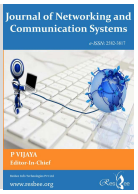
Vol.7 No.4 Oct 2024

Journal of Networking and Communication Systems

Received 13 August, Revised 19 August, Accepted 10 October



Deep Learning for Cybersecurity using SWideRNet with Cross Correntropy Loss in Malware Detection and Mitigation

A. Vinothavasuki

Assistant Professor, Department of Computer Science and Engineering,
J.P. College of Engineering, Tamil Nadu
vasukivinotha@gmail.com.

Abstract: Data integrity, cybersecurity, and defense against more advanced cyber threats all depend on effective malware detection. The majority of signature-based techniques used by current malware detection systems are ineffective against novel and polymorphic malware. Consequently, systems are exposed to serious security risks and suffer from high false negative rates. Moreover, heuristic and behavior-based approaches, though more flexible, frequently produce false positives, disrupting legitimate user activities and minimizing user trust in security solutions. By protecting sensitive information and vital systems from malicious attacks, this research aims to improve cybersecurity. Initially, input log file data for analysis is taken from the dataset. Subsequently, data normalization is executed through z-score normalization to standardize feature values. Next, feature selection is performed utilizing Mutual Information (MI) to identify the most relevant features. Following this, malware detection is conducted using the Modified Scaling Wide Residual Network (Modified SWideRNet) framework to identify potential threats. If an attack is detected, the effect of the identified malware is mitigated by applying the Modified SWideRNet, and the overall effectiveness of the process is evaluated based on the outcomes. Additionally, the devised Modified SWideRNet has outperformed existing techniques by acquiring accuracy, precision, and recall with values of 91.435%, 90.133%, and 92.344%.

Keywords: Cyber Threat Intelligence, Malware Data, Malware Detection, Malware Mitigation, Deep Learning

Nomenclature

Abbreviation	Expansion
DL	Deep Learning
ML	Machine Learning
ECDLP	Ensemble Classification with Deep Learning Parallelism
RHSODL-AMD	Rock Hyrax Swarm Optimization with deep learning-based Android malware detection
IoT	Internet of Things
RMDNet	Robust Malware Detection Network
WRN	Wide Residual Networks
SAC	Switchable Atrous Convolution
SE	Squeeze-and-Excitation

1. Introduction

Our lives are easier and more convenient in the digital age, but they are also vulnerable to cyberattacks due to the quick advancement of computing technologies. Recently, one of the biggest threats has emerged as a result of the rapid increase in cyberattacks and the associated damages [13]. Cyberattacks result in economic losses amounting to trillions of dollars on a global scale [2]. The framework of cybersecurity faces a major change due to the increasing complexity of malware threats. With advanced strategies and tools at their service, attackers are always testing the limits of accepted security practices. The complexities of the malicious code itself have increased, but so have the methods employed to prevent detection, exploit weakness, and remain within infected schemes [5]. Malware is frequently used by cybercriminals to initiate cyberattacks. Malware is any software that executes on victim machines in an unwanted or suspicious manner. There are many different kinds of malware, including viruses, worms, Trojan horses, rootkits, ransomware, etc., [11]. Cybercriminals create malware to execute unwanted tasks on victims' computers. Numerous malware, such as viruses, ransomware, worms, etc., can damage computer systems, steal confidential information, and hide from their targets' systems. Additionally, the trust that users place

https://doi.org/10.46253/jnacs.v7i4.a3

in malware serves as a vector for infection. The malware detection and security industry is valued at billions of dollars and is constantly expanding due to the harmful and dangerous effects of malware [2]. Malicious or benign files and activities can be identified based on many factors, including infection potential, malware structure, and speed rate. To protect legitimate users, threats and malware attacks should be instantly detected as soon as they affect computer systems. Researchers have created classification techniques for malware detection for this purpose. Malware detection methods can be divided into four categories, such as heuristic, signature, behavior, and model-checking. However, most of the methods failed to identify advanced malware variants effectively [2]. DL models have been used to address the problems of current malware detection techniques, and malware detection is no exception [14]. DL is a branch of ML that works with multi-layered artificial neural networks to automatically learn and excerpt complicated patterns and features from data. DL is an appealing strategy for addressing the difficulties associated with malware analysis. Finding and comprehending malicious software's behavior, goals, and capabilities is the main goal of malware analysis. Static analysis is the process of analyzing the content and structure of malware samples without processing them, while dynamic analysis is the process of observing the behavior of malware when it executes in a controlled environment [5]. DL models have several benefits over traditional ML models, including the capacity to handle huge databases and the automatic generation of high-quality features. However, ensemble methods can be applied to ML/DL models to improve their classification task performance [15] [2].

The significant aim of this work is to devise a proficient technique using Modified SWideRNet for the detection and mitigation of malware. The detection process is performed by acquiring input log file data from the specified database first. Next, data normalization is carried out utilizing Z-score normalization to standardize the data. After normalization, feature selection is carried out using MI, which helps to identify the most relevant data points for analysis. Following this, malware detection is performed by exploiting the Modified SWideRNet framework to identify potential threats. Finally, Modified SWideRNet is also utilized for mitigating the detected malware.

The important contribution of this work is enumerated beneath,

- ***Modified SWideRNet for detection and mitigation of malware:*** An effectual method based on Modified SWideRNet is introduced for the detection and mitigation of malware. Malware detection is performed by exploiting the SWideRNet framework with Cross correntropy for identifying potential threats. Finally, SWideRNet is utilized for mitigating the detected malware.

The remaining sections of this work are ordered as follows: The literature survey on malware detection and mitigation is explained in Section 2, section 3 presents the system model of malware variant traffic identification model and section 4 deliberates the Modified SWideRNet model for effective malware detection and mitigation. In addition, Section 2 demonstrates the outcomes of the Modified SWideRNet technique, while Section 6 specifies the conclusion.

2. Motivation

Detecting new and unknown malware variants is challenging for current malware detection systems, as they frequently depend on signature-based detection. In addition, the dynamic behavior of malware may not be taken into account based on static analysis techniques. Consequently, there is an urgent requirement for more reliable, flexible, and accurate malware detection techniques that can successfully identify known and unknown threats. In this work, an effectual method called Modified SWideRNet is devised for malware detection.

2.1 Literature Survey

Chaganti, R., *et al.* [1] devised the EfficientNetB1 model for effective malware classification. In this case, the method achieved a high-performance accuracy with fewer network parameters and greater computation efficiency. However, this framework was too complex to precisely categorize all samples utilizing image-based representation. Al-Andoli, M.N., *et al.* [2] designed an ECDLP approach for malware detection. Here, this approach detected malware with high scalability and efficiency. Nevertheless, this method failed to execute in large-scale parallel and distributed computing systems with numerous connected machines. To detect and classify malware, Albakri, A., *et al.* [3] established the RHSODL-AMD framework. Here, this technique produced more accurate and reliable consequences in the classification of malware. Nonetheless, the method was unable to handle Ethereum fraud transactions and crime data prediction. A visualization-based malware detection framework was developed by Marzouk, M.A. and Elkholy, M. [4]. Here, this approach required less time to test the samples. Furthermore, this approach processed malware analyses on a large scale and small scale with greater accuracy. However, this method was unable to lower false negatives. Puneeth, S., *et al.* [5] developed a RMDNet for effective malware

detection and classification. Here, the method achieved low computational complexity and optimal performance. Nevertheless, this method did not use DL models to classify malware more accurately.

2.2 Major Challenges

The key difficulties faced by the conventional approaches for malware detection and mitigation are enumerated beneath,

- In [1], the EfficientNetB1 model was devised for effective malware classification. Here, this approach handled overfitting problems and gained superior training efficacy. However, this method failed to mitigate the malware due to its inability to efficiently detect attacks.
- An effectual method called ECDLP was developed in [2] for malware detection. This technique was able to improve the computational efficacy and achieved minimal computation time. Nonetheless, this framework failed to consider feature selection techniques for attaining higher accuracy and better generalization.
- For identifying the malware variants, the RMDNet technique was established in [5]. This approach had minimum computation time and overfitting. However, this model failed to consider efficient databases for handling large-scale data storage.
- Even with the changing threat landscape, malware analysis and detection remain critical to ensuring the security of networks and computer systems. The rapidly evolving nature of malware makes traditional approaches inadequate. Advanced detection methods have a large overhead that may impact user experience and system performance.

3. System Model

Conventional malware detection models cannot be deployed in IoT gateways [10] because malware variants behave differently from general malware when they generate network traffic. A small amount of label data is gathered based on the current source domain data (general malware traffic flow) to prevent the massive collection of label data for these newly emerging unknown malware variants. The malware variants can be effectively identified with minimal label data of the unknown malware variants. Assume that in the aforementioned scenarios, the Modified SWideRNet model is installed at the IoT edge gateway. The gateway intercepts and records the communication traffic that is passing. Without requiring any expert knowledge, DL algorithms are used to extract features of the captured traffic. Fig. 1 displays the system model for malware variant traffic identification in IoT.

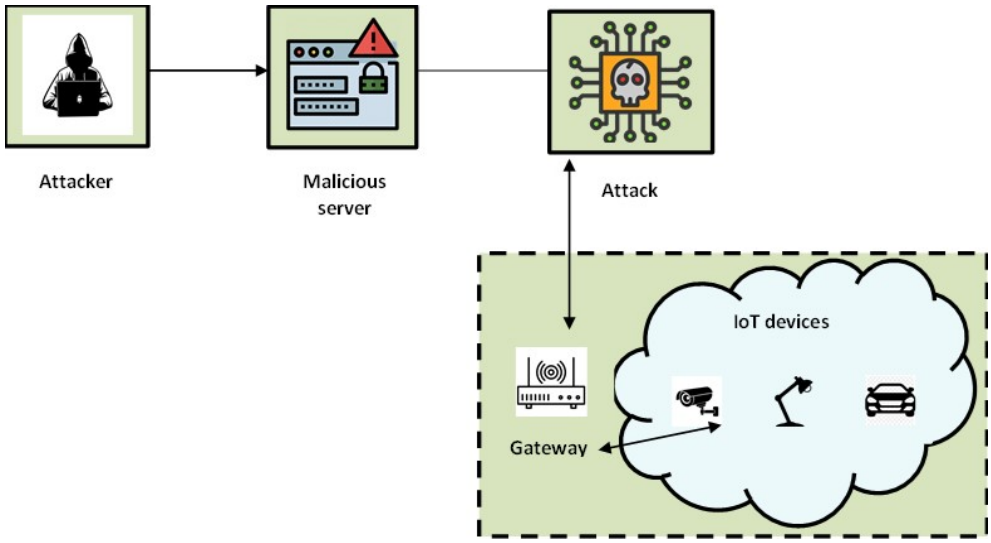


Fig. 1. *System model of malware variant traffic identification model*

4. Proposed Modified Scaling Wide Residual Network (SWideRNet) for malware detection

Malware detection is the process of locating and evaluating malicious software to stop it from harming systems. If malware is detected, then it is necessary to apply mitigation techniques to neutralize or eradicate threats found to lessen their influence on the attacked systems. The key intention of this work

is to devise an effectual method called Modified SWideRNet for malware detection and mitigation. The process begins by collecting input log file data from the dataset. Next, z-score normalization is used to standardize the data in the data normalization process. After that, feature selection is done using MI, which helps identify the most important features for the analysis. Then, malware detection is performed with the Modified SWideRNet framework to spot potential threats. Here, the SWideRNet is modified by including a Cross correntropy loss function. Finally, the same Modified SWideRNet model is used to mitigate the impact of the detected malware. Fig. 2 illustrates the schematic view of the proposed Modified SWideRNet for malware detection.

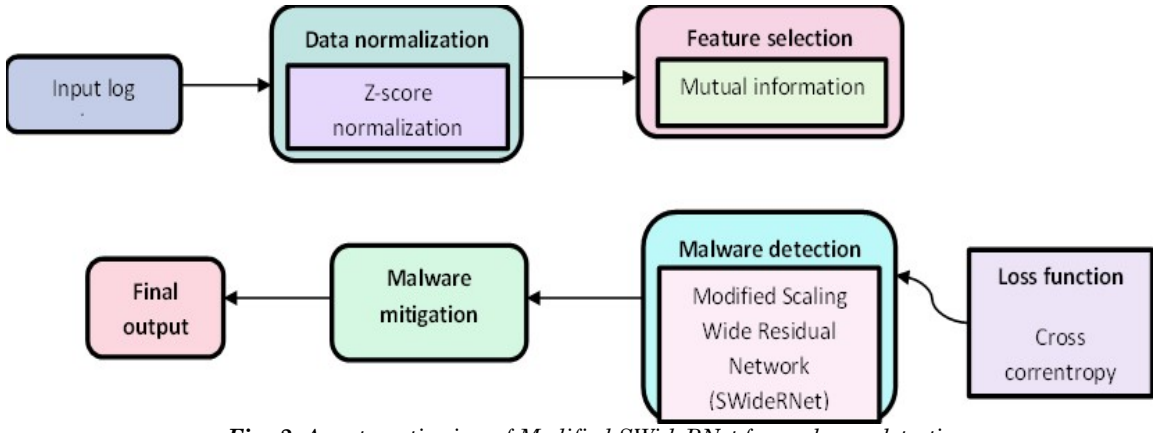


Fig. 2. *A systematic view of Modified SWideRNet for malware detection*

4.1 Log Data Acquisition

The input log file data consisting of IP address and log data is taken from the malware detection dataset [9]. The dataset is articulated as follows,

$$G = \left[G_1, G_2, ..., G_k, ..., G_q \right] \tag{1}$$

In the above equation, the malware detection database is denoted as G , the total number of log files is specified as q , and k^{th} log file considered for malware detection is demonstrated as G_k .

4.2 Data Normalization

The process of aligning a dataset's values to a common scale without distorting the ranges of values within it is known as data normalization. Typically, data normalization is performed to clean up the data in the input data G_k and assist in efficient decision-making. In this research, Z-score normalization [6] is exploited for normalizing the data.

Z-Score Normalization

Data can be transformed to have a standard deviation of one and a mean of zero by employing the Z-Score Normalization method [6]. This technique is frequently used in ML to ensure that features contribute equally to model training. It is especially helpful when the data has a normal distribution. Further, the expression for h_k data normalized by Z-score normalization is stated beneath,

$$h_k = \frac{G_k - \varpi}{m} \tag{2}$$

In this case, input is denoted as G_k , the mean of the dataset is represented as ϖ , and the standard deviation of the dataset is signified by m .

4.3 Feature Selection

The process of finding and choosing a subset of relevant features is known as feature selection. Its objective is to enhance the performance of the DL models by removing irrelevant or unnecessary features. Here, MI [7] is used to select the relevant features from the normalized log data file G_k .

Mutual information

MI [7] is a measure from information theory that enumerates the amount of information acquired about one random variable through another random variable. The degree to which knowing one variable lowers

uncertainty about the other is indicated by the assessment of the dependency between two variables. The MI $\mu(W,d)$ among two distinct random variables W and d is specified beneath,

$$\mu(W;d)=\sum_{\zeta\in W}\sum_{\varepsilon\in d}b(\zeta,\varepsilon)\log_2\frac{b(\zeta,\varepsilon)}{b(\zeta)b(\varepsilon)} \tag{3}$$

In the above equation, the target is designated by ε, the feature is specified by ζ, and $b(\zeta,\varepsilon)$ represents the joint probability distribution of W and d. Also, $b(\zeta)$ indicates marginal probability distribution of W and $b(\varepsilon)$ signifies marginal probability distribution of d. The feature selected outcome is demonstrated by the term g_R.

4.4 Malware Detection

The process of detecting malicious software, or malware, that can compromise systems, steal data, or interfere with operations is known as malware detection. Detecting malware is essential for protecting data and systems from online threats. Maintaining user trust in digital environments, ensuring compliance with security regulations, and preventing data breaches are possible by effective detection. Robust cybersecurity strategies require an advanced detection method as malware constantly evolving. In this paper, an effective technique called Modified SWideRNet is employed for detecting malware efficiently. The SWideRNEt [8] is modified by incorporating a Cross correntropy loss function to attain the Modified SWideRNet model. Here, the feature-selected output g_R is fed as an input for detecting the malware effectively.

4.4.1 Architecture of SWideRNet

WRNs are an enhancement of conventional Residual Networks (ResNets) designed to enhance training efficacy and model performance, particularly for DL tasks. SWideRNet framework can scale efficiently with increasing complexity and data size. WRNs can be trained faster while maintaining performance, leading to improved efficiency in model training. Outstanding malware detection performance has been shown by the WRN. The de facto network backbone for malware detection has been the Wide-ResNet-38 (WR-38), which has been improved by numerous human-crafted networks and extensive experiments. By eliminating the final residual block and repeating the second last residual block twice, Wide-ResNet-41 (WR-41) outperforms the WR-38 in both accuracy and speed.

Expanding upon WR-41, the baseline model is formed by including the SAC and the SE module, which are simplified. In specific terms, the channel attention map J within the SE module is calculated in this way:

$$J=H\left(\chi\varsigma\right) \tag{4}$$

Here, ς stands for the input feature map that is globally average-pooled, and the fully connected layer's weights are represented by χ. Additionally, the hard sigmoid function is described as follows,

$$H\left(\eta\right)=\frac{\mathrm{Re}\,LU6\left(\eta+3\right)}{6} \tag{5}$$

Essentially, features computed with various Atrous rates are gathered by the SAC operation. Especially, $s=Conv(\eta,\iota,\ell)$ is employed to signify the convolutional operation with input η, weights ι, output s, and Atrous rate ℓ. SAC combines two feature maps using the switch function ω:

$$\left(1-\omega(\eta)\right)\cdot conv(\eta,\iota,1)+\omega(\eta)\cdot conv(\eta,\iota,3) \tag{6}$$

Here, $\ell=1$ and 3 are exploited for two convolutions. Both location and input determine the switch function ω, which is carried out as a 1×1 convolution after an 5×5 average pooling. Two global context modules should also be inserted before and after the SAC's primary operation. These global context modules are executed as fully connected layers after global average pooling, and they are lightweight.

Scaling factors: To scale the baseline model's network capacity, the scaling factors $(\iota_1,\iota_2,\varphi)$ are adopted. Here, the channels of the primary two phases are represented by the scaling factor ι_1, the remaining stages designated as conv3, conv4, conv5, and conv6 are scaled by ι_2 and φ in terms of channels and layers. SWideRNet with $(\iota_1,\iota_2,\varphi)$ scaling factors is the resultant family of networks, which is known as SWideRNet-$(\iota_1,\iota_2,\varphi)$. The output produced from SWideRNet is symbolized by λ_S. Fig. 3 portrays the architecture of the SWideRNet model.

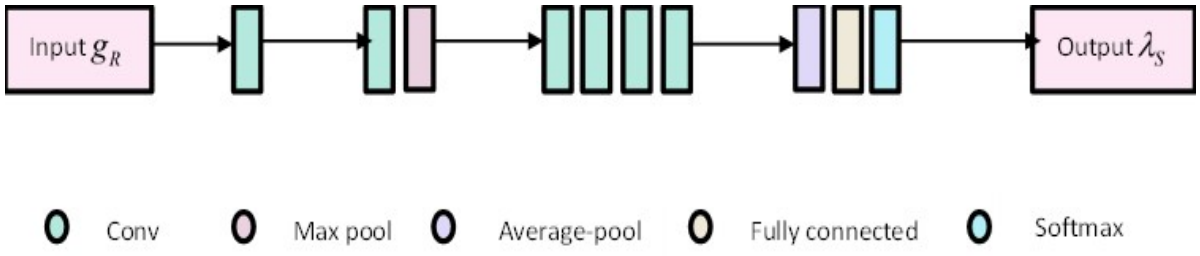


Fig. 3. *The architecture of SWideRNet model*

4.4.2 Cross Correntropy

The degree of similarity between two random variables within a constrained area defined by the kernel width β is expressed as correntropy [12]. It is important to select the loss function so that maximizing correntropy minimizes the expected risk. The Correntropy is specified as the C-loss function γ_C or induced loss function is shown below,

$$\gamma_C\left(K,t(p)\right)=\hbar\left[1-Q_x\left(K-t(p)\right)\right] \tag{7}$$

where, the positive scaling constant is represented as $\hbar$, the kernel width parameter is indicated as x, the true label is indicated as K, and the predicted label is signified by $t(p)$.

4.5 Malware Mitigation

Several techniques are used in malware mitigation to lessen the harm that malicious software causes to networks and systems. Employing detection tools to find malware early and having a response plan to efficiently handle any breaches are important measures. Antivirus software and firewalls are used to prevent infections. Users can also be protected against vulnerabilities by updating software frequently and learning about safe online practices. Organizations can improve their malware defenses and reduce potential damage by placing these techniques into practice. In this research, the Modified SWideRNet model is employed to mitigate the malware effectively. The detailed description of the Modified SWideRNet model is already specified in section 4.4.1.

5. Results and Discussion

This section provides an elaborate description of the experimental outcomes of the devised Modified SWideRNet. Furthermore, the experimental setup, evaluation measures, and comparative assessment of the designed technique are elucidated elaborately.

5.1 Experimental Setup

The experimentation of the introduced Modified SWideRNet is conducted using a Python tool on a malware detection dataset [9].

5.2 Dataset Description

Samples collected for training and assessing malware detection models are called malware detection datasets [9]. By including both benign and malicious files in their datasets, it is possible to create algorithms that can differentiate between safe and dangerous software. A wide range of behaviors and characteristics are covered by the dataset, which includes both benign applications and different types of malware. This dataset also includes 51 submissions from 14 teams, 16 participants, and 36 entrants.

5.3 Evaluation Metrics

The efficacy of the developed Modified SWideRNet is evaluated based on various parameters, including accuracy, precision, and recall. A summary of these metrics is provided below.

a) Accuracy

Accuracy is defined as the percentage of true results that are both true positives and true negatives from the total number of cases examined. The expression for accuracy is articulated as follows,

$$Accuracy = \frac{r_{tp} + r_{tn}}{r_{tp} + r_{tn} + r_{fp} + r_{fn}} \tag{8}$$

where r_{tp} specifies correctly identified malware cases, r_{tn} indicates correctly identified benign instances, r_{fp} represents incorrectly identified benign instances, and r_{fn} exemplifies incorrectly identified malware instances.

b) Precision

The percentage of actual positive outcomes among all of the model's positive predictions is known as precision. Moreover, precision represents the accuracy of the positive predictions, and the expression is stated as follows,

$$precision = \frac{r_{tp}}{r_{tp} + r_{fp}} \tag{9}$$

c) Recall

The percentage of true positive outcomes among instances detected as positive is measured by recall. It shows the degree to which the model can identify malware cases. The recall is expressed as follows.

$$recall = \frac{r_{tp}}{r_{tp} + r_{fn}} \tag{10}$$

5.4 Deviation Results

Fig. 4 indicates the assessment contemplating the deviation of the different features contained in the malware detection dataset used by the Modified SWideRNet in malware detection and mitigation. In this case, the graph is displayed between features and the number of samples. With a sample size of 60,000, the value of the free_area_cache feature ranges from 0 to 0.5, while the value of the mm_users feature varies from 1.4 to 2.7. Additionally, the value of nivcsw is within the range of 0 to 0.5, the value of the reserved_vm feature ranges from 0 to 1.7, and the value of gtime feature ranges from 0 to 0.1.

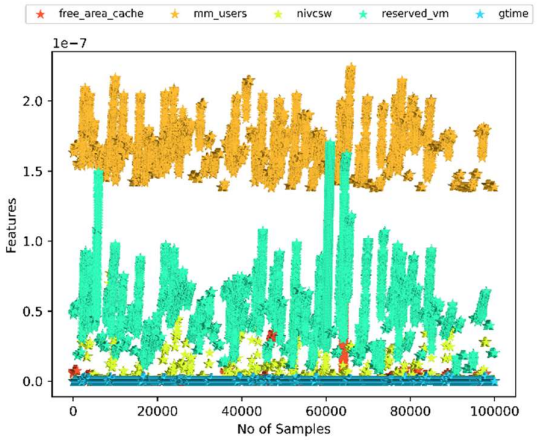


Fig. 4. *Deviation results of Modified SwideRNet*

5.5 Performance Analysis

Fig. 5 illustrates the performance evaluation of the Modified SWideRNet model by using various evaluation parameters. The examination of the Modified SWideRNet model about accuracy metric is specified in Fig. 5a). By assuming the learning set as 90%, the accuracy attained by Modified SWideRNet is 83.457%, 85.101%, 89.346%, and 91.435% for the epochs 10, 20, 30, and 40. The investigation of Modified SWideRNet concerning precision is demonstrated in Fig. 5b). In this case, Modified SWideRNet measured precision of 82.045%, 85.135%, 88.135%, and 90.133% for the epochs of 10, 20, 30, and 40 by considering the learning set as 90%. Fig. 5c) depicts the evaluation of Modified SWideRNet regarding recall. The Modified SWideRNet framework calculated recall of 85.768% for 10 epochs, 88.455% for 20 epochs, 90.134% for 30 epochs, and 92.344% for 40 epochs for the learning set of 90%.

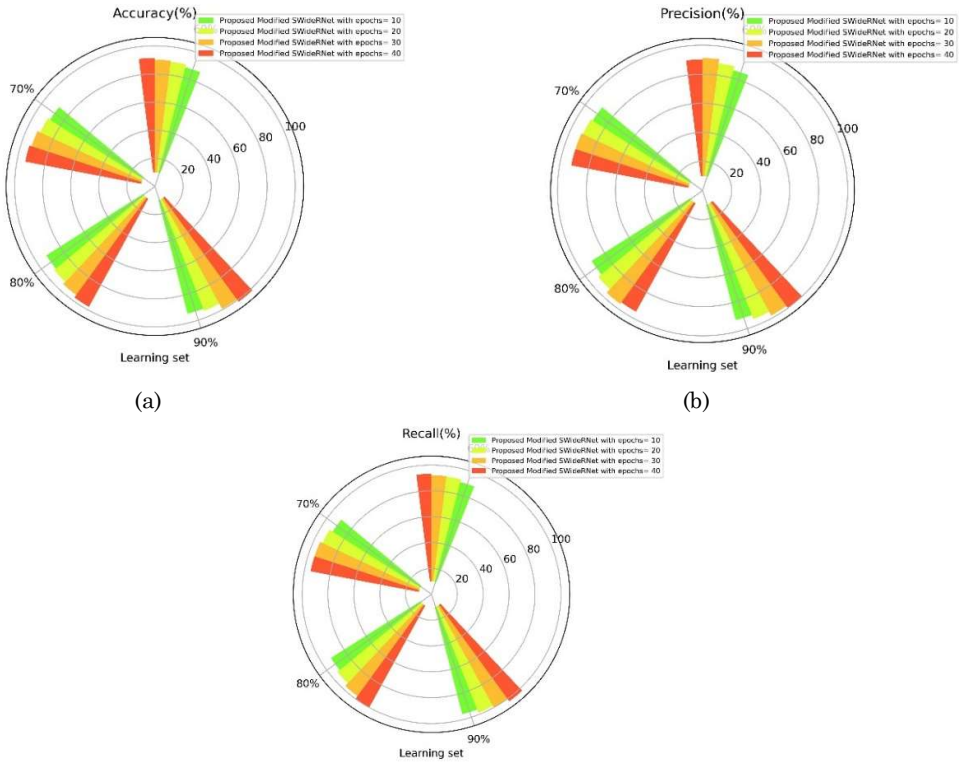


Fig. 5. *Valuation of Modified SWideRNet by employing several epochs, a) accuracy, b) precision, and c) recall*

5.6 Comparative Techniques

The effectiveness of the developed Modified SWideRNet approach is examined by contrasting it with other schemes, such as EfficientNetB1 [1], ECDLP [2], RHSODL-AMD [3], and RMDNet [5].

5.7 Comparative Analysis

The devised Modified SWideRNet is examined for its efficiency based on several measures, such as accuracy, precision, and recall by altering the learning set. The comparative evaluation of Modified SWideRNet is executed for the determination of performance by exploiting various evaluation metrics and is illustrated in Fig. 6. The valuation of Modified SWideRNet concerning accuracy is displayed in Fig. 6a). By contemplating the learning set of 90%, the Modified SWideRNet computed accuracy of 91.435%, while existing approaches quantified accuracy of EfficientNetB1 is 79.546%, ECDLP is 82.001%, RHSODL-AMD is 84.134%, and RMDNet is 88.540%. The Modified SWideRNet attained a higher performance of 2.935% when correlated with the RMDNet model. Fig. 6(b) displays the examination of Modified SWideRNet regarding precision. Here, precision recorded by Modified SWideRNet is 90.133%, whereas classical techniques gained precision are 78.646%, 80.245%, 83.772%, and 86.346% by EfficientNetB1, ECDLP, RHSODL-AMD, and RMDNet for the learning set of 90%. The Modified SWideRNet attained a higher performance of 6.361% as correlated with the RHSODL-AMD technique. The investigation of Modified SWideRNet concerning recall is demonstrated in Fig. 6(c). By assuming a learning set of 90%, the malware detection techniques, such as EfficientNetB1, ECDLP, RHSODL-AMD, RMDNet, and Modified SWideRNet calculated recall of 0.438, 0.391, 0.352, 0.304, and 0.286. The Modified SWideRNet attained an extreme performance of 7.758% as contrasted with the ECDLP model.

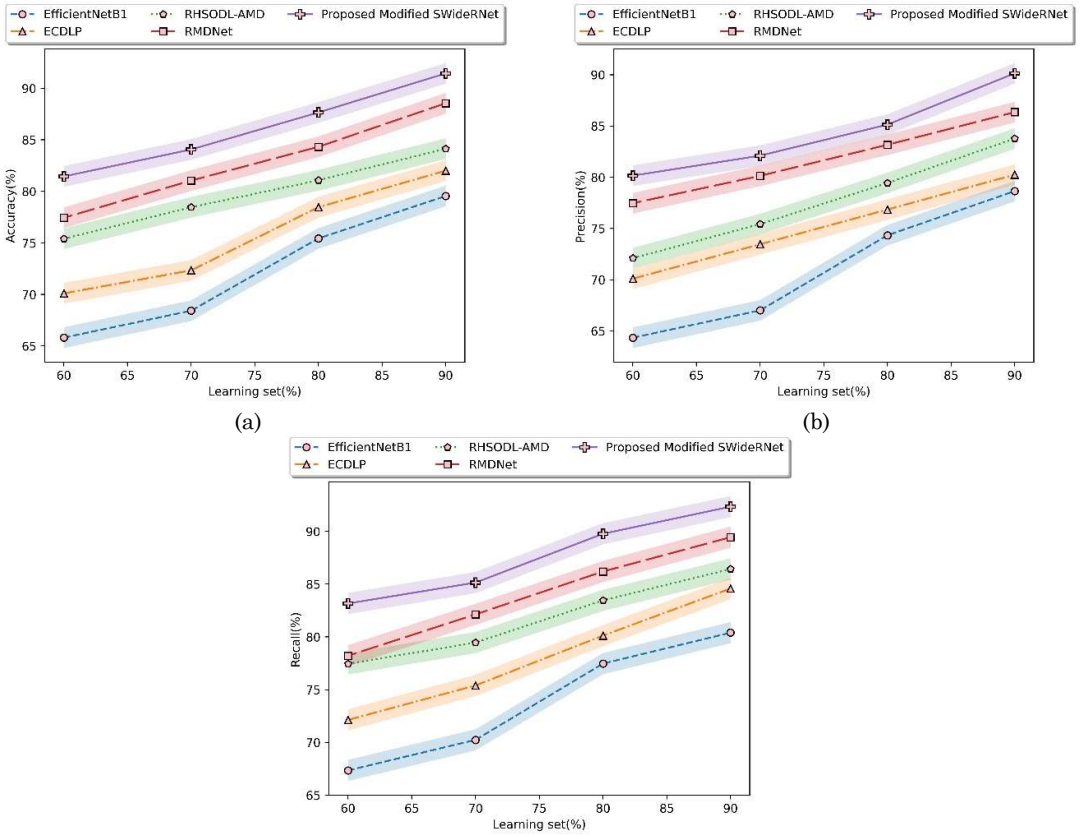


Fig. 6. *Assessment of Modified SWideRNet by altering learning set, a) accuracy, b) precision, and c) recall*

5.8 Comparative Discussion

The experimental outcomes computed by Modified SWideRNet and prevailing malware detection methods are displayed in Table 1. The Modified SWideRNet model employed for detecting malware attained effective results with an accuracy of 91.435%, precision of 90.133%, and recall of 92.344%. The accuracy quantified by the existing approaches of EfficientNetB1 is 79.546%, ECDLP is 82.001%, RHSODL-AMD is 84.134%, and RMDNet is 88.540%. The precision acquired by prevailing approaches like EfficientNetB1, ECDLP, RHSODL-AMD, and RMDNet is 78.646%, 80.245%, 83.772%, and 86.346%. Moreover, the malware detection techniques, such as EfficientNetB1, ECDLP, RHSODL-AMD, and RMDNet calculated recall of 80.401%, 84.586%, 86.432%, and 89.446%. Therefore, the Modified SWideRNet model exhibited scalability, making it appropriate for diverse environments and changing data types, while Cross-correntropy is less sensitive and is suitable for noisy environments.

Table 1. *Comparative discussion of Modified SWideRNet*

Parameters	EfficientNetB1	ECDLP	RHSODL-AMD	RMDNet	Proposed Modified SWideRNet
			For learning set as 90%		
Accuracy (%)	79.546	82.001	84.134	88.540	**91.435**
Precision (%)	78.646	80.245	83.772	86.346	**90.133**
Recall (%)	80.401	84.586	86.432	89.446	**92.344**

6. Conclusion

The increasing prevalence of malware poses significant risks to computer systems and networks, leading to data breaches and operational disruptions. Malware is evolving so quickly and becoming more sophisticated to the point where it can avoid detection with conventional techniques, cybersecurity is facing enormous challenges. This work aims to develop an effective system for detecting and mitigating malware threats, ensuring enhanced security and resilience against potential attacks. At first, input log file data is obtained from the given database. Afterward, data normalization is executed through z-score normalization for standardizing the feature values. Next, feature selection is carried out employing MI to

identify the most relevant features. Following this, malware detection is done using the Modified SWideRNet framework to recognize potential threats. Lastly, the identified malware is mitigated by the Modified SWideRNet framework. Furthermore, the devised Modified SWideRNet has outperformed prevailing models by attaining accuracy, precision, and recall with values of 91.435%, 90.133%, and 92.344%. Future work aims to contemplate hybrid DL techniques to enhance the accuracy and speed of malware detection. Moreover, effective datasets will be included to enhance the performance of the technique.

Compliance With Ethical Standards

Conflicts of interest: Authors declared that they have no conflict of interest.

Human participants: The conducted research follows ethical standards and the authors ensured that they have not conducted any studies with human participants or animals.

References

[1] R. Chaganti, V. Ravi, and T. D. Pham, "Image-based malware representation approach with EfficientNet convolutional neural networks for effective malware classification", Journal of Information Security and Applications, vol. 69, pp.103306, 2022.
[2] M. N. Al-Andoli, K. S. Sim, S. C. Tan, P. Y. Goh, and C. P. Lim, "An ensemble-based parallel deep learning classifier with PSO-BP optimization for malware detection", IEEE Access, 2023.
[3] A. Albakri, F. Alhayan, N. Alturki, S. Ahamed, and S. Shamsudheen, "Metaheuristics with deep learning model for cybersecurity and Android malware detection and classification", Applied Sciences, vol. 13, no. 4, pp.2172, 2023.
[4] M. A. Marzouk, and M. Elkholy, "Deep image: An efficient image-based deep conventional neural network method for android malware detection", J Adv Inf Technol, vol. 14, no. 4, pp.838-45, 2023.
[5] S. Puneeth, S. Lal, Singh, and B. S. Raghavendra, "RMDNet-Deep Learning Paradigms for Effective Malware Detection and Classification", IEEE Access, 2024.
[6] H. Henderi, T. Wahyuningsih, and E. Rahwanto, "Comparison of Min-Max normalization and Z-Score Normalization in the K-nearest neighbor (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer", International Journal of Informatics and Information Systems, vol. 4, no. 1, pp.13-20, 2021.
[7] J. Gonzalez-Lopez, S. Ventura, and A. Cano, "Distributed multi-label feature selection using individual mutual information measures", Knowledge-Based Systems, vol. 188, pp.105052, 2020.
[8] L. C. Chen, H. Wang, and S. Qiao, "Scaling wide residual networks for panoptic segmentation", arXiv preprint arXiv:2011.11675, 2020.
[9] The malware detection dataset is taken from "https://www.kaggle.com/datasets/nsaravana/malware-detection", accessed on October 2024.
[10] X. Hu, C. Zhu, G. Cheng, R. Li, H. Wu, and J. Gong, "A deep subdomain adaptation network with attention mechanism for malware variant traffic identification at an IoT edge gateway", IEEE Internet of Things Journal, vol. 10, no. 5, pp.3814-3826, 2022.
[11] Ö. Aslan, and A. A. Yilmaz, "A new malware classification framework based on deep learning algorithms", IEEE Access, vol. 9, pp.87936-87951, 2021.
[12] A. Singh, R. Pokharel, and J. Principe, "The C-loss function for pattern classification", Pattern Recognition, vol. 47, no. 1, pp.441-453, 2014.
[13] A. Jamal, M. F. Hayat, and M. Nasir, "Malware detection and classification in IoT network using ANN", Mehran University Research Journal of Engineering & Technology, vol. 41, no. 1, pp.80-91, 2022.
[14] M. S. Akhtar, and T. Feng, "Detection of malware by deep learning as CNN-LSTM machine learning techniques in real time", Symmetry, vol. 14, no. 11, pp.2308, 2022.
[15] R. Damaševičius, A. Venčkauskas, J. Toldinas, and Š. Grigaliūnas, "Ensemble-based classification using neural networks and machine learning models for Windows PE malware detection", Electronics, vol. 10, no. 4, pp.485, 2021.