Computer Systems Science & Engineering
DOI: 10.32604/csse.2023.033970
Article

Enhanced Gorilla Troops Optimizer with Deep Learning Enabled Cybersecurity Threat Detection

Fatma S. Alrayes[1], Najm Alotaibi[2], Jaber S. Alzahrani[3], Sana Alazwari[4], Areej Alhogail[5], Ali M. Al-Sharafi[6], Mahmoud Othman[7] and Manar Ahmed Hamza[8,*]

[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]Prince Saud Al Faisal Institute for Diplomatic Studies, Riyadh, Saudi Arabia
[3]Department of Industrial Engineering, College of Engineering at Alqunfudah, Umm Al-Qura University, Saudi Arabia
[4]Department of Information Technology, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif, 21944, Saudi Arabia
[5]Department of Information Systems, College of Computer and Information Sciences, King Saud University, Saudi Arabia
[6]Department of Computer Science, College of Computers and Information Technology, University of Bisha, Saudi Arabia
[7]Department of Computer Science, Faculty of Computers and Information Technology, Future University in Egypt, New Cairo, 11835, Egypt
[8]Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, AlKharj, Saudi Arabia
*Corresponding Author: Manar Ahmed Hamza. Email: ma.hamza@psau.edu.sa
Received: 02 July 2022; Accepted: 18 August 2022

Abstract: Recent developments in computer networks and Internet of Things (IoT) have enabled easy access to data. But the government and business sectors face several difficulties in resolving cybersecurity network issues, like novel attacks, hackers, internet criminals, and so on. Presently, malware attacks and software piracy pose serious risks in compromising the security of IoT. They can steal confidential data which results in financial and reputational losses. The advent of machine learning (ML) and deep learning (DL) models has been employed to accomplish security in the IoT cloud environment. This article presents an Enhanced Artificial Gorilla Troops Optimizer with Deep Learning Enabled Cybersecurity Threat Detection (EAGTODL-CTD) in IoT Cloud Networks. The presented EAGTODL-CTD model encompasses the identification of the threats in the IoT cloud environment. The proposed EAGTODL-CTD model mainly focuses on the conversion of input binary files to color images, where the malware can be detected using an image classification problem. The EAGTODL-CTD model pre-processes the input data to transform to a compatible format. For threat detection and classification, cascaded gated recurrent unit (CGRU) model is exploited to determine class labels. Finally, EAGTO approach is employed as a hyperparameter optimizer to tune the CGRU parameters, showing the novelty of our work. The performance evaluation of the EAGTODL-CTD model is assessed on a dataset comprising two class labels namely malignant and benign. The experimental values reported the supremacy of the EAGTODL-CTD model with increased accuracy of 99.47%.

Keywords: Cybersecurity; computer networks; threat detection; internet of things; cloud computing; deep learning

1 Introduction

The Internet of Things (IoT) is a type of network which enables objects connected by utilizing communication protocols. Connected objects are in all forms (i.e., smart fridges, watches, and scooters) and consist of sensor as well as actuator functions. The presence of IoT gadgets was increasing in modern lives and discovered applications in a broad range of environments [1]. As per Cisco report, by 2030, the quantity of connected gadgets is expected to go beyond 500 billion [2]. The IoT development has resolved several problems and developed numerous sectors [3]. The IoT enabled technologies is utilized for developing e-banking, smart cities, e-shopping, education system, managing industry, protecting human beings, and entertaining [4,5]. The IoT gadgets are employed for an open attack because of their availability on the network. Software piracy refers to the advancement of software by re-using source codes unlawfully from somebody's work and cover as the new version [6].

The machine learning (ML) ideology appeared in the mid of the 20th century, yet, it was not till the 1990s that the application took off [7]. This revolt has automated and simplified several tasks in a field plethora like industry, marketing, or medicine. Certainly, the latter provides various advantages such as the capability to process huge volumes of complicated data and manage repetitive and tedious tasks in record time [8]. The conventional methods employed in cybersecurity at the time of threat detection were majorly dependent upon manual statistical rules and data analytics that needs substantial duration [9]. So, the usage of ML methods turns out to be a crucial component in this field because of several benefits. This new supporter makes threat detection very rapid, immediate, and reactive, whereas restricting incorrect positives and uninterrupted. ML thus discovered numerous applications in cybersecurity field such as malware analysis and spam detection [10,11].

The rise in the usage of ML in the future would result in an important change in the threat landscape in 2 noteworthy manners [12]. One is the expansion of prevailing attacks. Certainly, by compiling ML techniques in present cyberattacks, these would be less recognizable, more resistant, and more reactive to the prevailing detection techniques [13]. This new generation of assaults would target the victim's vulnerability and adapt to changes in its atmosphere [14]. Another one indicates the creativity of innovative threats, till then, are not reachable because of the huge demand for data or its extreme manual processing period time. Also, the use of ML in the protection systems denotes a new vector that can be exploited for designing advanced assaults [15].

This article presents an Enhanced Artificial Gorilla Troops Optimizer with Deep Learning Enabled Cybersecurity Threat Detection (EAGTODL-CTD) in IoT Cloud Networks. The presented EAGTODL-CTD model encompasses the identification of the threats in the IoT cloud environment. The proposed EAGTODL-CTD model mainly focuses on the conversion of input binary files to color images, where the malware can be detected using an image classification problem. The EAGTODL-CTD model pre-processes the input data to transform to a compatible format. For threat detection and classification, cascaded gated recurrent unit (CGRU) model is exploited to determine class labels. Finally, EAGTO algorithm is employed as a hyperparameter optimizer to tune the CGRU parameters. The experimental validation of the EAGTODL-CTD model is tested using a dataset comprising two class labels namely malignant and benign.

2 Related Works

In [16], the evaluation of the case of transforming information among the cloud and the end-user dew gadgets combined with the linked vehicles is performed. And analyses the application and organizational

techniques in relation to Dew Computing, in which the processing is closer to the user than other IoT computing patterns. This work intends in presenting an IoT threat analysis and employs a DL technique for countering cyber anomalies, afterward authenticating it by scrutinizing its metrics. An improved version of SAE can be used which enhances the accurateness of identifying the defined assaults, utilizing the loss over the training data as a threshold. In [17], the authors developed a method to meet the IoT cybersecurity menaces in a smart city, an Anomaly Detection-IoT (AD-IoT) system can be suggested, that was an intellectual anomaly detection related to RF and ML methods. The suggested solution could effectually identify compromised IoT gadgets at distributed fog nodes. Roopak et al. [18] suggested an IDS discovered on blending a Jumping Gene adapted NSGA-II multi-objective optimization technique for data dimension reduction and CNN compiling LSTM and DL approaches to classify the attack.

In [19], a sequential method is a key point, and novel techniques were suggested by the model features. The method could make a collection of features from the network layer through TCP dump packets and application layer through system routines. Li et al. [20] recommend a new federated DL approach, termed DeepFed, for identifying cyberattacks contrary to industrial CPSs. To be Specific, firstly devise a novel DL-related ID method for industrial CPSs, by using a gated recurrent unit and CNN. Secondly, advance a federated learning structure, enabling multiple industrial CPSs to jointly construct a full ID method in a privacy preserving way. In [21], the authors offered the complete enhancement of a novel intellectual and autonomous DL related detection and classifier mechanism for cyberattack in IoT networks which uses the power of CNN, abbreviated as IoT-IDCS-CNN (IoT related ID and Classification System using CNN). The suggested IoT-IDCS-CNN employs higher performance computing which uses the effective Compute Unified Device Architectures (CUDA) related parallel processing and Nvidia GPUs (Graphical Processing Units) which leverage high-speed I9-core-based Intel CPUs. In [22], a smart IDS suitable for detecting IoT-related assaults was implemented. Particularly, to identify malicious IoT network traffic, a DL method was utilized. An IDS is one such popular form of network security technology that can be utilized for network security.

3 The Proposed Model

In this article, an EAGTODL-CTD approach has been developed for threat detection in the IoT cloud environment. The suggested EAGTODL-CTD model concentrated on the transformation of the conversion of input binary files to color images, where the malware can be detected using an image classification problem. Primarily, the EAGTODL-CTD model pre-processes the input data to transform it into a compatible format. To detect and classify threats, the CGRU model is exploited to determine class labels. Finally, EAGTO algorithm is applied as a hyperparameter optimizer to tune the CGRU parameters. Fig. 1 depicts the block diagram of EAGTODL-CTD approach.

3.1 Data Pre-processing

The color image is produced from raw binary file to convert the malware detection issue through an image classification issue. It distinguishes the presented study from the latest techniques, where malware binary files transform into grayscale images with 256 colors. This technique doesn't reliant on reverse engineering tools like decompiling and disassembler. The color image retrieves more effective features than grayscale image with 256 colors. Furthermore, the effective feature of malware image outperforms in the classification of malware family. Previously, several malware detection techniques depending on ML algorithm provided good results through grayscale images. The color image is converted into grayscale visualization, and later feature extraction technique was utilized for classifying malware types. The classification accuracy can be enhanced by feature reduction method to reduce the feature set.

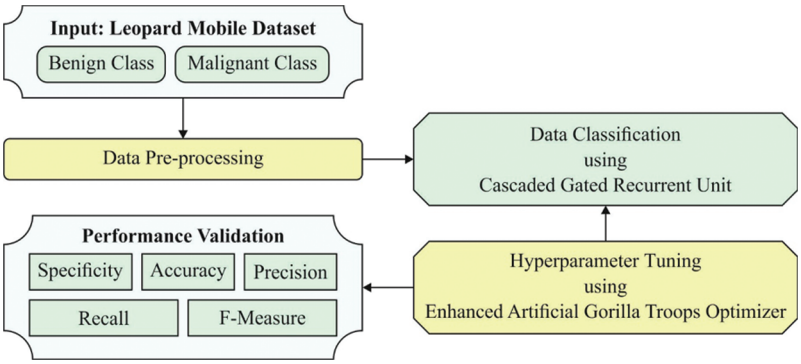

Figure 1: Block diagram of EAGTODL-CTD approach

The outcome shows that ML algorithm is not a preferable option for the detection of malware since it produces exponential value with color images. The DL algorithm outperforms big malware data as this type of approach makes use of the filter to automatically reduce noise. Therefore, the usage of color images generates good outcomes through DL technique. The transformation of malware binary files to color images encompasses four stages. Firstly, the hexadecimal strings (0–15) are generated from raw binary files. Subsequently, the eight bit vector is transformed into a 2D matrix space. At last, every eight bit integer produced from 2D space is designed with blue, red, and green shaded colors.

3.2 CGRU Based Threat Detection and Classification

To detect and classify threats, the CGRU model is exploited to determine class labels. LSTM is a different kind of RNN that is capable of learning long-term dependency. The LSTM is planned for avoiding long term dependency problem that is prevalent from RNN [23]. It is obtained great praise in the domain of ML and speech detection. Some NN contains dependencies problems, however, the LSTM has overcome the problem of dependencies by adjusting the data flow utilizing input, resultant, and forget gates. An input gate controlled the flow of input activation to memory cell. The resultant gate controls the outcome flow of cell activations as the rest of networks. Assume that trained data takes N equipment of similar creatures and kind that offer failure data, and all the equipment offer set multi-variate time sequence dataset in the sensor of equipment. Besides, let us r sensor of similar kinds on all the equipment. Next, the data gathered in all the equipment are demonstrated from the matrix procedure $X_n = [x_1,\ x_2,\ \ldots,\ x_t,\ \ldots,\ x_{T_n}] \in \mathbb{R}^{r\times T_n}(n = 1,\ \ldots,\ N)$ whereas T_n is time of failures and at time t the r-dimension vector of sensor measurements are $x_t = \left[s_t^1,\ \ldots,\ s_t^r\right] \in \mathbb{R}^{r\times 1}$, $t = 1,\ 2,\ \ldots,\ T_n$. The data of all the equipment's from X_n was provided to LSTM network and network learned that method the entire sequenced interms of target RUL.

At time t, LSTM network proceeds r-dimension sensor dataset x_t and provides forecast RUL_t. Consider the LSTM cell takes q nodes, next $c_t \in \mathbb{R}^{q\times 1}$ refers to the resultant of cell states, $h_t \in \mathbb{R}^{q\times 1}$ signifies the resultant of LSTM cell, $\mathrm{o_t}\times \in \mathbb{R}^{q\times 1}$ implies the resultant gate, $i_t \in \mathbb{R}^{q\times 1}$ denotes the input gate, and $f_t \in \mathbb{R}^{q\times 1}$ demonstrates the forget gate at time t. At $t-1$ time, the output h_{t-1}, and hidden state c_{t-1} are help as input to LSTM cell at t time. The input x_t is offered as input to cells. In LSTM, the normalizing data is computed utilizing the subsequent formulas:

$$i_t = \sigma(W_i \cdot [h_{t-1},\ x_t] + b_i), \tag{1}$$

$$f_t = \sigma\left(W_f \cdot [h_{t-1},\ x_t] + b_f\right), \tag{2}$$

$$o_t = \sigma(W_o \cdot [h_{t-1},\ x_t] + b_o), \tag{3}$$

$$\tilde{c}_t = act(W_c \cdot [h_{t-1},\ x_t] + b_c), \tag{4}$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t, \tag{5}$$

$$h_t = o_t * act(c_t), \tag{6}$$

whereas σ refers to the sigmoid layer. c_t and $\tilde{c}_t$ denotes all the internal memory cells and temporary value for making a novel internal memory cell at time t. This elementwise multiplication of two vectors. It functions by utilizing reset and update gates. Fig. 2 depicts the framework of GRU.

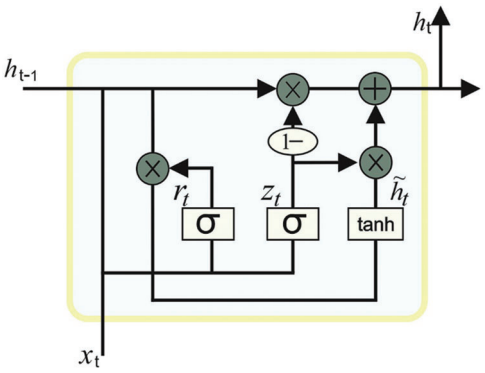

Figure 2: Structure of GRU

The GRU is an enhanced version of typical RNNs. Related to LSTM unit, the GRU takes gating units that control the data flows, but, without containing a distinct memory cell. The performance of GRU is based on specific tasks of polyphonic music and speech signal modeling was found to be same as LSTM. GRU has been found to display good performance on specific small datasets. The memory block of GRU is simple when compared to LSTM. Output, forget and input gates are replaced by the reset and update gates. As well, GRU integrates the internal memory cell and the hidden state. In GRU, the normalized dataset is evaluated by the subsequent formula:

$$z_t = \sigma(W_z \cdot [h_{t-1},\ x_t] + b_z), \tag{7}$$

$$r_t = \sigma(W_r \cdot [h_{t-1},\ x_t] + b_r), \tag{8}$$

$$\tilde{h}_t = act(W \cdot [r_t * h_{t-1},\ x_t] + b_h), \tag{9}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t, \tag{10}$$

From the equation, z_t and r_t refers to the update and reset gates at t time, correspondingly. h_t denotes a temporary value for making novel hidden state at t time. The CGRU model is developed by stacking a set of GRU models to enhance the classification performance.

3.3 Hyperparameter Tuning Using EAGTO Algorithm

In the final stage, the EAGTO algorithm is employed as a hyperparameter optimizer to tune the CGRU parameters. The AGTO approach contains the exploration and exploitation phases, in Eqs. (1)–(13) explain the basic concept of technique [24]. The exploration phase was mostly utilized for performing a global search of spaces. It utilizes 3 distinct processes contains migrate to unknown location, migrate to known location, and move to place of other gorillas. Eq. (11) inspires the exploitation phase.

$$GX(t+1) = \begin{cases} (ub - lb) \times r_1 + lb, & r < p, \\ (r_2 - C) \times X_r(t) + L \times H, & r \geq 0.5, \\ X(i) - L \times (L \times (X(t) - GX_r(t)) + r_3 \times (X(t) - GX_r(t))), & r < 0.5. \end{cases} \tag{11}$$

In Eq. (11), $X(t)$ signifies the gorilla's present place and $GX(t+1)$ refers to the gorilla's place from the $t+1$ iteration. p signifies the parameter amongst zero and one, which defines the migration process to select. lb and ub denote the lower and upper limits, X_r denotes the arbitrarily chosen gorilla member in the population and GX_r indicates the arbitrarily chosen gorilla candidate place vector. r_1, r_2, r_3, and r implies the random values from the range zero to one upgraded on all the iterations. In addition, L and H are computed in the subsequent formulas.

$$C = F \times \left(1 - \frac{It}{\text{Max } It}\right), \tag{12}$$

$$F = cos(2 \times r_4) + 1, \tag{13}$$

$$L = C \times l, \tag{14}$$

$$H = Z \times X(t), \tag{15}$$

$$Z = [-C,\ C]. \tag{16}$$

In Eq. (12), It denotes the present amount of iterations and $MaxIt$ represents the entire amount of iterations of the technique. In Eqs. (13) and (14), r_4 and l signifies the random values amongst zero to one upgraded on all the iterations. In Eq. (16), Z signifies the arbitrary value from the range $-C$ to C. After the exploration phase, this technique computed the fitness value of every GX solution, and when the fitness values are $X(t) < X(t)$, the $X(t)$ solution was exchanged by $GX(t)$ solution.

The exploitation mechanism of AGTO technique utilized 2 procedures, after silverback gorillas and competing with adult female gorillas. The process was chosen by relating the C value computed by Eq. (12) with the $\mathcal{W}$ parameter set from advance. When $C \geq W$, the AGTO technique utilizes the subsequent silverback gorilla process, however, if $C < W$, competing with adult female gorillas were chosen. Eq. (17) has been utilized for simulating Follow the silverback gorilla.

$$GX(t+1) = L \times M \times (X(t) - X_{silverback}) + X(t), \tag{17}$$

$$M = (|\frac{1}{N}\sum_{i=1}^{N} GX_i(t)|^g)^{\frac{1}{g}}, \tag{18}$$

$$g = 2^L. \tag{19}$$

In Eq. (17), $X_{silverback}$ denotes the silverback gorilla place. In Eq. (18), $GX_i(t)$ refers to the place of all the candidate gorillas from the iteration t and N indicates the entire amount of gorillas. Besides, Eq. (20) was utilized for simulating competition with adult female gorillas.

$$GX(i) = X_{silverback} - (X_{silverback} \times Q - X(t) \times Q) \times A, \tag{20}$$

$$Q = 2 \times r_5 - 1, \tag{21}$$

$$A = \beta \times E, \tag{22}$$

$$E = \begin{cases} N_1, & r \geq 0.5, \\ N_2, & r < 0.5. \end{cases} \tag{23}$$

In Eq. (21), r_5 refers to the arbitrary value between zero to one upgraded on every iteration. In Eq. (22), β denotes the parameter that provided value. In Eq. (23), if $rand \geq 0.5$, E denotes the arbitrary values from the normal distribution and dimensional of problems, however, if $rand < 0.5$, E denotes the arbitrary value selected in normal distributions. After the exploitation phase, this technique computed the fitness values of every GX solution. When the fitness value is $X(t) < X(t)$, the $X(t)$ solution was exchanged by $GX(t)$ solutions, and optimum solutions chosen from the total populations are considered as the silverback gorillas.

To enhance the efficacy of the AGTO algorithm, the EAGTO algorithm has been integrated into the Pinhole Imaging Opposition-Based Learning (PIOBL). For assisting the algorithm from getting trapped into local optima, few research workers have tried to integrate opposition-based learning (OBL) with intelligent optimization algorithms to extend the searching range by evaluating the reverse solution of the present likely solution, and as a result, discover the solution candidate at optimum position. As per the concept, and effectively applied the pinhole imaging opposition-based learning to improve the convergence speed and accuracy of the AGTO algorithm. An approach has been employed to improve the possibility of EAGTO from getting trapped into local optimal. Here, the upper and lower limits of the coordinate axes are, b. There is a smaller aperture screen located at the base point O. X_{best} (the present global optimum solution) signifies the projection of light source P that height is h on the *x-axis*, once the light source through the smaller aperture will get an inverted image p^* of height h^* at the imaging screen, where time the projection of p' on the χ-axis is X^*_{best} (the recently produced inverse solution). Based on the geometric relationships of the line segment in the figure, it is given by:

$$\frac{(a+b)/2 - X_{best}}{X^*_{best} - (a+b)/2} = \frac{h}{h^*} \tag{24}$$

Consider $h/h^* = K$ be substituted with the abovementioned formula, and the variation produces the equation for X^*_{besi}:

$$X^*_{best} = \frac{(a+b)}{2} + \frac{(a+b)}{2K} - \frac{X_{best}}{K} \tag{25}$$

While the algorithm is resolving a higher dimensional complex function, the smaller aperture inverse learning solution is calculated as follows:

$$X^*_{best,j} = \frac{a_j + b_j}{2} + \frac{a_j + b_j}{2K} - \frac{X_{best,j}}{K} \tag{26}$$

In Eq. (26), $X_{best,j}$ denotes the optimum solution in the *j-th* variable, $X^*_{best,j}$ denotes the inverse solution of $X_{best,j}$, and a_j and b_j represents the minimal and maximal values in the *j-th* parameter. If $K = 1$, Eq. (27) is expressed as follows:

$$X^*_{best} = a + b - X_{besi} \tag{27}$$

It is apparent that if $K = 1$ the PIOBL is the common OBL strategy. The candidate solution attained by the common OBL strategy is usually fixed however a wide range of inversion positions is attained by interchanging the distance between the imaging screen and the pinhole plate for adjusting the scaling factor K in the PIOBL approach. The EAGTO method derives a fitness function to accomplish better classification accuracy. It describes a positive integer to characterize the effective performance of the solution candidate. Here, the reduction of classification error rate can be regarded as the fitness function. The optimum solution has a lesser error rate and the worse solution accomplishes a higher error rate.

$$fitness(x_i) = Classifier\ Error\ Rate(x_i)$$
$$= \frac{number\ of\ misclassified\ samples}{Total\ number\ of\ samples} * 100 \quad (28)$$

4 Performance Validation

The proposed model is simulated using Python 3.6.5 tool. The proposed model is experimented on PC i5-8600k, GeForce 1050Ti 4 GB, 16 GB RAM, 250 GB SSD, and 1 TB HDD. This section inspects the threat classification performance of EAGTODL-CTD method using a Leopard mobile dataset. For experimental validation, we have taken a set of 14733 samples under malware class and 2486 samples under benign class as shown in Table 1.

Table 1: Dataset details

Class	No. of images
Malware	14733
Benign	2486
Total number of images	17219

Fig. 3 demonstrates the confusion matrices generated by the EAGTODL-CTD method under dissimilar epochs. The figure implied EAGTODL-CTD model has recognized samples effectually under both classes. For example, with 200 epochs, the EAGTODL-CTD model has identified 14640 samples under malw are class and 2471 samples under benign class. In addition, with 400 epochs, the EAGTODL-CTD method has recognized 14682 samples under malware class and 1630 samples under benign class. Moreover, with 600 epochs, the EAGTODL-CTD model has recognized 14646 samples under malware class and 2321 samples under benign class. Followed, with 800 epochs, the EAGTODL-CTD mechanism has identified 14645 samples under malware class and 2474 samples under benign class. Eventually, with 1000 epochs, the EAGTODL-CTD method has recognized 14654 samples under malware class and 2475 samples under benign class.

Table 2 and Fig. 4 report a detailed threat detection performance of the EAGTODL-CTD model under dissimilar epochs. The experimental values reported that the EAGTODL-CTD model has resulted in better outcomes under every epoch. For instance, on 200 epochs, the EAGTODL-CTD model has obtained an average $accu_y$ of 99.37%, $prec_n$ of 98.14%, $reca_l$ of 99.38%, $spec_y$ of 99.38%, and $F_{measure}$ of 98.75%. Along with that, on 400 epochs, the EAGTODL-CTD method has gained an average $accu_y$ of 94.73%, $prec_n$ of 95.73%, $reca_l$ of 82.61%, $spec_y$ of 82.61%, and $F_{measure}$ of 87.62%. Meanwhile, on 600 epochs, the EAGTODL-CTD mechanism has attained an average $accu_y$ of 98.54%, $prec_n$ of 97.64%, $reca_l$ of 96.39%, $spec_y$ of 96.39%, and $F_{measure}$ of 97%. Eventually, on 800 epochs, the EAGTODL-CTD technique has attained an average $accu_y$ of 99.42%, $prec_n$ of 98.24%, $reca_l$ of 99.46%, $spec_y$ of 99.46%, and $F_{measure}$ of 98.84%. Finally, on 1000 epochs, the EAGTODL-CTD approach has achieved an average $accu_y$ of 99.48%, $prec_n$ of 98.42%, $reca_l$ of 99.51%, $spec_y$ of 99.51%, and $F_{measure}$ of 98.95%.

The training accuracy (TA) and validation accuracy (VA) accomplished using the EAGTODL-CTD method on testing data are depicted in Fig. 5. The experimental result indicates that the EAGTODL-CTD approach has accomplished maximal values of TA and VA. Particularly, the VA seemed to be high than TA.

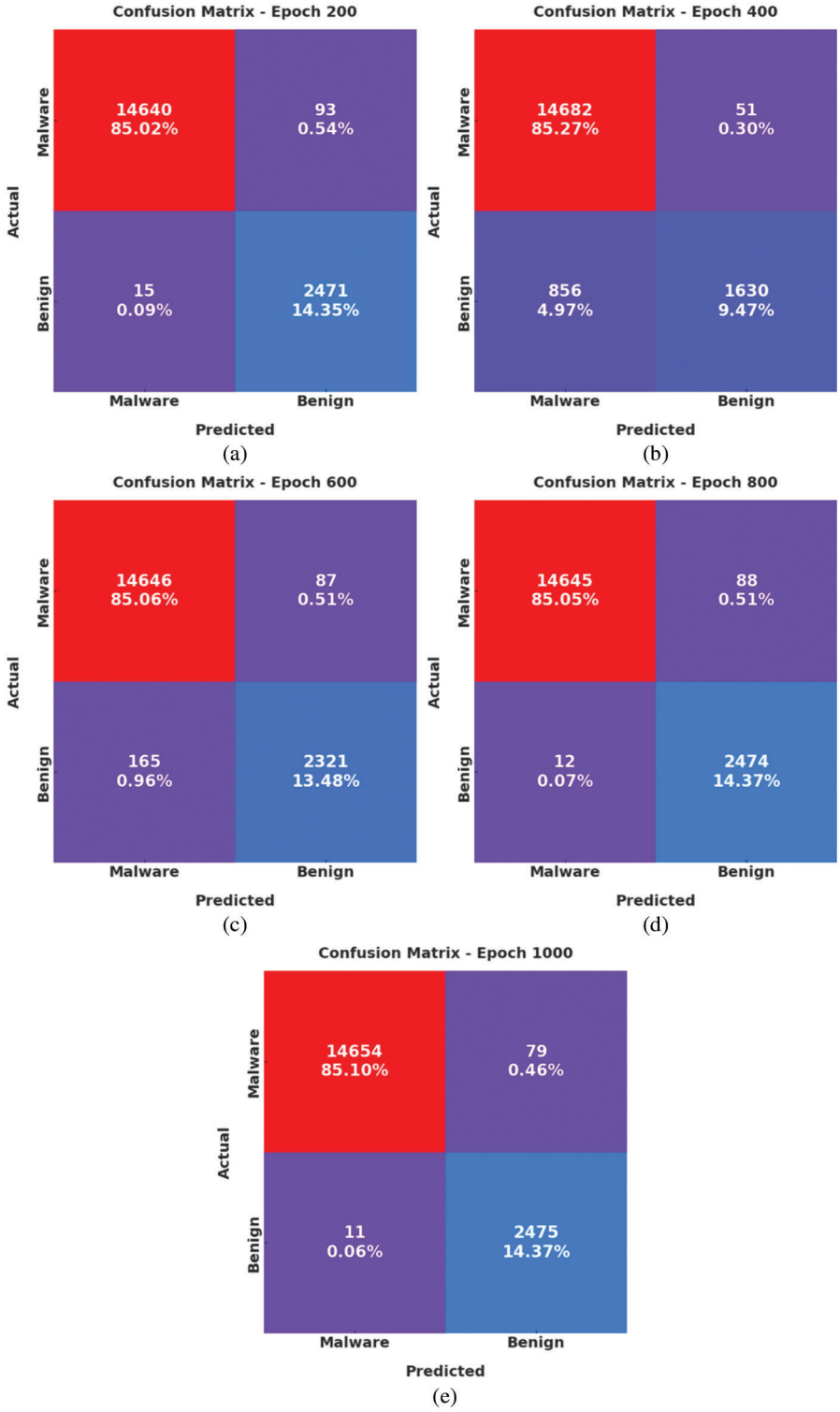

Figure 3: Confusion matrices of EAGTODL-CTD approach (a) Epoch 200, (b) Epoch 400, (c) Epoch 600, (d) Epoch 800, and (e) Epoch 1000

Table 2: Result analysis of EAGTODL-CTD approach with measures and epochs

Labels	Accuracy	Precision	Recall	Specificity	F-measure
Epoch-200					
Malware	99.37	99.90	99.37	99.40	99.63
Benign	99.37	96.37	99.40	99.37	97.86
Average	**99.37**	**98.14**	**99.38**	**99.38**	**98.75**
Epoch-400					
Malware	94.73	94.49	99.65	65.57	97.00
Benign	94.73	96.97	65.57	99.65	78.23
Average	**94.73**	**95.73**	**82.61**	**82.61**	**87.62**
Epoch-600					
Malware	98.54	98.89	99.41	93.36	99.15
Benign	98.54	96.39	93.36	99.41	94.85
Average	**98.54**	**97.64**	**96.39**	**96.39**	**97.00**
Epoch-800					
Malware	99.42	99.92	99.40	99.52	99.66
Benign	99.42	96.57	99.52	99.40	98.02
Average	**99.42**	**98.24**	**99.46**	**99.46**	**98.84**
Epoch-1000					
Malware	99.48	99.92	99.46	99.56	99.69
Benign	99.48	96.91	99.56	99.46	98.21
Average	**99.48**	**98.42**	**99.51**	**99.51**	**98.95**

The training loss (TL) and validation loss (VL) accomplished using the EAGTODL-CTD technique on testing data are illustrated in Fig. 6. The experimental result shows that the EAGTODL-CTD method has obtained minimum values of TL and VL. Especially, the VL is lower than TL.

A clear precision-recall analysis of the EAGTODL-CTD methodology on testing data is described in Fig. 7. The figure showed that the EAGTODL-CTD system has resulted in enhanced values of precision-recall values under all classes.

A brief ROC examination of the EAGTODL-CTD technique on testing data is portrayed in Fig. 8. The results indicated the EAGTODL-CTD methodology has shown its ability in classifying dissimilar classes on the testing dataset.

To illustrate the effective performance of the EAGTODL-CTD model, a wide-ranging comparative assessment is made in Table 3 [1]. Fig. 9 exhibits the comparative classification accuracy of the EAGTODL-CTD technique with recent models. The figure implied that the LBP-SVM method has reached ineffectual performance with lower $accu_y$ of 77.93%. Simultaneously, the GIST-SVM approach has exhibited somewhat improved outcomes with $accu_y$ of 86.30%. Moreover, the GCLM-SVM model has tried to depict moderate performance with $accu_y$ of 92.27%. Next to that, the DL model has gained reasonable performance with $accu_y$ of 97.15%. However, the EAGTODL-CTD model has surpassed the other models with higher $accu_y$ of 99.48%.

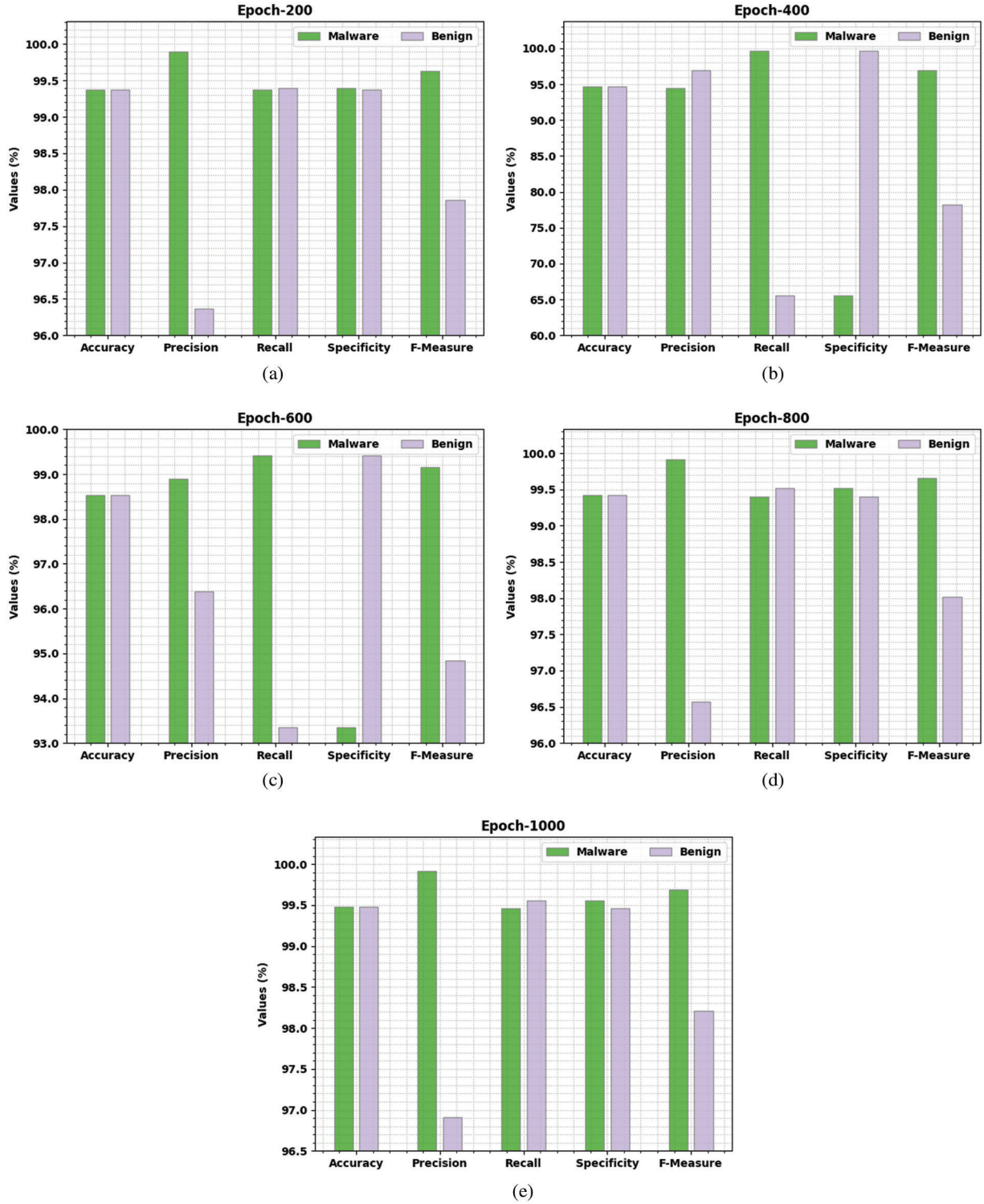

Figure 4: Result analysis of EAGTODL-CTD approach (a) Epoch 200, (b) Epoch 400, (c) Epoch 600, (d) Epoch 800, and (e) Epoch 1000

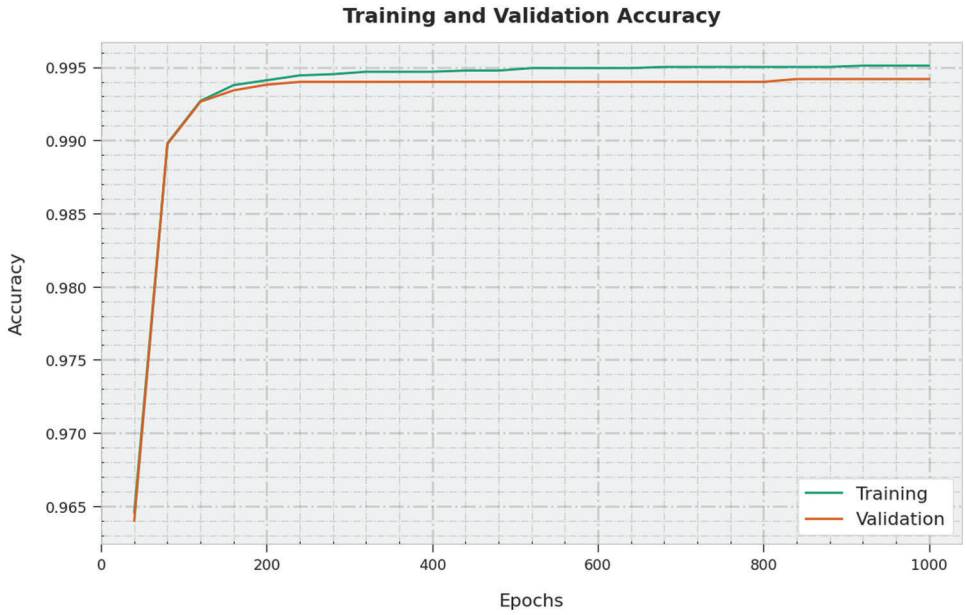

Figure 5: TA and VA analysis of EAGTODL-CTD approach

Figure 6: TL and VL analysis of EAGTODL-CTD approach

Fig. 10 shows the comparative $F_{measure}$ of the EAGTODL-CTD method with current techniques. The results denote that the LBP-SVM methodology has obtained ineffectual outcomes with lower $F_{measure}$ of 77.70%. Simultaneously, the GIST-SVM method has displayed somewhat enhanced results with $F_{measure}$ of 85.64%. Furthermore, the GCLM-SVM approach has tried to portray reasonable performance with $F_{measure}$ of 91.98%. Following, the DL method has accomplished moderate performance with $F_{measure}$ of 97.29%. But, the EAGTODL-CTD approach has surpassed the other methods with greater $F_{measure}$ of 98.95%.

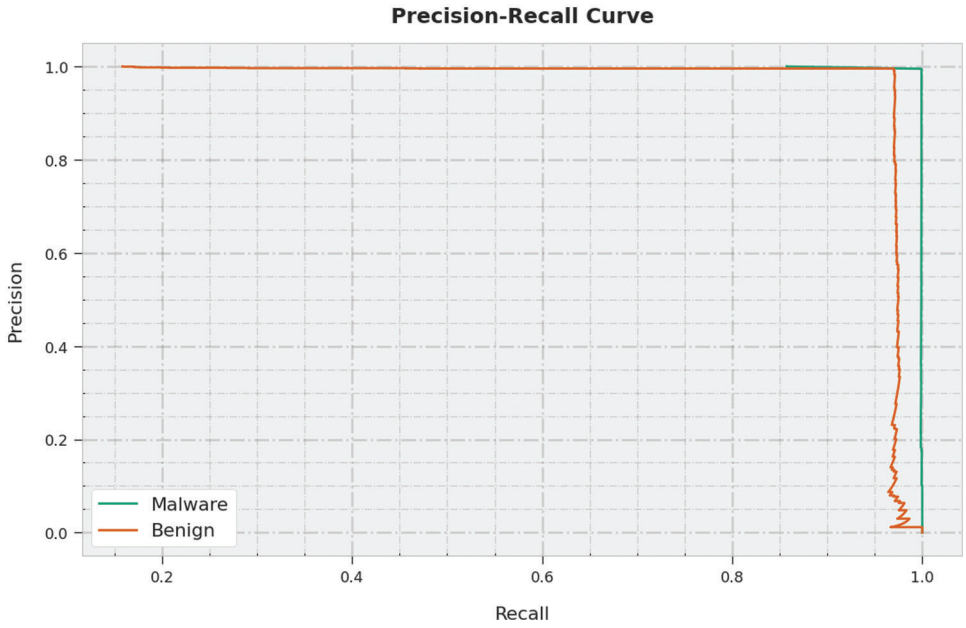

Figure 7: Precision-recall curve analysis of EAGTODL-CTD approach

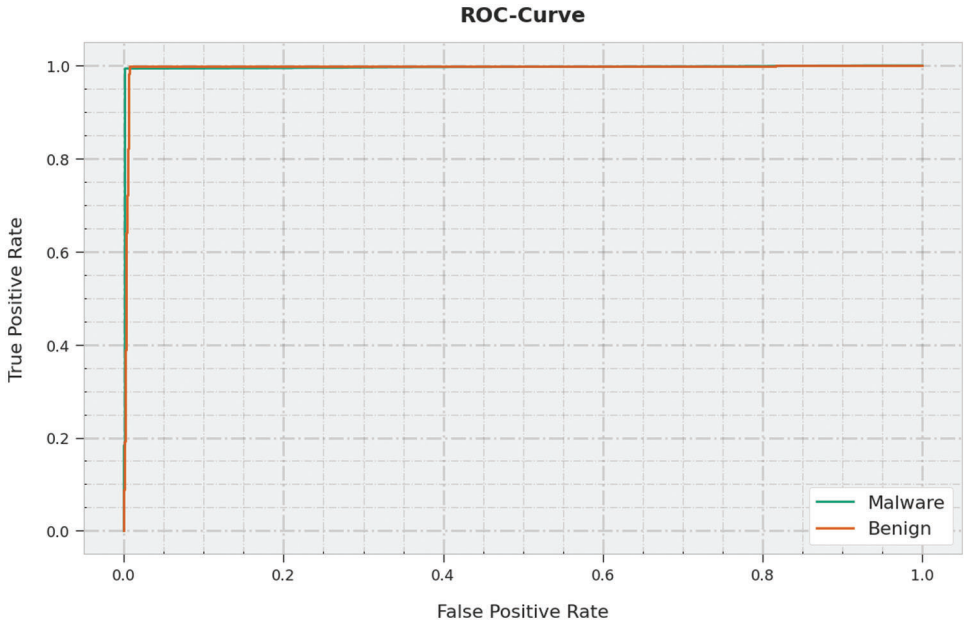

Figure 8: ROC curve analysis of EAGTODL-CTD approach

From the abovementioned results, it is clear that the EAGTODL-CTD model has resulted in enhanced results over other models.

Table 3: Comparison study of EAGTODL-CTD approach with current methodologies

Techniques	Classification accuracy (%)	F measure (%)
GIST-SVM model	86.30	85.64
LBP-SVM model	77.93	77.70
GCLM-SVM model	92.27	91.98
DL model	97.15	97.29
EAGTODL-CTD	99.48	98.95

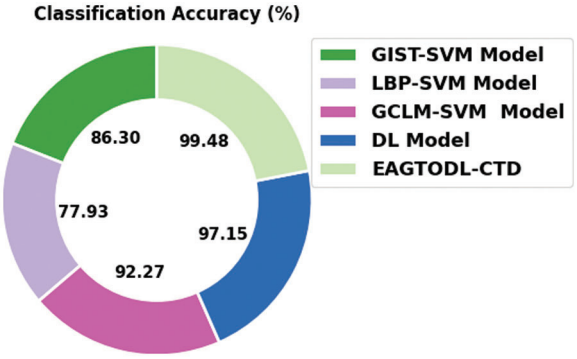

Figure 9: $Accu_y$ analysis of EAGTODL-CTD with current methodologies

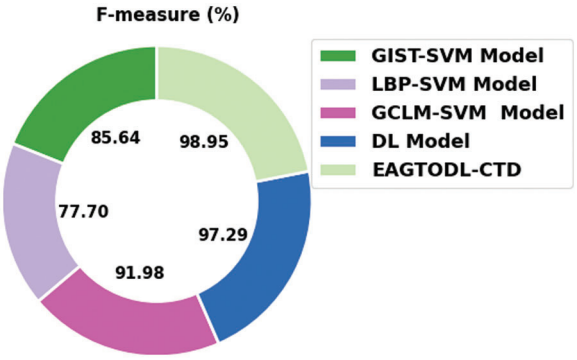

Figure 10: $F_{measure}$ analysis of EAGTODL-CTD approach with recent methodologies

5 Conclusion

In this article, an EAGTODL-CTD methodology has been developed for threat detection in the IoT cloud environment. The projected EAGTODL-CTD model concentrated on the transformation of the conversion of input binary files to color images, where the malware can be detected using an image classification problem. Primarily, the EAGTODL-CTD model pre-processes the input data to transform it into a compatible format. To detect and classify threats, the CGRU model is exploited to determine class labels. Finally, EAGTO technique is applied as a hyperparameter optimizer to tune the CGRU parameters. The performance evaluation of the EAGTODL-CTD model is assessed on a dataset comprising two class labels namely malignant and benign. The experimental values reported the supremacy of the EAGTODL-CTD model with increased accuracy of 99.47%. In the future, hybrid DL models can be exploited to improvise the detection efficiency of the presented EAGTODL-CTD model.

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: 22UQU4340237DSR41.

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

References

[1] F. Ullah, H. Naeem, S. Jabbar, S. Khalid, M. A. Latif *et al.,* "Cyber security threats detection in internet of things using deep learning approach," *IEEE Access*, vol. 7, pp. 124379–124389, 2019.

[2] M. Roopak, G. Yun Tian and J. Chambers, "Deep learning models for cyber security in iot networks," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conf. (CCWC)*, Las Vegas, NV, USA, pp. 0452–0457, 2019.

[3] J. Asharf, N. Moustafa, H. Khurshid, E. Debie, W. Haider *et al.,* "A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions," *Electronics*, vol. 9, no. 7, pp. 1177, 2020.

[4] A. A. Albraikan, S. B. H. Hassine, S. M. Fati, F. N. Al-Wesabi, A. Mustafa Hilal *et al.,* "Optimal deep learning-based cyberattack detection and classification technique on social networks," *Computers, Materials & Continua*, vol. 72, no. 1, pp. 907–923, 2022.

[5] K. A. P. da Costa, J. P. Papa, C. O. Lisboa, R. Munoz and V. H. C. de Albuquerque, "Internet of things: A survey on machine learning-based intrusion detection approaches," *Computer Networks*, vol. 151, pp. 147–157, 2019.

[6] A. Al-Qarafi, F. Alrowais, S. Alotaibi, N. Nemri, F. N. Al-Wesabi *et al.,* "Optimal machine learning based privacy preserving blockchain assisted internet of things with smart cities environment," *Applied Sciences*, vol. 12, no. 12, pp. 1–17, 2022.

[7] J. C. S. Sicato, S. K. Singh, S. Rathore and J. H. Park, "A comprehensive analyses of intrusion detection system for IoT environment," *Journal of Information Processing Systems*, vol. 16, no. 4, pp. 975–990, 2020.

[8] 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.

[9] 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.

[10] 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.

[11] M. A. Ferrag, L. Shu, O. Friha and X. Yang, "Cyber security intrusion detection for agriculture 4.0: Machine learning-based solutions, datasets, and future directions," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 3, pp. 407–436, 2022.

[12] K. V. V. N. L. S. Kiran, R. N. K. Devisetty, N. P. Kalyan, K. Mukundini and R. Karthi, "Building a intrusion detection system for IoT environment using machine learning techniques," *Procedia Computer Science*, vol. 171, pp. 2372–2379, 2020.

[13] A. Fatani, A. Dahou, M. A. A. Al-qaness, S. Lu and M. A. Abd Elaziz, "Advanced feature extraction and selection approach using deep learning and aquila optimizer for IoT intrusion detection system," *Sensors*, vol. 22, no. 1, pp. 140, 2021.

[14] S. Tsimenidis, T. Lagkas and K. Rantos, "Deep learning in IoT intrusion detection," *Journal of Network and Systems Management*, vol. 30, no. 1, pp. 8, 2022.

[15] V. Morfino and S. Rampone, "Towards near-real-time intrusion detection for IoT devices using supervised learning and apache spark," *Electronics*, vol. 9, no. 3, pp. 444, 2020.

[16] M. M. Moussa and L. Alazzawi, "Cyber attacks detection based on deep learning for cloud-dew computing in automotive IoT applications," in *2020 IEEE Int. Conf. on Smart Cloud (SmartCloud)*, Washington DC, WA, USA, pp. 55–61, 2020.

[17] I. Alrashdi, A. Alqazzaz, E. Aloufi, R. Alharthi, M. Zohdy *et al.,* "AD-IoT: Anomaly detection of IoT cyberattacks in smart city using machine learning," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conf. (CCWC)*, Las Vegas, NV, USA, pp. 0305–0310, 2019.

[18] M. Roopak, G. Y. Tian and J. Chambers, "An intrusion detection system against ddos attacks in IoT networks," in *2020 10th Annual Computing and Communication Workshop and Conf. (CCWC)*, Las Vegas, NV, USA, pp. 0562–0567, 2020.

[19] M. Zhong, Y. Zhou and G. Chen, "Sequential model based intrusion detection system for IoT servers using deep learning methods," *Sensors*, vol. 21, no. 4, pp. 1113, 2021.

[20] B. Li, Y. Wu, J. Song, R. Lu, T. Li *et al.,* "DeepFed: Federated deep learning for intrusion detection in industrial cyber–physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5615–5624, 2021.

[21] Q. A. A. Haija and S. Zein-Sabatto, "An efficient deep-learning-based detection and classification system for cyber-attacks in IoT communication networks," *Electronics*, vol. 9, no. 12, pp. 2152, 2020.

[22] N. Yadav, S. Pande, A. Khamparia and D. Gupta, "Intrusion detection system on IoT with 5G network using deep learning," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–13, 2022.

[23] N. Gruber and A. Jockisch, "Are GRU cells more specific and LSTM cells more sensitive in motive classification of text?," *Frontiers in Artificial Intelligence*, vol. 3, pp. 40, 2020.

[24] B. Abdollahzadeh, F. S. Gharehchopogh and S. Mirjalili, "Artificial gorilla troops optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems," *International Journal of Intelligent Systems*, vol. 36, no. 10, pp. 5887–5958, 2021.