AIMS Mathematics

https://www.aimspress.com/journal/Math

AIMS Mathematics, 9(3): 6784–6802.
DOI: 10.3934/math.2024331
Received: 05 December 2023
Revised: 19 January 2024
Accepted: 23 January 2024
Published: 19 February 2024

Research article

Integrating Ebola optimization search algorithm for enhanced deep learning-based ransomware detection in Internet of Things security

Ibrahim R. Alzahrani[1] **and Randa Allafi**[2,*]

[1] Department of Computer Science and Engineering, College of Computer Science and Engineering, University of Hafr Al Batin, Al Jamiah, Hafar Al Batin 39524, Saudi Arabia

[2] Department of Computers and Information Technology, College of Sciences and Arts, Northern Border University, Arar, Saudi Arabia

* **Correspondence:** Email: randa.allafi@nbu.edu.sa.

Abstract: With the widespread use of Internet, Internet of Things (IoT) devices have exponentially increased. These devices become vulnerable to malware attacks with the enormous amount of data on IoT devices; as a result, malware detection becomes a major problem in IoT devices. A reliable and effective mechanism is essential for malware detection. In recent years, research workers have developed various techniques for the complex detection of malware, but accurate detection continues to be a problem. Ransomware attacks pose major security risks to corporate and personal information and data. The owners of computer-based resources can be influenced by monetary losses, reputational damage, and privacy and verification violations due to successful assaults of ransomware. Therefore, there is a need to swiftly and accurately detect the ransomware. With this motivation, the study designs an Ebola optimization search algorithm for enhanced deep learning-based ransomware detection (EBSAEDL-RD) technique in IoT security. The purpose of the EBSAEDL-RD method is to recognize and classify the ransomware to achieve security in the IoT platform. To accomplish this, the EBSAEDL-RD technique employs min-max normalization to scale the input data into a useful format. Also, the EBSAEDL-RD technique makes use of the EBSA technique to select an optimum set of features. Meanwhile, the classification of ransomware takes place using the bidirectional gated recurrent unit (BiGRU) model. Then, the sparrow search algorithm (SSA) can be applied for optimum hyperparameter selection of the BiGRU model. The wide-ranging experiments of the EBSAEDL-RD approach are performed on benchmark data. The obtained results highlighted that the EBSAEDL-RD algorithm reaches better performance over other models on IoT security.

Keywords: Ransomware detection; Internet of Things; Sparrow Search Algorithm; deep learning; feature selection
Mathematics Subject Classification: 11Y40

1. Introduction

In recent times, the Internet of Things (IoTs) has exponentially increased with the usage of smart devices. IoT devices allow us to access from anywhere such as homes, vehicles, and offices to make day-to-day tasks simple [1] and are utilized in smart cities, care services, health, smart homes, smart grids, vehicular networks, and other industries. Also, they have special features, namely lower energy consumption, lighter protocols, and compact size which adapt them better [2]. Extended transportation of smart devices in advertising along with declined trust in identifying devices has made the web of things more and more versatile [3]. Malicious attacks or applications, like ransomware and malware families, constantly pose crucial security problems to cybersecurity and can result in catastrophic losses to the web, data centers, mobile applications, and computer systems across several businesses and industries [4]. Ransomware is mainly developed to prevent and block victims from accessing system databases by using a robust encrypting method that can be decrypted by attackers [5].

Removing the ransomware will lead the targeted victim to permanently lose data, therefore, targeted victims are compelled to comply with the attacker's demand [6]. Attackers transform traditional ransomware into new ransomware families through modern technology, which makes it more challenging to reverse the ransomware infection [7]. Ransomware is a variant and sophisticated threat affecting users around the world that limits users from accessing the data or system by encrypting or locking the system screen and the user files unless a ransom is paid [8]. Locker ransomware and crypto-ransomware are the two different types of ransomware based on attack strategies. Crypto ransomware prevents access to data or files and the access is denied to the device or computer in locker ransomware [9].

Conventional ransomware detection methods, like data-centric-based, event-based, and statistical-based approaches, are not suitable to combat the attacks. Thus, the high level of security and protection implemented by adopting innovative technology against these malware attacks has gained immense attention from researchers [10]. Due to their fixed architecture, classical machine learning (ML) techniques are unable to distinguish complicated cyberattacks from ever-growing cyber threats and adversaries' or attacker's resources and capabilities. The objective is to provide security on the device from different attacks by using the latest and advanced technologies that are capable of detecting the attacks with recognition accuracy in less time [11]. In this context, deep learning (DL) shows the real face of cyber data, either attack or legitimate, by identifying the slight changes or differences. Therefore, DL may quickly identify the anomalies and facilitate an in-depth analysis of network data [12]. Therefore, a DL-driven detection technique becomes cost-effective, adaptive, and highly scalable without exhausting the primitive devices, which is a breakthrough invention in cyber-security [13].

Alohali et al. [14] developed a sine cosine algorithm with a DL-based ransomware detection and classification (SCADL-RWDC) algorithm in the IoT platform. This algorithm employs the SCA-feature selection (SCA-FS) system to increase the recognition accuracy. Also, the proposed method implements the Hybrid Grey Wolf Optimizer GWO (HGWO) with the GRU technique to classify ransomware. The author in [15], introduced a new method to avoid crypto-ransomware by identifying block cipher techniques for IoT environment. This method has extracted the features in the opcode of binary documents for the microcontroller named 8-bit Alf and Vegard's RISC (AVR) processor.

In [16], developed a static analysis model based on N-gram opcodes with DL algorithm. At first, the proposed method splits the N-gram sequence into numerous patches as well as provides every patch to self-attention-based CNN (Conventual Neural Network) (SA-CNN). Next, the efficiency of SA-CNNs must be combined and implemented in a bi-directional SA network to achieve the outcome of ransomware classification. In [17], an IoT-based IDS and classification system based-CNN (IoT-IDCS-CNN) method was presented. The performance assessment utilizes parallel processing to use strong compute unified device architectures (CUDA) based Nvidia graphical processing unit (GPU) and high speed I9-core-based Intel CPU.

In [18], an optimum graph-CNN-enabled ransomware detection (OGCNN-RWD) method was developed for cyber-security in the IoT infrastructure. This study presents learnable enthusiasm to teach learning-based optimizer (LETLBO) techniques for the subcategory of the FS method. The GCNN architecture has been employed to classify ransomware, and hyperparameters should be effectively preferred by the harmony search algorithm (HSA). In [19], the main objective is to examine a lightweight DL method that increases the detection rate with a decreased computation rate for confirming the real-time application of malware monitoring in limited IoT devices. The architecture has been employed for RNN, LSTM, and the bi-directional-LSTM-DL method under a vanilla configuration trained with conventional malware databases.

Basnet et al. [20] projected the DL-based ransomware identification technique in SCADA-controlled electric vehicle charging stations (EVCS) with evaluation studies of 3 DL techniques such as LSTM-RNN, 1D-CNN, and DNN. Ransomware was determined the Distributed Denial-of-Service DDoS (distributed denial-of-service) attack prefers to change the state of charge (SOC) configuration by surpassing the control threshold of SOC. In [21], various assessment of malware evaluation of sample was determined. The 3 malware identification algorithms based on visualization methods (i.e., clustering technique, probabilistic method, and DL algorithm) were developed. Afterwards, a developed measure depends on the risk of instances that could be utilized for evaluation.

In the domain of IoT cybersecurity, researchers like Alohali et al. ([14]) have proposed innovative approaches, such as the SCADL-RWDC algorithm integrating sine cosine slgorithm and deep learning, while Basnet et al. ([20]) focused on DL-based ransomware identification in SCADA-controlled electric vehicle charging stations. These studies collectively offer a variety of methodologies, from SCA-FS to OGCNN-RWD, contributing to the advancement of ransomware detection and overall cybersecurity in IoT environments.

The presented article develops an EBSAEDL-RD approach in IoT security. In order to achieve this, the EBSAEDL-RD approach utilizes min-max normalization to scale input data effectively and incorporates the EBSA method for optimal feature selection. Ransomware classification is performed using the bidirectional gated recurrent unit (BiGRU) method, with the sparrow search algorithm (SSA) employed for fine-tuning hyperparameters. Extensive experiments employing the EBSAEDL-RD approach are conducted on a benchmark dataset.

2. The proposed model

In this study, we design a new EBSAEDL-RD algorithm in IoT security. The purpose of the

EBSAEDL-RD technique is to recognize and classify the ransomware to achieve security in the IoT platform. To achieve this, the EBSAEDL-RD technique contains different types of processes, namely min-max normalization, EBSA-based feature selection, BiGRU classification, and SSA-based hyperparameter tuning. Figure 1 illustrates the working flow of the EBSAEDL-RD technique.

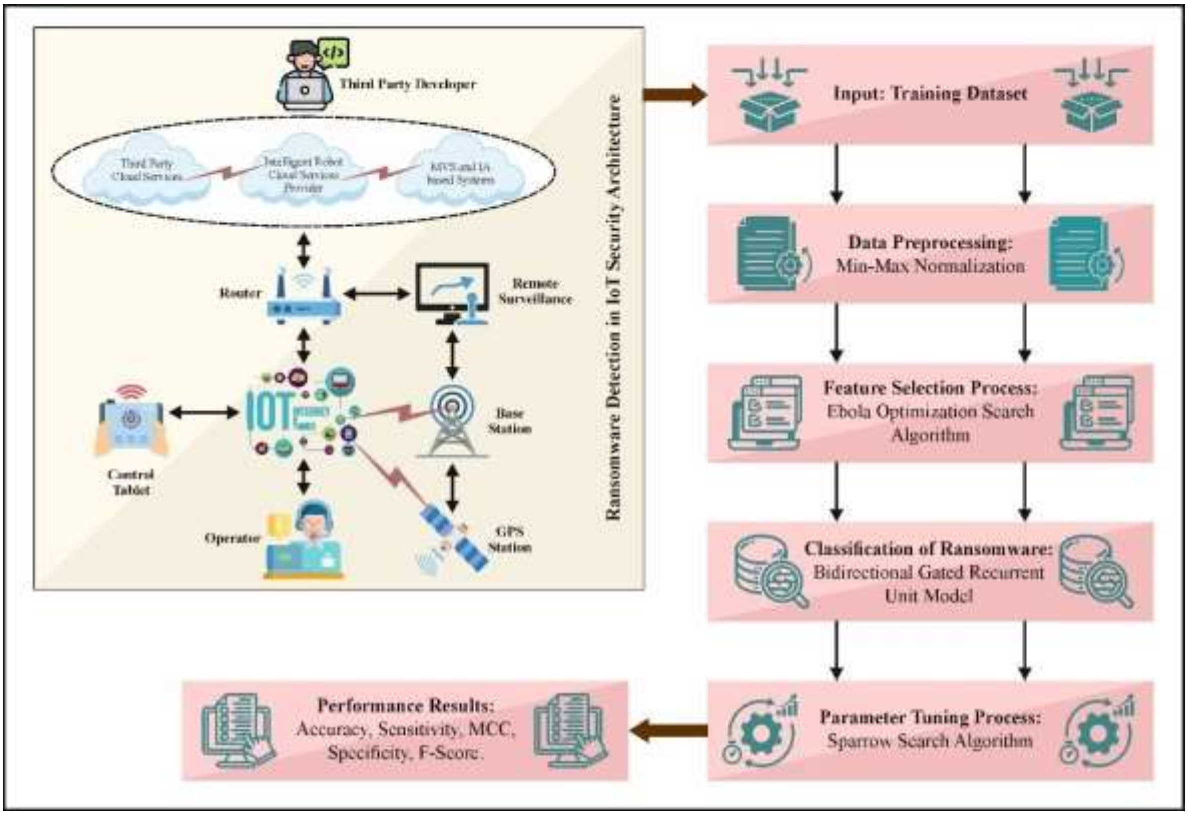

Third Party Developer
Third Party Cloud Services
Intelligent Robot Cloud Services Provider
MVS and IA based Systems
Router
Remote Surveillance
IOT
Base Station
Control Tablet
Operator
GPS Station
Ransomware Detection in IoT Security Architecture
Input: Training Dataset
Data Preprocessing: Min-Max Normalization
Feature Selection Process: Ebola Optimization Search Algorithm
Classification of Ransomware: Bidirectional Gated Recurrent Unit Model
Parameter Tuning Process: Sparrow Search Algorithm
Performance Results: Accuracy, Sensitivity, MCC, Specificity, F-Score

Figure 1. Workflow of EBSAEDL-RD technique.

2.1. Min-Max normalization

Initially, the EBSAEDL-RD method exploits min-max normalization. In the context of ransomware detection, min-max normalization is a preprocessing stage for IoT security [22]. This method is used to standardize and scale mathematical features within a certain range, between 0 and 1. In the field of IoT security, where the recognition of ransomware threats is of great significance, normalizing input data ensures that dissimilar feature sizes do not excessively impact the performance of ML algorithms. The min-max normalization facilitates the effective utilization of diverse features in detecting patterns indicative of ransomware attacks by transforming the data into a consistent scale. This normalization method improves the accuracy and robustness of prediction techniques, contributing to the general efficiency of ransomware detection systems in protecting the IoT environment from possible security attacks.

2.2. Feature selection

The EBSAEDL-RD technique makes use of the EBSA technique to select an optimum set of features. Ebola optimization search algorithm (EOSA), a recent meta-heuristic technique, draws inspiration from the propagation model of Ebola virus disease introduced by Oyelade and Ezugwu [23]. The explanation of the EOSA technique is discussed below:

1) Set each scalar and vector quantity which are parameters and individuals. Individuals in the set: Infected (I), Susceptible (S), Vaccinated (V), Dead (D), Recovered (R), Hospitalized (H), and Quarantine (Q) with the initial value.
2) The index case (I) is randomly generated from inclined individuals.
3) The index case is set as global and local optimum and the fitness values.
4) When the iteration count is not exhausted infected individuals exist,
    a. Generate and update their location depending on their movement for every susceptible individual. Note that the infected state is further moved, then there exists more infection, hence short displacement defines exploitation or else exploration.
        i. Generate diseased individual (nI) depend on (a).
        ii. Add that case to the I
    b. Calculate the individual number and add it to H, D, R, B, V, and Q through the corresponding rate based on the dimension of I
    c. Update S and I based on I.
    d. Pick the present finest from I and compute it with global finest.
    e. If the terminating criteria are not met, then return to step 6.
5) Return global best and each solution.

The mathematical modelling is given as follows: update of Funeral (F), Exposed (E), S, I, H, V, R, Q, and D are directed by a method of difference equation derived. The differential calculus aims to get the rates of change of quantities in terms of time t:

$$\frac{\partial S(t)}{\partial t} = \pi - (\beta_1 I + \beta_3 D + \beta_4 R + \beta_2 (\mathrm{PE})\eta) S - (\tau S + \mathrm{r} I) \tag{1}$$

$$\frac{\partial I(t)}{\partial t} = (\beta_1 I + \beta_3 D + \beta_4 R + \beta_2 (PE)\lambda) S — (\mathrm{r} + \gamma) I — (\tau\ ) S \tag{2}$$

$$\frac{\partial H(t)}{\partial t} = \alpha I - (\gamma + \varpi) H \tag{3}$$

$$\frac{\partial R(t)}{\partial t} = \gamma I - \mathrm{r} R \tag{4}$$

$$\frac{\partial V(t)}{\partial t} = \gamma I - (\mu + \vartheta) V \tag{5}$$

$$\frac{\partial D(t)}{\partial t} = (\tau S + \mathrm{r} I) - \delta D \tag{6}$$

$$\frac{\partial Q(t)}{\partial t} = \big(\pi I - (\gamma R + \mathrm{r} D)\big) - \xi Q \tag{7}$$

In the EBSA approach, the fitness function (FF) is intended to have a balance between the number of features chosen in every solution (minimum) and the classifier outcome (maximum) attained, Eq (8) shows the FF to calculate the solution.

$$Fitness = \alpha\gamma_R(D) + \beta\frac{|R|}{|C|} \tag{8}$$

In Eq (8), α and β are the significance of classifier quality and subset length, $\in [1,0]$ and $\beta = 1 - \alpha.\gamma_R(D)$ indicates the classifier error rate, $|R|$ stands for the cardinality of the selected subset, and $|C|$ refers to the overall amount of features in the dataset (parameters).

2.3. BiGRU based classification

In this phase, the classification of ransomware takes place using the BiGRU model. BiGRU is an RNN that has been effectively utilized for solving time-series sequence data challenges due to its bi-directional learning system that improves the learning of temporal designs from the time-sequence data [24]. All the BiGRU blocks comprise a cell that stores data. All the blocks are composed of update and reset gates and the cells assist in addressing the disappearing gradient problems. BiGRU contains 2 GRU units: reset and update gates. The reset gate integrates novel input with preceding memory and the update gate determines the preceding memory to recollect. The input dataset is fed into feedback and feedforward networks in terms of time, and these two are linked to one resultant layer. The BiGRU gates are planned to store data extensively in either backward or forward ways if the optimum solution than feedforward networks. The bi-directional method offers the ability to employ either past or future contexts from the sequences. BGRU has been formulated as:

$$h_t = \left[\overrightarrow{h_t}, \overleftarrow{h_t}\right] \tag{9}$$

where $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$are the feedforward and the backward blocks, respectively.
The last resultant layer at time t is:

$$y_t = \sigma\left(W_y h_t + b_y\right) \tag{10}$$

where σ stands for the activation function, W_y denotes the weighted, and b_y represents the bias vector.

Every GRU block is composed of 4 modules: reset gate r_I with equivalent weights and biases W_r, U_r, b_r, input vector x_l with equivalent weights and biases, output vector h_t with its weights and biases W_h, U_h, b_h, and update gate z_I with equivalent weights and biases W_z, U_z, b_z. The gating units are defined as follows:
Primarily, for $t = 0$, the resultant vector is $h_0 = 0$

$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z) \tag{11}$$

$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r) \tag{12}$$

$$h_t = z_t h_{t-1} + (1 - z_t) \otimes \emptyset h(W_h x_t + U_h(r_t \otimes h_{t-1}) + b_h) \tag{13}$$

where $W, U, and\ b$ denote the parameter matrices and vectors, σ_g defines the sigmoid function, $\otimes$ indicates the Hadamard product, σ_g and $\emptyset h$ imply the activation functions, and $\emptyset h$ signifies the hyperbolic tangent. Figure 2 defines the infrastructure of the BiGRU model.

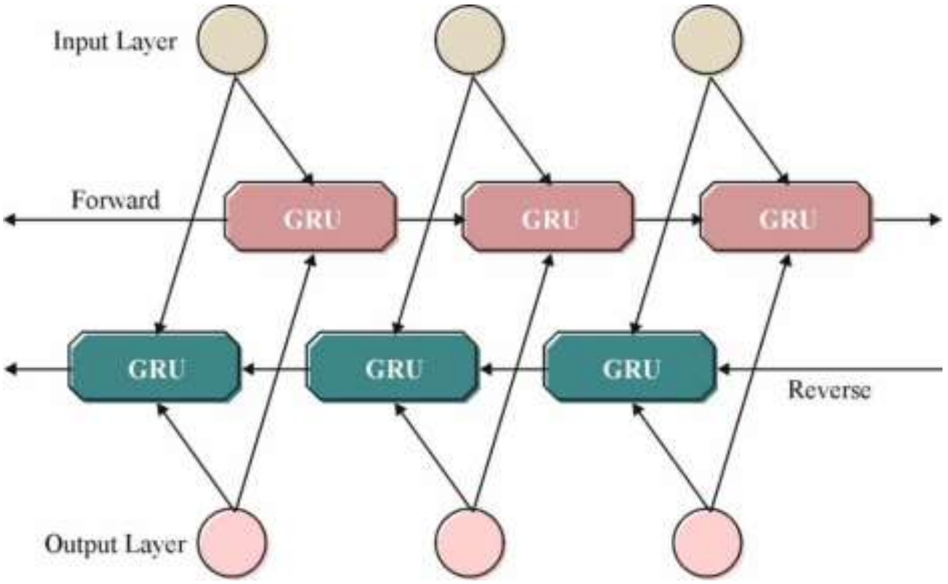


Figure 2. BiGRU architecture.

Initially, the BGRU cells have been generated for the outcome of feedforward has been calculated (F_t) and the feedback propagation (B_t) is combined. These 4 approaches combine the solution, multiplication, concatenation (default), average, and summation. In this case, it is related to the solution of the entire combining model. The combining is defined as:

$$O_t^1 = concat\left((\overrightarrow{F_t}), (\overleftarrow{B_t})\right)$$

Such that

$$(\overrightarrow{F_t}) = (\overrightarrow{h_1}, \overrightarrow{h_2}, \overrightarrow{h_3}, \dots, \overrightarrow{h_t})$$

and

$$(\overleftarrow{B_t}) = (\overleftarrow{h_t}, \overleftarrow{h_{t+1}}, \overleftarrow{h_{t+2}}, \overleftarrow{h_{t+3}}, \dots \overleftarrow{h_n}) \quad (14)$$

Then, the FC layer has been utilized to increase the BiGRU solution with its bias and weight. Afterwards, a Softmax regression layer generates a predictive utilization in the FC layer. The weighted classification layer has been utilized for computing the weighted cross-entropy loss function to predict score and training target that assists in addressing the class imbalanced problems. The next loss can be utilized as:

$$(p_{,t}) = -(1 - (p_t)^{\gamma}) \log_2(p_t) * \theta_i \quad (15)$$

where $(p_{,t})$ defines the assessed probability of all the classes, $\gamma \geq 0$ refers to the discount factor parameter that is tuned to better evaluate, and θ_i refers to the logic weight of all the classes.

2.4. Hyperparameter tuning using SSA model

Finally, the SSA can be applied for optimal hyperparameter selection of the BiGRU model. SSA developed that pretends to antipredatory and predatory performance of sparrows [25]. In the SSA model, the individuals are separated into producers by huge energy assets, joiners discover food through producers and vigilantes who are highly answerable for cautionary. The uniqueness of finders and joiners is not stable. Any individual who finds a superior food source becomes a producer while others become a joiner. Since the producer's ratio to joiners is constant in a cluster, during the foraging procedure, producers are highly responsible for searching regions for plentiful food and delivering guidelines to other joiners who constantly discover producers by optimal food. As soon as vigilantes discover a hunter, they guide an alarm sign via song and the producer takes the joiner far away to a protected region once a sign attains a definite threshold. At the edge of the cluster, other sparrows rapidly moved to the security area, but the sparrows who were in the middle had to move arbitrarily in confidence of receiving nearer to other sparrows. Let us assume that the complete number of sparrows is m, j signifies spatial distribution, the ratio of the producer to joiner is between 7:1 and 3:1, and W_s denotes protection threshold of cautionary signal, Then, $S_{i,j} = \left(S_{1,j},\ S_{2,j}, \ldots,\ S_{\mathrm{m},j}\right)$ refers to the location of i-th sparrow in flight. So, the location of producer, joiner, and vigilante upgraded affording to Eqs (16–18). $R_2 \geq W_s$ in Eq. (16), signifies vigilantes discover a hunter, all sparrows must rapidly fly to harmless places, and $R_2 < W_s$ then the producer continues its search in a wider region. If $\mathrm{I} > \frac{m}{2}$ in Eq (17), then $the\ i^{th}$ joiner with inferior fitness value is most probably a hungry sparrow. If $f_i = f_b$ shows that the sparrow is in mid of the swarm, and $f_i > f_b$ then the sparrow is at the edge of the swarm in Eq (18).

$$S_{i,j}^{k+1} = \begin{cases} S_{i,j}^{\mathrm{k}} + Q \cdot L, R_2 \geq W_s \\ S_{i,j}^{k} \cdot \exp\left(\frac{-i}{\alpha \cdot iter_{\max}}\right), R_2 < W_s \end{cases} \tag{16}$$

$$S_{\mathrm{i,j}}^{k+1} = \begin{cases} Q \cdot \exp\left(\frac{S_{worst}^{k} - S_{\mathrm{i,j}}^{k}}{i^2}\right), i > \frac{\mathrm{m}}{2} \\ S_b^{\mathrm{k+1}} + \left|S_{i,\mathrm{j}}^{\mathrm{k}} - S_b^{\mathrm{k+1}}\right| \cdot A^{+} \cdot L, i \leq \frac{\mathrm{m}}{2} \end{cases} \tag{17}$$

$$S_{\mathrm{i,j}}^{k+1} = \begin{cases} S_{best}^{k} + \beta \cdot \left|S_{\mathrm{i,j}}^{k} - S_{best}^{k}\right|, f_i > f_b \\ S_{i,j}^{\mathrm{k}} + k \cdot \left(\frac{\left|S_{\mathrm{i,j}}^{k} - S_{worst}^{k}\right|}{(f_i - f_w) + \varepsilon}\right), f_i = f_b \end{cases} \tag{18}$$

where $S_{i,j}^{\mathrm{k+1}}$ signifies the location of j^{th} element of i^{th} sparrow at $(k+1)$-th iteration, $S_{i,j}^{\mathrm{k}}$ represents the position of j^{th} dimension of i^{th} sparrow at k^{th} iteration, $S_{best}^{\mathrm{k+1}}$ denotes location of best producer at $(k+1)$-th iteration, L demonstrates a medium of every element inside is 1, S_{best}^{k} is the global optimum solution at the k^{th} iteration, S_{worst}^{k} is the global worst place at $the\ k^{th}$ iteration, Q denotes an arbitrary amount that follows the standard distribution, R_2 directs the value of the alarm signal for all sparrows, W_s signifies the protection threshold of the alarm signal that is equivalent to 0.8, f_i and f_b are said to be present and global best fitness value respectively, $iter_{\max}$ refers to the maximal amount of iterations, α and κ signify random numbers in [0,1], ε defines the error constant, and β denotes the control parameter.

The SSA method derives an FF to acquire higher efficiency of classification. It describes a positive integer to characterize the enhanced accuracy of candidate solutions. Here, the decline of the classifier error rate is regarded as an FF,

$$fitness(x_i) = ClassifierErrorRate(x_i)$$

$$= \frac{No.of\ misclassified\ samples}{Total\ No.of\ samples} * 100 \qquad (19)$$

3. Results, analysis, and discussion

The ransomware detection outcomes of the EBSAEDL-RD method are tested using a dataset [26] encompassing 840 samples as defined by Table 1.

Figure 3 defines the confusion matrices achieved by the EBSAEDL-RD algorithm under epochs from 500 to 3000. The experimental values imply that the EBSAEDL-RD algorithm has efficient recognition of the goodware and ransomware samples under two classes.

Table 1 Details of dataset.

Classes	No. of Instances
Goodware	420
Ransomware	420
Total Instances	**840**

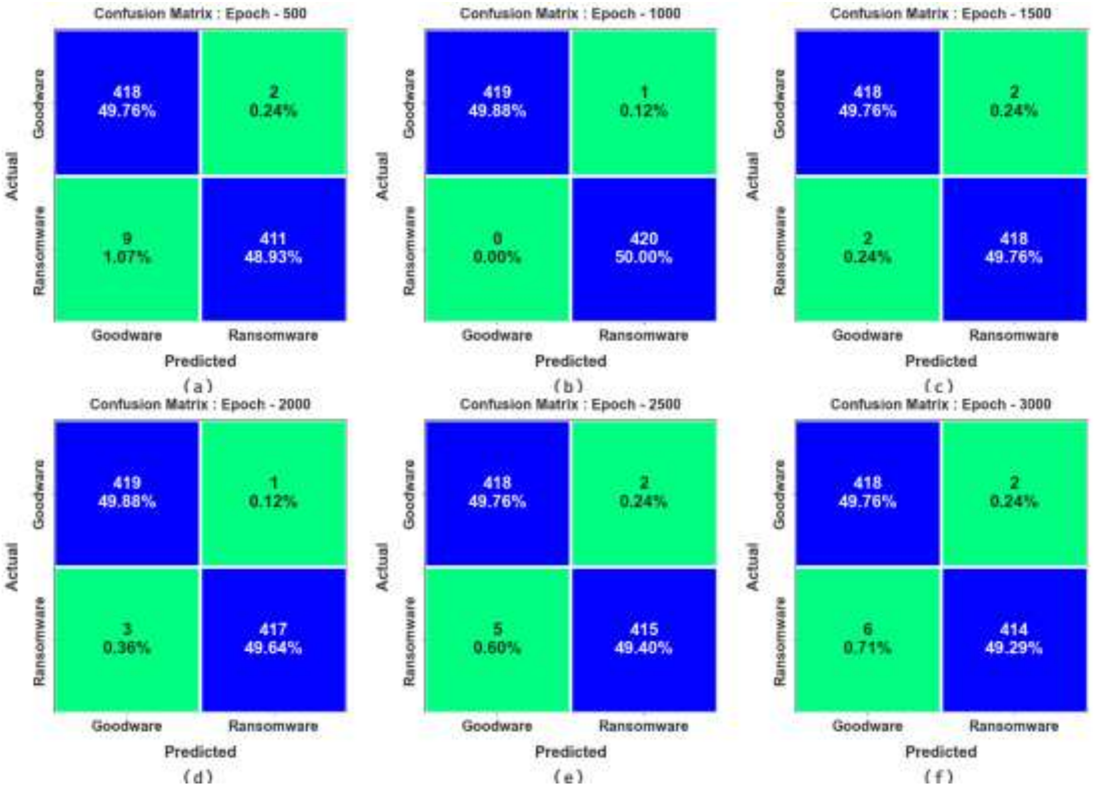


Figure 3. Confusion matrices of the EBSAEDL-RD model (a–f) epochs 500–3000.

Table 2 and Figure 4 show the ransomware detection of the EBSAEDL-RD technique is investigated under distinct epochs. The outcome inferred that the EBSAEDL-RD method reaches effectual detection of the goodware and ransomware. On 500 epochs, the EBSAEDL-RD method attains an average $accu_y$ of 98.69%, $sens_y$ of 98.69%, $spec_y$ of 98.69%, F_{score} of 98.69%, and MCC of 97.39%. On 1000 epochs, the EBSAEDL-RD system achieved an average $accu_y$ of 99.88%, $sens_y$ of 99.88%, $spec_y$ of 99.88%, F_{score} of 99.88%, and MCC of 99.76%. On 2000 epochs, the EBSAEDL-RD methodology reached an average $accu_y$ of 99.52%, $sens_y$ of 99.52%, $spec_y$ of 99.52%, F_{score} of 99.52%, and MCC of 99.05%. On 2500 epochs, the EBSAEDL-RD algorithm achieved an average $accu_y$ of 99.17%, $sens_y$ of 99.17%, $spec_y$ of 99.17%, F_{score} of 99.17%, and MCC of 98.34%. Lastly, on 3000 epochs, the EBSAEDL-RD technique obtained an average $accu_y$ of 99.05%, $sens_y$ of 99.05%, $spec_y$ of 99.05%, F_{score} of 99.05%, and MCC of 98.10%.

Table 2. Ransomware detection of the EBSAEDL-RD system under different epochs.

Classes	$Accu_y$	$Sens_y$	$Spec_y$	F_{score}	MCC
Epoch - 500					
Goodware	99.52	99.52	97.86	98.70	97.39
Ransomware	97.86	97.86	99.52	98.68	97.39
Average	**98.69**	**98.69**	**98.69**	**98.69**	**97.39**
Epoch - 1000					
Goodware	99.76	99.76	100.00	99.88	99.76
Ransomware	100.00	100.00	99.76	99.88	99.76
Average	**99.88**	**99.88**	**99.88**	**99.88**	**99.76**
Epoch - 1500					
Goodware	99.52	99.52	99.52	99.52	99.05
Ransomware	99.52	99.52	99.52	99.52	99.05
Average	**99.52**	**99.52**	**99.52**	**99.52**	**99.05**
Epoch - 2000					
Goodware	99.76	99.76	99.29	99.52	99.05
Ransomware	99.29	99.29	99.76	99.52	99.05
Average	**99.52**	**99.52**	**99.52**	**99.52**	**99.05**
Epoch - 2500					
Goodware	99.52	99.52	98.81	99.17	98.34
Ransomware	98.81	98.81	99.52	99.16	98.34
Average	**99.17**	**99.17**	**99.17**	**99.17**	**98.34**
Epoch - 3000					
Goodware	99.52	99.52	98.57	99.05	98.10
Ransomware	98.57	98.57	99.52	99.04	98.10
Average	**99.05**	**99.05**	**99.05**	**99.05**	**98.10**

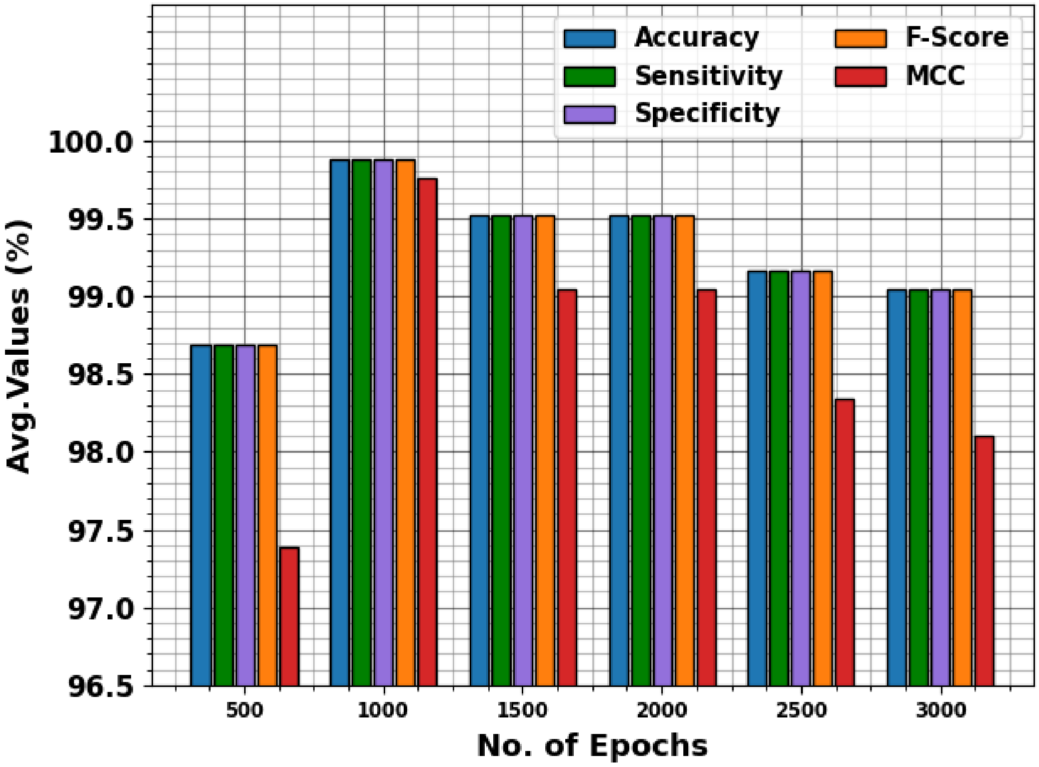


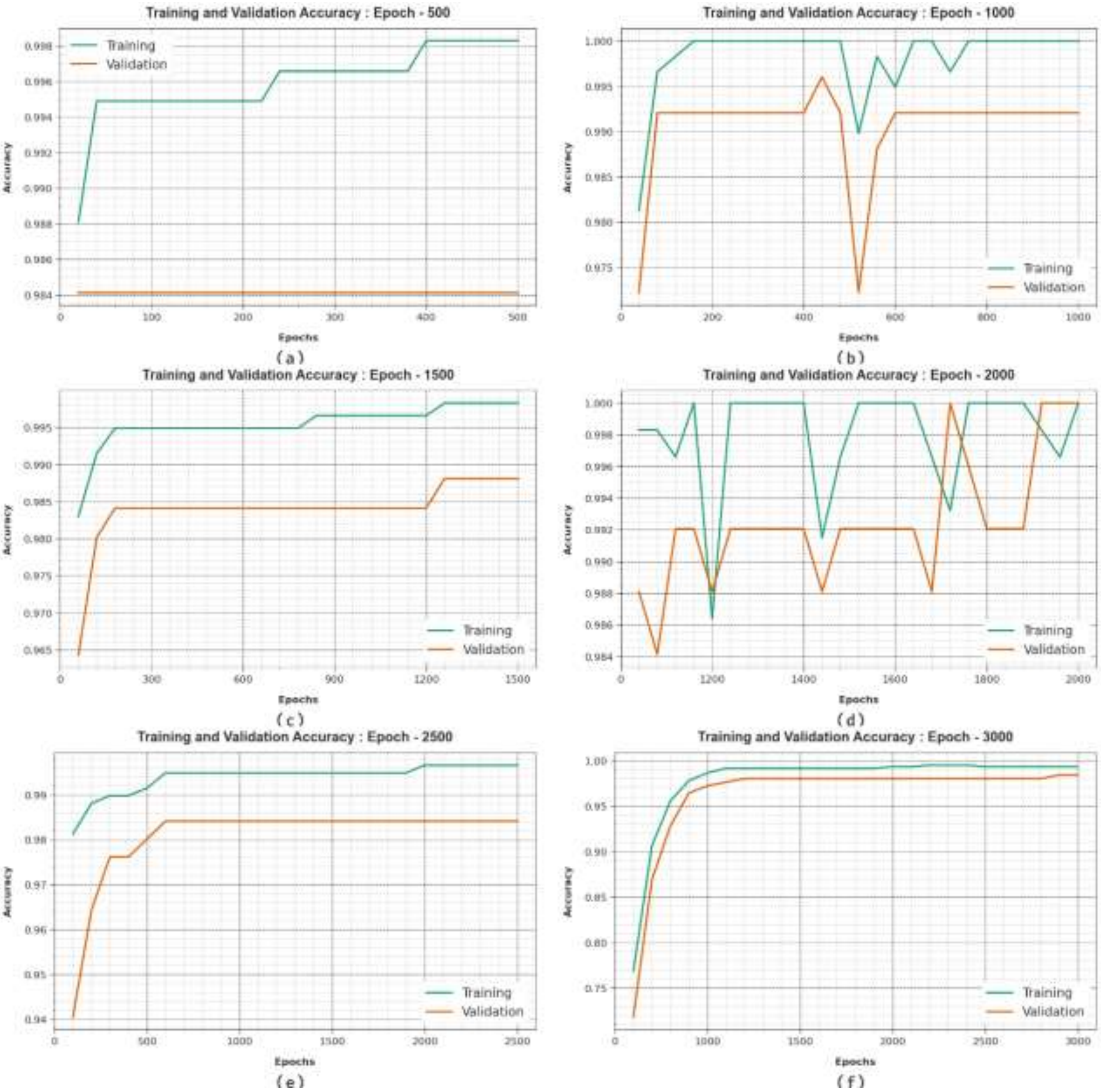
Figure 4. Average outcome of EBSAEDL-RD system under various epochs.

The $accu_y$ curves for training (TR) and validation (VL) depicted in Figure 5 for the EBSAEDL-RD approach under epochs 500–3000 offer appreciated insights into its outcome. Specifically, there is a consistent development in both TR as well as TS $accu_y$ with maximum epochs, demonstrating the model's ability to learn and distinguish designs in both TR and TS data. The rising trend in TS $accu_y$ underlines the model's adaptability to the TR dataset and its capability to create accurate predictions on unnoticed data, emphasizing robust generalized abilities.

Figure 6 offers a widespread outline of the TR and TS loss performances for the EBSAEDL-RD system on distinct epochs 500–3000. The TR loss constantly diminishes as the model increases its weights to reduce classifier errors on both databases. The loss curves exemplify the model's alignment with the TR data, emphasizing its proficiency to capture designs successfully in both databases. The continuous refinement of parameters in the EBSAEDL-RD approach is noticeable, intended to diminish discrepancies among predictions and actual TR labels.

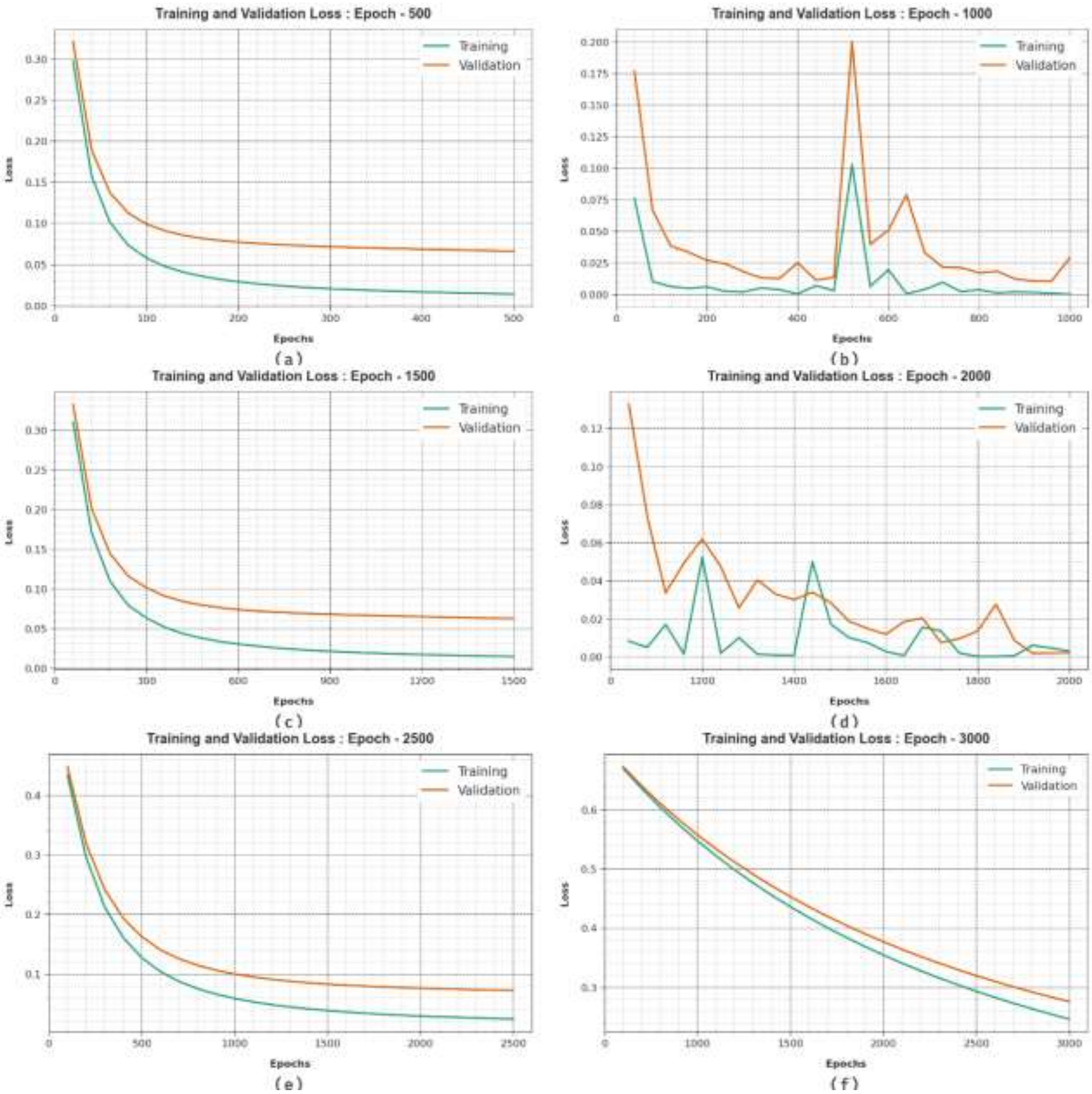
Concerning the PR curve existing in Figure 7, the findings affirm that the EBSAEDL-RD methodology under epoch 1000 consistently achieves improved PR values across each class. These results underscore the model's effective capacity for discriminating between various classes, highlighting its effectiveness in correctly distinguishing classes.

Additionally, in Figure 8, we existing ROC curves generated by the EBSAEDL-RD algorithm under epoch 1000, demonstrating its proficiency in distinguishing among class labels. These curves provide appreciated insights into how the tradeoff between TPR and FPR differs across dissimilar classification epochs and thresholds. The results underscore the model's correct classification solution under two class labels, highlighting its efficacy in addressing diverse classification tests.



(a) (b) (c) (d) (e) (f)

Figure 5. $Accu_y$ curve of the EBSAEDL-RD method (a–f) epochs 500–3000.



Training and Validation Loss : Epoch - 500

(a)

Training and Validation Loss : Epoch - 1000

(b)

Training and Validation Loss : Epoch - 1500

(c)

Training and Validation Loss : Epoch - 2000

(d)

Training and Validation Loss : Epoch - 2500

(e)

Training and Validation Loss : Epoch - 3000

(f)

Figure 6. Loss curve of the EBSAEDL-RD system (a–f) epochs 500–3000.

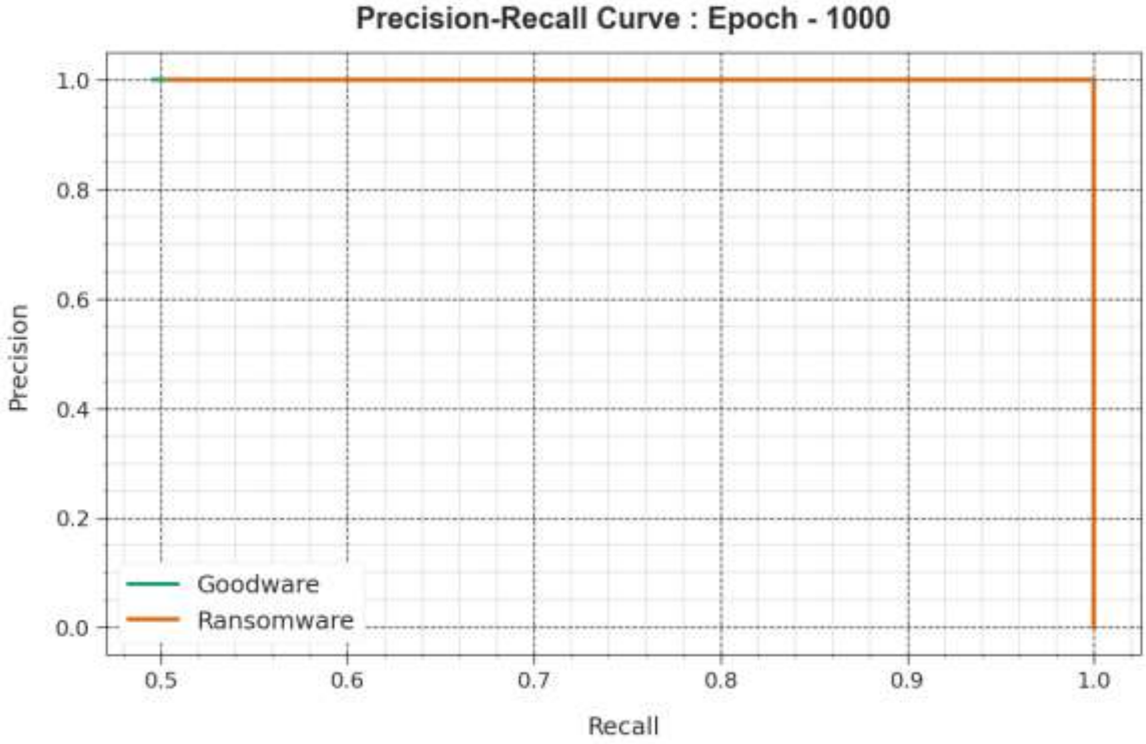


Figure 7. PR curve of the EBSAEDL-RD algorithm under epoch 1000.

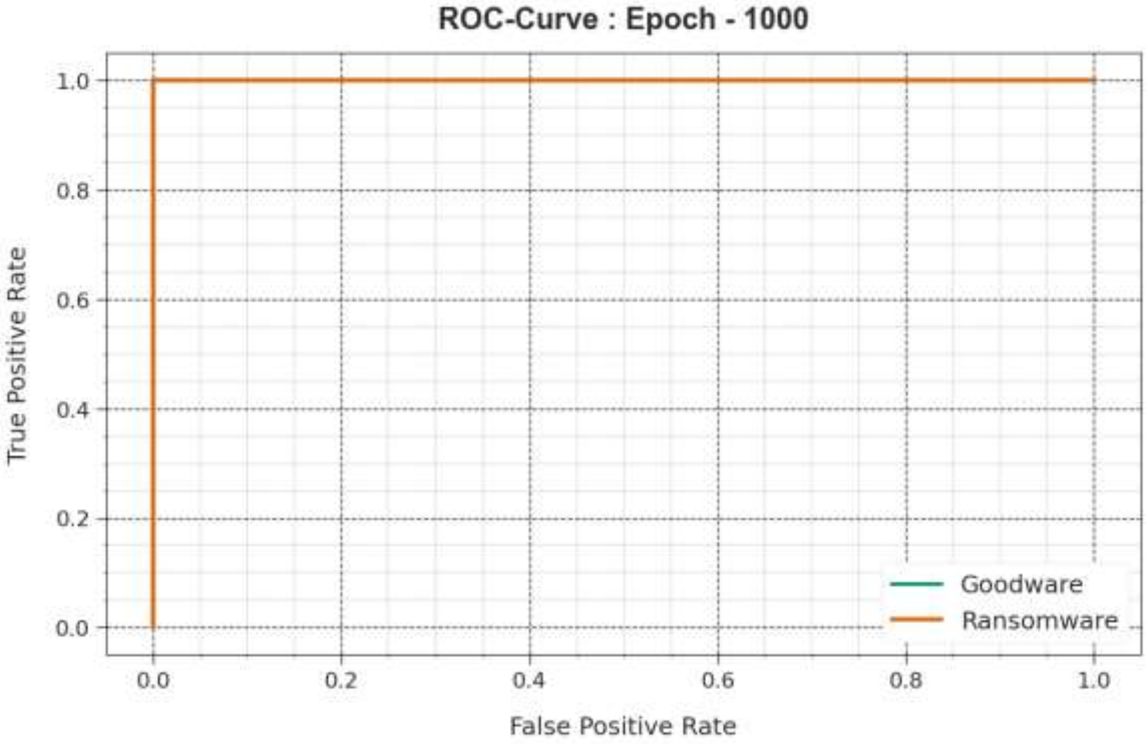


Figure 8. ROC curve of the EBSAEDL-RD technique under epoch 1000.

In Table 3, the comparative results of the EBSAEDL-RD technique are portrayed [18]. Figure 9 investigates the comparison study of the EBSAEDL-RD technique in terms of $accu_y$. The outcomes show that the EBSAEDL-RD method gains improved $accu_y$ values. Based on $accu_y$, the EBSAEDL-RD technique offers the greatest $accu_y$ of 99.88% whereas the OGCNN-RWD, DWOML, Bagging, AdaBoostM1, ROF, DT, and RF systems offer lesser $accu_y$ values of 99.67%, 99.12%, 98.53%, 96.19%, 95.87%, 97.71%, and 98.86%, respectively.

Table 3. Comparison analysis of the EBSAEDL-RD method with other techniques.

Methods	$Accu_y$	$Sens_y$	$Spec_y$
EBSAEDL-RD	99.88	99.88	99.88
OGCNN-RWD	99.67	99.68	99.68
DWOML	99.12	99.49	99.24
Bagging	98.53	93.74	96.14
AdaBoostM1	96.19	94.56	94.67
Rotation Forest (ROF)	95.87	96.81	97.44
Decision Tree (DT)	97.71	97.87	98.20
Random Forest (RF)	98.86	98.82	98.32

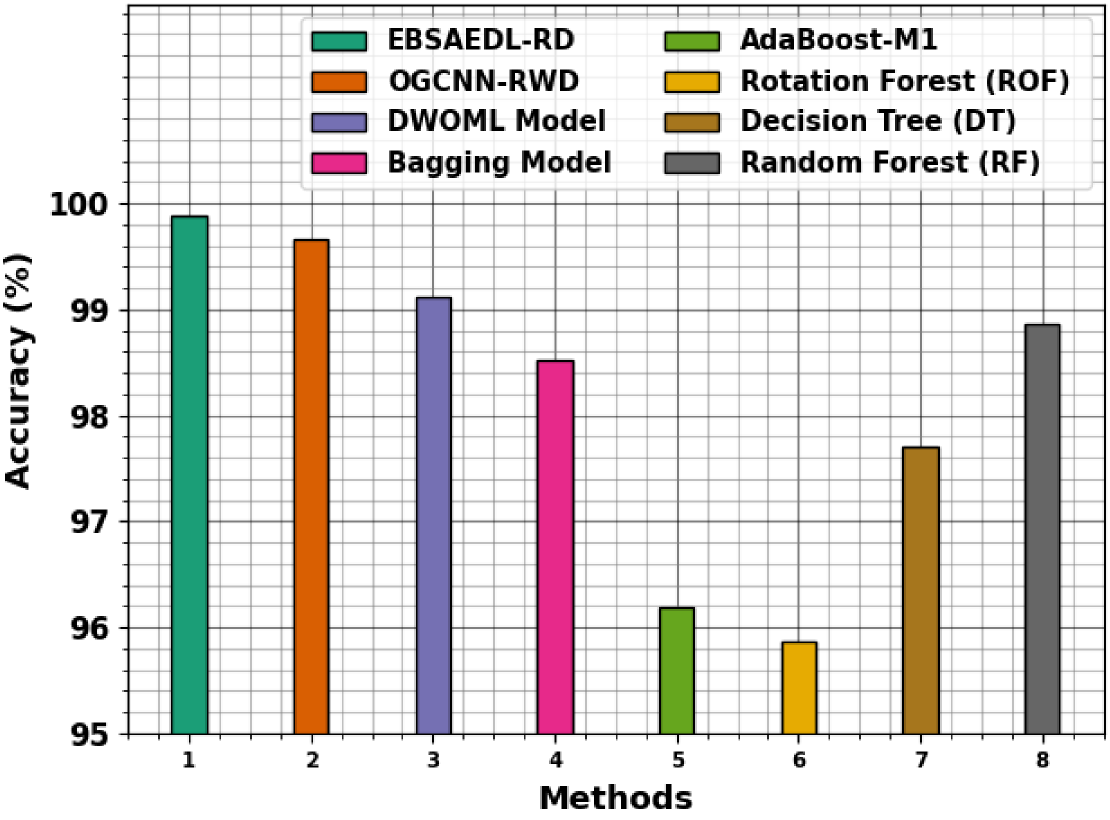


Figure 9. $Accu_y$ of the EBSAEDL-RD method compared with other systems.

Figure 10 scrutinizes the comparison analysis of the EBSAEDL-RD algorithm with respect to $sens_y$ and $spec_y$. The outcome means that the EBSAEDL-RD methodology obtains superior $sens_y$ and $spec_y$ values. Based on $sens_y$, the EBSAEDL-RD method offers a higher $sens_y$ of 99.88% whereas the OGCNN-RWD, DWOML, Bagging, AdaBoostM1, ROF, DT, and RF algorithms attain lower $sens_y$ values of 99.68%, 99.49%, 93.74%, 94.56%, 96.81%, 97.87%, and 98.82%, respectively. According to $spec_y$, the EBSAEDL-RD system offers an enhanced $sens_y$ of 99.88% whereas the OGCNN-RWD, DWOML, Bagging, AdaBoostM1, ROF, DT, and RF systems reach reduced $spec_y$ values of 99.68%, 99.24%, 96.14%, 94.67%, 97.44%, 98.20%, and 98.32%, respectively. Accordingly, the EBSAEDL-RD system has been executed for enhanced ransomware detection.

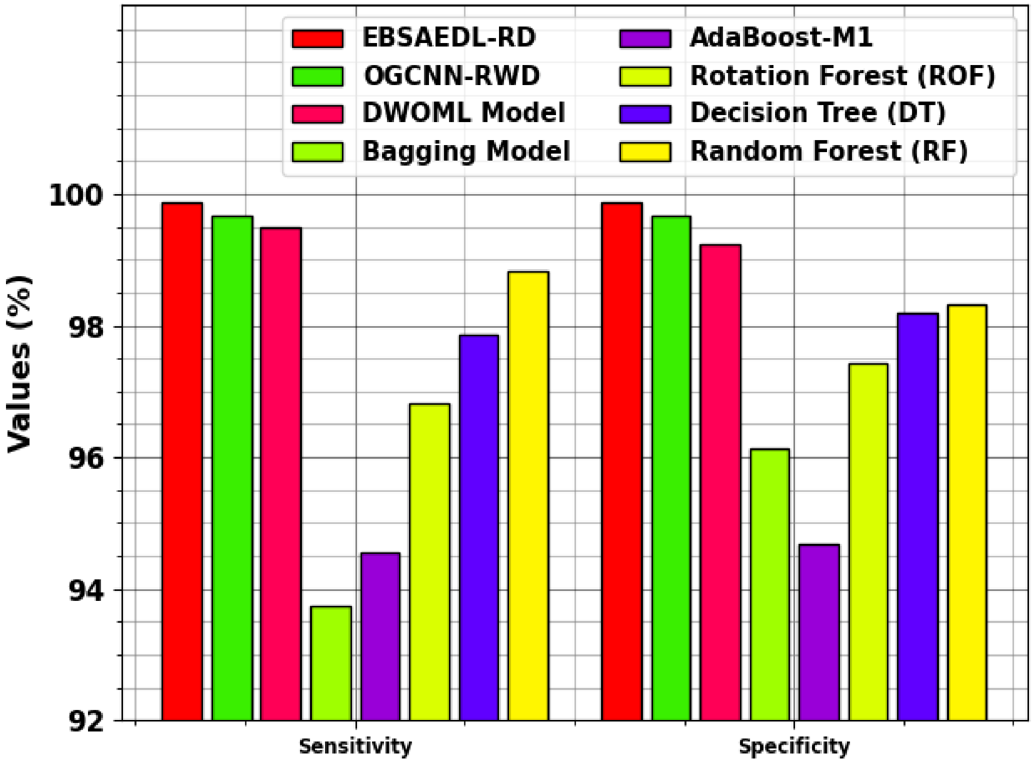


Figure 10. $Sens_y$ and $Spec_y$ of the EBSAEDL-RD technique compared with other approaches.

4. Conclusions

In this study, we design a new EBSAEDL-RD method in IoT security. The purpose of the EBSAEDL-RD technique is to recognize and classify the ransomware to achieve security in the IoT platform. To achieve this, the EBSAEDL-RD technique contains different types of processes, namely min-max normalization, EBSA-based feature selection, BiGRU classification, and SSA-based hyperparameter tuning. Initially, the EBSAEDL-RD technique employs min-max normalization to scale the input data into useful format. Then, the EBSAEDL-RD technique makes use of the EBSA method to select an optimum set of features. Meanwhile, the classification of ransomware takes place using the BiGRU model. At last, SSA can be applied for optimum hyperparameter selection of the BiGRU model. The wide-ranging experiments of the EBSAEDL-RD approach are performed on benchmark data. The obtained results highlighted that the EBSAEDL-RD method reaches better performance over other models on IoT security.

Acknowledgments

The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA for funding this research work through the project number "NBU-FFR-2024-170-01".

References

1. C. W. Tien, S. W. Chen, T. Ban, S. Y. Kuo, Machine learning framework to analyze iot malware using elf and opcode features, *Digit. Threat. Res. Pract.*, **1** (2020), 1–19. https://doi.org/10.1145/3378448
2. S. I. Bae, G. B. Lee, E. G. Im, Ransomware detection using machine learning algorithms, *Concurr. Comput. Pract. Exp.* **31** (2020), e5422.
3. S. Sharma, C. R. Krishna, R. Kumar, Android Ransomware Detection using Machine Learning Techniques: A Comparative Analysis on GPU and CPU. In Proceedings of the 2020 21st International Arab Conference on Information Technology (ACIT), Giza, Egypt, 28–30 November 2020; IEEE: Piscataway, NJ, USA, 2020, 1–6. https://doi.org/10.1109/ACIT50332.2020.9300108
4. D. W. Fernando, N. Komninos, T. Chen, A study on the evolution of ransomware detection using machine learning and deep learning techniques, *IoT*, **1** (2020), 551–604. https://doi.org/10.3390/iot1020030
5. U. Urooj, B. A. S. Al-rimy, A. Zainal, F. A. Ghaleb, M. A. Rassam, Ransomware detection using the dynamic analysis and machine learning: A survey and research directions, *Appl. Sci*. **12** (2021), 172. https://doi.org/10.3390/app12010172
6. R. Damaševiˇcius, A. Venˇckauskas, J. Toldinas, S. Grigaliunas, Ensemble-Based classification using neural networks and machine-learning models for windows pe malware detection, *Electronics,* **10** (2021), 485. https://doi.org/10.3390/electronics10040485
7. M. A. Almaiah, O. Almomani, A. Alsaaidah, S. Al-Otaibi, N. Bani-Hani, A. K. A. Hwaitat, et al., Performance investigation of principal component analysis for intrusion detection system using different support vector machine kernels, *Electronics*, **11** (2022), 3571. https://doi.org/10.3390/electronics11213571
8. A. H. Mohammad, T. Alwada'n, O. Almomani, S. Smadi, N. ElOmari, Bio-Inspired hybrid feature selection model for intrusion detection, *Comput. Mater. Contin*., **73** (2022), 133–150. https://doi.org/10.32604/cmc.2022.027475
9. Y. Dion, S. N. Brohi, An experimental study to evaluate the performance of machine learning alogrithms in ransomware detection, *J. Eng. Sci. Technol*., **15** (2020), 967–981.
10. F. Noorbehbahani, F. Rasouli, M. Saberi, Analysis of machine learning techniques for ransomware detection, In Proceedings of the 2019 16th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC), Mashhad, Iran, 28–29 August 2019; IEEE: Piscataway, NJ, USA, 2019, 128–133. https://doi.org/10.1109/ISCISC48546.2019.8985139

11. X. Deng, M. Cen, M. Jiang, M. Lu, Ransomware early detection using deep reinforcement learning on portable executable header, *Cluster Comput.*, 2023, 1–15. https://doi.org/10.1007/s10586-023-04043-5
12. Z. Yao, Z. Wang, T. Wu, W. Lu, A hybrid data-driven deep learning prediction framework for lake water level based on fusion of meteorological and hydrological multi-source data, *Nat. Resour. Res.*, 2023, 1–28. https://doi.org/10.1007/s11053-023-10284-3
13. L. Almomani, A. Alkhayer, W. El-Shafai, E2E-RDS: Efficient End-to-End ransomware detection system based on Static-Based ML and Vision-Based DL approaches, *Sensors*, **23** (2023), 4467. https://doi.org/10.3390/s23094467
14. M. A. Alohali, M. Elsadig, F. N. Al-Wesabi, M. Al Duhayyim, A. M. Hilal, A. Motwakel, Optimal deep learning based ransomware detection and classification in the internet of things environment, *Comput. Syst. Sci. Eng.*, **46** (2023). https://doi.org/10.32604/csse.2023.036802
15. H. Kim, J. Park, H. Kwon, K. Jang, H. Seo, Convolutional neural network-based cryptography ransomware detection for low-end embedded processors, *Mathematics*, **9** (2021), 705. https://doi.org/10.3390/math9070705
16. B. Zhang, W. Xiao, X. Xiao, A. K. Sangaiah, W. Zhang, J. Zhang, Ransomware classification using patch-based CNN and self-attention network on embedded N-grams of opcodes, *Future Gener. Comp. Sy.*, **110** (2020), 708–720. https://doi.org/10.1016/j.future.2019.09.025
17. Q. Abu Al-Haija, S. Zein-Sabatto, An efficient deep-learning-based detection and classification system for cyber-attacks in IoT communication networks, *Electronics*, **9** (2020), 2152. https://doi.org/10.3390/electronics9122152
18. H. Khalid, K. Mahmood, M. Khalid, M. Othman, M. Al Duhayyim, A. E Osman, et al., Optimal graph convolutional neural network-based ransomware detection for cybersecurity in IoT environment, *Appl. Sci.*, **13** (2023), 5167. https://doi.org/10.3390/app13085167
19. A. R. Khan, A. Yasin, S. M. Usman, S. Hussain, S. Khalid, S. S. Ullah, Exploring lightweight deep learning solution for malware detection in IoT constraint environment, *Electronics*, **11** (2022), 4147. https://doi.org/10.3390/electronics11244147
20. M. Basnet, S. Poudyal, M. H. Ali, D. Dasgupta, Ransomware detection using deep learning in the SCADA system of electric vehicle charging station, In 2021 IEEE PES Innovative Smart Grid Technologies Conference-Latin America (ISGT Latin America), IEEE, 1–5. https://doi.org/10.1109/ISGTLatinAmerica52371.2021.9543031
21. M. Ghahramani, R. Taheri, M. Shojafar, R. Javidan, S. Wan, Deep Image: A precious image based deep learning method for online malware detection in IoT Environment, 2022. arXiv preprint arXiv:2204.01690.
22. D. Singh, B. Singh, Investigating the impact of data normalization on classification performance, *Appl. Soft Comput.*, **97** (2020), 105524. https://doi.org/10.1016/j.asoc.2019.105524
23. O. N. Oyelade, A. E. Ezugwu, A bioinspired neural architecture search based convolutional neural network for breast cancer detection using histopathology images, *Sci. Rep.*, **11** (2021), 19940. https://doi.org/10.1038/s41598-021-98978-7
24. M. D. Dangut, I. K. Jennions, S. King, Z. Skaf, A rare failure detection model for aircraft predictive maintenance using a deep hybrid learning approach, *Neural Comput. Appl.*, **35** (2023), 2991–3009. https://doi.org/10.1007/s00521-022-07167-8

25. C. Li, J. Zhou, K. Du, D. Dias, Stability prediction of hard rock pillar using support vector machine optimized by three metaheuristic algorithms, *Int. J. Min. Sci. Technol.*, **33** (2023), 1019–1036. https://doi.org/10.1016/j.ijmst.2023.06.001
26. K. A. Alissa, D. H. Elkamchouchi, K. Tarmissi, A. Yafoz, R. Alsini, O. Alghushairy, et al., Dwarf Mongoose Optimization with machine-learning-driven ransomware detection in internet of things environment, *Appl. Sci.*, **12** (2022), 9513. https://doi.org/10.3390/app12199513