Computer Systems
Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2023.034034
Article

Learning-Based Artificial Algae Algorithm with Optimal Machine Learning Enabled Malware Detection

Khaled M. Alalayah[1], Fatma S. Alrayes[2], Mohamed K. Nour[3], Khadija M. Alaidarous[1], Ibrahim M. Alwayle[1], Heba Mohsen[4], Ibrahim Abdulrab Ahmed[5] and Mesfer Al Duhayyim[6,*]

[1]Department of Computer Science, College of Science and Arts, Najran University, Sharurah, Saudi Arabia
[2]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia
[3]Department of Computer Sciences, College of Computing and Information System, Umm Al-Qura University, Saudi Arabia
[4]Department of Computer Science, Faculty of Computers and Information Technology, Future University in Egypt, New Cairo, 11835, Egypt
[5]Computer Department, Applied College, Najran University, Najran, 66462, Saudi Arabia
[6]Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Al-Kharj, 16273, Saudi Arabia
*Corresponding Author: Mesfer Al Duhayyim. Email: m.alduhayyim@psau.edu.sa

Received: 04 July 2022; Accepted: 22 November 2022

Abstract: Malware is a ‘malicious software program that performs multiple cyberattacks on the Internet, involving fraud, scams, nation-state cyberwar, and cybercrime. Such malicious software programs come under different classifications, namely Trojans, viruses, spyware, worms, ransomware, Rootkit, botnet malware, etc. Ransomware is a kind of malware that holds the victim’s data hostage by encrypting the information on the user’s computer to make it inaccessible to users and only decrypting it; then, the user pays a ransom procedure of a sum of money. To prevent detection, various forms of ransomware utilize more than one mechanism in their attack flow in conjunction with Machine Learning (ML) algorithm. This study focuses on designing a Learning-Based Artificial Algae Algorithm with Optimal Machine Learning Enabled Malware Detection (LBAAA-OMLMD) approach in Computer Networks. The presented LBAAA-OMLMD model mainly aims to detect and classify the existence of ransomware and goodware in the network. To accomplish this, the LBAAA-OMLMD model initially derives a Learning-Based Artificial Algae Algorithm based Feature Selection (LBAAA-FS) model to reduce the curse of dimensionality problems. Besides, the Flower Pollination Algorithm (FPA) with Echo State Network (ESN) Classification model is applied. The FPA model helps to appropriately adjust the parameters related to the ESN model to accomplish enhanced classifier results. The experimental validation of the LBAAA-OMLMD model is tested using a benchmark dataset, and the outcomes are inspected in distinct measures. The comprehensive comparative examination demonstrated the betterment of the LBAAA-OMLMD model over recent algorithms.

Keywords: Computer networks; machine learning; security; malware detection; feature selection; ransomware

1 Introduction

The network has changed greatly with the advancement in Network Function Virtualization (NFV) and Software Defined Networks (SDN). It produces a large volume of data in day-to-day life routines [1]. Thus, it becomes highly complicated to physically examine every piece of data by the network specialist and to determine whether the network is sufficiently good for managing every piece of data or if a modification is needed. Additionally, security problems have been consistently rising in the network due to several malicious users and hackers. Thus, a strong security system mandates protecting data from malicious users and hackers [2]. At present, Machine Learning (ML) systems are utilized for the management of the network by numerous authors. Cybersecurity is a primary concern in advancing networking technologies and computers [3,4]. The criminals are being smart and launching new menaces; a substantial investigation was placed to expand the counteractions to rescue organizations and individuals from these damages. Crypto-viruses and Crypto virology ideas were launched back in 1996 [5].

Malware is a malicious program focused on accumulating delicate data, occurring destruction or causing trouble to sole or many users [6]. It can be initially monitored in the late 1970s. It generally has accessibility to legal sources to make problems for performing ordinary activities. Ransomware is regarded as a type of malware that affects the user by encoding data without the user's knowledge. It confines the legal accessibility to user data [7,8]. It halts the accessibility of users to their sources, that is, data. Ransomware assaults are scattered because of their monetary incentives and lethal effects [9,10]. As it utilizes the lethal grouping of 2 policies to assault, this malware is very hard to manage. Few ransomware assaults utilize asymmetric cryptography for encoding in addition to erasing the shadow copies and recovery points. One such important characteristics of ransomware are that it resembles a benevolent program, becoming complex to differentiate ransomware code from legal encryption applications [11].

Ransomware assaults might be troublesome in dispersed atmospheres to halt undisturbed work between heterogeneous data centres. Such systems contain complicated structures of algorithms and corpora. These surroundings that are data centres have large scales of data and could pay money to ignore the reputation and corruption of data [12,13]. ML could potentially identify malware not just in Windows operating systems but also in Android systems [14]. Additional ML research into malware recognition as an alternative to the usage of signs is offered to screen the efficacy of utilizing ML-related detection than signature-related methods. The choice to assess ML and deep learning (DL) methods as against other non-ML related methods was taken due to its flexibility and robust ability to spot hidden ransomware malware samples [15].

This study focuses on designing the Learning-Based Artificial Algae Algorithm with Optimal Machine Learning Enabled Malware Detection (LBAAA-OMLMD) approach in Computer Networks. The presented LBAAA-OMLMD model derives a Learning-Based Artificial Algae Algorithm based Feature Selection (LBAAA-FS) model to reduce the curse of dimensionality problems. Besides, the Flower Pollination Algorithm (FPA) with Echo State Network (ESN) Classification model is applied. The FPA model helps to appropriately adjust the parameters related to the ESN model to accomplish enhanced classifier results. The experimental validation of the LBAAA-OMLMD model is tested using a benchmark dataset, and the results are inspected under special measures.

2 Related Works

In Aurangzeb et al. [16], a BigRC-EML technique is proposed to detect and classify ransomware dependent upon static and dynamic characteristics. It can utilize ensemble ML methodologies on big datasets to enhance the detection of ransomware's performance. Even though several ML techniques were employed in ransomware detection, but still, the assessment of the ensemble model hasn't been inspected. Furthermore, a new FS technique based on principal component analysis (PCA) is proposed to decrease the feature's dimension. Egunjobi et al. [17] illustrated a classification model incorporating static and dynamic features for improving the performance of classification and detection of ransomware. Then trained supervised ML algorithm with a testing set and applied a confusion matrix for observing accuracy, which enables a systematic comparison of all the algorithms.

In Khan et al. [18], a DNAact-Ran, a Digital deoxyribonucleic acid (DNA) Sequencing Engine, is proposed to detect Ransomware through ML. DNAact-Ran makes use of k-mer frequency vector and Digital DNA sequencing design constraint. Daku et al. [19] utilize an ML classifier to identify an adapted ransomware version. To carry out the research, behavioural reports of 150 ransomware samples were employed from ten distinct ransomware categories. An iterative method is utilized for identifying optimal behavioural attributes utilized for achieving the optimal accuracy of the classification. Lee et al. [20] make use of an entropy model for measuring the characteristics of the encrypted file (uniformity). The ML method is utilized to classify the infected file based on the analysis of file entropy. The presented technique recovers the new file from the backup systems by identifying ransomware-affected files that are synchronizing to the backup systems, even when the user system is affected by ransomware.

Kok et al. [21] developed a pre-encryption detection approach (PEDA) to detect crypto-ransomware preceding the existence of encryption. The PEDA contains 2 levels of recognition. The initial one is a signature repository (SR) that recognizes the signature matches with known ransomware. The next one is a learning algorithm (LA) that detects known and unknown crypto-ransomware. Sharma et al. [22] presented an architecture which utilizes the novel feature of Android ransomware, employs an ML model to categorize ransomware and benign applications, and implements a comparison analysis to evaluate the computation time needed by the ML model for detecting Android ransomware.

3 The Proposed Model

In this study, an effectual LBAAA-OMLMD model has been developed to identify and classify ransomware in Computer Networks. The presented LBAAA-OMLMD model follows a three-stage process: LBAAA-based feature selection, ESN-based classification, and FPA-based parameter tuning. Fig. 1 illustrates the block diagram of the LBAAA-OMLMD approach.

3.1 Design of LBAAA-FS Model

At the initial stage, the LBAAA-OMLMD model employed the LBAAA-FS model to reduce the curse of dimensionality problems. AAA is the newly designed population-based optimization technique stimulated by the survival skill of algae [23]. Evolutionary, adaption, and helical movement establishes the AAA are the three basic processes. The algae position receives adequate light and is regarded as an optimum global point. Beforehand entering the algorithm, the process begins with the primary solution, and after that, the fitness is calculated. Next, the colony size of algae has been calculated by the following equations.

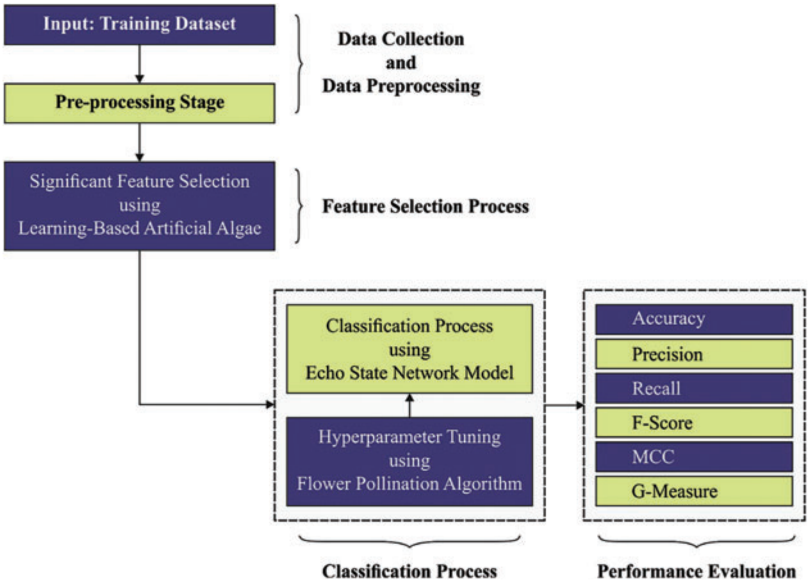

Input: Training Dataset
Pre-processing Stage
Data Collection and Data Preprocessing
Significant Feature Selection using Learning-Based Artificial Algae
Feature Selection Process
Classification Process using Echo State Network Model
Hyperparameter Tuning using Flower Pollination Algorithm
Accuracy
Precision
Recall
F-Score
MCC
G-Measure
Classification Process
Performance Evaluation

Figure 1: Block diagram of LBAAA-OMLMD technique

$$CS' = \mu_i \times CS \tag{1}$$

$$\mu = \frac{\mu_{\max} S}{K_s + S} \tag{2}$$

Let CS be the size of *the* i^{th} algal colony, μ indicates the growth rate, $\mu_{\max}$ denotes the maximal growth rate, S represents the nutrient amount, i.e., the fitness value, and K_s refers to a constant demonstrating substrate half saturation of colonies. The helical movement is the movement that the algal cell undertakes from the present location to the surface of the water to absorb adequate light based on the consumed energy level and the frictional surface. Therefore, once the algal cells approach the surface, it implies that it has more energy consumption when compared to others. Different from the abovementioned case, the friction surface is lesser; they explore better globally and cover a long distance. The AAA considers that the gravity is 0. The location of certain algal cells depends on the force dragging their motion in the liquid, that is, shear force and the friction surface. The higher algae (CS') size, the greater the shear force.

$$\tau\left(x_i\right) = 2\pi \left(\sqrt[3]{\frac{3CS}{4\pi}}\right)^2 \tag{3}$$

In AAA, a novel solution candidate was produced by simulating the helical motion of algal cells that involves angular and linear movements. The tournament selection model recognizes the neighbour. Next, the weight difference is employed on arbitrarily chosen three variables.

$$w_{ip}\left(t+1\right) = w_{ip} + \left(w_{jp} - w_{ip}\right)\left(\Delta - \tau\left(w_i\right)\right)\rho \tag{4}$$

$$w_{iq}\left(t+1\right) = w_{iq} + \left(w_{jq} - w_{iq}\right)\left(\Delta - \tau\left(w_i\right)\right) cos\alpha \tag{5}$$

$$w_{ir}\left(f+1\right) = w_{ir} + \left(w_{jr} - w_{ir}\right)\left(\Delta - \tau\left(w_i\right)\right) \sin\beta \tag{6}$$

Let w_{ip}, w_{iq}, w_{ir} be the present solution chosen randomly, and w_j stands for neighbor algal colony recognized using tournament selection; $\alpha,\ \beta C\ \ [0, 2\pi]$, Δ indicates the shear force; τ (w_i) denotes the friction surface region of i^{th} algal cells and $\rho\ \in\ [-1, 1]$. The fitness is estimated, and depending on the fitness values; greedy choices take place for deciding the best solution among the new and current solutions.

In the evolutionary process, there are sufficient nutrients, and the colony gets adequate light, an alga reproduced into two novel algal cells. Or else the algal colony passes away afterwards sometimes. The algal colony continues to grow bigger as it provides better solutions continuously. The subsequent expression recognizes the smallest and biggest colonies.

$$Biggest\ Colony = \ \max\ (CS) \tag{7}$$

$$Smallest\ Colony = \ \min\ (CS) \tag{8}$$

An arbitrarily designated algal cell between the smallest colonies is a candidate for reproduction. The evolutionary procedure was decided afterwards; the bigger group talked over the location of the smaller one and arranged by size.

$$Smallest\ Colony = Biggest\ Colony \tag{9}$$

Adaption is a method where an algal colony hasn't grown adequately to attempt to survive. The colony that has a good solution continues to grow. However, the colony doesn't result in good solutions and becomes more starved; thus, the A_i starvation level is increased. The algal cell that is starved most is selected for adoption as follows.

$$Starving = \ \max\ (A_i) \tag{10}$$

$$Starving\ \ (t + 1) = starving + \Gamma alnd \times (biggest - starving) \tag{11}$$

From the equation, A_i represents the starvation value of i^{th} algal colony; starving characterizes the colony with maximal starvation levels. The adoption variable, A_p, refers to the constant value ranges from [0, 1] and defines whether to get into the adoption process or not.

The fitness function (FF) of LBAAA assumes classifier accuracy and the number of selected features. It maximizes the classifier accuracy and minimizes the set size of chosen features. Thus, the subsequent FF has been utilized for evaluating individual solutions, as illustrated in Eq. (12).

$$Fitness = \alpha * ErrorRate + (1 - \alpha) * \frac{\#SF}{\#All\_F} \tag{12}$$

Whereas $ErrorRate$ implies the classifier error rate utilizing the chosen features. $ErrorRate$ has been computed as the percentage of incorrect classification to the number of classifiers made, formulated as the value between zero and one. ($ErrorRate$ is the complement of classifier accuracy), $\#SF$ denotes the amount of chosen features and $\#All\_F$ indicates the entire amount of attributes from the original dataset.

3.2 Ransomware Classification Using ESN Model

Once the features are chosen, they are given as input into the ESN model for effective ransomware classification. ESN is comprised of a reservoir, output, and input layers. The reservoir comprises hundreds of sparsely linked neurons, and links weighted amongst neurons were arbitrarily created and set [24]. The state, as well as the resultant formula of ESN, are as follows:

$$x(t) = \phi\left(W_{in}\, u(t) + W_x x(t-1) + W_{back}\, y(t-1)\right) \tag{13}$$

$$y(t) = f_{out}(W_{out}\left(u(t), x(t), y(t-1)\right)) \tag{14}$$

Whereas $u(t) \in R^{M\times 1}$ represents the input vector, $y(t) \in R^{M\times 1}$ stands for the resultant vector, $b_x \in R^{N\times 1}$ signifies the input bias, and $b \in R^{M\times 1}$ denotes the resultant bias. The state $x(t) \in R^{N\times 1}$ at present was computed in the input vector $u(t)$ at present τ and the state of reservoir $x(t-1)$ at the preceding time $t-1$. $\phi(\cdot)$ denotes the activation function of neurons that is choosing the Sigmoid or *tanh* functions. The component of input-reservoirs linked to the weighted matrix $W_{in} \in R^{N\times K}$ is in the interval of -1 and 1. $W_x \in R^{N\times N}$ implies the internal linking weighted matrix of reservoirs. $W_{back} \in R^{N\times L}$ denotes the output-reservoir linking weighted matrix. $W_{out} \in R^{L\times(K+N\times L)}$ defines the resultant linking weighted matrix. W_{in}, W_x, and W_{back} were created arbitrarily and endured unchanged in the trained stage of ESN. The network only requires training the resultant linking weighted matrix W_{out} that decreases the computational complexity [25].

The reservoir encompasses masses of sparsely related neurons, and the connection weights among neurons were fixed and randomly generated. The principle of ESN is the reservoir. The efficiency is based on the 4 crucial hyperparameters: R, the spectral radius, N size of the reservoir, S input scale, and D sparse degree. How to choose this hyperparameter is highly significant.

(1) Size of reservoir: here, the paramount hyperparameter that affects the efficiency of ESN is the number of neurons from the reservoirs. The large the neuron number is, the superior the classifier accuracy. Overfitting will be caused when the neuron number is overlarge.
(2) Spectral radius: R spectral radius refers to the arbitrary values of the maximal eigenvalue of internal relation weight matrixes Wx of the reservoir. R < 1 is an essential state to guarantee network stability.
(3) Sparse degree: D sparse degree specifies the sparsity of neuron connection. The neuron in the reservoir is sparsely linked instead of fully connected. The large the value is, the strong the non-linear approximate capability.
(4) Input scale: S input scale denotes the scaling factor beforehand the dataset is inputted to the reservoir and signifies the range of input linking weight. Usually, the range of activation function is [0, 1].

The ESN performance heavily relies on the abovementioned hyperparameter, and outcomes attained by the hyperparameter configuration differ considerably.

3.3 Parameter Optimization Process

Finally, the FPA model helps to appropriately adjust the parameters related to the ESN model to accomplish enhanced classifier results. Global and local pollination are the two major phases of the FPA [26]. In global pollination, pollen grain is carried by pollinators like insects, and pollen grain travels longer distances since insects could move and fly for a longer distance. So, rules No. (1) and (3) are mathematically expressed in the following equation:

$$X_i^{t+1} = X_i^t + \gamma L(\lambda)\left(X_{best} - X_i^t\right) \tag{15}$$

where

X_i^t: The solution for X_i in cycle t

X_{best}: The optimal solution attained in cycle t, viz., the optimum solution that has been found amongst each solution in the existing generation.

γ: Scaling factor for controlling step size.

$L(\lambda)$: Pollination strength parameter.

X_i^{t+1}: The new solution.

L indicates the standard gamma function, and the distribution is utilized since it is appropriate for larger steps of the swarm.

$$L \sim \frac{\lambda\Gamma(\lambda)\sin\left(\frac{\pi\lambda}{2}\right)}{\pi}\frac{1}{s^{1+\lambda}}, \quad (s > s_0 > 0) \tag{16}$$

The value of s is evaluated by the following equation:

$$s = \frac{u}{|v|^{\lambda-1}} \tag{17}$$

$$u \sim N\left(0,\sigma^2\right), \; v \sim N(0,1) \tag{18}$$

$$\sigma^2 = \left[\frac{\Gamma(1+\lambda)}{\lambda\Gamma\left(\frac{1+\lambda}{2}\right)}\frac{\sin\ (\pi\lambda/2)}{2(\lambda-1)/2}\right]^{1/\lambda} \tag{19}$$

$$S\left(X_i^j(t)\right) = \frac{1}{1+e^{-X_i^j(t)}} \tag{20}$$

Generally the value of $s_0 = 0.1$.

In local pollination, the pollination is self-pollinating. It characterizes rule No. (2) and (3) arithmetically as:

$$X_i^{t+1} = X_i^t + \varepsilon\left(X_j^t - X_k^t\right) \tag{21}$$

where

X_i^t: The solution for X_i in cycle t

X_i^{t+1}: The new solution.

X_j^t, X_k^t: Pollen from mixed flowers on the same plant, k, j are arbitrarily designated. ε: an arbitrary parameter ranges from $U(0,1)$.

Afterwards the global and local rounds of pollination, the best flower was taken, and intensive exploitation was made, as follows:

$$X_i^{t+1} = X_{best} + H(\varepsilon_1 - [(\varepsilon_2 - \varepsilon_3)\,X_{best}] \tag{22}$$

where

H: control parameter that is evaluated as:

$$H = \begin{cases} 1, & if\ \varepsilon_4 < p \\ 0, & otherwise \end{cases} \tag{23}$$

where

$\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4$: random variable that ranges from $U(0,1)$. Fig. 2 illustrates the flowchart of FPA.

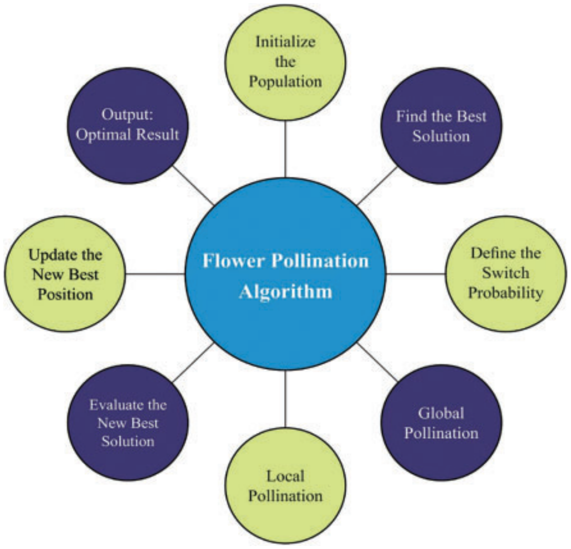

Initialize the Population
Find the Best Solution
Define the Switch Probability
Global Pollination
Local Pollination
Evaluate the New Best Solution
Update the New Best Position
Output: Optimal Result
Flower Pollination Algorithm

Figure 2: Flowchart of FPA

The FPA algorithm resolves a FF to achieve enhanced classifier efficiency. During the case, the minimized classifier error rate was regarded as FF offered in Eq. (24).

$$
\begin{aligned}
&fitness\left(x_i\right) = ClassifierErrorRate\left(x_i\right)\\
&= \frac{numberof\,misclassifiedsamples}{totalnumberof\,samples} * 100
\end{aligned}
\tag{24}
$$

4 Performance Validation

The proposed LBAAA-OMLMD model is tested using an open-access dataset (available at https://github.com/PSJoshi/Notes/wiki/). The dataset contains 1524 records and 30970 features, of which 582 are ransomware, and 942 are goodware applications. The proposed model is simulated using the Python tool.

Table 1 and Fig. 3 highlight the FS outcomes of the LBAAA-FS with other optimization algorithms. The results indicated that the binary cuckoo search (BCS) model had shown poor results with the maximum selection of 218 features. Followed by the multi-objective grey wolf optimization (MOGWO) algorithm has chosen a set of 55 features, whereas the DNAact-Ran model has elected reasonable 26 features. But the LBAAA-FS model has chosen a minimum of 23 features.

Table 1: FS analysis of LBAAA-FS technique with existing methods

Methods	No. of features selected
MOGWO	55
BCS	218
DNAact-Ran	26
LBAAA-FS	23

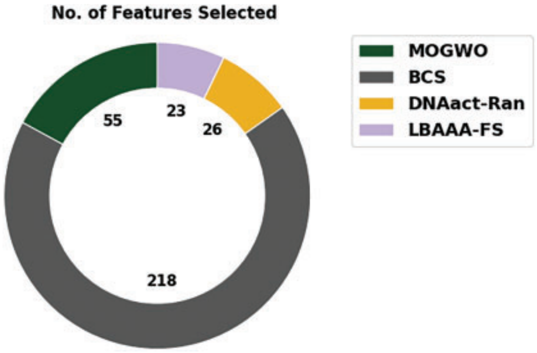


Figure 3: FS analysis of LBAAA-FS technique with existing methods

Fig. 4 illustrates the confusion matrices created by the LBAAA-OMLMD model under distinct folds. On fold-1, the LBAAA-OMLMD model has categorized a total of 863 samples into goodware and 530 samples into malware. In addition, on fold-4, the LBAAA-OMLMD technique has categorized a total of 893 samples into goodware and 533 samples into malware. Meanwhile, on fold-6, the LBAAA-OMLMD system has categorized a total of 888 samples into goodware and 551 samples into malware. Eventually, on fold-8, the LBAAA-OMLMD algorithm has categorized a total of 890 samples into goodware and 526 samples into malware. Next, on fold-10, the LBAAA-OMLMD method has categorized a total of 872 samples into goodware and 561 samples into malware.

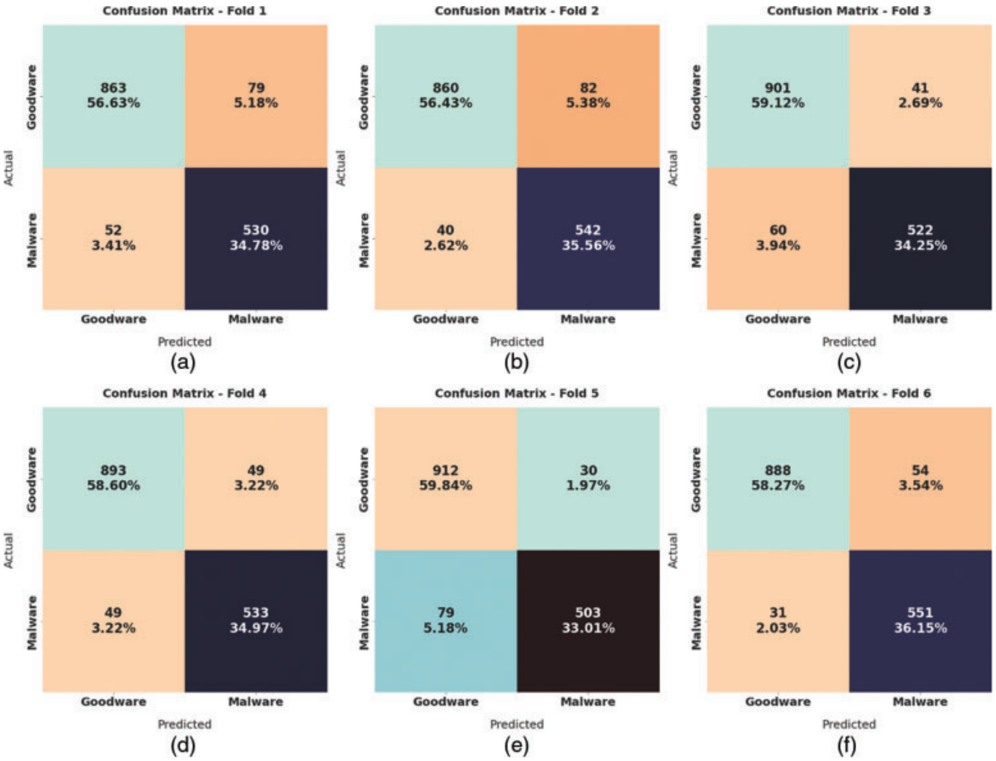
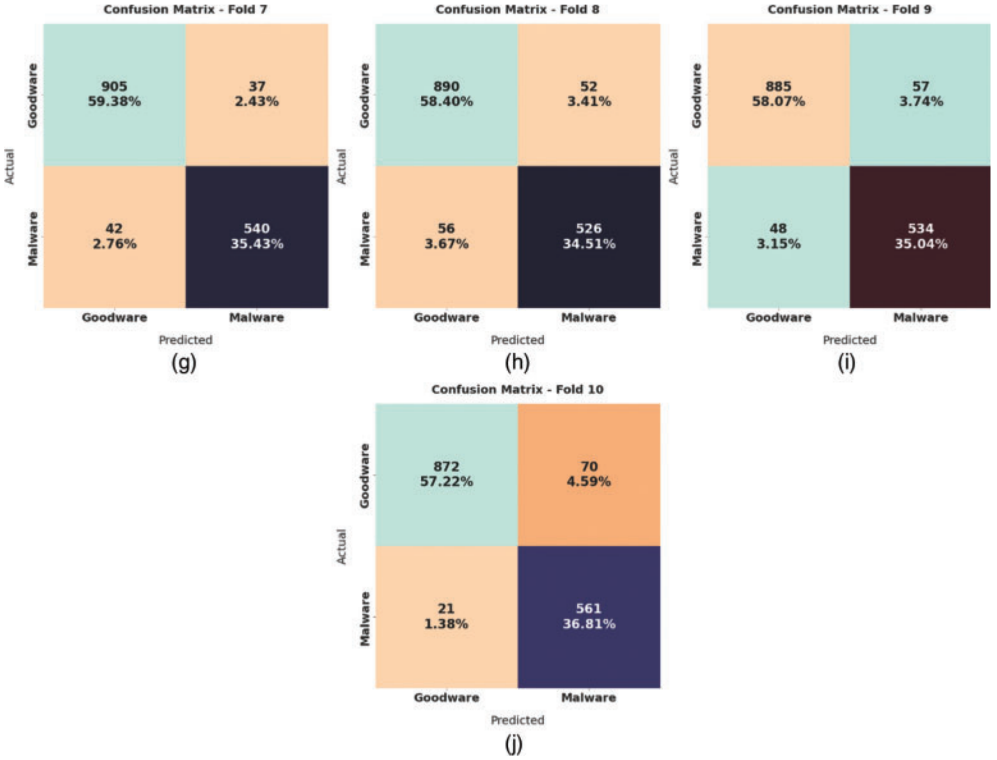


Figure 4: (Continued)



(g) (h) (i)

(j)

Figure 4: Confusion matrices of LBAAA-OMLMD technique (a) Fold 1, (b) Fold 2, (c) Fold 3, (d) Fold 4, (e) Fold 5, (f) Fold 6, (g) Fold 7, (h) Fold 8, (i) Fold 9, and (j) Fold 10

Fig. 5 reports the overall ransomware classification outcomes of the LBAAA-OMLMD model. The results exposed that the LBAAA-OMLMD model has gained effectual outcomes under all classes. For instance, on fold-1, the LBAAA-OMLMD model has identified samples with average $accu_y$, $prec_n$, $reca_l$, F_{score} Mathew Correlation Coefficient (MCC), and $G_{measure}$ of 91.40%, 90.67%, 91.34%, 90.97%, 82.01%, and 90.99% respectively. Followed by, on fold-4, the LBAAA-OMLMD technique has identified samples with average $accu_y$, $prec_n$, $reca_l$, F_{score}, MCC, and $G_{measure}$ of 93.37%, 93.24%, 92.67%, 92.94%, 85.90%, and 92.94% correspondingly. Then, on fold-6, the LBAAA-OMLMD method has identified samples with average $accu_y$, $prec_n$, $reca_l$, F_{score}, MCC, and $G_{measure}$ of 92.85%, 93.20%, 91.62%, 92.29%, 84.81%, and 92.35% correspondingly. Moreover, on fold-8, the LBAAA-OMLMD algorithm has identified samples with average $accu_y$, $prec_n$, $reca_l$, F_{score}, MCC, and $G_{measure}$ of 94.82%, 94.58%, 94.43%, 94.50%, 89%, and 94.50% correspondingly. At last, on fold-10, the LBAAA-OMLMD approach has identified samples with average $accu_y$, $prec_n$, $reca_l$, F_{score}, MCC, and $G_{measure}$ of 93.11%, 92.61%, 92.85%, 92.72%, 85.46%, and 92.73% correspondingly.

The training accuracy (TA) and validation accuracy (VA) attained by the LBAAA-OMLMD system on the test dataset is demonstrated in Fig. 6. The experimental outcome implied that the LBAAA-OMLMD algorithm had gained maximum values of TA and VA. In specific, the VA seemed to be higher than TA.

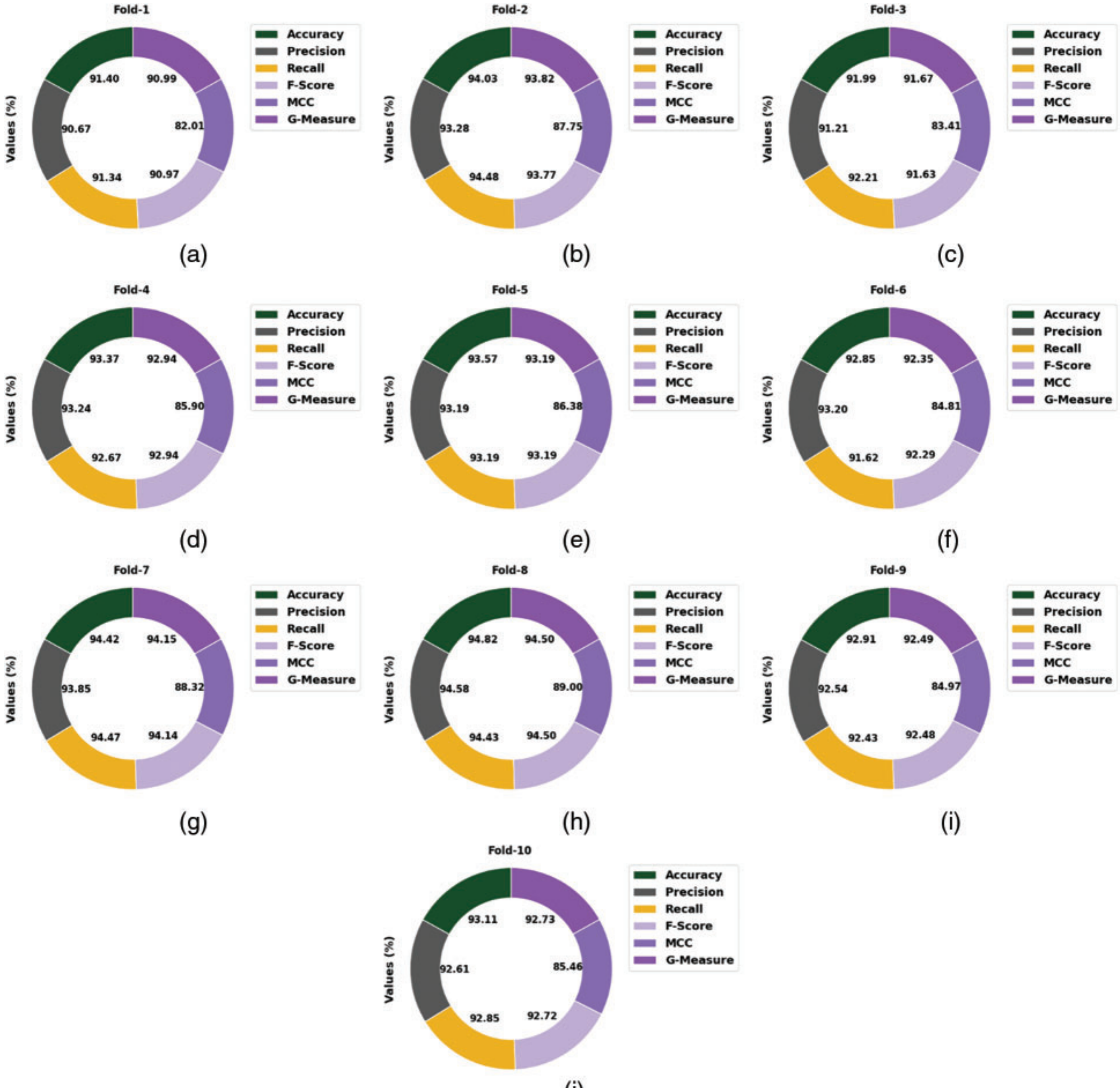


Figure 5: Average analysis of LBAAA-OMLMD technique (a) Fold 1, (b) Fold 2, (c) Fold 3, (d) Fold 4, (e) Fold 5, (f) Fold 6, (g) Fold 7, (h) Fold 8, (i) Fold 9, and (j) Fold 10

The training loss (TL) and validation loss (VL) achieved by the LBAAA-OMLMD approach on the test dataset are established in Fig. 7. The experimental outcome inferred that the LBAAA-OMLMD method had accomplished the least values of TL and VL. In specific, the VL seemed to be lower than TL.

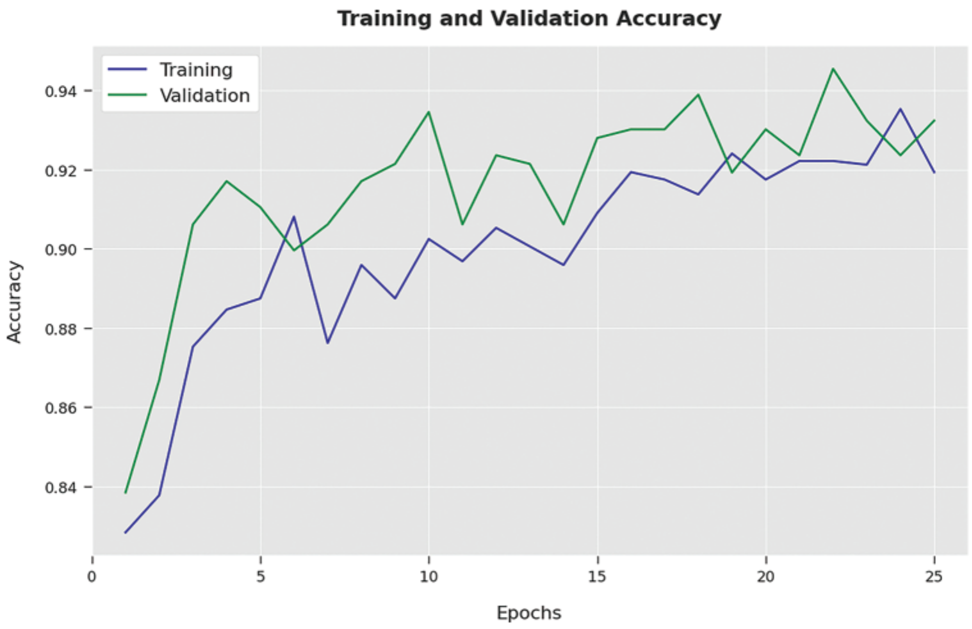


Figure 6: TA and VA analysis of LBAAA-OMLMD technique

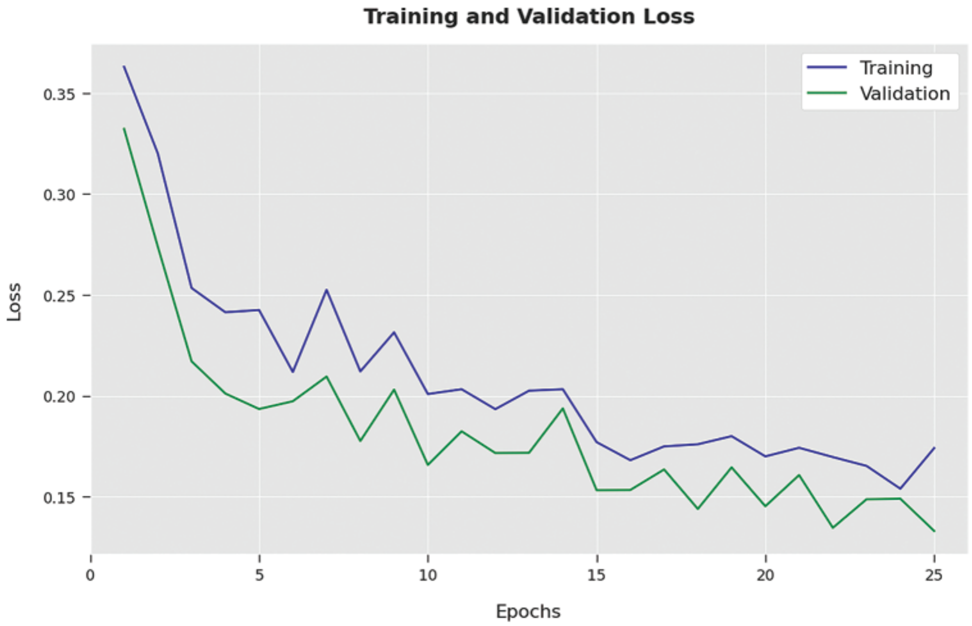


Figure 7: TL and VL analysis of LBAAA-OMLMD technique

A brief precision-recall examination of the LBAAA-OMLMD system on the test dataset is portrayed in Fig. 8. By observing the figure, it is noticed that the LBAAA-OMLMD model has accomplished maximum precision-recall performance under all classes. A detailed ROC investigation of the LBAAA-OMLMD methodology on the test dataset is portrayed in Fig. 9. The results indicated that the LBAAA-OMLMD algorithm exhibited its ability to categorize two different classes on the test dataset.

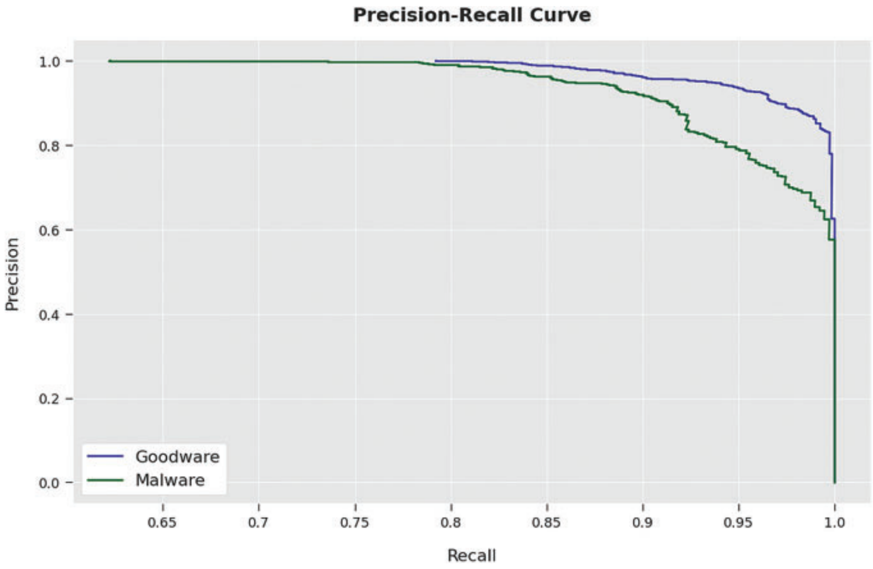


Figure 8: Precision-recall curve analysis of LBAAA-OMLMD technique

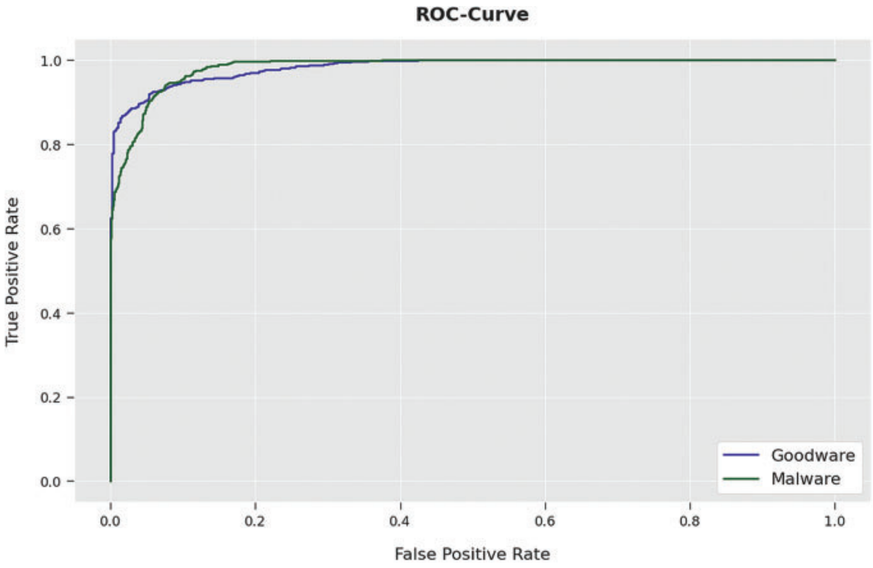


Figure 9: ROC curve analysis of LBAAA-OMLMD technique

In the final stage, the comparative study of the LBAAA-OMLMD model with existing models has carried out in Table 2 and Fig. 10. The experimental results indicated that the LBAAA-OMLMD model had gained effectual classification results. Concerning $\boldsymbol{accu_y}$, the LBAAA-OMLMD model has offered an increased $\boldsymbol{accu_y}$ of 94.82%, whereas the Naïve Bayes (NB), random forest (RF), spider monkey optimization (SMO), decision stump, AdaBoost, and DNAact-Ran models have obtained reduced $\boldsymbol{accu_y}$ of 78.52%, 84.43%, 85.68%, 75.83%, 83.22%, and 87.91%.

Table 2: Comparative analysis of LBAAA-OMLMD technique with recent algorithms

Methods	Accuracy	Precision	Recall	F-Score
NB algorithm	78.52	79.75	79.99	78.20
RF algorithm	84.43	83.34	81.09	78.05
SMO algorithm	85.68	82.36	80.73	83.55
Decision stump	75.83	81.09	81.73	82.63
AdaBoost	83.22	82.49	80.44	81.98
DNAact-Ran	87.91	83.01	78.53	78.74
LBAAA-OMLMD	94.82	94.58	94.43	94.50

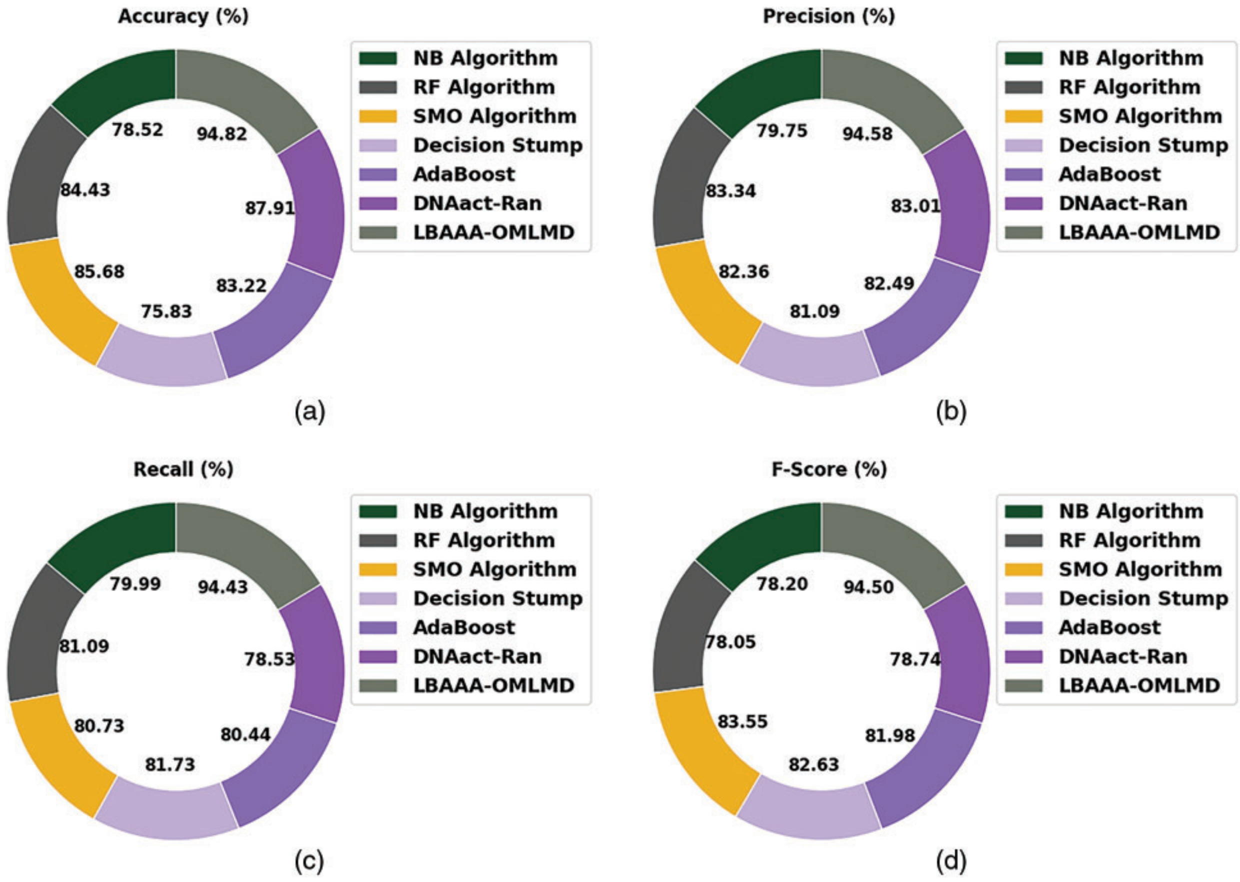


Figure 10: Comparative analysis of LBAAA-OMLMD algorithms (a) $Accu_y$, (b) $Prec_n$, (c) $Reca_l$, and (d) F_{score}

Also, concerning $reca_l$, the LBAAA-OMLMD method has rendered an increased $reca_l$ of 94.43%, whereas the NB, RF, SMO, decision stump, AdaBoost, and DNAact-Ran algorithms have gained reduced $reca_l$ of 79.99%, 81.09%, 80.73%, 81.73%, 80.44%, and 78.53%. Besides, for F_{score}, the LBAAA-OMLMD system has provided an increased F_{score} of 94.50%, whereas the NB, RF, SMO, decision stump, AdaBoost, and DNAact-Ran techniques have acquired a reduced F_{score} of 78.20%, 78.05%, 83.55%, 82.63%, 81.98%, and 78.74%. The above-mentioned result analysis confirmed the better performance of the LBAAA-OMLMD model over other models.

5 Conclusion

In this study, an effective LBAAA-OMLMD model was developed for the identification and classification of ransomware in Computer Networks. The presented LBAAA-OMLMD model follows a three-stage process, namely feature selection, classification, and parameter tuning. Initially, the LBAAA-OMLMD model employed the LBAAA-FS model to reduce the curse of the dimensionality problem. Followed the ESN-based classification with an FPA-based parameter tuning process performed. The FPA model helps to appropriately adjust the parameters related to the ESN model to accomplish enhanced classifier results. The experimental validation of the LBAAA-OMLMD model is tested using a benchmark dataset, and the outcomes are inspected under distinct measures. The comprehensive comparative examination demonstrated the betterment of the LBAAA-OMLMD model over recent algorithms. In future, advanced deep learning models can be integrated into the LBAAA-OMLMD model to improve classification efficiency.

Funding Statement: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R319), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: 22UQU4310373DSR34. The authors are thankful to the Deanship of Scientific Research at Najran University for funding this work under the Research Groups Funding program Grant Code (NU/RG/SERC/11/4).

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

References

[1] S. Saad, W. Briguglio and H. Elmiligi, “The curious case of machine learning in malware detection,” arXiv preprint arXiv:1905.07573, 2019.

[2] S. I. Bae, G. B. Lee and E. G. Im, “Ransomware detection using machine learning algorithms,” *Concurrency and Computation: Practice and Experience*, vol. 32, no. 18, pp. 1–11, 2020.

[3] D. W. Fernando, N. Komninos and T. Chen, “A study on the evolution of ransomware detection using machine learning and deep learning techniques,” *Internet of Things*, vol. 1, no. 2, pp. 551–604, 2020.

[4] M. A. Hamza, S. B. Haj Hassine, I. Abunadi, F. N. Al-Wesabi, H. Alsolai *et al.,* “Feature selection with optimal stacked sparse autoencoder for data mining,” *Computers Materials & Continua*, vol. 72, no. 2, pp. 2581–2596, 2022.

[5] S. Poudyal, D. Dasgupta, Z. Akhtar and K. Gupta, “A Multi-level ransomware detection framework using natural language processing and machine learning,” in *14th Int. Conf. on Malicious and Unwanted Software MALCON*, Nantucket, MA, pp. 1–8, 2019.

[6] M. A. Alohali, F. N. Al-Wesabi, A. M. Hilal, S. Goel, D. Gupta *et al.,* “Artificial intelligence enabled intrusion detection systems for cognitive cyber-physical systems in industry 4.0 environment,” *Cognitive Neurodynamics*, 2022. https://doi.org/10.1007/s11571-022-09780-8

[7] F. Alrowais, A. S. Almasoud, R. Marzouk, F. N. Al-Wesabi, A. M. Hilal *et al.,* "Artificial intelligence based data offloading technique for secure mec systems," *Computers, Materials & Continua*, vol. 72, no. 2, pp. 2783–2795, 2022.

[8] U. Urooj, B. A. S. Al-rimy, A. Zainal, F. A. Ghaleb and M. A. Rassam, "Ransomware detection using the dynamic analysis and machine learning: A survey and research directions," *Applied Sciences*, vol. 12, no. 1, pp. 172, 2021.

[9] A. M. Hilal, J. S. Alzahrani, I. Abunadi, N. Nemri, F. N. Al-Wesabi *et al.,* "Intelligent deep learning model for privacy preserving iiot on 6 g environment," *Computers, Materials & Continua*, vol. 72, no. 1, pp. 333–348, 2022.

[10] H. Zhang, X. Xiao, F. Mercaldo, S. Ni, F. Martinelli *et al.,* "Classification of ransomware families with machine learning based on N-gram of opcodes," *Future Generation Computer Systems*, vol. 90, pp. 211–221, 2019.

[11] A. M. Hilal, M. A. Alohali, F. N. Al-Wesabi, N. Nemri, H. J. Alyamani *et al.,* "Enhancing quality of experience in mobile edge computing using deep learning based data offloading and cyberattack detection technique," *Cluster Computing*, 2021. https://doi.org/10.1007/s10586-021-03401-5

[12] A. Cohen and N. Nissim, "Trusted detection of ransomware in a private cloud using machine learning methods leveraging meta-features from volatile memory," *Expert Systems with Applications*, vol. 102, pp. 158–178, 2018.

[13] I. Bibi, A. Akhunzada, J. Malik, G. Ahmed and M. Raza, "An effective android ransomware detection through multi-factor feature filtration and recurrent neural network," in *2019 UK/China Emerging Technologies (UCET)*, Glasgow, United Kingdom, pp. 1–4, 2019.

[14] J. Zhu, J. J. Jaccard, A. Singh, I. Welch, H. AI-Sahaf *et al.,* "A Few-shot meta-learning based siamese neural network using entropy features for ransomware classification," *Computers & Security*, vol. 117, pp. 102691, 2022.

[15] I. Bello, H. Chiroma, U. A. Abdullahi, A. Y. U. Gital, F. Jauro *et al.,* "Detecting ransomware attacks using intelligent algorithms: Recent development and next direction from deep learning and big data perspectives," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 9, pp. 8699–8717, 2021.

[16] S. Aurangzeb, H. Anwar, M. A. Naeem and M. Aleem, "BigRC-EML: Big-data based ransomware classification using ensemble machine learning," *Cluster Computing*, pp. 1–18, 2022. https://doi.org/10.1007/s10586-022-03569-4

[17] S. Egunjobi, S. Parkinson and A. Crampton, "Classifying ransomware using machine learning algorithms," in *Int. Conf. on Intelligent Data Engineering and Automated Learning*, Lecture Notes in Computer Science book series, Springer, Cham, vol. 11872, pp. 45–52, 2019.

[18] F. Khan, C. Ncube, L. K. Ramasamy, S. Kadry and Y. Nam, "A digital dna sequencing engine for ransomware detection using machine learning," *IEEE Access*, vol. 8, pp. 119710–119719, 2020.

[19] H. Daku, P. Zavarsky and Y. Malik, "Behavioral-based classification and identification of ransomware variants using machine learning," in *2018 17th IEEE Int. Conf. on Trust, Security and Privacy in Computing and Communications/12th IEEE Int. Conf. on Big Data Science and Engineering (TrustCom/BigDataSE)*, New York, NY, USA, pp. 1560–1564, 2018.

[20] K. Lee, S. Y. Lee and K. Yim, "Machine learning based file entropy analysis for ransomware detection in backup systems," *IEEE Access*, vol. 7, pp. 110205–110215, 2019.

[21] S. H. Kok, A. Azween and N. Jhanjhi, "Evaluation metric for crypto-ransomware detection using machine learning," *Journal of Information Security and Applications*, vol. 55, pp. 102646, 2020.

[22] S. Sharma, C. R. Krishna and R. Kumar, "Android ransomware detection using machine learning techniques: A comparative analysis on GPU and CPU," in *2020 21st Int. Arab Conf. on Information Technology (ACIT)*, Giza, Egypt, pp. 1–6, 2020.

[23] M. Kumar and J. S. Dhillon, “Hybrid artificial algae algorithm for economic load dispatch,” *Applied Soft Computing*, vol. 71, pp. 89–109, 2018.

[24] M. Han and M. Xu, “Laplacian echo state network for multivariate time series prediction,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 1, pp. 238–244, 2018.

[25] M. Zhang, W. Sun, J. Tian, X. Zheng and S. Guan, “An internet traffic classification method based on echo state network and improved salp swarm algorithm,” *PeerJ Computer Science*, vol. 8, pp. e860, 2022.

[26] D. Chakraborty, S. Saha and O. Dutta, “DE-FPA: A hybrid differential evolution-flower pollination algorithm for function minimization,” in *2014 Int. Conf. on High Performance Computing and Applications (ICHPCA)*, Bhubaneswar, India, pp. 1–6, 2014.