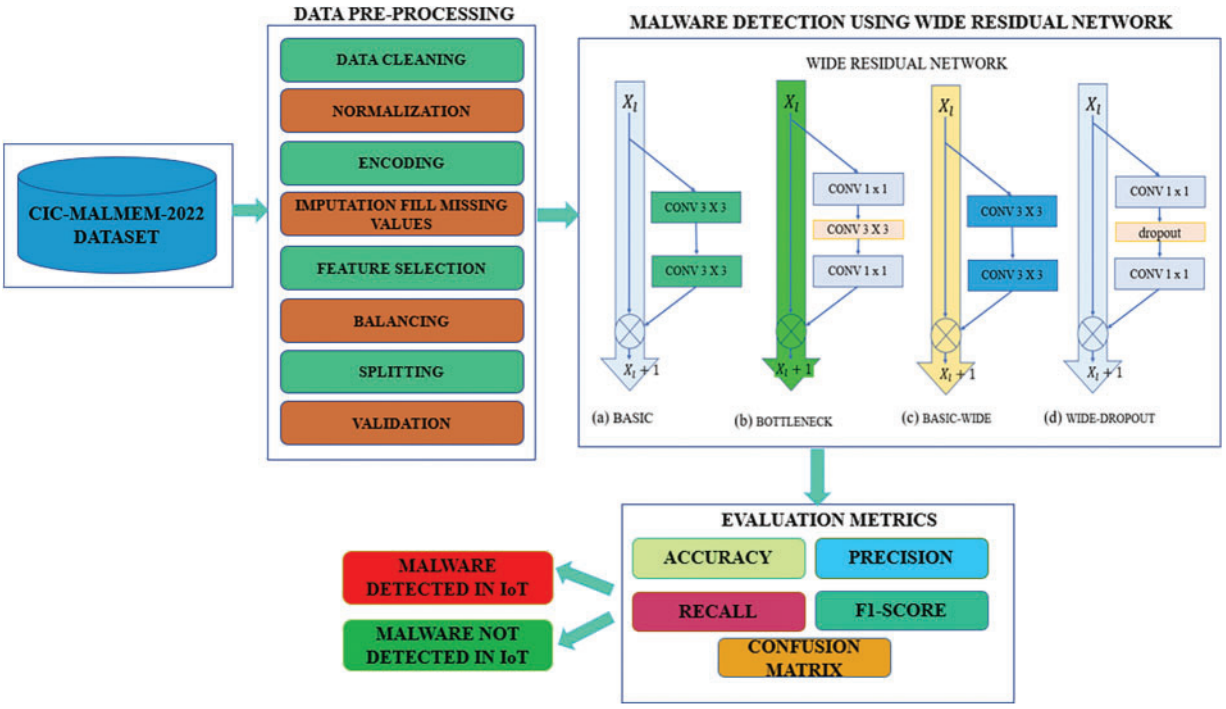

Figure 1: Framework for detecting malware using wide residual network in IoT devices

The purpose of our research is to enhance the current knowledge base by focusing on the following objectives:

1. Introduces a pioneering WRN methodology tailored for detecting concealed malware within IoT devices. This novel approach represents a significant advancement in cybersecurity, particularly addressing the challenges posed by obfuscated malware in IoT environments.
2. Achieves remarkable detection accuracy in identifying obscured malware samples. This exceptional precision underscores the effectiveness of the proposed WRN-based detection model, capable of discerning harmful code even when concealed through obfuscation techniques.
3. Focuses on bolstering the security of IoT ecosystems by implementing proactive malware detection mechanisms at the network edge, where IoT devices operate. This proactive approach mitigates the risk of malware infiltration, thereby fortifying the overall security posture of IoT deployments.
4. Prioritizes safeguarding user privacy within IoT contexts by identifying and neutralizing potential malware threats that may compromise personal data.
5. By utilizing deep neural networks, advanced machine learning techniques—in particular, WRNs (wide residual networks)—showcase their effectiveness in cybersecurity. This invention makes a substantial contribution to the creation of strong defenses that can successfully fend off the constantly changing dangers posed by malware. Through the improvement of cybersecurity resilience in networked environments, these strategies provide enhanced defense against advanced cyberattacks.

In order to accomplish these goals, we utilize the MalMemAnalysis dataset, which is a comprehensive collection of obfuscated malware samples linked to privacy breaches on IoT devices. Through the examination of this dataset, our objective is to get a more profound comprehension of the tactics utilized by malware creators to avoid detection and undermine user privacy.

In addition, Section 2 explores prior study on the subject and dives into pertinent literature. Section 3 then goes on to explain the data collection process, while Section 4 describes the chosen approach. The total results are shown in Section 5 and are followed by a thorough explanation in Section 6. Section 7 provides a summary of the major conclusions and ideas from the review.

2 Literature Review

With the growing use of Internet of Things (IoT) devices in the current technological environment, there has been an unprecedented increase in the level of connection and convenience. On the other hand, this development raises a big concern: the potential breach of privacy and security that may occur as a consequence of the presence of hidden dangerous software. In light of these challenges, researchers have been actively studying novel techniques to identify and limit the hazards that are presented by malware that pretends to be something else on Internet of Things devices.

Recent research has investigated several datasets and machine learning algorithms to identify malware in IoT devices. The study described in Reference [7] specifically examines the CIC-Malmem-2022 OMM dataset. The researchers utilize a model based on a combination of Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BiLSTM). Additionally, they incorporate the CompactCBL and RobustCBL techniques. Nevertheless, this study is restricted by certain constraints. Firstly, the validation of the system is limited to particular IoT devices, which restricts its capacity to be used with a wide variety of devices that are important for real-world deployment scenarios. Furthermore, the issue of scalability has not been well investigated, which could potentially hinder its efficacy in bigger IoT networks. In addition, the model's performance comparison is limited to specific datasets and does not include a full evaluation *vs.* the most advanced

approaches available. Notwithstanding these limitations, it attains commendable levels of accuracy in detecting rates, specifically 72.6% and 71.42%.

Similarly, Reference [10] uses the same dataset but applies the Pearson correlation coefficient for detecting malware. Although this approach achieves a multi-class detection accuracy of 77.5%, its generalizability is limited due to specific feature engineering decisions. Furthermore, the absence of a comparison with the most advanced techniques raises doubts about its usefulness in the present scenario of malware detection. The selection of the dataset may introduce potential biases, which in turn, undermine the reliability and strength of the conclusions. Further, Reference [13] differs by using a malware dataset that contains obfuscated string patterns and uses layered RNNs and CNNs to extract features. Although it has achieved a remarkable detection accuracy of 97.7%, this method encounters difficulties in accurately depicting real-world situations because it relies on the presence of obfuscated patterns. Furthermore, the reliability of its performance in diverse real-world conditions is questionable, which could restrict its dependability beyond controlled settings. On the other hand, Reference [18] examines the VirusShare malware repository and utilizes the Random Forest approach to achieve a detection accuracy of 89.8%. Nevertheless, this approach presupposes constraints on the attackers' capacity to obscure malware while maintaining its functionality, and the outcomes may be influenced by biases in the representativeness of the dataset.

In Reference [21], the CIC-Malmem-2022 OMM dataset is examined, and Naïve Bayes and KNN models are used to detect malware. The accuracy achieved by the Naïve Bayes model is 92%, while the KNN model achieves an accuracy of 95%. However, the efficiency of this approach in dealing with new and complex methods of hiding information and its capacity to be applied to other types of malware have not been sufficiently discussed, which presents difficulties in its practical use. Moreover, Reference [24] utilizes deep belief networks to classify malware, with a detection accuracy of 95.8%. Nevertheless, the limits of this approach lie in its dependence on precisely depicting the behavior of malware families and capturing their behavioral changes.

Likewise, given studies [26,28,32,33] provide distinct perspectives by employing diverse models and datasets. However, they face certain obstacles, including reliance on certain dataset characteristics, potential overfitting, and restricted applicability to real-world situations.

Unlike the current approaches, our methodology presents a new framework for identifying malware in IoT environments. This is achieved by utilizing WRNs that are specifically designed for this purpose. WRNs are selected based on their more complex structures, which have the ability to capture nuanced characteristics in comparison to simpler models such as CNNs or standard neural networks. The richness of our model allows it to reveal more intricate patterns in the behavior of malware, which could potentially improve the ability to detect malicious activity across a wide range of IoT devices [34,35]. Further, our solution tackles scalability difficulties by enhancing the WRN architecture through the utilization of techniques like transfer learning and IoT-specific data augmentation. This approach guarantees that our model retains a high level of detection accuracy while successfully adapting to bigger IoT networks, which is a crucial improvement compared to past research.

In addition, our validation system incorporates a wide range of IoT devices, guaranteeing strong performance across various device categories and real-world deployment situations. Our thorough validation technique differs from earlier research that frequently lacked diversity in device validation, therefore enhancing the reliability and applicability of our findings. Furthermore, our methodology strives to effectively apply to new and changing malware strains by utilizing the potential of WRNs to immediately acquire intricate characteristics from unprocessed data [36,37]. This decreases dependence

on manually designed characteristics and prejudiced dataset choices, thus improving flexibility and generalizability [38].

Finally, our approach is subjected to thorough performance benchmarking, comparing it not just to typical machine learning models but also to cutting-edge deep learning approaches employed in malware detection. This guarantees a comprehensive evaluation of the efficiency of our methodology and establishes it as a possible standard in the field.

Our methodology is a notable improvement in detecting malware for IoT devices, as it effectively tackles important challenges such as scalability, generalizability, and adaptation to new malware strains. Our goal is to set a new benchmark in malware detection accuracy and reliability for IoT environments by utilizing vast residual networks. This will help us address the problems of protecting IoT ecosystems from emerging cyber threats.

A thorough summary of previous references is included in Table 1 of the paper, along with information on the datasets used, the approach used in comparable investigations, any constraints found, and the relevant outcomes. This well-organized review provides a useful summary of previous studies, pointing out areas of weakness and future directions for malware detection and IoT security research.

Table 1: List of past references including datasets, methodology, limitations, and results

References	Dataset	Methodology	Limitations	Results
[7]	CIC-Malmem-2022 OMM dataset is used	CNN-BiLSTM based Model, CompactCBL and RobustCBL	Limited validation on diverse IoT devices; lacks real-world deployment assessment; scalability challenges unexplored; performance comparison limited to specific datasets.	Detection rate accuracy of 72.6% and 71.42%
[10]	CIC-Malmem-2022 OMM dataset is used	Pearson Correlation Coefficient	Limited generalizability due to specific feature engineering and model choices. Lack of comparison with state-of-the-art methods. Potential bias in dataset selection.	Multi-class detection accuracy is 77.5%
[13]	Malware dataset containing obfuscated string patterns	Stacked RNNs and CNNs applied for feature extraction	Dependency on available obfuscated patterns, potential bias, and performance variations in real-world scenarios.	Detection accuracy is 97.7%

(Continued)

Table 1 (continued)

References	Dataset	Methodology	Limitations	Results
[18]	VirusShare malware repository	Random Forest technique	Assumes attackers cannot obfuscate crucial parts without affecting malware functionality; relies on VirusShare dataset representativeness.	It has a detection accuracy of 89.8%
[21]	CIC-Malmem-2022 OMM dataset is used	Naïve Bayes, KNN	Framework effectiveness may vary with evolving obfuscation techniques; dataset simulation might not capture all real-world scenarios.	These models have detection accuracies of 92% and 95%
[24]	Malware classification	Deep belief networks	Dependency graphs may not capture all variations in malware behavior; effectiveness reliant on accurate representation of family behavior.	This model has an accuracy detection of 95.8%
[26]	CIC-MalMem-2022 dataset	RF, MLP, KNN	Dependency on available memory dump data, potential overfitting, and the challenge of keeping up with evolving malware obfuscation techniques.	These models have detection accuracy of 93.95%, 82.11% and 91.21%
[28]	CIC-MalMem-2022 dataset	DNN, Ensemble-Learning	Dependency on available memory dump data, potential bias from oversampling techniques, and challenges in keeping classifiers updated against evolving malware tactics.	Accuracy detection is 67.9% and for Ensemble Classifier it is 66.0%. Whereas, AUC is 93.4%
[32]	CIC-MalMem-2022 dataset	CNN	Dependency on available dataset, potential bias towards known malware families, and challenges in generalizing results to new malware variants.	Accuracy detection for this dataset is 75.0%

(Continued)

Table 1 (continued)

References	Dataset	Methodology	Limitations	Results
[33]	CIC-MalMem-2022 dataset	MLP, Naïve Bayes	Dependency on dataset quality, potential overfitting, and challenges in generalizing results to real-world scenarios.	Detection accuracy of 97.67% and 98.42%

To summarize, studies on safeguarding privacy on IoT devices by detecting obfuscated malware highlight the effectiveness of machine learning methods. Nevertheless, these techniques encounter obstacles such as bias in the dataset, restricted applicability to different contexts, and difficulties in scaling up. Subsequent investigations should strive to mitigate these constraints in order to augment the efficacy of malware detection on IoT devices, thus enhancing user privacy and security.

3 Experimental Dataset

3.1 MalMemAnalysis Datasets

The efficacy of ML algorithms is significantly influenced by the quality of the training datasets. A major barrier to the progress of detection algorithms is the absence of a standardized dataset for privacy threat detection. We can employ a diverse range of datasets to examine machine learning techniques in various domains, such as biomedical, business, and language translation. Security and privacy issues are the main factors that contribute to the limited availability of attack detection datasets. Furthermore, most publicly available datasets are outdated, often anonymized, and may not correctly represent current network vulnerabilities. To address these issues and ensure the effectiveness of the proposed machine learning models, we employ a rigorous approach by meticulously replicating real-world malware that has been encrypted using the MalMemAnalysis dataset [7,10,13]. In debug mode, we utilize the memory dump approach to handle this dataset. The memory dump procedure is hidden within the memory dumps. Fig. 2 depicts the distribution of four classes within the data. The benign class comprises 29,298 samples, whereas the spyware, Trojan, and ransomware classes each include fewer than 10,000 samples.

Containment strategies are used by obfuscated malware to evade detection and elimination. The obfuscated malware dataset's objective is to assess methods for locating obfuscated malware in memory. Realistic results were the primary goal in creating the dataset. This was made possible by the use of widely used malware that exists in the actual world. A balanced dataset that includes Trojan horses, ransomware, and spyware can be used to assess an obfuscated malware detection system. Real-world scenarios are faithfully replicated in the dataset. Fig. 3 displays a dataset divided by malware family types to aid in understanding the prevalence and characteristics that are crucial for IoT device security and malware detection strategies.

With an equitable allocation of harmless and harmful memory dumps. According to Table 2, out of the total of 58,596 records, 29,298 are classified as benign and 29,298 are classified as malicious.

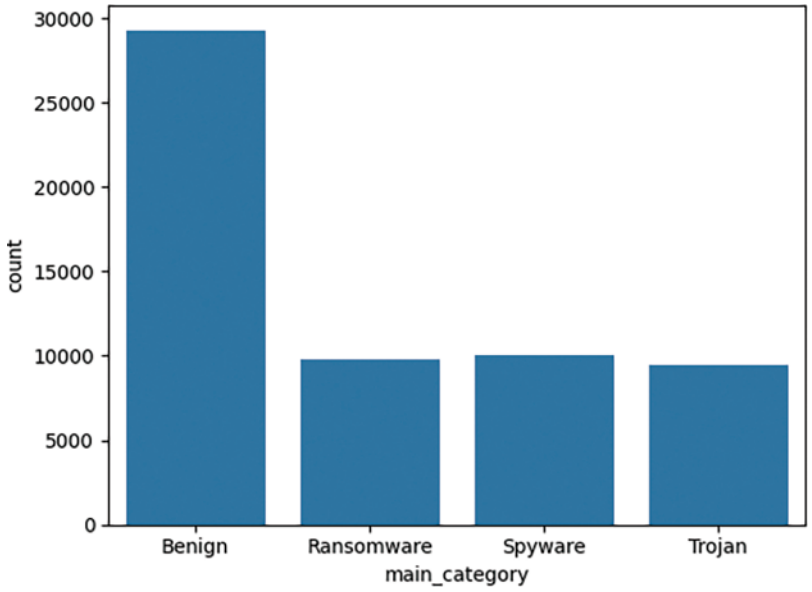

Figure 2: Four classes distribution in dataset

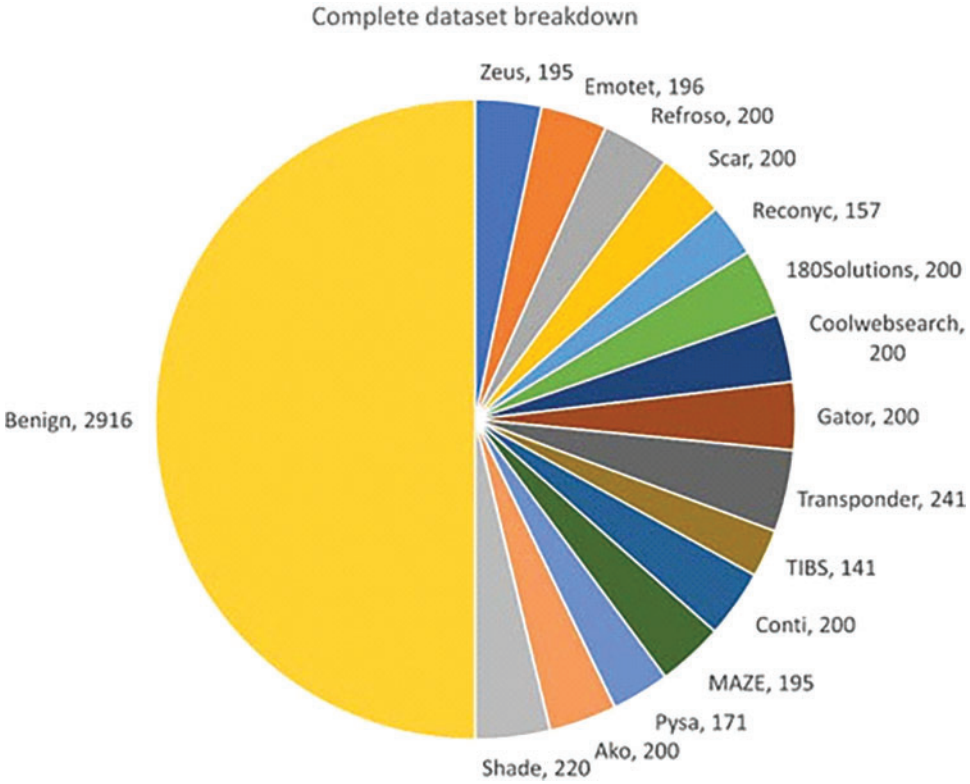

Figure 3: Complete dataset breakdown by malware family of each category

Table 2: Malware category and family distribution

Malware category	Malware family	Count
Trojan horse	Zeous	195
	Emotet	296
	Refroso	200
	Scar	200
	Reconyc	157
Spyware	180 solutions	200
	Cool Web Search	200
	Gatar	200
	Transponder	241
	TIBS	141
Ransomware	Conti	200
	Maze	195
	Pysa	171
	Ako	200
	Shade	200

The interception of harmful communication was facilitated through the utilization of malicious memory dumps, resulting in the collection of 2916 malware samples from Virus Total. The collected samples, as depicted in Fig. 3, exhibit a diverse range of malware categories, including Trojan horses, spyware, and ransomware. In a similar vein, it was shown that users exhibited usual behavior when employing multiple virtual machine applications to generate harmless memory dumps.

3.2 Malware Memory Analysis|CIC-MalMem-2022 Dataset Analysis

In this section, we will analyze various aspects of IoT network traffic, each of which presents distinct privacy risks. The aim of this work is to comprehensively examine the aforementioned features and develop a robust machine learning model for the purpose of detecting these potential risks. The primary attributes encompass callbacks, servers, handles, and processes. The following elucidates their correlation with various forms of assaults.

The scatter plot depicted in Fig. 4 showcases the fluctuations in specific attributes, such as threads and handlers, in relation to different types of assaults. The data reveals that ransomware attacks constitute the predominant form of attacks, as denoted by the color orange. As denoted by the color blue, there exists a minuscule assemblage of these characteristics where vehicular movement is forbidden. This approach will facilitate the training of our model to accurately identify instances of ransomware, spyware, or Trojan infections whenever the threshold values are present within this certain cluster. The protection of IoT detection will be ensured through the implementation of measures aimed at preventing the ingress of such traffic and exclusively permitting valid or forbidden traffic.

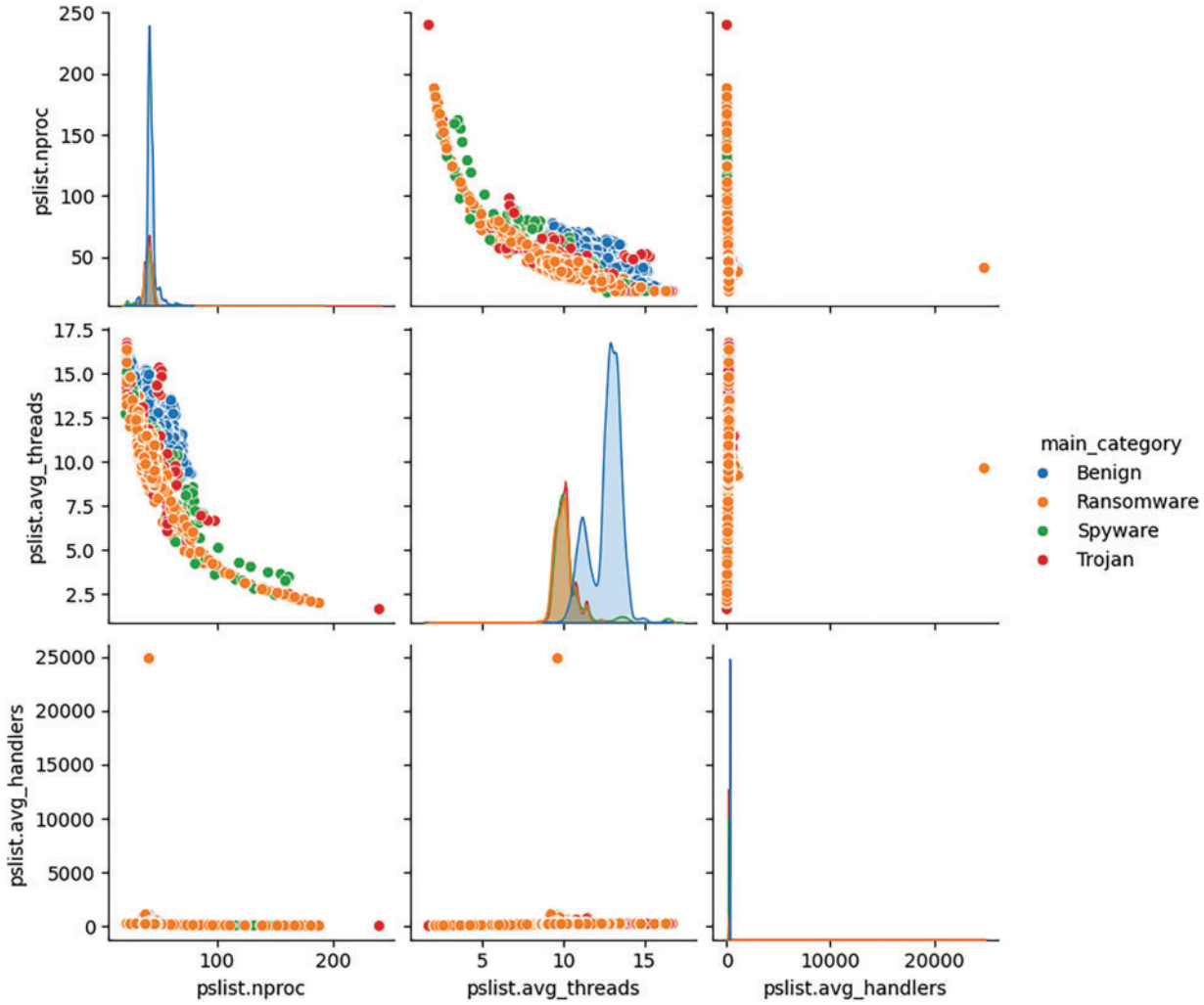

Figure 4: Different features distribution with respect to malware category

In addition, we analyzed the fundamental attributes and their impact solely on the sort of assault. The overall number of characteristics is approximately fifty-five. A significant portion of these features possess a singular distinct value as shown in Fig. 5a, while Fig. 5b exhibits continuous numeric values, category values, or a combination of both.

As depicted in Fig. 5, certain features exhibit limitations within a defined range, whilst others have a far broader range. This statement elucidates the significance of data quality and the prevalence of assaults targeting IoT systems. These characteristics unveil the vulnerability, indicating that a feature will not impact the training of the model unless its value is associated with the assault in either a positive or negative manner. Fig. 6 illustrates the essential attributes and their corresponding significance in relation to the type of attack.

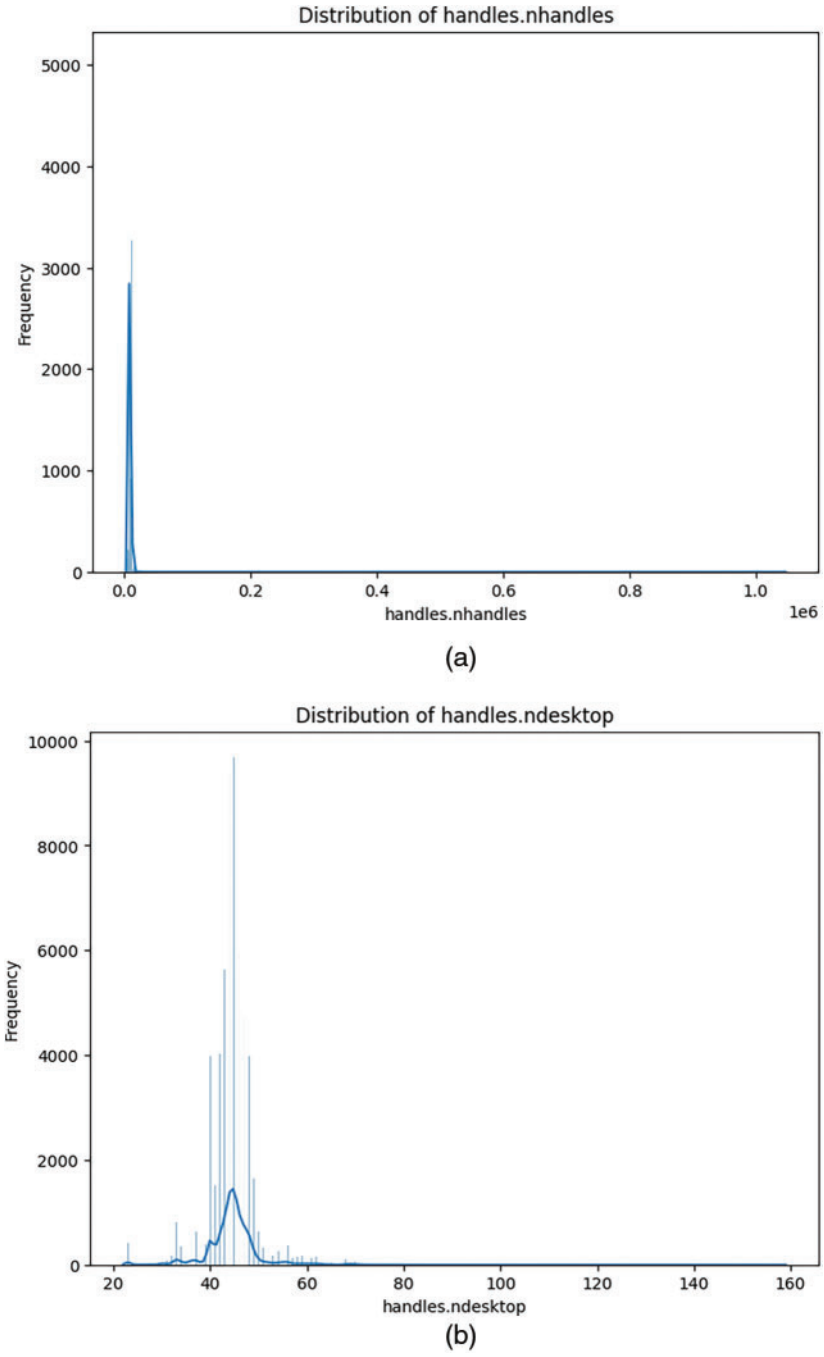

Figure 5: Different continuous, categorical features distribution (a) Feature distribution with singular distinct value, (b) Features distribution with far broader range

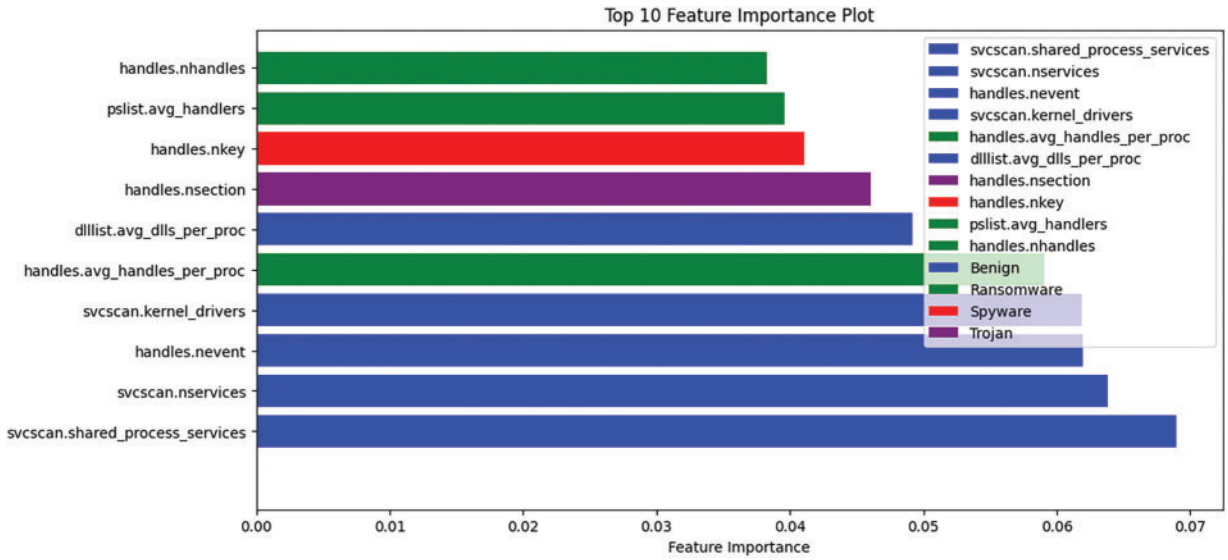

Figure 6: Important features by malware class with respect to impact on prediction

It is evident that specific characteristics exhibited in blue block traffic, whereas characteristics exhibited in green—approximately three characteristics—are linked to ransomware, suggesting their active involvement in such assaults. The initiation of such attacks is attributed to a singular feature present in spyware and trojans. Now, we will construct a robust deep learning model that will be trained using these attributes to predict different types of privacy breaches in IoT systems.

4 Proposed Methodology

4.1 IoT Threats Vulnerabilities

The IoT enables a diverse array of applications that significantly improve individuals' quality of life. Notwithstanding the appealing objective, the preservation of users' privacy on these IoT devices is a significant apprehension. IoT devices are vulnerable because of the rapid spread of the IoT, their use in critical infrastructure and commerce, and their susceptibility to viruses and hackers [7]. The main goal of our project is to identify malicious software, such as Trojan horses, ransomware, and spyware, on IoT devices by utilizing machine learning models. Our proposal involves utilizing machine learning techniques to identify privacy breaches and generate test patterns for the IoT. We have the capability to train the machine learning model to accurately forecast behavioral architecture. We conducted tests and analysis on the "MalMemAnalysis" datasets, which simulate obfuscated malware related to privacy in real-world scenarios [13,18,21].

Data from individual users is collected, as depicted in Fig. 7. In addition, in certain situations, individual user data may include confidential and personal information about the user, such as email addresses and passwords for online accounts, private photos, sensitive contacts and phone numbers, contracts, and other important documents, payment card numbers, and other financial information.

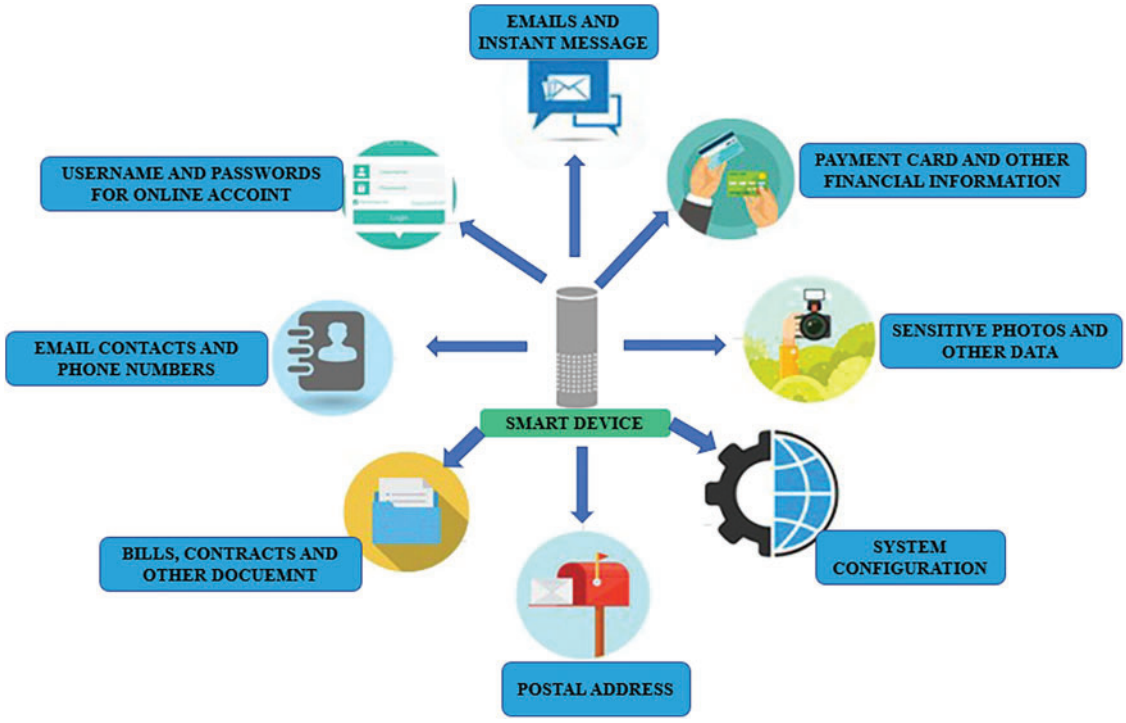

Figure 7: Types of information that cyber criminals can gain through different IoT privacy and security attacks

4.1.1 IoT Architecture

The most well-liked and commonly recognized three-layer design is shown in Fig. 8, despite the fact that there isn't a single, widely accepted model for IoT architecture.

It contains the following layers:

- **Perception layer:** The physical layer of the architecture is known as the perception layer. The sensors and associated equipment are used to gather varying amounts of data, depending on the project's requirements. Drives, sensors, and edge systems interacting with their surroundings are examples of these.
- **Network layer:** The information gathered by each of these devices is transferred and processed by the network layer. These devices are linked to network devices, servers, and other intelligent objects. Its other responsibility is data transport.
- **Application layer:** There is communication between the user and the application layer. It is in charge of providing the user with particular application services. An example of this might be in an intelligent house, where users may turn on a coffee maker by tapping an app.

It is now feasible to launch attacks that compromise the privacy of an Internet of Things system in a number of ways, including ransomware, Trojan, and spyware attacks that result in data manipulation, database reconstruction, and model theft. The proposed methodology requires building a deep learning model for a vast residual network in order to detect these types of attacks. The steps for the suggested methodology are as follows:

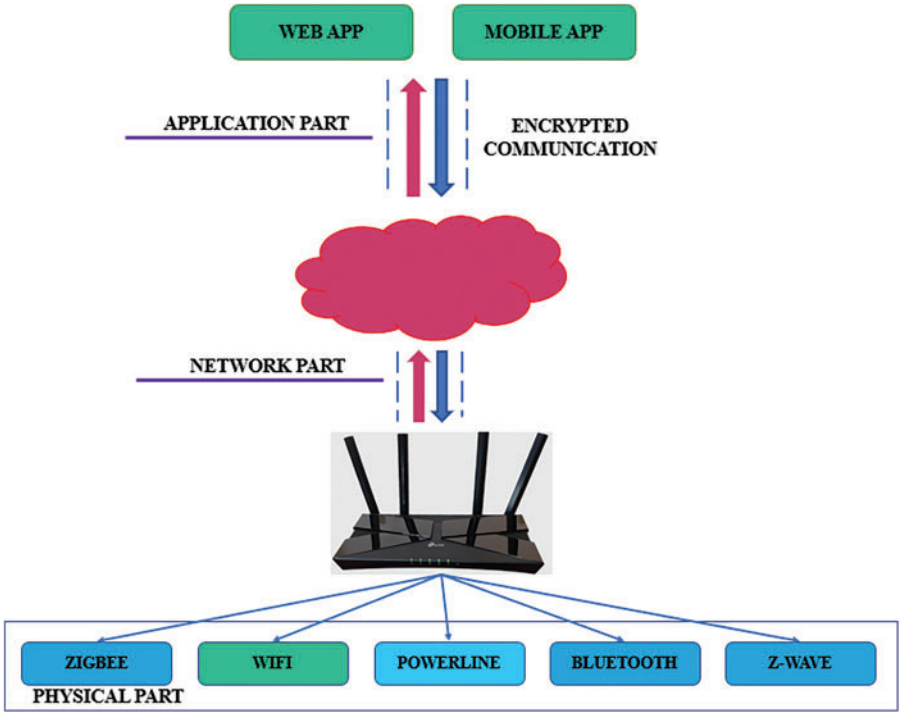

Figure 8: Three layers of IoT architecture

4.1.2 Model Architecture Design WRN

- Create the WRN architecture specifically for detecting malware in IoT devices.
- Adjust the WRN's width and depth according to the dataset's complexity and the processing power at your disposal.
- To address the vanishing gradient issue and facilitate the training of deeper networks, make use of residual connections.
- To expedite training and enhance the model's capacity for generalization, include batch normalization layers.
- To avoid overfitting and improve the model's resistance to noise, add dropout layers.

4.1.3 Model Training

- Divide the preprocessed dataset into test, validation, and training sets in the proper proportions.
- Utilizing the training data, train the WRN model by optimizing the selected objective function (such as cross-entropy loss) using an appropriate optimizer.
- Make use of strategies like early stopping and learning rate scheduling to enhance convergence and avoid overfitting.
- During training, keep an eye on the model's performance on the validation set to make necessary hyperparameter adjustments and identify any problems.

4.1.4 Model Evaluation

- Analyze the performance of the trained WRN model with the test set to see how well it detects malware that has been disguised.

- To measure the model's performance, compute evaluation metrics including F1-score, ROC-AUC, accuracy, precision, and recall.
- Examine the confusion matrix of the model to learn about its advantages and disadvantages in identifying malicious and safe samples.
- Perform more experiments to evaluate the resilience of the model against adversarial assaults and differences in IoT device settings.

4.1.5 Model Optimization and Fine-Tuning

- Optimize hyperparameters and fine-tune the WRN model using the knowledge gathered from the evaluation phase.
- Investigate methods like data augmentation and transfer learning to improve the model's functionality and capacity for generalization.
- To attain the targeted degree of precision and dependability in identifying obfuscated malware in Internet of Things devices, repeat the training and assessment procedures.

This study is using a novel technique with the chosen model, which is a model based on vast residual networks.

4.2 Pseudo Algorithm for Our Proposed Methodology

The proposed methodology uses the WRN to detect obfuscated malware, hence addressing the important issue of privacy preservation on IoT devices [30,31,34]. By identifying and mitigating potential malware dangers, the algorithm's main goal is to safeguard private data and preserve the dependability of IoT devices. Preparing the input data for convolutional layers and splitting the dataset into distinct training and testing sets are both part of the preprocessing step. After that, the WRN architecture is developed, launched, and constructed by combining residual connections and wide residual blocks. Adaptive dropout regularization is used to achieve model resilience, while normal initialization is used to initialize convolutional kernels [35,37,38]. Appropriate parameters are inserted into the model for training purposes. Following training, testing data is used to evaluate the model's performance, and a collection of comprehensive performance metrics is produced for analytical purposes. This methodology represents a methodical way to protect user privacy and improve IoT security in the face of constantly evolving cybersecurity threats.

4.2.1 Input

- Dataset D = {(x_1, y_1), (x_2, y_2),..., (x_n, y_n)} of malware samples and IoT device attributes
- Parameters: depth (D), width (W), dropout rate (p)

4.2.2 Preprocessing

- Split the dataset into training (D_train) and testing (D_test) sets
- Encode class labels to binary class matrices: y_train, y_test
- Reshape input data for convolutional layers: X_train_reshaped, X_test_reshaped

4.2.3 Model Architecture

- Define Wide Residual Network (WRN) architecture:
  WRN (D, W) = {Wide Residual Blocks with Residual Connections}
- Initialize with input shape, depth, width, number of classes, and dropout rate:

Initialize Model (input_shape, D, W, num_classes, p)

- Construct wide residual blocks with residual connections
- Wide_Residual_Blocks (input_shape, D, W, p)
- Utilize adaptive dropout regularization
- Adaptive_Dropout_Regularization (p)
- Initialize convolutional kernels using He normal initialization
- He_Normal_Initialization ()
- Compile the model with Adam optimizer, binary crossentropy loss, and accuracy metrics
- Compile_Model (optimizer='Adam', loss='binary_crossentropy', metrics=['accuracy'])

4.2.4 Model Training

- Train the WRN model on the training data
- Train_Model (X_train_reshaped, y_train, batch_size, epochs, validation_split)

4.2.5 Model Evaluation

- Evaluate the trained model on the testing data
- Evaluate_Model (X_test_reshaped, y_test)

4.2.6 Performance Analysis

- Generate classification report, confusion matrix, heatmap, and ROC curve
- Generate_Performance_Metrics ()

4.3 Widening Residual Network (RESNET)

Widening of the residual network (ResNet) permits a shallower network while preserving or enhancing accuracy. Shallower networks facilitate the lowering of layer count.

It's also feasible to work out for shorter periods of time. A higher dropout rate is also investigated. In the field of machine learning, the argument over shallow *vs.* deep networks has lasted for a time. Citations from the literature on circuit complexity theory show that circuits with 10 times as many components in shallow circuits as those with deeper circuits. The designers of residual networks tried to make their blocks as thin as feasible in an attempt to decrease parameter counts and increase depth. To further thin the ResNet blocks, they even included a "bottleneck" block.

Remaining networks can have some disadvantages, as we can see in Fig. 9, even though the residual block with identity mapping makes training very deep networks easier.

The gradient that runs through the network allows it to avoid learning anything during training and does not force it to pass through residual block weights. Therefore, it is feasible that just a small number of blocks learn valuable representations, or that many blocks contribute little to the ultimate aim. The initial phrasing of the problem was limited feature reuse [20,22,25]. The authors suggested that residual blocks be arbitrarily disabled during training in an effort to address this problem. This method can be understood as a particular case of dropout, where each residual block receives a dropout according to its identity scalar weight. The efficiency of this strategy lends credence to the above hypothesis.

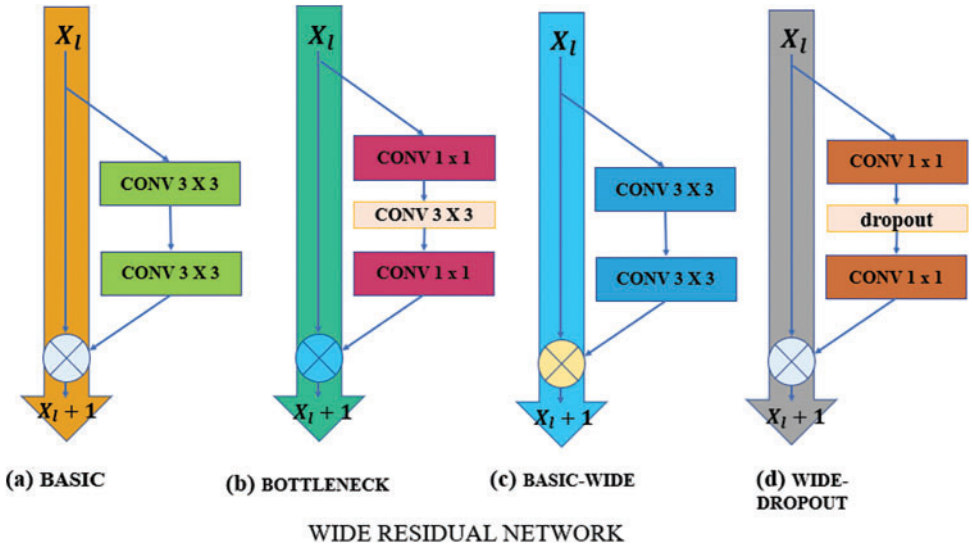

Figure 9: Wide *vs*. different dropout types. (a) BASIC (b) BOTTLENECK (c) BASIC-WIDE (d) WIDE-DROPOUT

We would like to look at regularization techniques when the number of components rises due to widening. For residual networks, batch normalization already exists and results in a regularization effect; however, it requires extensive data augmentation, which is not always possible and something we would like to avoid. We add a dropout layer to each residual block between convolutions and after ReLU to prevent batch normalization in the subsequent residual block and prevent it from overfitting. The issue of diminishing feature reuse in very deep residual networks should be partially resolved by enforcing learning in separate residual blocks.

Let's examine the single linear layer of a neural network. This layer is sometimes called a "linear layer" since it considers the behavior of a linear activation function, f(x) = x. The weighted total of all the inputs is computed to construct the final neuron in this layer (Eq. (1)). The output of this layer is this neuron. For many non-linear networks, this simple mathematical explanation is supported by empirical evidence; however, it is important to keep in mind that model estimate involves minimizing a loss function. The ordinary least squares (OLS) loss.

$$Ln = \frac{1}{2}\left(t - \sum_{i=1}^{n} w_i I_i\right)^2 \tag{1}$$

$$L_D = \frac{1}{2}\left(t - \sum_{i=1}^{n} \delta_i w_i I_i\right)^2 \tag{2}$$

Eq. (1) represents the loss function used in general neural networks, while Eq. (2) corresponds to the loss function specific to dropout networks. The Bernoulli distribution governs the dropout rate, denoted by δ in the dropout network, which has a parameter p. This implies that 0 else if p is the value that δ accepts. While the letter "w" shows the weight of each input, the letter "I" denotes the input in the network. We employ the gradient descent technique for backpropagation during network training. The gradient of the dropout network, which feeds into the regular network, is calculated using Eq. (2):

$$\frac{dL_D}{dw_i} = -t\delta_i I_i + w_i \delta_i^2 I_j^2 + n \sum_{j=1, j\neq i}^{n} w_j \delta_i \delta_j I_i I_j \tag{3}$$

The relationship between the gradients of a regular network and a dropout network is expressed by Eq. (1), where w' and w represent the parameters of the dropout network and the regular network, respectively, and p denotes the dropout probability variable.

$$LD = \frac{1}{2} 2 \left(t - \sum_{i=1}^{n} p_i w_i I_i \right)^2 \tag{4}$$

Calculating Eq. (4)'s derivative:

$$\frac{dL_D}{dw_i} = -tp_i I_i + w_i p_i^2 I_j^2 + \sum_{j=1, j\neq i}^{n} w_j p_i p_j I_i I_j \tag{5}$$

Compute the derivative of the given equation. Continue to the subsequent stage. After performing the calculation of the gradient expectation for the dropout network, the resulting expression is as follows:

$$L\left[\frac{dL_D}{dw_i}\right] = \frac{dL_N}{dw_i} + w_i p_i \left(1 - pi\right) I_i^2 \tag{6}$$

According to Eq. (6), the gradient of the regular neural network, when scaled by p, is equivalent to the expectation of the gradient with dropout, which is denoted as:

$$w' = p \times w \tag{7}$$

Expanding the width of residual networks, together with adjusting their depths, can greatly enhance their effectiveness. This improvement can persist until an excessive quantity of parameters is reached, at which juncture regularization becomes progressively imperative. Notably, networks with a larger size have demonstrated the capacity to acquire similar or superior representations in comparison to networks with a smaller size, even when the number of parameters is the same. This observation indicates that residual networks with very deep depths do not always result in a regularization impact. Moreover, broad networks have the capability to acquire knowledge with a parameter count that is twice as large as that of thin networks. Achieving equivalent learning capacity using thin networks necessitates doubling their depth, a process that frequently proves problematic or impracticable during training. The mathematical representation of a residual block with identity mapping is commonly denoted as:

$$x_l + 1 = x_l + F\left(x_l, w_l\right) \tag{8}$$

where F is a residual function, w_l are block parameters, and $x_l + 1$ and x_l are the input and output of the l-th unit in the network. The blocks that make up a residual network are placed one after the other.

There were two different kinds of blocks in residual networks:

- **Basic:** conv3 × 3-conv3 × 3 is the name of the two consecutive 3 × 3 convolutions with batch normalization and ReLU preceding convolution.
- **Bottleneck:** conv1 × 1-conv3 × 3-conv1 × 1 is the 1 × 1 convolution layer surrounded by dimensionality-reducing and expanding convolution layers of size 3 × 3.

- Now the proposed model has very novel points which can enhance the performance of privacy breaches detection in a very efficient way. Some of the novel features which are main features are of model and are very effective are as follows:
- **Wide Residual Blocks with Residual Connections:** Introduction of residual connections into wide residual blocks. These blocks alleviate the vanishing gradient problem and facilitate information flow through the network, allowing the network to learn sophisticated representations of the input data.
- **Adaptive Dropout Regularization:** Use of adaptive dropout regularization, in which the complexity and learning progress of the network are used to dynamically modify dropout rates. The resilience of the model is increased, and overfitting is avoided by this adaptive method.
- **He Normal Initialization:** Convolutional kernels are initialized using the He normal initializer, which takes into account the receptive field size of neurons to promote convergence and preserve gradient stability during training.

Moreover, these are some of the salient features of our model as shown in Fig. 10 which are very novel as compared to many other deep learning models which utilizes the same CNN, batch normalization layers and their effectiveness is improved by deepening the network but here we enhance effectiveness by widening the network.

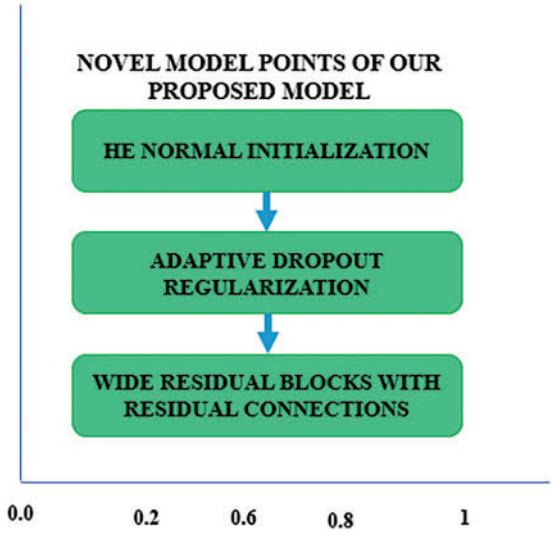

Figure 10: Model novel points

4.4 Model Architecture Design

The WRN has the following architecture to identify obfuscated malware in Internet of Things devices and layer distribution is also shown in Fig. 11.

4.4.1 Input Layer

- The input layer obtains the input data, which comprises properties derived from malware samples and attributes of Internet of Things devices.
- Usually, a convolutional layer is the first layer, then batch normalization and ReLU activation.
- This block extracts low-level features by processing the raw input data.

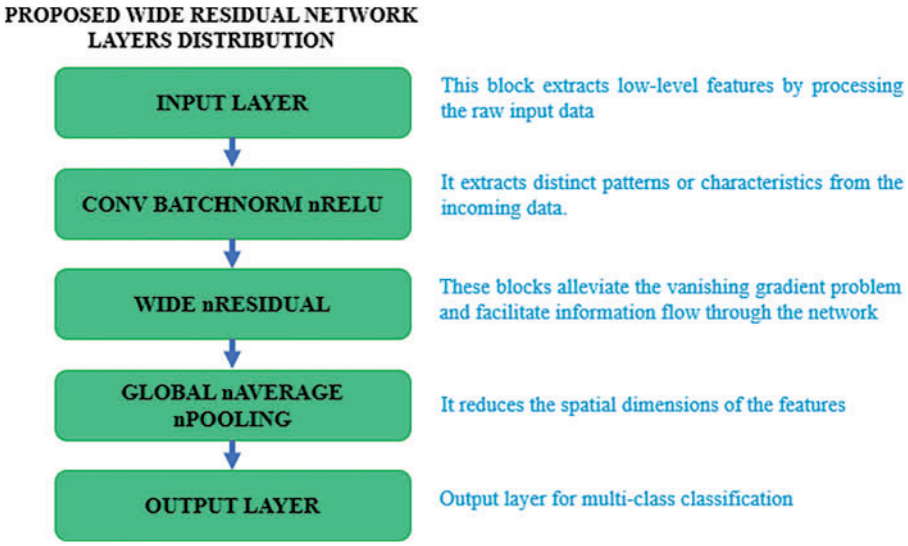

Figure 11: WRN layers distribution

4.4.2 Convolutional Layers

- It filters the input data using a set of sixteen filters.
- The filters are 3 × 3 in size, so they conduct convolutions by sliding over the input data in a 3 × 3 grid.
- Every filter extracts distinct patterns or characteristics from the incoming data.
- A collection of feature maps, each representing the activation of a single filter across the input data, is the convolutional layer's output.

4.4.3 Activation: ReLU

- The activation function that is applied following the convolution operation is called ReLU, or Rectified Linear Unit.
- By setting all negative values to zero and maintaining positive values unaltered, ReLU adds non-linearity to the network.
- This activation function speeds up the training process and aids in the network's ability to recognize intricate patterns in the input.

4.4.4 Batch Normalization

- Prior to the activation function and following the convolution procedure is the use of batch normalization.
- It ensures that the variance is close to one and the mean activation is close to zero by normalizing the activations of each layer.
- By decreasing internal covariate shift, batch normalization contributes to training process stabilization and acceleration.
- Additionally, it serves as a regularizer, enhancing the model's generalization and decreasing the requirement for methods like dropout.

4.4.5 Wide Residual Blocks

- Two convolutional layers, batch normalization, and ReLU activation come next in each block.

- The number of filters used by the first convolutional layer is the same as that of the preceding layer.
- The quantity of filters is doubled by the second convolutional layer.
- A 1×1 convolution is used to modify the input tensor's dimensions for the addition operation if the stride is more than 1.
- The input tensor is added to the second convolutional layer's output tensor by the residual connection.

Following the additional operation, ReLU activation is applied.

Fig. 12 illustrates a single wide residual block, which is accompanied by twelve more wide residual blocks. These blocks are constructed using varying amounts of batch normalization, convolution, and Relu activation layers.

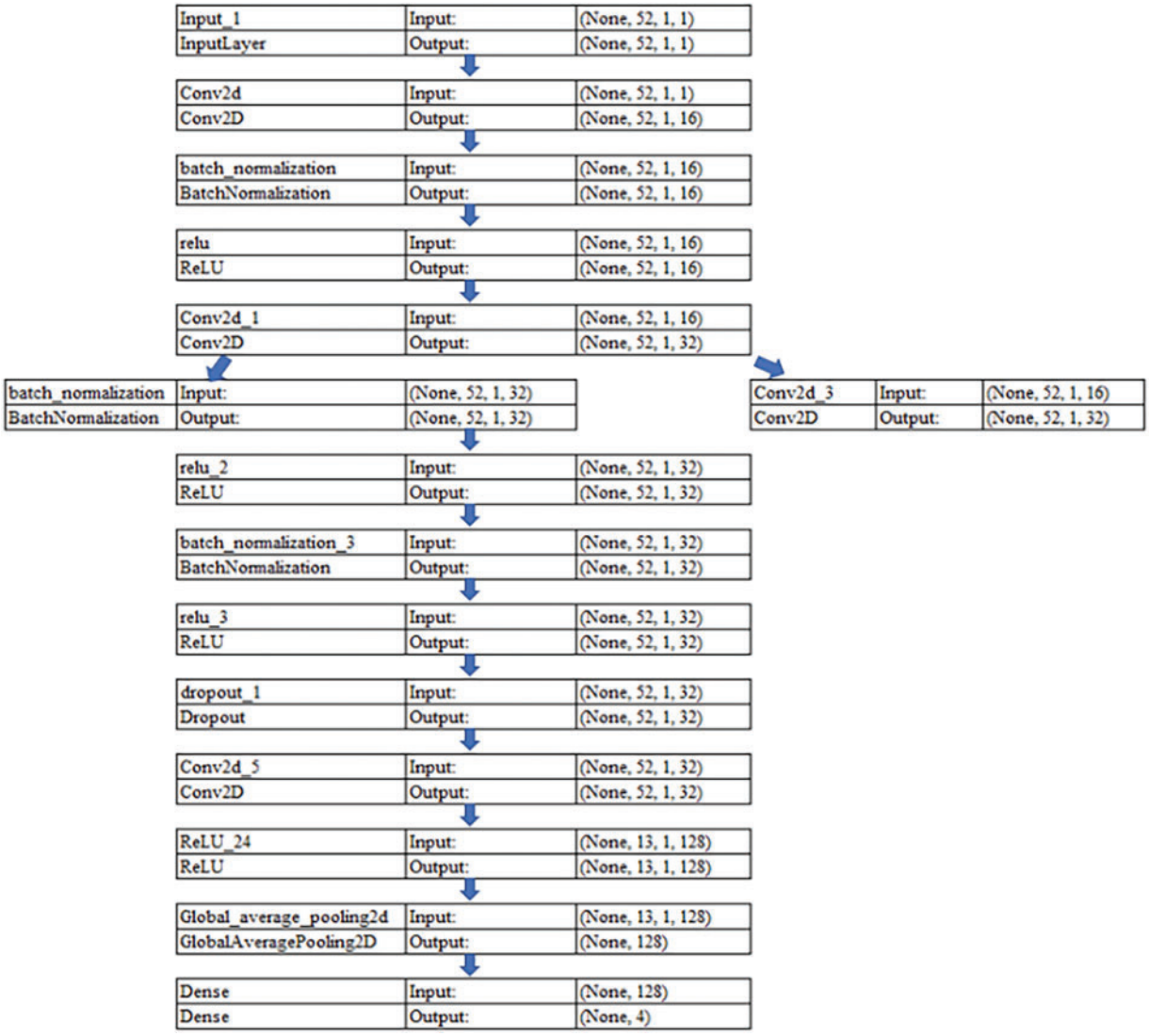

Figure 12: WRN last layers

- Depth: 28.
- Width: 2 (number of filter multiplicative factor).
- Dropout Rate: 0.3 (dropout regularization done optionally after each convolutional layer).
- Count of Blocks: Computed using the network's depth, or 12 Wide Residual Blocks.

4.4.6 Global Average Pooling

- By averaging each feature map, it reduces the spatial dimensions of the feature maps to a single vector, which aids in lowering the number of parameters and avoiding overfitting.

4.4.7 Output Layer

- The density layer uses the SoftMax activation function to create class probabilities for each malware category.
- Units: The number of classes (ascertained by the number of distinct malware categories).
- The likelihood of each class is represented in the SoftMax layer's output, which enables the model to categorize input data into distinct malware groups.

4.4.8 Model Compilation

- Adam is the optimizer.
- Categorical cross entropy is the loss function.
- Accuracy is the metrics.

This architectural design enables the model to accurately classify obfuscated malware in IoT environments by effectively capturing the complex relationships between malware attributes and features of IoT devices.

Below proposed methodology has made notable strides in enhancing privacy security within the realm of IoT by harnessing state-of-the-art machine learning techniques. The introduced approach variants have yielded multiple advancements, as delineated below:

i. **Developing a Novel WRN Approach:** Introduced an innovative WRN technique to detect concealed malware on IoT devices. This approach represents a significant advancement in the field of cybersecurity, particularly in addressing the unique challenges posed by malware specifically designed for IoT environments.
ii. **High Detection Accuracy:** The research conducted yielded remarkable outcomes, as it successfully identified concealed malware samples with an accuracy rate approaching 99%. The observed level of precision demonstrates the efficacy of the proposed WRN-based detection model in detecting harmful code, even in cases where it has been concealed by obfuscation methods.
iii. **Enhancing IoT Security:** Our research focuses on improving the security of IoT ecosystems by specifically targeting malware detection at the network edge, where IoT devices are used. The implementation of this proactive method enhances the overall security of IoT deployments by mitigating the probability of malware infiltration and subsequent breaches of privacy.
iv. **Preserving User Privacy:** The significance of protecting user privacy in IoT contexts, where linked devices gather and handle personal data, is underscored by our research findings. Our objective is to safeguard user privacy by identifying and eliminating malware threats that have the potential to illicitly access personal information.
v. **Advancing Machine Learning (AML) Techniques:** Our research showcases the potential of advanced machine learning techniques in cybersecurity applications by utilizing WRNs. Deep

neural networks have the potential to enhance malware detection capabilities, hence facilitating the development of more robust defensive systems capable of effectively countering ever-evolving threats.

vi. **Contributing to Cybersecurity Research:** Contributing to the Field of Cybersecurity Research: The issue of ensuring the security of IoT devices is increasingly pressing, and our research endeavors aim to address this matter with the intention of benefiting the broader community of cybersecurity researchers. Our objective is to augment the overall understanding of cybersecurity issues in networked systems through the identification and mitigation of malware threats inside IoT environments.

vii. **Potential for Real-World Implementation:** The findings of our study are relevant for cybersecurity professionals, industry professionals, and manufacturers of IoT devices. The integration of the developed detection model into security solutions can effectively mitigate malware attacks on IoT devices, hence safeguarding user privacy and enhancing trust in IoT technology.

Fig. 13 illustrates the fundamental contributions of our work. Furthermore, taking all factors into account, the primary contributions of our research encompass its innovative methodology for identifying malware, its notable precision in detecting such threats, its focus on enhancing security in the IoT and safeguarding user privacy, and its potential for practical use in mitigating cybersecurity vulnerabilities inside IoT environments.

Figure 13: Core contributions of our study

4.5 Hypothesis and Limitations of the Developed Method

Hypothesis:

i. **Effectiveness of WRN for Malware Detection:** Our research hypothesis is that Wide Residual Networks (WRN) can detect obfuscated malware on IoT devices. Since WRNs can manage

deep architectures with residual connections, they can capture sophisticated malware patterns even when obfuscated.

ii. **Enhanced Privacy Protection:** We hypothesize that WRN-based machine learning models can improve IoT privacy. These models can detect and prevent privacy-compromising Trojan horses, ransomware, and spyware, according to this idea.

iii. **Generalization across IoT Architectures:** We expect the methodology to work across many IoT architectures and scenarios. WRNs' scalability and versatility allow them to handle IoT deployments' diverse data and device configurations.

iv. **Reduction of False Positives:** We believe adaptive dropout regularization and other model optimization methods will lower malware detection false positives. We believe these methods can improve model robustness and tolerance to noise and adversarial attacks.

Limitations:

1. **Data Availability and Quality:** Data availability and quality are major limits for training and evaluation datasets. Data diversity and representativeness are crucial to machine learning algorithms. Data quality and lack of real-world IoT malware samples may impair the model's effectiveness and generalizability.
2. **Complexity of IoT Environments:** The complexity of IoT ecosystems is due to their heterogeneity, including devices having different computational capabilities, communication protocols, and security levels. The methodology may not work in all IoT contexts, making performance and scalability difficult.
3. **Adversarial Attacks:** Adaptive dropout and regularization strategies improve model robustness, yet sophisticated adversarial attacks still occur. To avoid discovery, attackers may exploit model architecture weaknesses or modify input data, compromising security and privacy.
4. **Interpretability of Results:** Deep learning models, especially WRNs, are difficult to interpret. In sensitive fields like cybersecurity, understanding model projections and guaranteeing decision-making transparency are crucial yet difficult.
5. **Computational Resources:** Training deep learning models, especially WRNs with complicated architectures, requires significant computer resources. Lack of high-performance computing infrastructure may limit the methodology's scalability and practicality.
6. **Deployment Challenges:** Successfully transferring from research to commercialization requires overcoming practical hurdles such integration with existing IoT infrastructure, regulatory compliance, and user acceptance. These deployment issues may hamper the methodology's general acceptance and influence.
7. **Ethical and Legal Implications:** Machine learning for cybersecurity poses data privacy, consent, and model bias issues. Responsible usage and deployment of the proposed methodology requires addressing these ethical and legal issues.

In conclusion, our research increases IoT cybersecurity with unique machine learning approaches, including WRNs, but we must identify and overcome the above shortcomings. To improve the methodology's dependability, scalability, and real-world impact on user privacy and IoT ecosystem security, these issues should be addressed.

5 Model Evaluation

The subsequent phase in the development of this model will involve evaluating its performance through the utilization of several assessment metrics. This evaluation aims to assess privacy attacks

on previously undisclosed data and to analyze the model's ability to generalize and make accurate predictions.

The six metrics utilized in this study offer a concise account:

i. **Accuracy:** Accuracy refers to the proportion of test cases that were properly classified out of all test samples.

$$Accuracy = TP + TN/TP + TN + FP + FN \tag{9}$$

ii. **Precision:** Precision is a metric that measures the ratio of accurately labeled test samples to the total number of instances gathered.

$$Precision = TP/TP + FP \tag{10}$$

iii. **Recall:** Recall, commonly referred to as sensitivity, true positive rate (TPR), or detection rate (DR), is widely recognized by various names. The term refers to the proportion of malware samples that have been correctly recognized out of all the malware samples in a given test set.

$$Recall = TP/TP + FN \tag{11}$$

iv. **F1-score:** The F1-score is a metric that calculates the harmonic average of recall and precision for a specific model.

$$F1 = 2 * Recall * Precision/Recall + Precision \tag{12}$$

v. **Confusion Matrix:** The confusion matrix is a commonly employed tabular representation that elucidates the performance of a classification model. Through the comparison of projected and true labels, this analysis offers a comprehensive assessment of a model's predictions for a certain dataset. The confusion matrix consists of four main components:
   - A True Positive (TP) refers to a situation when the model correctly anticipated the positive class.
   - The model properly predicted the negative class, resulting in a true negative (TN).
   - A False Positive (FP) refers to a Type I error that arises when the model incorrectly predicts the positive class.
   - A False Negative (FN) refers to a Type II error that arises when the model incorrectly predicts the negative class.

vi. **ROC Curve:** The receiver operating characteristic (ROC) curve is a visual representation that illustrates the extent to which the diagnostic capability of a binary classification model can be assessed by adjusting its discrimination threshold. The process involves graphing the true positive rate (TPR) against the FPR at various threshold levels.
   The TPR, also known as sensitivity or recall, measures the proportion of actual positive instances that the model accurately identifies.
   - **TPR:** TPR, which is often referred to as sensitivity or recall, quantifies the percentage of real positive cases that the model properly detects.
   - **FPR:** FPR quantifies the percentage of real negative cases that the model mistakenly classifies as positive.

The visual representation of the trade-off between TPR and FPR across various threshold values is provided by the ROC curve. A perfect classifier would have a ROC curve with a high sensitivity and low false positive rate that passes through the top-left corner of the plot (TPR = 1, FPR = 0).

We shall now evaluate training performances and outcomes in the next section.

5.1 Training and Evaluation Results

Within this section, we shall do an analysis of the performance exhibited by the training and evaluation outcomes. It is imperative to acknowledge that our experiment was conducted on two distinct settings. There exist two distinct categories, namely benign and malicious. There are four categories of malware: benign, ransomware, spy worm, and Trojan. In the initial scenario, all the malware categories are consolidated [23]. One notable advantage of binary classes is their superior performance compared to multiclasses. Fig. 14 displays the training and validation curves for accuracy and loss.

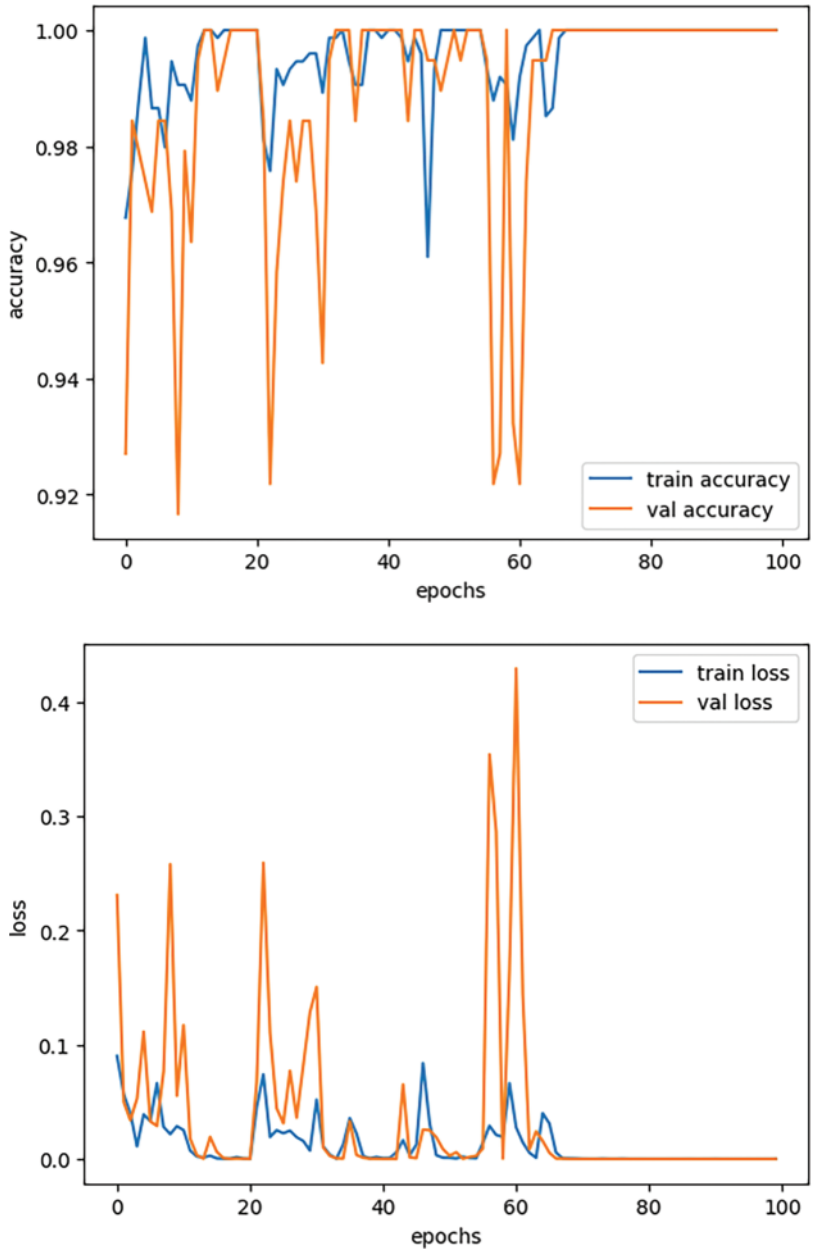

Figure 14: 4 multi class training and validation performance

These curves in Fig. 15 show the model training results and validation results for both of the classes which are a perfect fit with not much difference in validation and training performance. As in some cases the model makes a perfect fit for the training results, but validation performance is very poor due to the overfitting in the model. This issue from the start is kept away by fine tuning model architecture and ensuring that training and testing sets are properly prepared.

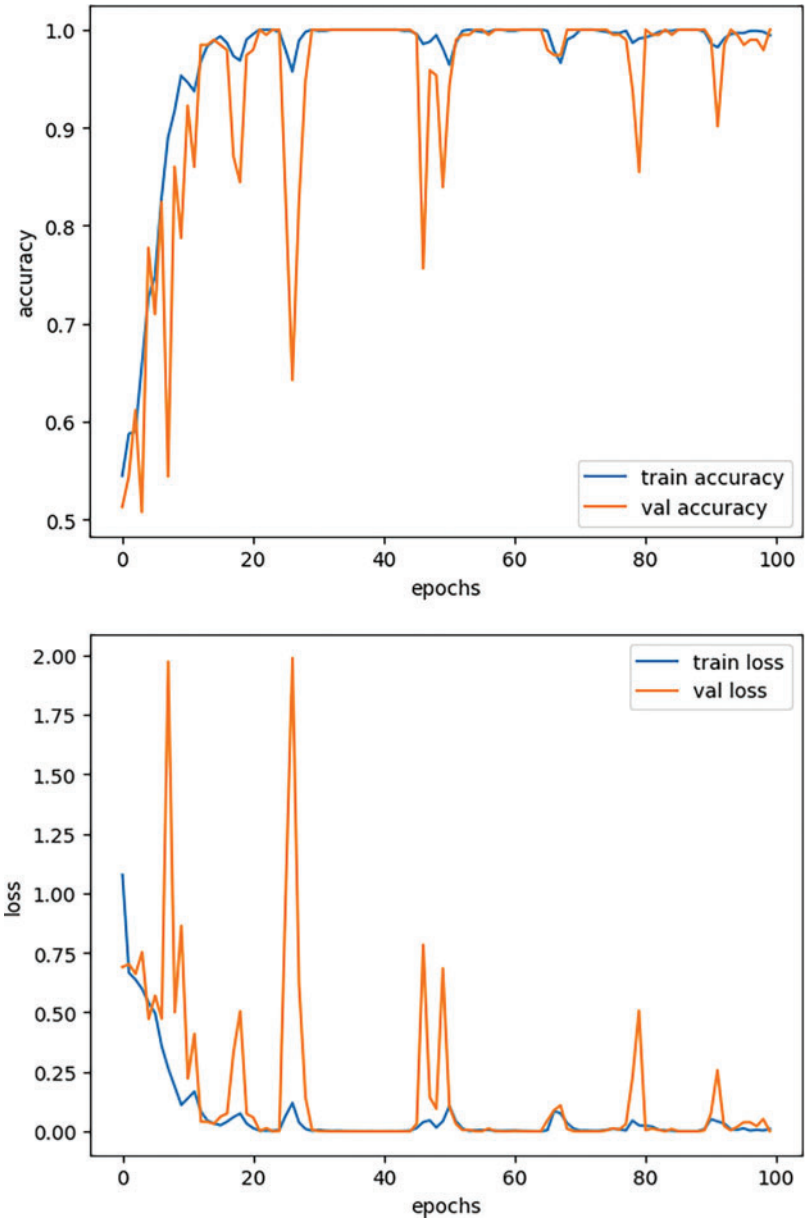

Figure 15: Binary class training and validation accuracy and loss performances

Now in the next section we will evaluate the performance of the model using unseen test data for both binary and multi class using different evaluation metrics.

The Tables 3 and 4 display the model's performance in terms of recall, F1-score, and precision. These metrics are exceptionally high for both binary and multiclass classifications. This remarkable outcome is being attained through the utilization of the cutting-edge wide residual network, which prioritizes the breadth rather than the depth of the neural network. The extensive residual network, consisting of multiple layers of residual blocks, enables our model to effectively extract features and achieve exceptional performance [24,26]. By utilizing these training outcomes and evaluation outcomes, we may employ this model in many scenarios across various IoT systems in smart cities, thereby guaranteeing the preservation of IoT safety and privacy. Such assaults can persist in IoT systems for an extended duration without being discovered and have the capability to illicitly duplicate information, so violating privacy. This concept can be utilized in contemporary IoT systems to guarantee their security and privacy.

Table 3: Evaluation metrics for test data multi class

Evaluation metric	Performance
Accuracy	0.9989
Loss	1e-5
Precision	0.9981
F1-score	0.9976
Recall	0.9982

Table 4: Evaluation metrics for test data binary class

Evaluation metric	Performance
Accuracy	0.9998
Loss	1e-6
Precision	0.9992
F1-score	0.9983
Recall	0.9991

In this section, we will demonstrate the exceptional efficacy of our model by employing the confusion matrix and ROC curve. These graphical representations will illustrate the number of classes that accurately predicted the test data and the corresponding number of incorrect predictions. Fig. 16 displays the ROC curve, which displays the rates of true positives and false positives.

As depicted in the aforementioned Fig. 17, there are no incorrect predictions made by any class in both binary and multiclass scenarios. The model's exceptional predictive capability is indicated by the fact that just a small fraction, perhaps 1 or 2 out of thousands, are inaccurately forecasted. The ROC curve for both classes is depicted in Fig. 18.

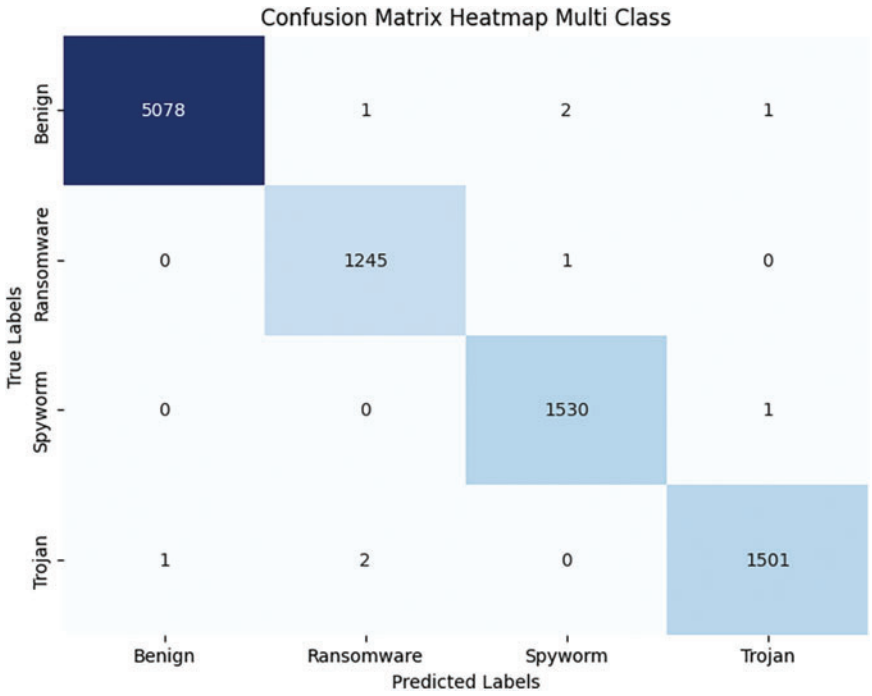

Figure 16: Multi class confusion chart

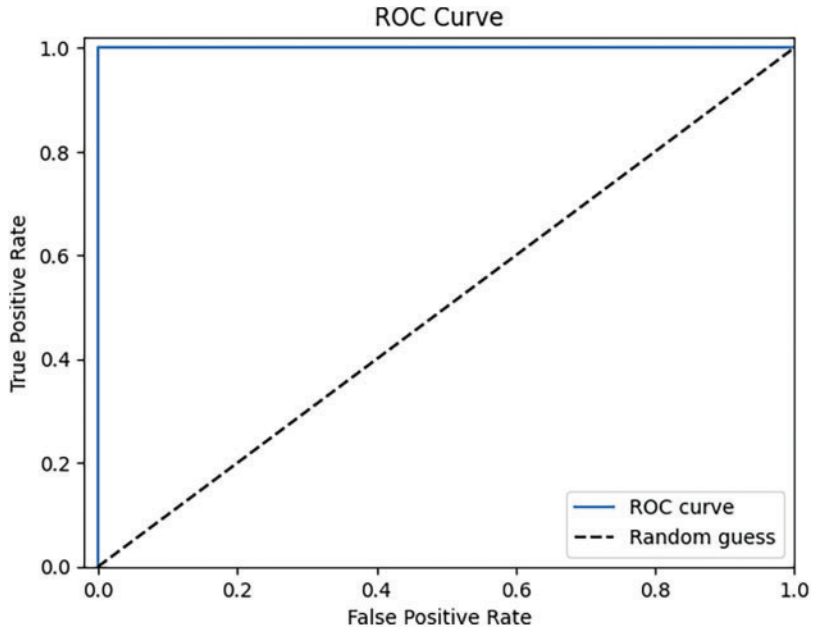

Figure 17: Multi class ROC curve

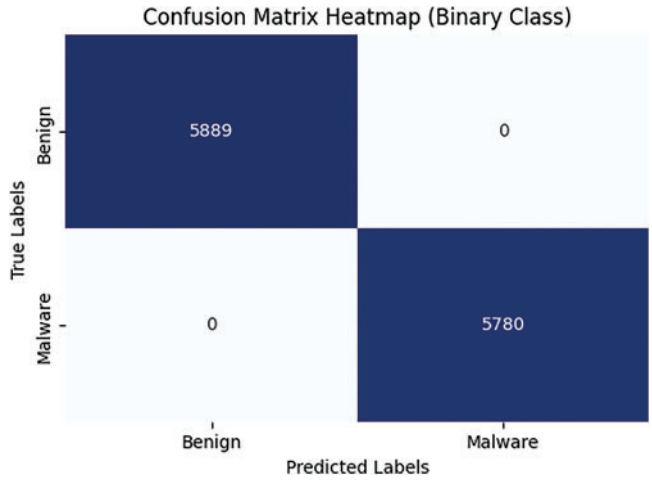

Figure 18: Binary class confusion matrix

The ROC curve depicted in Fig. 19 demonstrates the exceptional performance of our model. The ROC rate of 1.0 indicates a true positive rate of 1.0, signifying that the model accurately predicts nearly all of the classes. It is noteworthy to mention that the size of our dataset was sufficiently large, consisting of around 60,000 features and a total of 55 features. The substantial volume of data facilitated the model's acquisition and extraction of valuable features through the utilization of extensive residual blocks, so ensuring exceptional detection of privacy violations in the IoT domain. Various forms of assaults can impact the IoT in diverse manners. The primary objective of this study is to establish a resilient system capable of identifying obfuscated attacks that are challenging to detect. These attacks infiltrate the IoT system and compromise privacy information without being readily apparent. By utilizing this technology, it is possible to guarantee the security of our IoT systems against such malicious attacks.

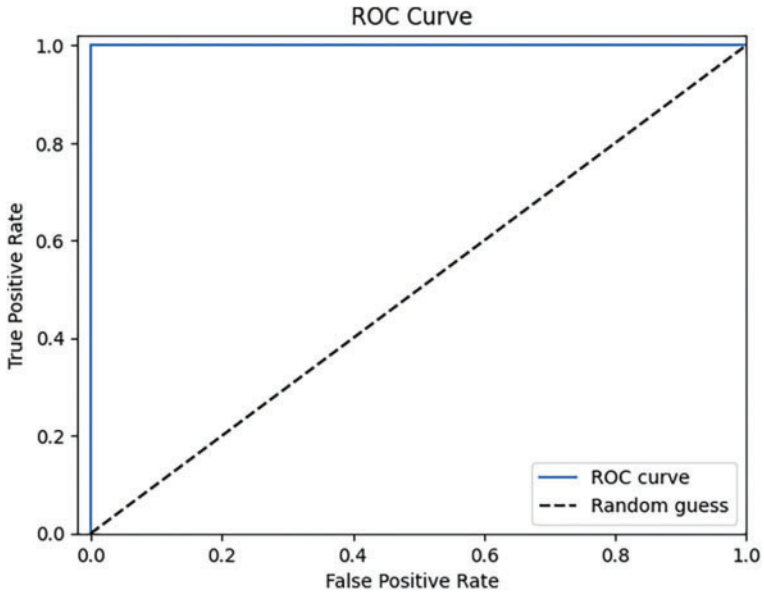

Figure 19: Binary ROC curve

5.2 Results and Method Advantages|Disadvantages

In both binary and multiclass classifications, the study on privacy preservation in IoT devices utilizing Wide Residual Networks (WRNs) to detect obfuscated malware has produced encouraging results. The efficacy of our methodology is demonstrated by the performance measures presented in Tables 3 and 4. The model's accuracy in the binary class scenario was 99.98%, with precision, recall, and F1-score all surpassing 99.9%. Similar to this, the model demonstrated strong performance in terms of precision, recall, and F1-score metrics—all of which above 99.7%—and high levels of accuracy (99.89%) for the multiclass arrangement. These results demonstrate how well WRNs can distinguish between benign activity and other types of malwares, such as worms, trojans, ransomware, and spyware.

Advantages of the Method

i. **High Accuracy and Precision:** WRNs are useful because they can sustain high precision levels (over 99.8%) in a variety of classifications. With minimal false positives, this precision guarantees the trustworthy identification of possible threats.
ii. **Robust Performance on Large Datasets:** Performance on Big Datasets: The WRN architecture performs robustly when it comes to feature extraction and model training, handling a dataset of about 60,000 features. The extensive feature learning made possible by the deep layers of residual blocks is essential for complex IoT security concerns.
iii. **Generalizability:** Robust generalizability is indicated by the model's performance on test data that hasn't been seen yet. This is essential for deployment in a variety of IoT scenarios, such smart cities, where a range of IoT networks and devices call for flexible security solutions.
iv. **Resilience to Obfuscated Attacks:** One of WRNs' most notable benefits is their ability to withstand obfuscated attacks. Although the goal of these attacks is to mask malware in order to avoid detection by security systems, the wide architecture of WRNs allows them to identify minute patterns that point to malevolent intent.
v. **Scalability:** The method is scalable with respect to the size of the dataset and the IoT network's complexity. When IoT systems grow, adding more devices and producing greater amounts of data, WRNs are able to effectively manage the higher processing demands without sacrificing functionality.

Disadvantages of the Method

i. **Computational Intensity:** Although WRNs perform exceptionally well, training and inference can be computationally demanding due to their deep architecture. This could need a significant amount of computational power, which could restrict implementation on IoT devices with limited resources.
ii. **Complexity in Implementation:** Neural network design and optimization techniques knowledge are necessary for the successful implementation and ongoing fine-tuning of WRNs. For practitioners who lack specialist expertise of deep learning approaches, this intricacy may provide difficulties.
iii. **Dependency on Quality and Quantity of Data:** The amount and quality of labeled data that is available for training determines how effective WRNs are. Model predictions may be skewed or performance may be subpar due to incomplete or biased datasets.
iv. **Interpretability:** WRNs and other deep neural networks are frequently regarded as "black-box" models, which makes it difficult to understand how they make decisions. This interpretability problem might reduce confidence in the model by making it more difficult to grasp how and why particular predictions are made.

To sum up, the use of large residual networks to identify disguised malware on Internet of Things devices shows a great deal of promise for enhancing privacy protection. The method's relevance for protecting IoT ecosystems against changing security threats is shown by its ability to achieve high accuracy and robust performance across many malware categories. Even if there are operational and computational difficulties, the benefits exceed these drawbacks, particularly when it comes to guaranteeing IoT security and privacy. Subsequent investigations may concentrate on augmenting comprehensibility and diminishing computational burden, therefore improving the implementation of WRNs in authentic IoT settings.

6 Discussion

This study examines the issue of privacy protection, which holds significant importance in the context of IoT devices. The growing popularity of IoT devices presents a potential risk of malware specifically targeting these devices, hence posing a significant threat to user privacy. The study enhances the security and privacy of IoT devices by the identification of malware, which is concealed utilizing advanced techniques such as WRN. Malware writers frequently employ obfuscation as a strategy to evade detection by conventional antivirus software. Employing WRN for malware detection is a beneficial approach for identifying obfuscated malware in the IoT domain. The ability of WRN to detect conduct, even when it is veiled by obfuscation, is attributed to its capacity to discern intricate patterns and features. The efficacy of the WRN design in addressing the intricacies associated with virus detection is exemplified through its utilization. The extensive and comprehensive architecture of WRN enables it to effectively extract both high-level and low-level attributes from input data, rendering it suitable for a diverse range of malware sample characteristics. Furthermore, the network's ability to acquire intricate representations is enhanced by incorporating more compact layers, hence enhancing detection performance. The obtained training results provide evidence of the efficacy of the proposed technique, exhibiting an accuracy rate in close proximity to 99%. The model's capacity to differentiate between harmless and harmful conduct on Internet of Things devices is evidenced by its exceptional precision. Moreover, the effectiveness of the model is additionally substantiated by its strong performance on the test set and its ability to generalize effectively to novel data.

The proposed approach has the potential to mitigate privacy breaches and safeguard sensitive data transmitted or stored by IoT devices through the accurate detection of concealed malicious software. The implementation of this measure serves to mitigate the risk of illegal entry into confidential data, personal information, and critical systems, hence yielding significant implications for individuals, corporations, and institutions reliant on IoT technology. Despite the positive results, there are still several challenges and areas that require more investigation. One challenge lies in the ever-evolving nature of malware, which constantly adapts to evade detection. To effectively address emerging threats, it is necessary to continuously monitor and adjust detection techniques. Additional research might be focused on examining the feasibility of using the proposed methodology in practical IoT environments and expanding its capacity to effectively manage large-scale datasets. It is crucial to tackle ethical concerns, such as ensuring transparency in the collection and utilization of data, safeguarding users' privacy rights, and mitigating any potential biases in the detection model. In addition, it is imperative to establish standards and laws that promote responsible deployment and security practices for IoT devices. This can be achieved through collaborative efforts among industry stakeholders, legislators, and cybersecurity professionals. The study finishes by presenting a novel approach to safeguarding privacy on IoT devices, which involves the utilization of a broad residual network to identify concealed malware. The results demonstrate the efficacy of the proposed methodology and its substantial impact on enhancing the security and privacy of IoT systems. In order to effectively address emerging risks and

ensure the resilience of IoT infrastructure against hostile attacks, it is imperative to conduct additional research and foster collaboration within this domain.

6.1 Comparative Analysis

The comparison research uncovers a wide range of approaches and performance indicators used by different studies in the field of network intrusion detection, with a specific emphasis on detecting disguised malware on IoT devices as shown in Table 5. Every reference offers several methods, including both conventional machine learning algorithms and more sophisticated deep learning frameworks. The evaluation criteria mainly focus on the detection accuracy attained on the CIC-Malmem-2022 OMM dataset, which offers insights into the effectiveness of various algorithms in recognizing harmful actions.

Table 5: Comparative analysis

References	Topic	Dataset	Technique used	Detection accuracy	Advantages	Disadvantages
[7]	Obfuscated Memory Malware Detection in Resource-Constrained IoT Devices for Smart City Applications	CIC-Malmem-2022 OMM	CompactCBL and RobustCBL based Model	72.6% and 71.42%	Compact and robust models	Accuracy not highest among peers
[10]	MalHyStack: A hybrid stacked ensemble learning framework with feature engineering schemes for obfuscated malware analysis	CIC-Malmem-2022 OMM	Pearson correlation coefficient	77.5%	Effective feature engineering	Moderate accuracy
[13]	SeqDroid: Obfuscated Android Malware Detection Using Stacked Convolutional and Recurrent Neural Networks	CIC-Malmem-2022 OMM	Stacked RNNs and CNNs applied for feature extraction	97.7%	High detection accuracy	Potential complexity in model interpretation
[18]	Obfusifier: Obfuscation-Resistant Android Malware Detection System	CIC-Malmem-2022 OMM	Random forest technique	89.8%	Robust against obfuscation	Lower accuracy compared to top performers
[21]	Detecting Obfuscated Malware using Memory Feature Engineering	CIC-Malmem-2022 OMM	Naïve Bayes, KNN	92% and 95%	Utilizes memory feature engineering	Mixed accuracy scores
[24]	A malware detection method based on family behavior graph	CIC-Malmem-2022 OMM	Deep Belief Networks	95.8%	Deep learning approach	Complexity in training and deployment
[26]	Obfuscated Malware Detection: Investigating Real-world Scenarios through Memory Analysis	CIC-Malmem-2022 OMM	RF, MLP, KNN	93.95%, 82.11% and 91.21%	Multiple model comparison	Mixed accuracy across different models
[28]	Obfuscated Privacy Malware Classifiers Based on Memory Dumping Analysis	CIC-Malmem-2022 OMM	DNN, Ensemble-Learning	67.9% 66.0%. Whereas, AUC is 93.4%	AUC performance	Lower detection accuracy
[32]	Obfuscated malware detection using dilated convolutional network	CIC-Malmem-2022 OMM	CNN	75.0%	Specialized network architecture	Lower accuracy compared to top models

(Continued)

Table 5 (continued)

References	Topic	Dataset	Technique used	Detection accuracy	Advantages	Disadvantages
[33]	Malware Detection Using Memory Analysis Data in Big Data Environment	CIC-Malmem-2022 OMM	MLP, Naïve Bayes	97.67% and 98.42%	High accuracy in big data environment	Moderate accuracy for certain methods
Our paper		CIC-Malmem-2022 OMM	WRN	99.98%	Exceptionally high detection accuracy	Potential dependency on specific dataset

The paper by Shafin et al. [7] presents a model based on CNN-BiLSTM, using CompactCBL and RobustCBL approaches. This model achieves detection accuracies of 72.6% and 71.42%, respectively. Although these accuracies indicate a satisfactory level of performance, they are not as impressive as more advanced methods like WRN. Roy et al. [10] employ the Pearson correlation coefficient as their detection mechanism, resulting in a somewhat greater accuracy of 77.5%. Nevertheless, this approach seems to be less efficient in comparison to WRN's outstanding performance.

Moreover, various studies have utilized a range of machine learning algorithms to analyze the CIC-Malmem-2022 OMM dataset. As an example, Reference [13] employs stacked RNNs and CNNs to extract features, resulting in an outstanding accuracy of 97.7%. This approach exploits the capabilities of recurrent and convolutional neural networks to capture temporal and spatial correlations in data, which are essential for identifying intricate infiltration patterns.

Likewise, in contrast, Reference [18] utilizes the Random Forest approach and achieves an accuracy rate of 89.8%. Random forests are renowned for their resilience and capacity to handle data with a large number of dimensions, making them well-suited for intrusion detection jobs that require feature importance and interpretability. Mezina et al. [20] employ a Convolutional Neural Network (CNN) structure, attaining a precision rate of 75.0%. Although convolutional neural networks have demonstrated potential in numerous tasks, their effectiveness in this particular situation is surpassed by WRN.

Further, Carrier et al. [21] utilize the Naïve Bayes and KNN algorithms, yielding remarkable detection accuracies of 92% and 95%, respectively. The results demonstrate the efficacy of conventional machine learning methods in detecting malware. Further, ensemble approaches, which include the combination of many models to enhance predictive performance, have been extensively investigated. Deep Belief Networks (DBN) were employed in the study referenced as [24], resulting in an accuracy rate of 95.8%. DBNs are highly effective in autonomously acquiring features and have demonstrated potential in managing extensive datasets with intricate feature interdependencies.

In a similar manner, Rakib et al. employ RF, MLP, and KNN algorithms, attaining accuracies of 93.95%, 82.11%, and 91.21%, respectively. Although these results are noteworthy, they do not exceed the performance of WRN, as revealed in our work. In addition to, Cevallos et al. [28] utilize deep neural network (DNN) and ensemble learning methods to achieve accuracies of 67.9% and 66.0%, respectively. They also obtained an area under the curve (AUC) of 93.4%. Although ensemble methods are used, the accuracy of their model is relatively poor. In another contribution to the field of malware detection, the authors in [32] explore and presente a review of the commonly used solutions to resolve privacy and confidentiality challenges through ML algorithms in the IoT systems to preserve privacy.

Likewise, Dener et al. [33] employ MLP and Naïve Bayes algorithms, attaining remarkably high accuracies of 97.67% and 98.42%, respectively. The results demonstrate the effectiveness of MLP in

identifying disguised malware. Nevertheless, the WRN model presented in our article surpasses even these remarkable accuracies, achieving an exceptional accuracy of 99.98%.

The comparison highlights the substantial progress achieved by the WRN model in the field of network intrusion detection. While current techniques show different levels of effectiveness, our suggested model sets a new standard in terms of accuracy and robustness. The outstanding success of WRN can be credited to its capability to capture complex patterns and traits that are characteristic of disguised malware, hence facilitating more accurate detection.

To summarize, the comparative research highlights the significance of ongoing innovation in the realm of network intrusion detection. Although current models perform well, our suggested WRN model sets a higher standard for accuracy, demonstrating its ability to improve cybersecurity and reduce the risks of network attacks. With the ever-changing nature of cybersecurity threats, it is crucial to design intrusion detection systems that are reliable and adaptable. Our work is a significant advancement in this pursuit. Moreover, the distinctive architecture, exceptional classification accuracy, and ability to identify intricate characteristics of malware make our WRN model a practical approach to enhance security and safeguard privacy in IoT settings. The comparison analysis with reference studies highlights the efficacy of our proposed strategy and its potential to advance research in malware detection in IoT environments.

6.2 Future Directions

In order to further enhance the detection of such privacy attacks on IoT systems there are many steps which can be taken in future to better understand IoT ecosystems, use of advanced learning techniques. Some of the suggested directions are as follows in Fig. 20.

i. **Enhancing Model Robustness:** The robustness of the detection mechanism against adversarial attacks can be improved by more research. Adversarial assaults are designed to gently alter input data in order to trick machine learning models. Examining methods like robust optimization and adversarial training can strengthen the model's defenses against these kinds of assaults.
ii. **Dynamic Malware Detection:** Dynamic detection systems are necessary since malware is still evolving quickly and must be able to instantly adjust to new and emerging threats. Creating dynamic malware detection systems that make use of online learning and reinforcement learning approaches can allow for proactive defense against malware variants that are constantly changing.
iii. **Exploring Transfer Learning:** Transfer learning has the potential to enhance malware detection performance, particularly from adjacent fields like computer vision or natural language processing. Subsequent investigations may examine the suitability of transfer learning methodologies to capitalize on expertise from many fields and augment the detection model's capacity for generalization.
iv. **Scaling to IoT Ecosystems:** It is imperative to expand the study to include a range of IoT ecosystems with different kinds of devices, different architectures, and different communication protocols. For a practical implementation, it is imperative to look for ways to modify the detection model to fit various IoT scenarios while taking performance needs and resource limitations into account.
v. **Behavioral Analysis:** The accuracy of malware detection can be increased by combining dynamic behavioral analysis with static analysis. Finding unusual patterns that point to

malicious activities by analyzing the runtime behavior of IoT devices can be a useful tool in the detection of advanced malware strains.

vi. **User Awareness and Education:** Users of IoT devices can greatly reduce their risk of malware infestations by learning the value of cybersecurity hygiene and best practices. In order to enable users to properly protect their IoT devices, future directions may involve creating user-friendly educational materials, security awareness campaigns, and interactive training modules.

vii. **Collaboration and Information Sharing:** A collective defense against malware threats can be facilitated by establishing cooperative frameworks for information sharing and threat intelligence exchange across cybersecurity companies, security researchers, and manufacturers of IoT devices. Promoting openness and cooperation can hasten the identification and neutralization of new threats.

viii. **Regulatory Frameworks:** Encouraging the creation of standards and regulatory frameworks for IoT security can encourage adherence to security best practices and foster accountability. An IoT ecosystem can be made safer by implementing policy efforts that force baseline security criteria for IoT devices and promote vulnerability disclosure.

ix. **Continuous Evaluation and Improvement:** It is crucial to benchmark detection methods against developing malware samples and to continuously evaluate them. Standardized evaluation methods and datasets can promote ongoing detection performance improvement and make comparative analysis easier.

x. The work can significantly advance the field of privacy preservation in IoT devices and strengthen cybersecurity resilience in a world where connectivity is growing by pursuing these future approaches.

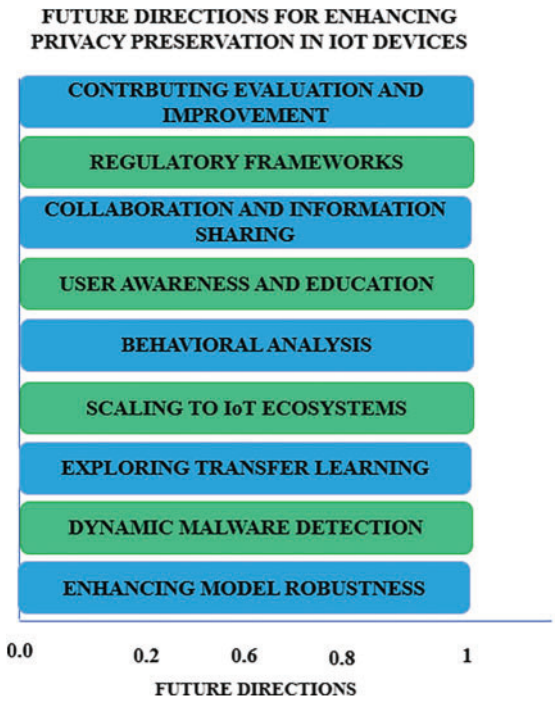

Figure 20: Future directions for privacy preservation

7 Conclusion

This study examines the important area of privacy protection in IoT ecosystems, specifically looking at how Wide Residual Network (WRN) approaches can be used to identify hidden malware. The widespread incorporation of Internet of Things (IoT) devices into everyday life has unquestionably improved connectivity and convenience. Nevertheless, the extensive acceptance of this technology has also heightened susceptibility to security risks, highlighting the need for strong safeguards. Our research highlights the efficacy of utilizing deep neural networks and advanced machine learning (ML) techniques to counteract progressively intricate malware, particularly those utilizing obfuscation to elude traditional detection methods. By utilizing WRNs, we have successfully attained an impressive accuracy rate of 99.98% in detecting concealed malware samples. This outcome confirms the effectiveness of our method as a proactive measure against new risks in IoT networks.

Based on the evaluation metrics obtained from our experiments, our WRN-based malware detection system demonstrates strength in several areas, such as recall, F1-score, accuracy, and precision. These findings routinely surpass the 99.98% barrier for both binary and multi-class classifications, demonstrating the effectiveness of our technology and its potential to significantly enhance the safety of IoT installations. In addition, our research clarifies the distinct security and privacy problems presented by IoT devices. IoT devices, in contrast to traditional computing systems, frequently function with limited resources and inadequate security measures, making them vulnerable to malicious assaults. By specifically targeting these weaknesses and suggesting customized remedies, we make a substantial contribution to strengthening the ability of IoT ecosystems to withstand cyberattacks.

The CIC-Malware-2022 OMM dataset is crucial for our study as it provides a genuine and comprehensive means to evaluate malware detection techniques in IoT environments. Utilizing this dataset not only strengthens the validity of our conclusions but also emphasizes the practical significance of our research in addressing real-world cybersecurity concerns. Our findings provide a foundation for future research focused on improving the security of IoT devices and protecting user privacy. Investigating alternative approaches such as behavioral analysis, transfer learning, and privacy-preserving tactics shows potential for improving malware detection systems and strengthening their ability to effectively combat new threats.

Ultimately, our research supports the implementation of proactive measures to strengthen user privacy and improve the security resilience of IoT devices. Addressing malware risks at the network edge, where IoT devices are deployed, is essential for establishing strong defenses against ever-changing cyber threats. Furthermore, our research emphasizes the necessity for cooperation between academic institutions, industries, and government entities to collaboratively address the security issues arising from the extensive deployment of IoT devices. The results of our analysis signify a substantial advancement in improving privacy protection and security in IoT environments. By utilizing cutting-edge machine learning methods and implementing a proactive risk management strategy, we can enhance the security strength of IoT devices while also respecting individuals' privacy rights. To ensure the long-term durability of IoT deployments, it is crucial to remain watchful and flexible in adapting to emerging technologies.

Prospective Endeavors

Subsequent research efforts can utilize our insights to enhance the security capabilities of IoT ecosystems. An avenue worth exploring is the enhancement of our detection models using continuous learning frameworks that can adapt to the changing behaviors of malware. Behavioral analysis

approaches show potential in this area by allowing the early detection of abnormal device activities that may indicate security breaches.

Furthermore, investigating transfer learning techniques designed for IoT-specific datasets could improve the scalability and applicability of malware detection systems in various IoT installations. This approach utilizes knowledge acquired from one Internet of Things (IoT) scenario to enhance the accuracy of detection in a different scenario, thereby maximizing the consumption of resources and improving performance.

Moreover, the incorporation of privacy-preserving techniques into current detection systems is a crucial field that requires further exploration. Methods such as federated learning and differential privacy might enable cooperative model training across decentralized IoT networks while maintaining data confidentiality, hence improving overall security resilience.

It is crucial to have collaboration among academics, industry, and politicians in order to drive these breakthroughs forward. Through the cultivation of interdisciplinary collaborations, upcoming research can efficiently tackle emerging cybersecurity concerns and guarantee the ongoing security and privacy of IoT ecosystems in the face of rapid technological advancement.

Acknowledgement: The authors would like to thank Princess Nourah bint Abdulrahman University for funding this project through the Researchers Supporting Project (PNURSP2024R435) and Prince Sultan University for covering the article processing charges (APC) associated with this publication. Special acknowledgment to Automated Systems & Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia. The authors would also like to thank Researchers Supporting Project number (RSPD2024R1107), King Saud University, Riyadh, Saudi Arabia.

Funding Statement: The authors would like to thank Princess Nourah bint Abdulrahman University for funding this project through the researchers supporting project (PNURSP2024R435) and this research was funded by the Prince Sultan University, Riyadh, Saudi Arabia.

Author Contributions: Study conception and design: Deema Alsekait, Mohammed Zakariah, Syed Umar Amin; data collection: Mohammed Zakariah, Syed Umar Amin, Zafar Iqbal Khan; analysis and interpretation of results: Deema Alsekait, Syed Umar Amin, Zafar Iqbal Khan; draft manuscript preparation: Mohammed Zakariah, Jehad Saad Alqurni, Syed Umar Amin. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Dataset is available on reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Q. Abu Al-Haija, A. Odeh, and H. Qattous, "PDF malware detection based on optimizable decision trees," *Electronics*, vol. 11, no. 19, 2022, Art. no. 3142. doi: 10.3390/electronics11193142.

[2] M. S. Akhtar and T. Feng, "Malware analysis and detection using machine learning algorithms," *Symmetry*, vol. 14, no. 11, pp. 1–10, 2022. doi: 10.3390/sym14112304.

[3] A. Amira, A. Derhab, E. B. Karbab, and O. Nouali, "A survey of malware analysis using community detection algorithms," *ACM Comput. Surv.*, vol. 56, no. 2, pp. 1–29, 2024. doi: 10.1145/3610223.
[4] 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, no. 6, 2021, Art. no. 102490. doi: 10.1016/j.cose.2021.102490.
[5] K. Brezinski and K. Ferens, "Metamorphic malware and obfuscation: A survey of techniques, variants, and generation kits," *Secur. Commun. Netw.*, vol. 2023, no. 2, pp. 1–41, 2023. doi: 10.1155/2023/8227751.
[6] Z. Chen and X. Ren, "An efficient boosting-based windows malware family classification system using multi-features fusion," *Appl. Sci.*, vol. 13, no. 6, 2023, Art. no. 4060. doi: 10.3390/app13064060.
[7] S. S. Shafin, G. Karmakar, and I. Mareels, "Obfuscated memory malware detection in resource-constrained IoT devices for smart city applications," *Sensors*, vol. 23, no. 11, 2023, Art. no. 5348. doi: 10.3390/s23115348.
[8] I. Almomani, M. Ahmed, and W. El-shafai, "Android malware analysis in a nutshell," *PLoS One*, vol. 17, no. 7, Jul. 2022, Art. no. e0270647. doi: 10.1371/journal.pone.0270647.
[9] A. Ara, "Privacy preservation in cloud based cyber physical systems," *American Sci. Pub.*, vol. 16, no. 10, pp. 4320–4327(8), Jan. 2019. doi: 10.1166/jctn.2019.8520.
[10] K. S. Roy, T. Ahmed, P. B. Udas, M. E. Karim, and S. Majumdar, "MalHyStack: A hybrid stacked ensemble learning framework with feature engineering schemes for obfuscated malware analysis," *Intell. Syst. with Appl.*, vol. 20, no. 2, 2023, Art. no. 200283. doi: 10.1016/j.iswa.2023.200283.
[11] A. M. Hilal *et al.*, "Intelligent deep learning model for privacy preserving IIoT on 6G environment," *Comput. Mater. Contin.*, vol. 72, no. 1, pp. 333–348, 2022. doi: 10.32604/cmc.2022.024794.
[12] K. Lee, J. Lee, and K. Yim, "Classification and analysis of malicious code detection techniques based on the APT attack," *Appl. Sci.*, vol. 13, no. 5, 2023, Art. no. 2894. doi: 10.3390/app13052894.
[13] W. Y. Lee, J. Saxe, and R. Harang, "SeqDroid: Obfuscated android malware detection using stacked convolutional and recurrent neural networks," in *Deep Learning Applications for Cyber Security. Advanced Sciences and Technologies for Security Applications*, M. Alazab, M. Tang (Eds.), Springer, Cham. 2019, pp. 197–210 doi: 10.1007/978-3-030-13057-2.
[14] A. H. Lashkari, B. Li, T. L. Carrier, and G. Kaur, "VolMemLyzer: Volatile memory analyzer for malware classification using feature engineering," in *2021 Reconcil. Data Anal., Automat., Priv. Secur.: A Big Data Challe. (RDAAPS)*, IEEE, 2021, pp. 1–8. doi: 10.1109/RDAAPS48126.2021.9452028.
[15] Y. Qian, Y. Jiang, M. S. Hossain, L. Hu, G. Muhammad and S. U. Amin, "Privacy-preserving based task allocation with mobile edge clouds," *Inf. Sci.*, vol. 507, no. 11, pp. 288–297, 2020. doi: 10.1016/j.ins.2019.07.092.
[16] H. M. Farghaly and T. Abd El-Hafeez, "A high-quality feature selection method based on frequent and correlated items for text classification," *Soft. Comput.*, vol. 27, no. 16, pp. 11259–11274, 2023. doi: 10.1007/s00500-023-08587-x.
[17] H. H. R. Manzil and S. M. Naik, "Android malware category detection using a novel feature vector-based machine learning model," *Cybersecurity*, vol. 6, no. 1, 2023, Art. no. 6. doi: 10.1186/s42400-023-00139-y.
[18] Z. Li, J. Sun, Q. Yan, W. Srisa-an, and Y. Tsutano, "Obfusifier: Obfuscation-resistant android malware detection system," vol. 304, pp. 214–234, 2019. doi: 10.1007/978-3-030-37228-6_11.
[19] M. Al-Qudah, Z. Ashi, M. Alnabhan, and Q. Abu Al-Haija, "Effective one-class classifier model for memory dump malware detection," *J. Sens. Actuator Netw.*, vol. 12, no. 1, 2023, Art. no. 5. doi: 10.3390/jsan12010005.
[20] A. Mezina and R. Burget, "Obfuscated malware detection using dilated convolutional network," in *2022 14th Int. Cong. Ultra Modern Telecommun. Cont. Syst. Workshops (ICUMT)*, IEEE, 2022, pp. 110–115. doi: 10.1109/ICUMT57764.2022.9943443.
[21] T. Carrier, P. Victor, A. Tekeoglu, and A. Lashkari, "Detecting obfuscated malware using memory feature engineering," in *Proc. 8th Int. Conf. Inf. Syst. Secur. Priv.*, SciTePress-Science and Technology Publications, 2022, pp. 177–188. doi: 10.5220/0010908200003120.

[22] B. I. Mukhtar, M. S. Elsayed, A. D. Jurcut, and M. A. Azer, "IoT vulnerabilities and attacks: SILEX malware case study," *Symmetry*, vol. 15, no. 11, 2023, Art. no. 1978. doi: 10.3390/sym15111978.
[23] L. K. Vashishtha, K. Chatterjee, and S. S. Rout, "An Ensemble approach for advance malware memory analysis using Image classification techniques," *J. Inf. Secur. Appl.*, vol. 77, no. 5, 2023, Art. no. 103561. doi: 10.1016/j.jisa.2023.103561.
[24] Y. Ding, X. Xia, S. Chen, and Y. Li, "A malware detection method based on family behavior graph," *Comput. Secur.*, vol. 73, no. 10, pp. 73–86, 2018. doi: 10.1016/j.cose.2017.10.007.
[25] Q. -V. Dang, "Enhancing obfuscated malware detection with machine learning techniques," vol. 1688, pp. 731–738, 2022. doi: 10.1007/978-981-19-8069-5.
[26] S. M. Rakib and A. Dhakal, "Obfuscated malware detection: Investigating real-world scenarios through memory analysis," in *2023 IEEE International Conference on Telecommunications and Photonics (ICTP)*, Dhaka, Bangladesh, 2023, pp. 1–5, 2024. doi: 10.1109/ICTP60248.2023.10490701.
[27] E. M. Rudd, D. Krisiloff, S. Coull, D. Olszewski, E. Raff and J. Holt, "Efficient malware analysis using metric embeddings, digital threats," *Res. and Pract.*, vol. 5, no. 1, pp. 1–20, 2024. doi: 10.1145/3615669.
[28] D. Cevallos-Salas, F. Grijalva, J. Estrada-Jiménez, D. Benítez, and R. Andrade, "Obfuscated privacy malware classifiers based on memory dumping analysis," *IEEE Access*, vol. 12, no. 12, pp. 17481–17498, 2024. doi: 10.1109/ACCESS.2024.3358840.
[29] L. Mhamdi, D. McLernon, F. El-moussa, S. A. Raza Zaidi, M. Ghogho and T. Tang, "A deep learning approach combining autoencoder with one-class SVM for DDoS attack detection in SDNs," in *2020 IEEE Eighth Int. Conf. Commun. Netw. (ComNet)*, Hammamet, Tunisia, 2020.
[30] A. Nugraha and J. Zeniarja, "Malware detection using decision tree algorithm based on memory features engineering," *J. Appl. Intell. Sys.*, vol. 7, no. 3, pp. 206–210, 2022. doi: 10.33633/jais.v7i3.6735.
[31] M. A. Hossain and M. S. Islam, "Enhanced detection of obfuscated malware in memory dumps: A machine learning approach for advanced cybersecurity," *Cybersecurity*, vol. 7, no. 1, 2024, Art. no. 16. doi: 10.1186/s42400-024-00205-z.
[32] S. El-Gendy, M. S. Elsayed, A. Jurcut, and M. A. Azer, "Privacy preservation using machine learning in the internet of things," *Mathematics*, vol. 11, no. 16, 2023, Art. no. 3477. doi: 10.3390/math11163477.
[33] M. Dener, G. Ok, and A. Orman, "Malware detection using memory analysis data in big data environment," *Appl. Sci.*, vol. 12, no. 17, 2022, Art. no. 8604. doi: 10.3390/app12178604.
[34] A. Singh, R. Ikuesan, and H. Venter, "Ransomware detection using process memory," in *Int. Conf. Cyber Warfare and Secur.*, Albany, New York, NY, USA, 2022.
[35] F. S. Alrayes, M. Zakariah, S. U. Amin, Z. Iqbal Khan, and M. Helal, "Intrusion detection in IoT systems using denoising autoencoder," *IEEE Access*, vol. 12, pp. 122401–122425, 2024. doi: 10.1109/ACCESS.2024.3451726.
[36] H. Naeem, S. Dong, O. J. Falana, and F. Ullah, "Development of a deep stacked ensemble with process based volatile memory forensics for platform independent malware detection and classification," *Expert. Syst. Appl.*, vol. 223, no. 4, 2023, Art. no. 119952. doi: 10.1016/j.eswa.2023.119952.
[37] F. Concone, S. Gaglio, A. Giammanco, G. Lo Re, and M. Morana, "AdverSPAM: Adversarial SPam account manipulation in online social networks," *ACM Trans. Priv. Secur.*, vol. 27, no. 2, pp. 1–31, 2024. doi: 10.1145/3643563.
[38] M. M. Alani, A. Mashatan, and A. Miri, "XMal: A lightweight memory-based explainable obfuscated-malware detector," *Comput. Secur.*, vol. 133, no. 2, 2023, Art. no. 103409. doi: 10.1016/j.cose.2023.103409.