Computer Systems
Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2023.036802
Article

Optimal Deep Learning Based Ransomware Detection and Classification in the Internet of Things Environment

Manal Abdullah Alohali[1], Muna Elsadig[1], Fahd N. Al-Wesabi[2], Mesfer Al Duhayyim[3], Anwer Mustafa Hilal[4,*] and Abdelwahed Motwakel[4]

[1]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia
[2]Department of Computer Science, College of Science & Art at Mahayil, King Khalid University, Saudi Arabia
[3]Department of Computer Science, College of Sciences and Humanities- Aflaj, Prince Sattam bin Abdulaziz University, Saudi Arabia
[4]Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, AlKharj, Saudi Arabia
*Corresponding Author: Anwer Mustafa Hilal. Email: a.hilal@psau.edu.sa

Received: 12 October 2022; Accepted: 06 January 2023

Abstract: With the advent of the Internet of Things (IoT), several devices like sensors nowadays can interact and easily share information. But the IoT model is prone to security concerns as several attackers try to hit the network and make it vulnerable. In such scenarios, security concern is the most prominent. Different models were intended to address these security problems; still, several emergent variants of botnet attacks like Bashlite, Mirai, and Persirai use security breaches. The malware classification and detection in the IoT model is still a problem, as the adversary reliably generates a new variant of IoT malware and actively searches for compromise on the victim devices. This article develops a Sine Cosine Algorithm with Deep Learning based Ransomware Detection and Classification (SCADL-RWDC) method in an IoT environment. In the presented SCADL-RWDC technique, the major intention exists in recognizing and classifying ransomware attacks in the IoT platform. The SCADL-RWDC technique uses the SCA feature selection (SCA-FS) model to improve the detection rate. Besides, the SCADL-RWDC technique exploits the hybrid grey wolf optimizer (HGWO) with a gated recurrent unit (GRU) model for ransomware classification. A widespread experimental analysis is performed to exhibit the enhanced ransomware detection outcomes of the SCADL-RWDC technique. The comparison study reported the enhancement of the SCADL-RWDC technique over other models.

Keywords: Security; IoT network; ransomware attack; deep learning; metaheuristics

1 Introduction

The Internet of Things (IoT) is a copious amount of physical devices interconnected with the internet. Embedded with software and sensor nodes that can share and collect information online, IoT gadget is invaluable for improving productivity and enhancing a large number of processes all over the industry [1–3]. There is a considerable expansion in the severity and amount of cyber-based attacks. Generally, distinct variants of malware are the major reason for cyberattacks. Malware is software proposed to use network and computer systems' vulnerabilities to gain financial benefits and execute malicious activities [4]. Trojans, viruses, backdoors, worms, ransomware, and rootkits are popular examples of malware. Ransomware attack causes a problem for the distributed IoT platform and halts smooth work among heterogeneous data centre [5]. This mechanism contains the complex structure of method and corpora. The data centre environment has a large amount of information and pays money to avoid damage to the exploitation and reputation of information [6].

Even though various methodologies are developed for detecting malware [7], it is still a prominent topic amongst research workers because of the increasing number of new malware and its difficulty [8]. The conventional method for malware detection is signature-based. This method requires saving each known and existing malware signature to identify malware samples. The general problem with this method is that they needed help identifying unknown and new malware samples [9]. As well, database updating with new signatures takes considerable time, and at that time, malware is capable of performing malicious actions [1–3, 10].

Furthermore, storing each present malware's signature could be more efficient and costly [5–7, 11]. The heuristic machine learning (ML) method is commonly applied to overcome this limitation. First, this technique extracts feature from malware sample that describes the behaviour and content of the malware. Then, this feature is used for training a model to identify malware samples [8]. But this method requires the detection of significant features beforehand, which is sometimes impossible, costly, and time-consuming due to limited available malware samples, which is a major constraint in IIoT and IoT platforms [9,10].

This article develops a Sine Cosine Algorithm with Deep Learning based Ransomware Detection and Classification (SCADL-RWDC) technique in an IoT environment. In the presented SCADL-RWDC technique, the major intention exists in recognizing and classifying ransomware attacks in the IoT platform. The SCADL-RWDC technique uses the SCA feature selection (SCA-FS) model to improve the detection rate. Besides, the SCADL-RWDC technique exploits a hybrid grey wolf optimizer (HGWO) with a gated recurrent unit (GRU) model for the ransomware classification process. A widespread experimental analysis is performed to exhibit the enhanced ransomware detection outcomes of the SCADL-RWDC technique.

2 Related Works

The authors in [12] employed DL approaches to extract the latent representation of higher dimensional data to identify malicious performance accurately. Especially this method present was dependent upon a hybrid feature engineering system of traditional and VAEs. This system was utilized to reduce the data's dimensionality and extract an optimum representation of gathered model actions. Next, a novel feature vector has passed to a constructed classification dependent upon DNN and batch-normalized methods. Naeem [13] examined a further accurate and fast method to detect malware from IoT environments. The authors establish a Malware Threat Hunting System (MTHS) in the presented method. In MTHS, primary converts malware binary to colour image and conducts the

ML or DL studies for effectual malware detection. The authors lastly make a baseline for comparing the efficiency of MTHS with classic recent malware detection techniques.

Naeem et al. [14] proposed a structure for detecting malware attacks on the Industrial Internet of Things (MD-IIOT). This technique was presented for a comprehensive investigation of malware, depending upon the colour image visualized and deep CNN. The outcomes of the presented system are related to former techniques for malware detection. Moti et al. [15] presented MalGan, a structure to detect and generate novel malware instances dependent upon the raw byte code at the edge layer of IoT networks. CNN is employed to extract higher-level features, and a boundary-seeking Generative Adversarial Network (GAN) system generates novel malware instances. Therefore, even with some malware instances, an essential count of earlier unseen malware instances was detectable with maximum accuracy. For capturing the short- and long-term dependency of features, the authors utilized an attention-based method, an integration of CNN and LSTM. The attention system enhances the model's efficiency by increasing or decreasing attention for particular features. Kumar [16] introduces a new malware classifier with fine-tuning CNNs (MCFTCNN) method. This method utilizes deep transfer learning (DTL) to classify the malware image into its corresponding family. The presented method improves the ResNet50 technique by changing the final layer with a fully connected (FC) dense layer.

In [17], an automated ML-based ransomware classification model is derived. Using the malware life cycle on the Windows platform, real-time examination of ransomware samples is performed to identify various traits of harmful code patterns. The grid search hyperparameter optimizer is used to determine the optimal fit approaches, and the results are examined over the test dataset. The authors in [18] presented the major suggestion and schemes to mitigate ransomware. An automated indexing approach is developed to offer searching functions, similarity verification, classification, and clustering. The proposed model mainly aims at the original ransomware binary and the indexing engine based on the hybridized data from the static analyzer system. The proposed model tracks and classifies ransomware depending on the static features to determine the resemblance among various ransomware samples.

3 The Proposed Model

In this article, we have introduced automated ransomware detection using the SCADL-RWDC technique. The goal of the SCADL-RWDC technique lies in the recognition and classification of ransomware attacks in the IoT environment. It follows three processing stages: SCA-FS-based feature subset selection, GRU classification, and HGWO parameter tuning. Fig. 1 represents the block diagram of the SCADL-RWDC system.

3.1 Data Pre-Processing

In the primary stage, the min-max normalizes system was executed to transform the input database into a suitable format. Min-max normalized system is utilized for scaling the feature in zero and one with the subsequent expression.

$$v' = \frac{v - \min_A}{\max_A - \min_A} \tag{1}$$

In Eq. (1), $\min_A$ and $\max_A$ imply the lower and higher values of features A. The normalizing and original values of attributes, A, are considered as v and v' correspondingly. It is noticeable in the equation mentioned above that the higher and lower feature values are mapped to 0 and 1 correspondingly.

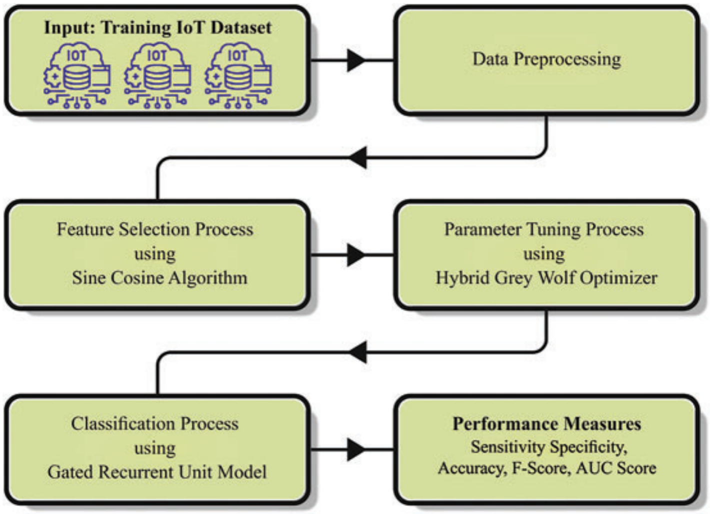

Figure 1: Block diagram of SCADL-RWDC system

3.2 Algorithmic Design of SCA-FS Technique

The SCADL-RWDC technique applied the SCA-FS model to choose features optimally in this work. SCA is a metaheuristic approach developed with shallow efficiency [19] and a population-based method that begins with searching for random solutions. Thus, each random optimization highlights the exploitation and exploration of the problem. In SCA, two distinct mathematical formulas are utilized for updating the solution for the balance of exploitation and exploration:

$$X_{ij}^{i+1} = \begin{cases} X_{ii}^{t} + r_1 \times sin\,(r_2) \times \left|r_3 P_j^t - X_{ii}^t\right|) & r_4 < 0.5 \\ X_{ij}^{t} + r_1 \times cos\,(r_2) \times \left|r_3 P_j^t - X_{ij}^t\right|) & r_4 \geq 0.5 \end{cases} \tag{2}$$

X_i^{ti} denotes the $j{-}th$ parameter of *the* $i{-}th$ location at the t generation population, r_1, r_2, and r_3 indicate each random number, and P_j^f represents the $j{-}th$ parameter of the terminal point at the t generation population.

Four variables should be presented. It reduced linearly from a to 0, which balances the exploitation and exploration. Furthermore, the variable $r_2 \in [0, 2\pi]$ is a random value for updating the following solution in the accurate direction. Lastly, the r_4 variable characterizes that the sine and cosine function was chosen in Eq. (1) with corresponding probability.

The comprehensive equation for variable r_1 is given by:

$$r_1 = a \times \left(1 - \frac{FEs}{\mathrm{Max}FEs}\right) \tag{3}$$

In Eq. (3), *FEs* specify the present computation, *MaxFEs* show the maximal amount of computations, and a is constant. Fig. 2 demonstrates the flowchart of SCA.

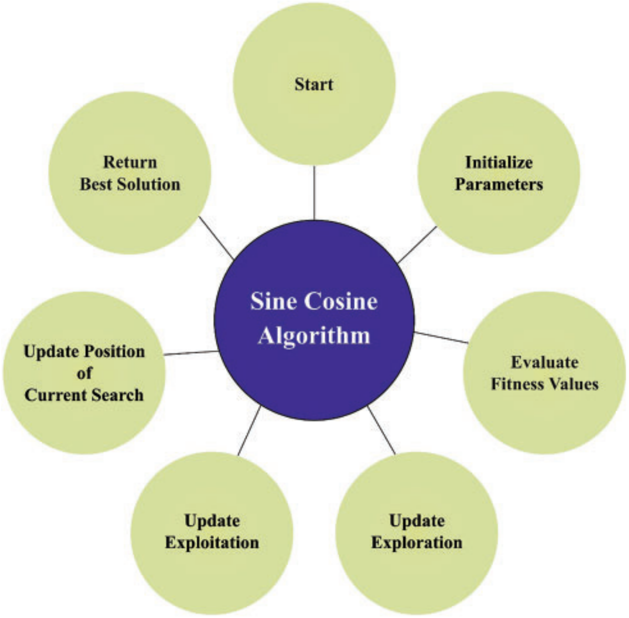

Figure 2: Flowchart of SCA

It demonstrates the present trend of the solution towards the target solution. The equation of SCA is presented to two arithmetical functions involving sine and cosine operations. The effects on those functions depend on how to attain the following solution and determine the location amongst the sensible and the existing solutions to the problem.

The effect of cosine and sine functions in a random integer on the following solution. Two trends are analyzed: the inner direction of the present and outside space.

The common step of SCA starts with the optimization procedure of a set of primary random solutions. With increasing assessments, the better solution is the target solution presently retained.

Consequently, the fitness function is utilized for evaluating individual solutions as follows:

$$Fitness = \alpha * \ ErrorRate + (1 - \alpha) * \frac{\#SF}{\#All\_F} \tag{4}$$

Whereas ErrorRate denotes the classification error rate.

3.3 Ransomware Attack Detection Using GRU Model

The SCADL-RWDC technique exploited the HGWO with the GRU model for ransomware classification. GRU network is a variant of the LSTM network in RNN that mostly replaces the two gating units of LSTM (input and forgetting gates) with one gating unit (update gate) [20]. It is well-designed and employed in the area of time series prediction.

Similarly, The GRU-NN works with the LSTM, where it has two gates, such as reset and update gates. These gates receive the h_{t-1} hidden state at the preceding moment and *the* x_t input dataset at the current moment. The output gating signal is represented as r_t and z_t, correspondingly. Also, the gating unit comprises sigmoid and dot product functions.

Short-time series dependency is attained through the reset gate:

$$r_t = \sigma\left(W_{xr}x_t + W_{hr}h_{t-1} + b_r\right). \tag{5}$$

Estimate update gate in the following:

$$z_t = \sigma\left(W_{xz}x_t + W_{hz}h_{t-1} + b_z\right). \tag{6}$$

The candidate vector can be attained afterwards by updating the following:

$$\widetilde{h}_t = \tanh\left(W_{x\tilde{h}}x_t + W_{h\tilde{h}}\left(r_t h_{t-1}\right) + b_c\right). \tag{7}$$

Update memory to obtain hidden layer output outcomes:

$$h_t = \left(1 - z_t\right) h_{t-1} + z_t\widetilde{h}_t, \tag{8}$$

In Eq. (7), σ denotes the sigmoid function, and tanh represents the tanh function. W_{xz}, W_{xr}, and W_{xc} and W_{hz}, W_{hr}, and $W_{hi\widetilde{ch}}$ indicates the weighted matrix from x_t and h_{t-1} to update gate, reset gate, and candidate hidden state, correspondingly; b_z, b_r, and b_c indicate the bias.

3.4 Hyperparameter Tuning

Finally, the HGWO algorithm is used for the hyperparameter tuning process. In the HGWO algorithm, for the position update, the target encircling nature of the grey wolf can be arithmetically modelled [21]. The distance between the target and grey wolf for different wolf groups is formulated as follows.

$$D = \left|C \times X_{target}(t) - X_{GW}(t)\right| \tag{9}$$

The location updating of a grey wolf for the following iteration is shown as follows:

$$X_{GW}(t+1) = X_{target}(t) - A \times D \tag{10}$$

Whereas X_{GW}, X_{target} signifies the location vector of grey wolf and target, t characterizes the iteration. A and C vectors are determined to introduce flexibility for the grey wolf for the general search of the target as follows:

$$A = 2 \cdot A \cdot \left(rand_1[0,1]\right) - \left(a\left\{f\left(\frac{r}{r_{\max}}\right)\right\}\right) \tag{11}$$

$$C = 2 \cdot \left(rand_2[0,1]\right) \tag{12}$$

Whereas 'a' corresponds to the linear conversion from the exploration to the exploitation stage in the following,

$$a\left\{f\left(\frac{r}{r_{\max}}\right)\right\} = 2 - \left(\frac{2 \cdot t}{t_{\max}}\right) \tag{13}$$

In Eq. (12), t_{mox} denotes the overall iteration count. The attacking or hunting nature of the grey wolf is defined linearly, differing from 2 to 0 since it accomplishes maximal. The hunting strategy of grey wolf groups is achieved in the following.

$$D_a\left|pho = \{^{*}_{*}\right| pha\forall r(lnd() \geq 0.5\forall r(lnd() < 0.5$$

$$D_{beta} = \begin{cases} rand()^{*}sin(rand()) & \\ *abs\left(C_1^{*}X_{alpha} - X_{GW}(t)\right) & \forall rand() < 0.5 \\ rand()^{*}cos(rand()) & \\ *abs\left(C_1^{*}X_{alpha} - X_{GW}(t)\right) & \forall rand() \geq 0.5 \end{cases} \tag{14}$$

$$D_{beta} = \begin{cases} rand()^{*}sin(rand()) & \\ *abs\left(C_2^{*}X_{beta} - X_{GW}(t)\right) & \forall rand() < 0.5 \\ rand()^{*}cos(rand()) & \\ *abs\left(C_2^{*}X_{beta} - X_{GW}(t)\right) & \forall rand() \geq 0.5 \end{cases} \tag{15}$$

$$D_{beta} = \begin{cases} rand()^{*}sin(rand()) & \\ *abs\left(C_3^{*}X_{delta} - X_{GW}(t)\right) & \forall rand() < 0.5 \\ rand()^{*}cos(rand()) & \\ *abs\left(C_3^{*}X_{delta} - X_{GW}(t)\right) & \forall rand() \geq 0.5 \end{cases} \tag{16}$$

$$X^{!}(t) = X_{alpha}(t) - A_1 \times D_{alpha}(t) \tag{17}$$

$$X^{!!}(t) = X_{beta}(t) - A_2 \times D_{beta}(t) \tag{18}$$

$$X^{!!!}(t) = X_{delta}(t) - A_3 \times D_{delta}(t) \tag{19}$$

Location updating for the next iteration is attained by:

$$X_{HGW0SCA}(t+1) = \frac{X^{!}(t) + X^{!!}(t) + X^{!!!}(t)}{3} \tag{20}$$

Eqs. (11), (12) are utilized for evaluating distinct factors in (14)–(19). Now, X_{alpha}, X_{beta}, and X_{delta} denote the better position at the t' iteration, leading the remaining population to the optimum solution with the best searching capability. However, sometimes, there is a risk of getting trapped in the local optima, which results in poor population diversity. The DLH method considers that the optimum fitness value might be positioned in the neighbourhood of the location update evaluated. Therefore, the fitness values with location update in (20) are compared to the fitness values of the neighbourhood position, and the better position is repositioned to the novel position when the neighborhood outcomes in optimum fitness value. This technique might prevent getting trapped in local optima and augment the performance; thus, it is called the HGWO model.

4 Results Analysis

The proposed model is simulated using Python 3.6.5 tool on PC i5-8600k, GeForce 1050Ti 4 GB, 16 GB RAM, 250 GB SSD, and 1 TB HDD. The parameter settings are learning rate: 0.01, dropout: 0.5, batch size: 5, epoch count: 50, and activation: ReLU. The ransomware classification results of the SCADL-RWDC model are assessed on a dataset with 840 samples, as depicted in Table 1. The dataset holds 420 goodware samples and 420 ransomware samples.

The confusion matrix gained by the SCADL-RWDC technique is portrayed in Fig. 3. On 80% of the TR database, the SCADL-RWDC model has categorized 318 samples into goodware and 336 samples into ransomware. Meanwhile, on 20% of the TS database, the SCADL-RWDC approach has classified 83 samples into goodware and 82 samples into ransomware. Finally, on 70% of the TR database, the SCADL-RWDC method has classified 293 samples into goodware and 292 samples into ransomware.

Table 1: Dataset details

Class	No. of sample images
Goodware	420
Ransomware	420
Total Number of Sample Images	**840**

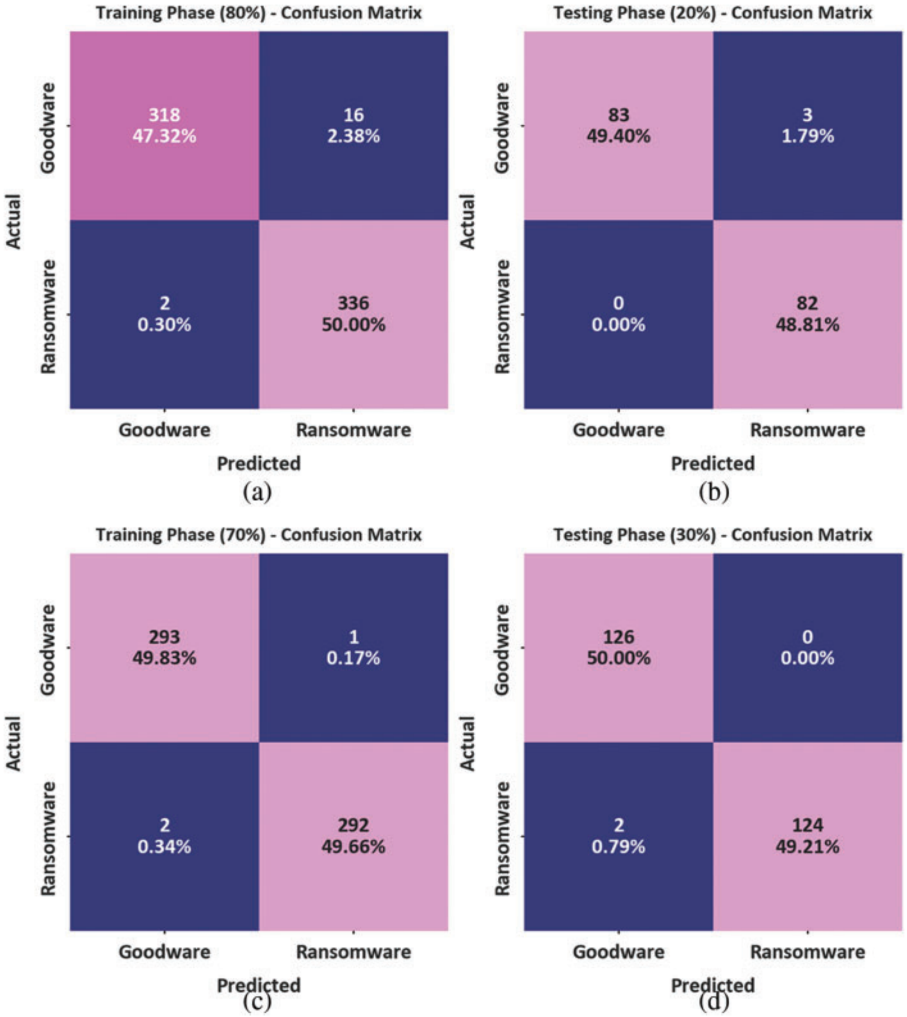

Figure 3: Confusion matrices of SCADL-RWDC system (a–b) TR and TS database of 80:20 and (c–d) TR and TS database of 70:30

Table 2 offers detailed results of the SCADL-RWDC model on 80% of TR and 20% of TS databases.

Fig. 4 reports an overall ransomware classification outcome of the SCADL-RWDC model on 80% of the TR database. In the goodware class, the SCADL-RWDC model has obtained $accu_{bal}$, $sens_y$,

$spec_y$, F_{-score}, and AUC_{scores} of 95.21%, 95.21%, 99.41%, 97.25%, and 97.31%, respectively. Eventually, in the Ransomware class, the SCADL-RWDC system attained $accu_{bal}$, $sens_y$, $spec_y$, F_{-score}, and AUC_{scores} of 99.41%, 99.41%, 95.21%, 97.39% and 97.31%, correspondingly.

Table 2: Result analysis of SCADL-RWDC system under 80:20 of TR/TS databases

Class	Accuracy$_{bal}$	Sensitivity	Specificity	F-Score	AUC Score
Training Phase (80%)					
Goodware	95.21	95.21	99.41	97.25	97.31
Ransomware	99.41	99.41	95.21	97.39	97.31
Average	**97.31**	**97.31**	**97.31**	**97.32**	**97.31**
Testing Phase (20%)					
Goodware	96.51	96.51	100.00	98.22	98.26
Ransomware	100.00	100.00	96.51	98.20	98.26
Average	**98.26**	**98.26**	**98.26**	**98.21**	**98.26**

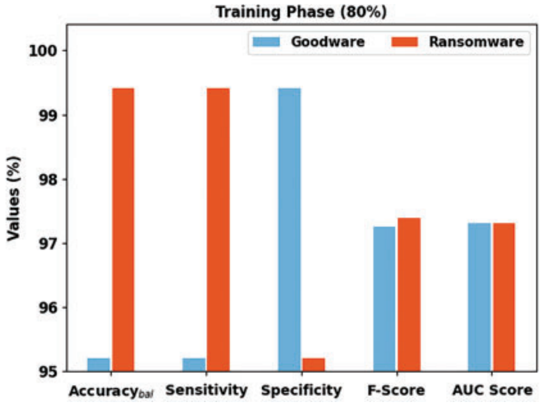

Figure 4: Result analysis of the SCADL-RWDC system in 80% of the TR database

Fig. 5 reports the overall ransomware classification outcomes of the SCADL-RWDC model on 20% of the TS database. In goodware class, the SCADL-RWDC algorithm has gained $accu_{bal}$, $sens_y$, $spec_y$, F_{-score}, and AUC_{score} of 96.51%, 96.51%, 100.00%, 98.22% and 98.26%, correspondingly. At last, in the Ransomware class, the SCADL-RWDC method has achieved $accu_{bal}$, $sens_y$, $spec_y$, F_{-score}, and AUC_{scores} of 100.00%, 100.00%, 96.51%, 98.20% and 98.26% correspondingly.

Table 3 provides a detailed outcome of the SCADL-RWDC approach on 70% of TR databases and 30% of TS databases.

Fig. 6 demonstrates an overall ransomware classification result of the SCADL-RWDC algorithm on 70% of TR data. In goodware class, the SCADL-RWDC approach has acquired $accu_{bal}$, $sens_y$, $spec_y$, F_{-score}, and AUC_{score} of 99.66%, 99.66%, 99.32%, 99.49% and 99.49%, correspondingly. Followed by, on Ransomware class, the SCADL-RWDC methodology has attained $accu_{bal}$, $sens_y$, $spec_y$, F_{-score}, and AUC_{scores} of 99.32%, 99.32%, 99.66%, 99.49% and 99.49% correspondingly.

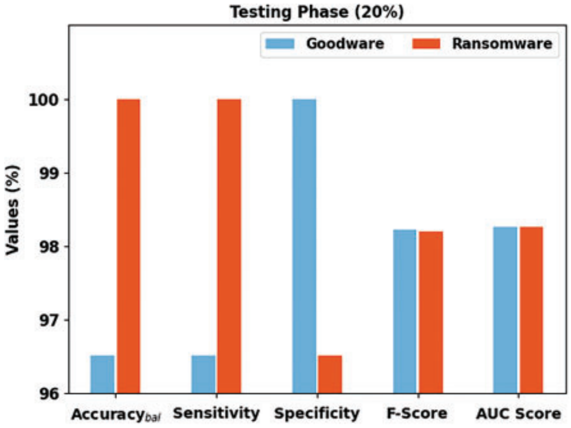

Figure 5: Result analysis of SCADL-RWDC system in 20% of the TS database

Table 3: Result analysis of SCADL-RWDC system under 70:30 of TR/TS databases

Class	Accuracy$_{bal}$	Sensitivity	Specificity	F-score	AUC score
Training Phase (70%)					
Goodware	99.66	99.66	99.32	99.49	99.49
Ransomware	99.32	99.32	99.66	99.49	99.49
Average	**99.49**	**99.49**	**99.49**	**99.49**	**99.49**
Testing Phase (30%)					
Goodware	100.00	100.00	98.41	99.21	99.21
Ransomware	98.41	98.41	100.00	99.20	99.21
Average	**99.21**	**99.21**	**99.21**	**99.21**	**99.21**

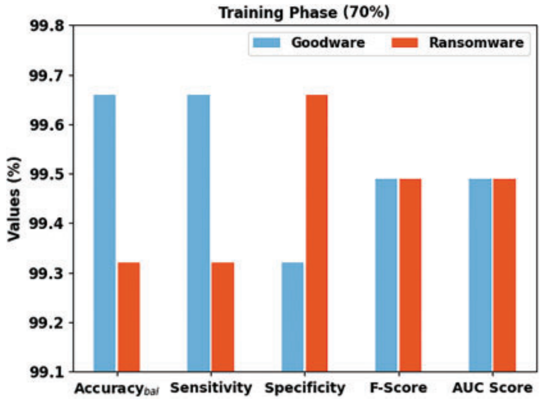

Figure 6: Result analysis of the SCADL-RWDC system in 70% of the TR database

Fig. 7 depicts an overall ransomware classification result of the SCADL-RWDC methodology on 30% of TS data. In goodware class, the SCADL-RWDC system has achieved $accu_{bal}$, $sens_y$, $spec_y$,

F_{-score}, and AUC_{scores} of 100.00%, 100.00%, 98.41%, 99.21% and 99.21%, correspondingly. In addition, in the Ransomware class, the SCADL-RWDC algorithm has reached $accu_{bal}$, $sens_y$, $spec_y$, F_{-score}, and AUC_{scores} of 98.41%, 98.41%, 100.00%, 99.20% and 99.21% correspondingly.

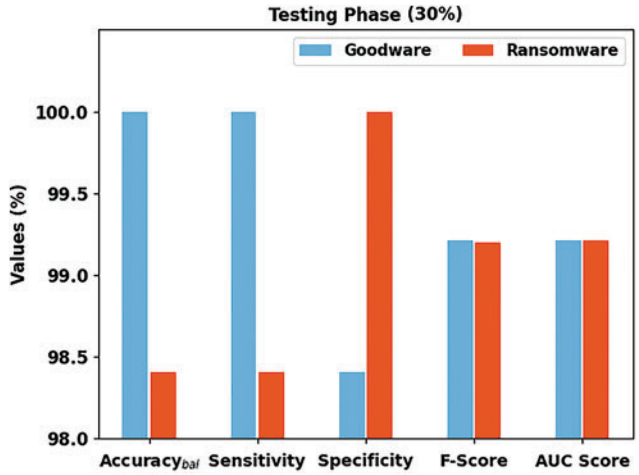

Figure 7: Result analysis of SCADL-RWDC system in 30% of the TS database

The training accuracy (TR_{acc}) and validation accuracy (VL_{acc}) obtained by the SCADL-RWDC system in the test database is displayed in Fig. 8. The simulation result shows that the SCADL-RWDC approach has realized superior values of TR_{acc} and VL_{acc}. Especially the VL_{acc} looked better than TR_{acc}.

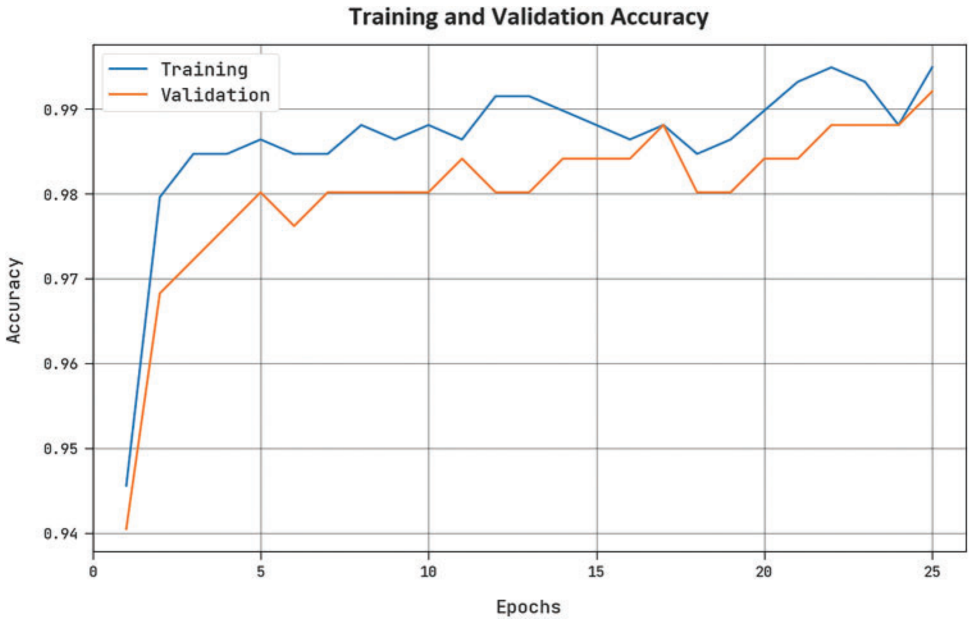

Figure 8: TR_{acc} and VL_{acc} analysis of SCADL-RWDC system

The training loss ($_{loss}$) and validation loss (VL_{loss}) gained by the SCADL-RWDC approach in the test database are portrayed in Fig. 9. The simulation result referred that the SCADL-RWDC system has attained lower values of TR_{loss} and VL_{loss}. In certain, the VL_{loss} is lesser than $_{the\ loss}$.

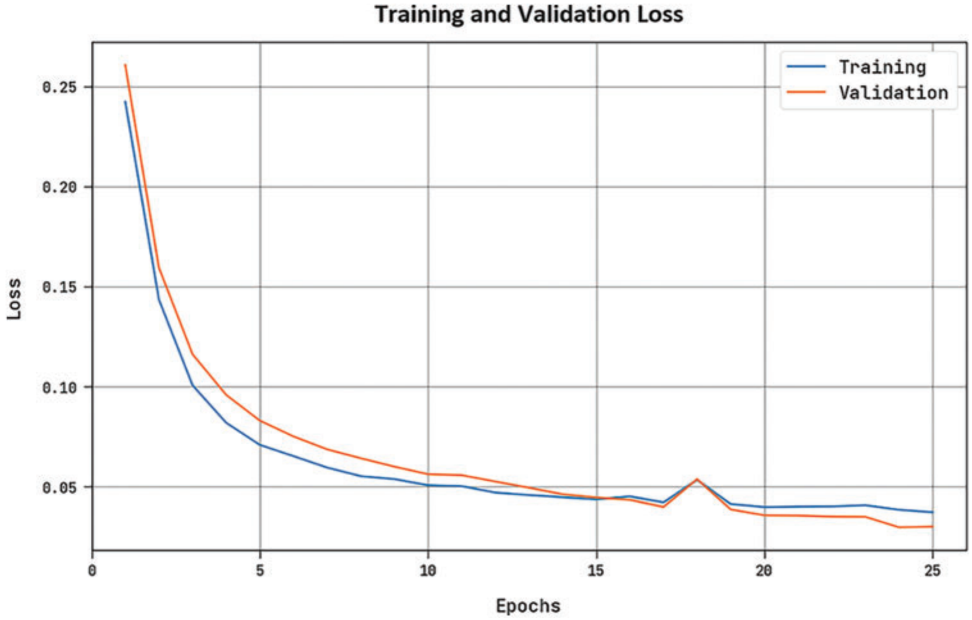

Figure 9: TR_{loss} and VL_{loss} analysis of SCADL-RWDC system

A comparative examination of the SCADL-RWDC with other ML approaches occurs in Table 4 [22].

Table 4: Comparative analysis of SCADL-RWDC algorithm with other current systems

Methods	Accuracy	Sensitivity	Specificity	F-score
SCADL-RWDC	99.49	99.49	99.49	99.49
DWOML-RWD	99.38	99.35	99.36	99.37
AdaBoost-M1 Algorithm	95.80	94.11	94.69	94.54
Bagging Algorithm	98.90	93.57	96.17	96.27
Rotation Forest Algorithm	96.17	96.41	97.50	97.30
RF Algorithm	98.63	98.96	98.57	98.12
DT Algorithm	97.51	98.24	98.23	98.06

Fig. 10 illustrates a comparison study of the SCADL-RWDC model with current ML approaches in terms of $accu_y$. The results indicated that the Adaboost-M1 and ROF models had reached a minimum $accu_{racy}$ of 95.80% and 96.17%, respectively. Then, the DT model resulted in moderate $accu_{racy}$ of 97.51%. In contrast, the DWOML-RWD, bagging, and RF approaches have obtained reasonable closer $accu_y$ values of 99.38%, 98.90%, and 98.63%, respectively. But the SCADL-RWDC model has shown higher $accu_{racy}$ of 99.49%.

Fig. 11 illustrates a comparison study of the SCADL-RWDC with current ML approaches in $sens_y$. The result indicates that the bagging and Adaboost-M1 approaches have reached a minimum

$sens_y$. of 93.57% and 94.11%, respectively. Next, the ROF methodology has resulted in a moderate $sens_y$ of 96.41%. In contrast, the DWOML-RWD, DT, and RF models have attained closer $sens_y$ values of 99.35%, 98.24% and 98.96%, correspondingly. But the SCADL-RWDC approach has shown a high $sens_y$ of 99.49%.

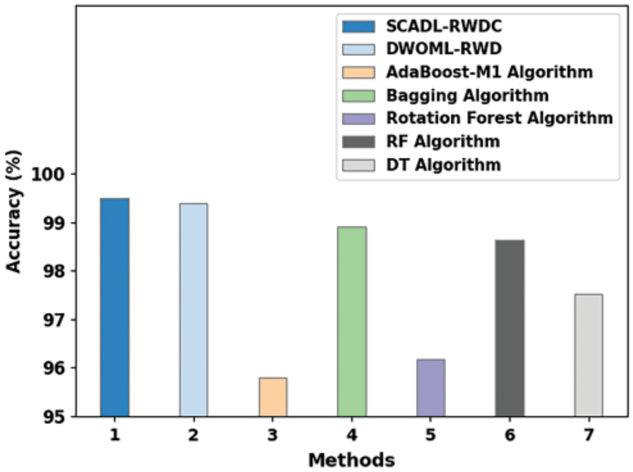

Figure 10: $Accu_{racy}$ analysis of SCADL-RWDC algorithm with other current systems

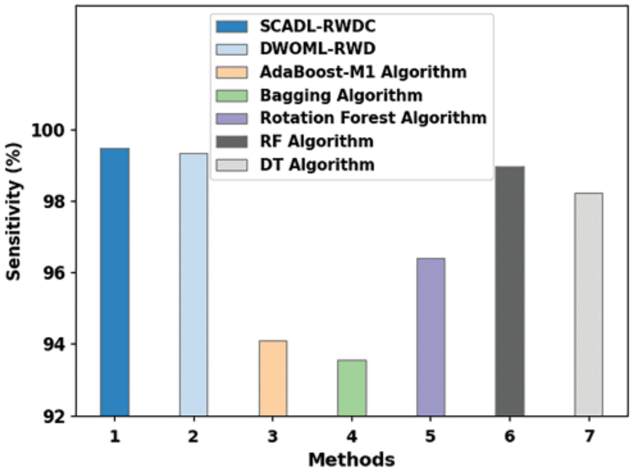

Figure 11: $Sens_y$ analysis of SCADL-RWDC technique with other present systems

Fig. 12 shows a comparison study of the SCADL-RWDC with current ML methods in $spec_y$. The result indicates that the Adaboost-M1 and bagging approaches have reached a minimum $spec_y$ of 94.69% and 96.17%, respectively. Next, the ROF model has resulted in a moderate $spec_y$ of 97.50%. In contrast, the DWOML-RWD, DT, and RF approaches have correspondingly attained closer $spec_y$ values of 99.36%, 98.23%, and 98.57%. But the SCADL-RWDC technique has demonstrated a high $spec_y$ of 99.49%.

Fig. 13 depicts a comparison study of the SCADL-RWDC with current ML techniques in terms of F_{-score}. The result indicates that the Adaboost-M1 and bagging techniques have reached a minimum F_{-score}. of 94.54% and 96.27%, respectively. Next, the ROF technique has resulted in a moderate F_{score}

of 97.30%. In contrast, the DWOML-RWD, DT, and RF methods have attained reasonably closer F_{-score} values of 99.37%, 98.06% and 98.12%, correspondingly. But the SCADL-RWDC approach has demonstrated a high F_{score} of 99.49%. These results show the enhanced performance of the SCADL-RWDC model.

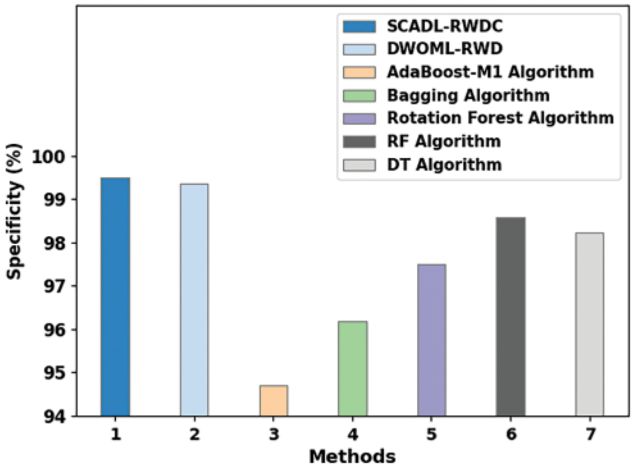

Figure 12: $Spec_y$ analysis of SCADL-RWDC algorithm with other existing systems

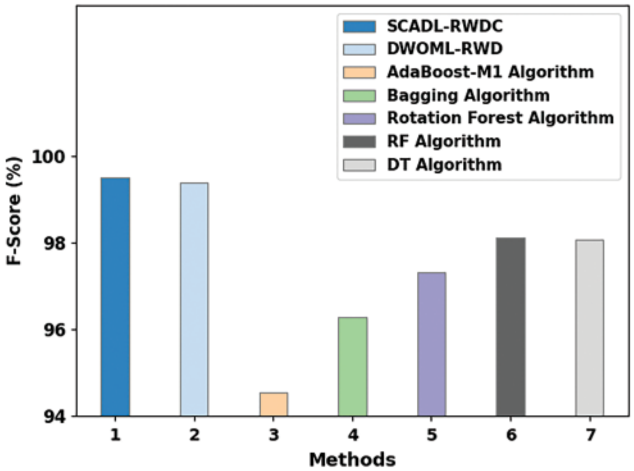

Figure 13: F_{-score} analysis of SCADL-RWDC algorithm with other existing systems

5 Conclusion

In this article, we have introduced automated ransomware detection using the SCADL-RWDC technique. The goal of the SCADL-RWDC technique lies in the recognition and classification of ransomware attacks in the IoT environment. The SCADL-RWDC technique was applied to the SCA-FS model to improve the detection rate. Besides, the SCADL-RWDC technique exploited the HGWO with the GRU model for the ransomware classification process. A widespread experimental analysis is performed to exhibit the enhanced ransomware detection outcomes of the SCADL-RWDC technique. The comparison study reported the enhancement of the SCADL-RWDC technique over other models with maximum accuracy of 99.49%. Thus, the presented SCADL-RWDC technique can be employed for the automated recognition of security attacks in the IoT platform. In the future, the performance of the SCADL-RWDC technique can be improvised using hybrid deep learning (DL) models.

Funding Statement: This work was funded by the Deanship of Scientific Research at Princess Nourah bint Abdulrahman University, through the Research Groups Program Grant No. (RGP-1443-0051).

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

References

[1] I. Bello, H. Chiroma, U. A. Abdullahi, A. Y. U. Gital, F. Jauro *et al.,* "Detecting ransomware attacks using intelligent algorithms: Recent development next direction from deep learning and big data perspectives," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 8699–8717, 2021.

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

[3] M. Humayun, N. Z. Jhanjhi, A. Alsayat and V. Ponnusamy, "Internet of things and ransomware: Evolution, mitigation and prevention," *Egyptian Informatics Journal*, vol. 22, no. 1, pp. 105–117, 2021.

[4] D. Sanvito, G. Siracusano, R. Gonzalez, and R. Bifulco, "Poster: MUSTARD–Adaptive behavioral analysis for ransomware detection," in *Proc. of the 2022 ACM SIGSAC Conf. on Computer and Communications Security*, Los Angeles CA USA, Nov, pp. 3455–3457, 2022.

[5] M. Hirano, R. Hodota and Kobayashi, "RanSAP: An open dataset of ransomware storage access patterns for training machine learning models," *Forensic Science International: Digital Investigation*, vol. 40, pp. 301314, 2022.

[6] S. Aurangzeb, H. Anwar, M. A. Naeem and M. Aleem, "BigRC-EML: Big-data based ransomware classification using ensemble machine learning," *Cluster Computing*, vol. 25, pp. 3405–3422, 2022.

[7] Y. A. Ahmed, S. Huda, B. A. S. Al-rimy, N. Alharbi, F. Saeed *et al.,* "A weighted minimum redundancy maximum relevance technique for ransomware early detection in industrial iot," *Sustainability*, vol. 14, no. 3, pp. 1231, 2022.

[8] B. M. Khammas, "Ransomware detection using random forest technique," *ICT Express*, vol. 6, no. 4, pp. 325–331, 2020.

[9] J. Du, S. H. Raza, M. Ahmad, I. Alam, S. H. Dar *et al.,* "Digital forensics as advanced ransomware pre-attack detection algorithm for endpoint data protection," *Security and Communication Networks*, vol. 2022, pp. 1–16, 2022.

[10] S. Usharani, P. M. Bala and M. M. J. Mary, "Dynamic analysis on crypto-ransomware by using machine learning: Gandcrab ransomware," *In Journal of Physics: Conference Series*, vol. 1717, no. 1, pp. 012024, 2021.

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

[12] Al-Hawawreh and E. Sitnikova, "Leveraging deep learning models for ransomware detection in the industrial internet of things environment," in *Military Communications and Information Systems Conf. (MilCIS)*, Canberra, ACT, Australia, pp. 1–6, 2019.

[13] H. Naeem, "Detection of malicious activities in internet of things environment based on binary visualization and machine intelligence," *Wireless Personal Communications*, vol. 108, no. 4, pp. 2609–2629, 2019.

[14] H. Naeem, F. Ullah, M. R. Naeem, S. Khalid, D. Vasan *et al.,* "Malware detection in industrial internet of things based on hybrid image visualization and deep learning model," *Ad Hoc Networks*, vol. 105, pp. 102154, 2020.

[15] Z. Moti, S. Hashemi, H. Karimipour, A. Dehghantanha, A. N. Jahromi *et al.,* "Generative adversarial network to detect unseen internet of things malware," *Ad Hoc Networks*, vol. 122, pp. 102591, 2021.

[16] S. Kumar, "MCFT-CNN: Malware classification with fine-tune convolution neural networks using traditional transfer learning in internet of things," *Future Generation Computer Systems*, vol. 125, pp. 334–351, 2021.

[17] S. Gokul Srinath, "Ransomware detection using machine learning and AI based re-enforcement learning method," *Journal of Optoelectronics Laser*, vol. 41, no. 11, pp. 128–133, 2022.

[18] B. Yamany, M. S. Elsayed, A. D. Jurcut, N. Abdelbaki, and M. A. Azer, "A new scheme for ransomware classification and clustering using static features," *Electronics*, vol. 11, no. 20, pp. 3307, 2022.

[19] W. Zhou, P. Wang, A. A. Heidari, X. Zhao and H. Chen, "Spiral Gaussian mutation sine cosine algorithm: Framework and comprehensive performance optimization," *Expert Systems with Applications*, vol. 209, pp. 118372, 2022.

[20] W. Lv, Y. Lv, J. Guo and J. Ma, "A lane-changing decision-making model of bus entering considering bus priority based on gru neural network," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–10, 2022.

[21] R. Devarapalli, B. V. Rao and A. Al-Durra, "Optimal parameter assessment of Solar Photovoltaic module equivalent circuit using a novel enhanced hybrid GWO-SCA algorithm," *Energy Reports*, vol. 8, pp. 12282–12301, 2022.

[22] K. A. Alissa, D. H. Elkamchouchi, K. Tarmissi, A. Yafoz, R. Alsina *et al.,* "Dwarf mongoose optimization with machine-learning-driven ransomware detection in internet of things environment," *Applied Sciences*, vol. 12, no. 19, pp. 9513, 2022.