applied sciences

MDPI

Article

Optimal Graph Convolutional Neural Network-Based Ransomware Detection for Cybersecurity in IoT Environment

Hend Khalid Alkahtani [1], Khalid Mahmood [2], Majdi Khalid [3], Mahmoud Othman [4], Mesfer Al Duhayyim [5,*], Azza Elneil Osman [6], Amani A. Alneil [6] and Abu Sarwar Zamani [6]

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 Information Systems, College of Science & Art at Mahayil, King Khalid University, Abha 62529, Saudi Arabia
3 Department of Computer Science, College of Computing and Information System, Umm Al-Qura University, Makkah 24382, Saudi Arabia
4 Department of Computer Science, Faculty of Computers and Information Technology, Future University in Egypt, New Cairo 11835, Egypt
5 Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Al-Kharj 16273, Saudi Arabia
6 Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, Al-Kharj 16278, Saudi Arabia
* Correspondence: m.alduhayyim@psau.edu.sa

Abstract: The fast development of the Internet of Things (IoT) and widespread utilization in a large number of areas, such as vehicle IoT, industrial control, healthcare, and smart homes, has made IoT security increasingly prominent. Ransomware is a type of malware which encrypts the victim's records and demands a ransom payment for restoring access. The effective detection of ransomware attacks highly depends on how its traits are discovered and how precisely its activities are understood. In this article, we propose an Optimal Graph Convolutional Neural Network based Ransomware Detection (OGCNN-RWD) technique for cybersecurity in an IoT environment. The OGCNN-RWD technique involves learning enthusiasm for teaching learning-based optimization (LETLBO) algorithms for the feature subset selection process. For ransomware classification, the GCNN model is used in this study, and its hyperparameters can be optimally chosen by the harmony search algorithm (HSA). For exhibiting the greater performance of the OGCNN-RWD approach, a series of simulations were made on the ransomware database. The simulation result portrays the betterment of the OGCNN-RWD system over other existing techniques with an accuracy of 99.64%.

Keywords: cybersecurity; ransomware; Internet of Things; feature selection; deep learning

Citation: Khalid Alkahtani, H.; Mahmood, K.; Khalid, M.; Othman, M.; Al Duhayyim, M.; Osman, A.E.; Alneil, A.A.; Zamani, A.S. Optimal Graph Convolutional Neural Network-Based Ransomware Detection for Cybersecurity in IoT Environment. *Appl. Sci.* **2023**, *13*, 5167. https://doi.org/10.3390/app13085167

Academic Editor: Gregory Epiphaniou

Received: 1 February 2023
Revised: 25 February 2023
Accepted: 1 March 2023
Published: 21 April 2023

1. Introduction

Recently, the use of interconnected smart devices commonly called the Internet of Things (IoT) has seen exponential growth [1]. IoT gadgets can be accessed from any place, home, vehicle and office to make daily tasks as simple as they can. Such smart devices are utilised in smart cities, healthcare services, vehicular networks, industries, smart grids, and smart homes [2]. These smart gadgets have unique features such as minimal power consumption and lighter protocols, weight and compact size which make them more adjustable. Expanded dispatch of smart gadgets in advertisements has decreased trust regarding detecting gadgets and has made the web of things increasingly versatile [3]. With the two downsides and upsides, the devices linked to the Internet are at risk of attacks and digital threats, prompting failure of the administration to more dreadful conveyed refusal of administration [4]. There are no confirmed security techniques that ensure the digital safety of such gadgets. IoT infrastructure is prone to terrible security threats and various attacks, because it lacks built-in security mechanisms and standard supporting systems [5].

IoT has become a capitated platform for invaders since it has the potential to launch all types of network attacks on the connected devices, which in most cases, result in some serious losses.

Ransomware can be referred to as a malware type that can be devised to block access to the user files, device, or operating system [6]. Figure 1 represents the process of ransomware detection for cybersecurity in IoT platforms. Ransomware is commonly found in the form of crypto-ransomware and locker ransomware. Crypto-ransomware encodes key documents on a system of the user, utilizing complicated encryption methods and demand payments, generally cryptocurrency for decoding the credentials of victims [7]. Locker ransomware displays a lock screen that stops the victim from opening their system and demands money for access to a computer. Ransomware is more destructive, prominent, and advanced [8]. In comparison to static code analysis methods, machine learning (ML) approaches have proven to be effectual. ML has high potential in finding malware in Android and Windows OS systems [9]. Further studies on ML in malware recognition as a substitute for the use of signs has shown efficiency regarding the use of ML-related detection over signature-related techniques [10]. The decision to assess ML and DL methods as opposed to other non-ML-related methods has been considered due to their strong ability and adaptability to find unseen samples of ransomware malware.

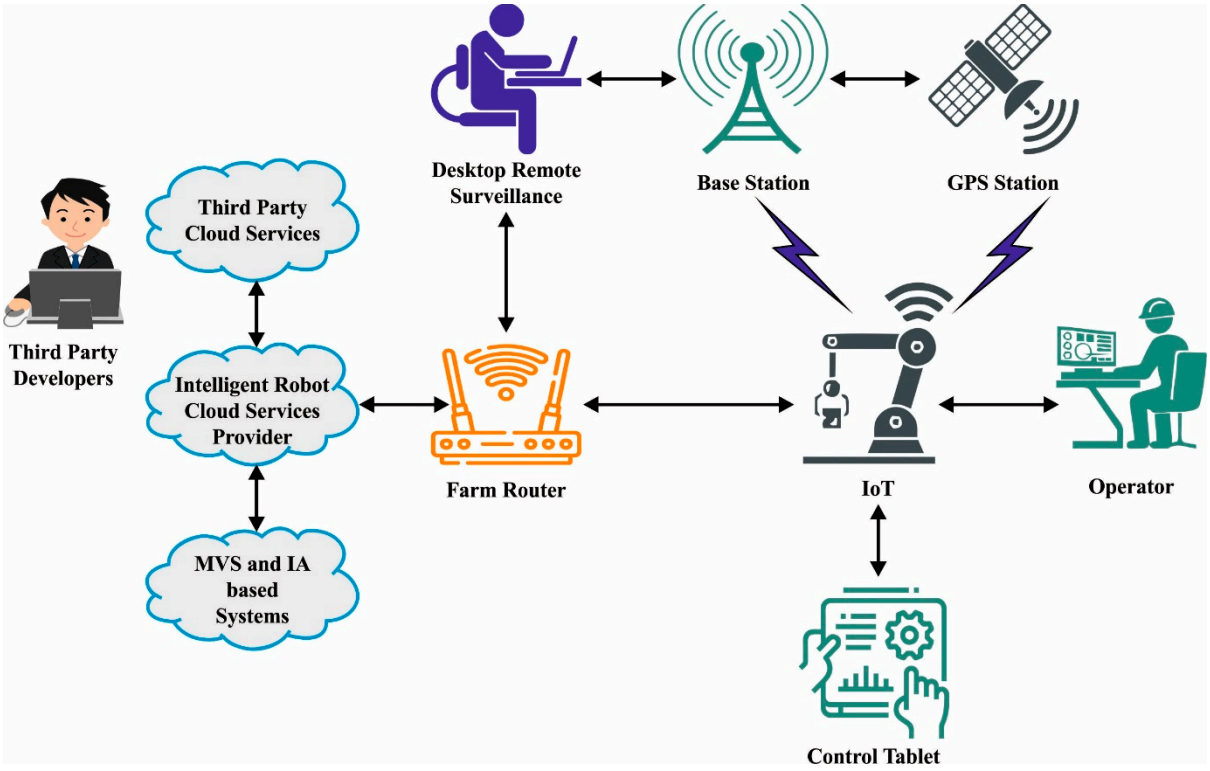

Figure 1. Ransomware detection for cybersecurity in IoT platform.

This article focuses on the development of an Optimal Graph Convolutional Neural Network based Ransomware Detection (OGCNN-RWD) technique for cybersecurity in an IoT environment. Primarily, the OGCNN-RWD technique involves learning enthusiasm for teaching learning-based optimization (LETLBO) algorithms for the feature selection procedure. Next, the GCNN model is used for ransomware classification, and its hyperparameters can be optimally chosen by the harmony search algorithm (HSA). To demonstrate the better results of the OGCNN-RWD system, a series of simulations were made on the ransomware database.

The rest of the paper is organized as follows. Section 2 provides the related works, and Section 3 offers the proposed model. Then, Section 4 gives the results analysis, and Section 5 concludes the paper.

2. Related Works

One author used deep learning (DL) approaches for the extraction of the latent representation of high dimensions of the gathered dataset for precisely finding malevolent behavior [11]. Specifically, this method rests on a hybridized feature engineering approach of a variational and traditional autoencoder (AE). This approach was utilised to minimalize the dimension of data and to extract a good representation of accumulated system activities. Afterwards, the novel feature vector was sent to classifiers that can be framed on batch normalization and deep neural network (DNN) methods. The authors in [12] presented a detection system relevant to the stacked variational AE (VAE) with a fully connected neural network (FC-NN) that learns the latent framework of system activities and exposes the ransomware performance. In addition, the author came up with a data augmentation approach that depends on VAE to produce novel datasets that can be utilised in training an FC network to enhance the generalized capabilities of the presented recognition model.

Al-Hawawreh et al. [13] modelled a new aimed ransomware detection method devised for the industrial IoT edge mechanism. It leverages DL and Asynchronous Peer-to-Peer Federated Learning (AP2PFL) methods as targeted ransomware recognition methods. The presented technique contains two modules. The Diagnostic and one Decision Module (DDM) was utilized for finding targeted ransomware and its phases depend on DNN and Batch Normalization (BN). Basnet et al. [14] introduced the DL-related new ransomware detection structure in supervisory control and data acquisition-controlled electric vehicle charging station (EVCS) with the performance analysis of three DL methods.

Alrawashdeh and Purdy [15] devised a fast ransomware identification approach utilizing Memory-based Stochastic-Dynamic-Fixed-Point arithmetic utilizing a four-layer deep belief network (DBN) architecture. The technique stored random bit-streams in storage for producing potential cross-correlation for stochastic computation in Field Programmable Gate Arrays (FPGAs). Mathane and Lakshmi [16] presented a context-aware ransomware predictive approach that leverages context ontology to derive data features (software updates, connection requests, etc.) and ML techniques to predict ransomware. The presented approaches rely on and focus on the initial detection and prediction of ransomware penetration attempts to resource-limited IoT mechanisms. A weighted minimum Redundancy maximum Relevance (WmRmR) algorithm has been modelled for superior feature impact prediction in datasets captured at the primary stages of a ransomware attack [17]. This presented approach can assess if the feature in the appropriate set was significant or not. It includes a smaller number of evaluations and low-dimensional complexity than the original mRmR approach.

Several models exist in the literature that perform the ransomware classification process. Although several ML and DL models for ransomware classification are available in the literature, a model is still needed that can enhance the classification performance. Owing to the continual deepening of the model, the number of parameters of DL models also increases quickly, which results in model overfitting. At the same time, different hyperparameters have a significant impact on the efficiency of the CNN model. Particularly, hyperparameters such as epoch count, batch size, and learning rate selection are essential to attain effectual outcomes. Since the trial and error method for hyperparameter tuning is a tedious and erroneous process, metaheuristic algorithms can be applied. Therefore, the HSA algorithm can be applied to the parameter selection of the GCNN model.

3. The Proposed Model

In this study, a novel OGCNN-RWD system has been developed for cybersecurity in the IoT platform. The OGCNN-RWD technique mainly intends to precisely distinguish ransomware from legitimate activities. In the presented OGCNN-RWD approach, the

LETLBO system is applied for the feature subset selection process. To classify ransomware, the GCNN model is used in this study, and its hyperparameters can be optimally chosen by the HSA. Figure 2 illustrates the workflow of the OGCNN-RWD system.

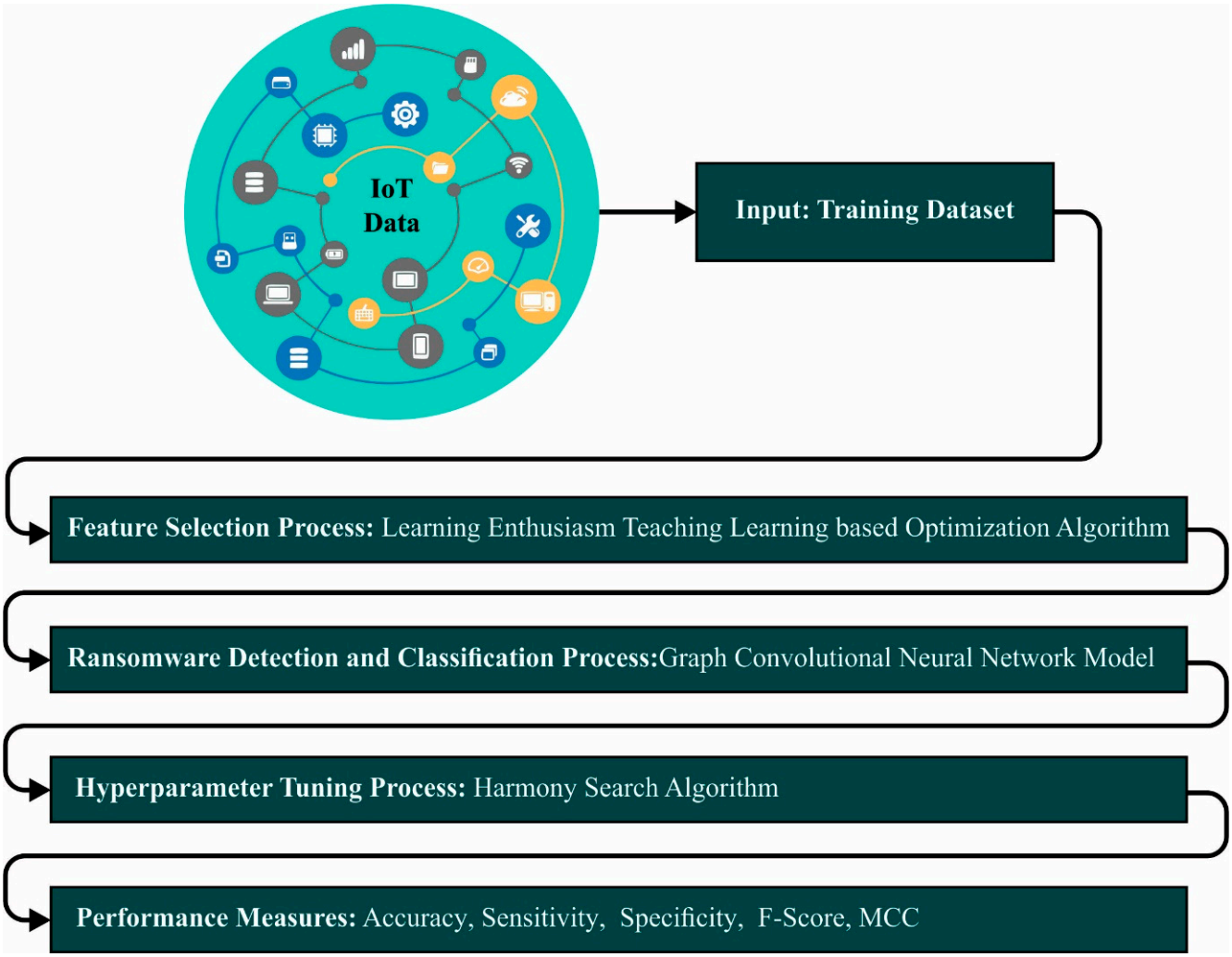

Figure 2. Workflow of OGCNN-RWD approach.

3.1. Feature Selection: LETLBO Algorithm

In this work, the LETLBO algorithm is exploited for an optimal subset of features. LETLBO is an improved version of the fundamental TLBO technique. A TLBO modification increases the ability for searching for better solutions. LETLBO combines two novel components such as the learning enthusiasm-based teacher and learner phases [18]. These are added for improving the typical of worse learners by utilizing the worst student tutoring stage and for raising searching potency. Based on the basic TLBO, all the learners have similar abilities for obtaining the knowledge of others. However, LETLBO reached their stimulus through the learning enthusiasm process, but all the learners have a unique group of capabilities and enthusiasm for learning. A primary step contains a population of NP learners (whereas the entire populations are referred to as x), with initialization as:

$$x_i^j = x^j{}_{\min} + a_b \times \left(x^j{}_{\max} - x^j{}_{\min} \right) \quad (1)$$

where $i \in \{1, 2, \ldots, NP\}, j \in \{1, 2, \ldots, D\}$, $x_{i,j}$ refers to the ith solution from the jth dimension; a_b represents a random number between 0 and 1, and $\chi_{\min/} x_{\max}$ defines the lower as well as upper bounds, respectively. Next, the learner population is initialized, and

every learner's fitness was calculated. The maximum fitness learner is termed the teacher, signified as the $x_{teacher}$. The stages of the LETLBO technique are defined below.

3.1.1. Learning Enthusiasm-Based Teacher Phase

LETLBO is a learning enthusiasm-based paradigm, but students with optimum estimations are more enthusiastic about learning, and thus, it is highly possible to learn with the instructor. The estimated student is less inspired to learn, and it is less possible to receive what the educator needs to tell.

During this stage, every learner can sort based on their fitness value:

$$f(x1) \leq f(x_2) \leq \cdots \leq f(x_{NP}) \tag{2}$$

The learner learning enthusiasm value was determined as:

$$LE_i = LE_{\min} + (LE_{\max} - LE_{\min}) \frac{NP - i}{NP},\ i = 1, 2, \ldots . NP \tag{3}$$

where LE_{max} indicates the maximal learning enthusiasm, and LE_{min} refers to the minimal learning enthusiasm, with referred values of $LE_{\max} = 1$ and $LE_{\min} \in [0.1, 0.5]$. The learning enthusiasm curve illustrates that the better learner reveals maximal learning enthusiasm and the worse learner displays minimal learning enthusiasm.

Due to the characteristics of learning enthusiasm, all the students are classified as both learning and gaining in the teacher and not learning in the instructor, depending on the learning enthusiasm value LE. It generates an irrational number $r_i \in [0, 1]$ for student x_i; if $r_i \leq LE_i$; afterwards, student x_i is advantageous for the educator; otherwise, student x_i neglects the instructor's teachings generally. If student x_i obtains the skill of the teacher, the position is restored by exploiting a change of upgraded displaying techniques in the subsequent situations:

$$x_{i,new}^d = \begin{cases} x_{i,old}^d + rand\left(x_{teacher}^d - T_F \cdot x_{mean}^d\right)\ if\ rand_1 < 0.5 \\ x_{r_1}^d + F \cdot x_{r_2}^d - x_{r_3}^d\ otherwise \end{cases} \tag{4}$$

where r_1, r_2 and $r_3 (r_1 \neq r_2 \neq r_3 \neq i)$ represent the arbitrarily created integers selected in $\{1, 2, \ldots, NP\}; d \in \{1, 2, \ldots, D\}; rand_1$ and $rand_2$ signifies the arbitrarily created numbers that are uniformly distributed in the range of zero and one, and F signifies the scaling factor from range 0 to 1. Equation (4) is observed as a hybrid method of TLBO and DE.

3.1.2. Learning Enthusiasm-Based Learner Phase

The learner system for learning is also learning enthusiasm-based from LETLBO. Related to the teaching approach, it integrates maximal learning enthusiasm to obtain better grades, and it may be a higher probability region to attain the data. During this learning enthusiasm-inspired learner stage, every learner can rank depending on the efficiency of the grades as determined in Equation (3).

The count is created randomly amongst $r_i \in [0, 1]$ for learner x_i; if $r_i \leq LE_i$, then learner x_i is learned by the other learner; otherwise, the data of the learners are ignored by learner x_i. If learner x_i acquires the data from the teacher, dependent upon a diversity-enhanced teaching manner, their position is upgraded as:

$$\chi_{i,new} = \begin{cases} x_{i,old} + rand \cdot (x - \chi_j),\ if\ f(\chi_i) \geq f(x_j) \\ x_{i,old} + rand \cdot (\chi_{\mathrm{j}} - x_j)\ if\ f(x_j) < f(x_j) \end{cases} \tag{5}$$

where $f(X_i)$ stands for the main function, and $x_{i,old}$ represents the preceding position of ith learners. If $x_{i,new}$ is fitter than $x_{i,old}$, then $x_{i,new}$ is accepted; otherwise, $x_{i,old}$ does not changed.

3.1.3. Poor Student Tutoring Phase

The basic TLBO can not be used for this stage; the initial purpose of this stage is for enhancing the grades of worse students. A similar procedure was utilized under this stage as well, with learners ranking from better to poor depending on their grades.

A learner assumes a worse learner when it exists in the bottom 10%. This stage used to is arbitrarily select learner x_T in all the worse students χ_i, whose rank exists at the top 50%, and the learning is dependent upon the subsequent formula:

$$x^d_{i,new} = x^d_{i,old} + rand \cdot \left(x^d_T - x^d_{i,old}\right) \tag{6}$$

If $x_{i,new}$ is superior to $\chi_{j,old}$, $x_{i,new}$ is accepted; if not, $X_{j,old}$ remains the same. The students with worse grades have a lesser probability of upgrading their position from the type of optimum students, but students with better grades take a comparatively superior probability of upgrading their position. The worse student tutoring stage plays a vital role in enhancing the grades of worse students into that of better students. This technique was appropriate to real-time teaching–learning procedures, but the worst students of all the time require tutorials for its enhancement, more tutorials than if related to other better students.

The fitness function of the LETLBO technique considers the count of selective features and the classifier performance. It minimises the set size of selective features and maximizes classification accuracy. Thus, the subsequent fitness function is utilized for evaluating individual solutions as follows:

$$Fitness = \alpha \times ErrorRate + (1 - \alpha) \times \frac{\#SF}{\#All\_F} \tag{7}$$

where *ErrorRate* denotes the classifier rate of errors exploiting the selective feature. *ErrorRate* can be evaluated as the percentage of inappropriate classifications to the count of classifiers developed in the formula as a value within $[0, 1]$. $\#SF$ implies the selective feature count, and $\#All\_F$ indicates the overall amount of features from the original data. α is exploited for controlling the importance of subset length and classifier quality. Here, α is fixed to 0.9.

3.2. Ransomware Detection: Optimal GCNN Model

To classify ransomware from legitimate activities, the GCNN model is used. The GCNN is a DL framework which works on graph-structured data. CNN is used to work on arbitrary graphs (with any number of edges and nodes, and graphs of some structure, cyclic or not) rather than on images [19]. Consider the image as a "grid graph" (all the nodes represent a pixel, and the pixel matrix of an image represents the adjacent matrix of grid graphs). To exploit the similar concept of filtering an image on the graph, rather than having a pixel that applies the data contained in its adjacent pixel to upgrade its value, it takes a node where it applies its adjacent node to upgrade its features.

The GCNN classifies the edges or examines the existence of a connection between two nodes, classifies every node individually, or classifies the overall graph. To construct a GCNN, we begin to construct the adjacent matrix A of the graphs. For instance, non-oriented graphs consider $A_{ij} = 1$ (with A_{ij} being a component of the adjacent matrix A) when there is a connection between the ith and jth nodes, and $A_{ij} = 0$ if ith and jth nodes are mess linked. In addition, the node matrix H is constructed that contains stored information or a message in all the nodes, and later constructs the matrix $H' = \sigma\left(\hat{D}^{-1}\hat{A}HW\right)$, where W indicates a learnable node-wise shared linear conversion (linear layer in a DL architecture), σ denotes the non-linear function, for example, ReLU, $\hat{A} = A + I$, where $\hat{A}$ does not remove the central node, it forces a node to stay connected with itself, $\hat{D}$ denotes the degree matrix, which provides the degree of all the nodes, $\hat{D}$ is incorporated into the equation for normalizing A and enforcing the feature not to explode, while summing is named as the mean-pooling upgrade rule:

$$H' = \sigma\left(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}HW\right) \tag{8}$$

The GCN update rule can be obtained using the above equation. Currently, this is the more commonly known graph convolution layer. Generally, nodes can transmit arbitrary messages alongside the edge $\vec{e_{ij}}$ and then aggregate each message it receives through the permutable-invariant function, where $\vec{m_{ij}}$ denotes the message transmitted from ith to jth nodes, evaluated by the message function f_e:

$$\vec{m_{ij}} = f_e\left(\vec{h_i}, \vec{h_j}, \vec{e_{ij}}\right), \tag{9}$$

Then, each message which enters the nodes is aggregated through a readout function as follows:

$$f_b : \vec{h_i'} = f_v\left(\vec{h_v}, \sum_{j \in N_i} \vec{m_{ji}}\right), \tag{10}$$

In Equation (10), N_i. represents the group of neighbors of ith nodes. This provides the message-passing neural network (MPNN), which applies only to smaller graphs. f_e and f_t are generally smaller multilayer perceptron and are generally expressed as follows:

$$\vec{h_i'} = \sigma\left(\sum_{j \in N_i} \alpha_{ij} W \vec{h_j}\right), \tag{11}$$

In Equation (11), α_{ij} denotes the coefficient that is explicitly determined to cause certain deficiencies, or

$$\alpha_{\mathrm{ij}} = \frac{\exp\left(a_{ij}\right)}{\sum_{j \epsilon N_i} \exp\left(a_{ik}\right)}, \tag{12}$$

where

$$a_{\mathrm{ij}} = a\left(\vec{h}_v, \vec{h}_j, \vec{e_{ij}}\right), \tag{13}$$

From the expression, a is a shared, learnable, self-attention model. It is named as the graph attention network upgrade rule.

Briefly, the presented graph was encoded using three matrices: W, A, H, and D. Using the aforementioned parameters, a matrix H' can be evaluated after the selected update rule formula. The description of a GCNN is the process of encoding the graphs as matrix H'.

At the final stage, the HSA is applied for the optimal hyperparameter selection process, a new intelligent optimized technique. Similar to how the SA simulates physical annealing, the GA simulates biological evolution, the harmony algorithm simulates the principles of concert performance, and the PSO algorithm [20] simulates flocks of birds. Briefly, for HSA, every solution vector (decision variable set) is stored in harmony memory (HM). The key parameter of HSA includes pitch adjusting rate (PAR), harmony memory size (HMS), distance bandwidth (BW), stopping criterion or several improvisations (NI), and harmony memory consideration rate (HMCR). Generally, the global optimization problems are discussed below. Minimize $f(x)$ subjected to

$$\chi_j \in X_i i = 1, 2, \cdots, N. \tag{14}$$

where $f(x)$ indicates the main function, χ denotes the group of decision parameters x_i, N shows the count of decision parameters, X_i represents the group of the potential range of value for every decision parameter, the upper boundary for every decision parameter is

$B(i)$, and the lower boundary is $LB(i)$; afterwards, $LB(i) \leq X_i \leq UB(i)$. The HM with the size of HMS is produced based on solution space.

$$HM = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_{N-1}^1 & x_N^1 \\ x_1^2 & x_2^2 & \dots & x_{N-1}^2 & x_N^2 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ x_1^{HMS-1} & x_2^{HMS-1} & \dots & x_{n-1}^{HMS-1} & x_N^{HMS-1} \\ x_1^{HMS} & x_2^{HMS} & \dots & x_{\mathrm{N}-1}^{HMS} & x_N^{HMS} \end{bmatrix} \tag{15}$$

Every decision parameter is produced by: $x_i^j = LB(i) + (UB(i) - LB(i)) * r$ for $i = 1, 2, \dots, N$ and $j = 1, 2, \dots, HMS$, where r denotes the arbitrary value within [0, 1]. A new harmony vector is produced using the following rules, such as pitch adjustment, random selection, and memory consideration. Initially, a random number r_1 is generated within $[0, 1]$ and compares r_1 with the initialized HMCR. When $r_1 < HMCR$, a random parameter in the initial HM is taken that is named memory consideration. Otherwise, it is attained by random selection (produced randomly between the search boundary). Lastly, a new harmony parameter is taken. Once it can be upgraded by the memory consideration, it should be attuned, and a parameter r_2 within [0, 1] is produced randomly as explained in Algorithm 1 below. When $r_2 < PAR$, the parameter based on the initial BW is adjusted and a newly generated parameter that is named pitch adjustment is obtained:

$$x_i^{Inew} = x_i^{new} \pm r * BiV \tag{16}$$

where r denotes the randomly generated value within [0, 1].

Algorithm 1 Pseudocode of HSA

```
Initialize the parameters HMCR, HMS, BW, PAR, Tax
Initialize the HM
Repeat
        Create a New Harmony as:
        for every i, perform
                x^new → { memory consideration with probability HMCR
                        { random selection with probability 1 − HMCR
        if x_i^new ∈ HM, then
        x_i^Inew = { x_i^new ± r * BW    with probability PAR
                   { x_i^new             with probability 1 − PAR
        end if
        end for
        if the new harmony vector is superior to that of the worse one in the novel HM, then
        Upgrade HM
    end if
Until T_max is satisfied
Return better harmony
```

The newly attainted harmony is evaluated by (x). Once the new harmony has the best main function solution when compared to the worst solution in the abovementioned HM, the new harmony substitutes the worst harmony from the HM. If the present amount of times of creation are attained, the abovementioned maximal times $T_{\max}$ of formation are checked. Fitness selection is a critical factor in the HSA technique. Solution encoding can be used to assess the aptitude (goodness) of the candidate solution. Here, the accuracy value is the main condition used to design a fitness function.

$$Fitness = \max(P) \tag{17}$$

$$P = \frac{TP}{TP + FP} \tag{18}$$

From the expression, TP represents the true positive, and FP denotes the false positive value.

4. Performance Validation

The proposed model is simulated using Python 3.6.5 tool on PC i5-8600k, GeForce 1050Ti 4GB, 16GB RAM, 250GB SSD, and 1TB HDD. The parameter settings are given as follows: learning rate: 0.01, dropout: 0.5, batch size: 5, epoch count: 50, and activation: ReLU. In this section, the ransomware classification performance of the OGCNN-RWD technique can be observed on a database comprising 840 samples [21] as represented in Table 1.

Table 1. Details of the dataset.

Class	Number of Instances
Goodware	420
Ransomware	420
Total No. of Samples	840

The confusion matrix of the OGCNN-RWD technique is demonstrated in Figure 3. The outcomes ensure that the OGCNN-RWD system has properly recognized goodware and ransomware samples. For instance, on 100 epochs, the OGCNN-RWD technique identifies 359 goodware and 386 ransomware samples. Moreover, on 200 epochs, the OGCNN-RWD method identifies 372 goodware and 401 ransomware samples. Furthermore, on 300 epochs, the OGCNN-RWD method identifies 372 goodware and 408 ransomware samples. Lastly, on 500 epochs, the OGCNN-RWD approach identifies 417 goodware and 420 ransomware samples.

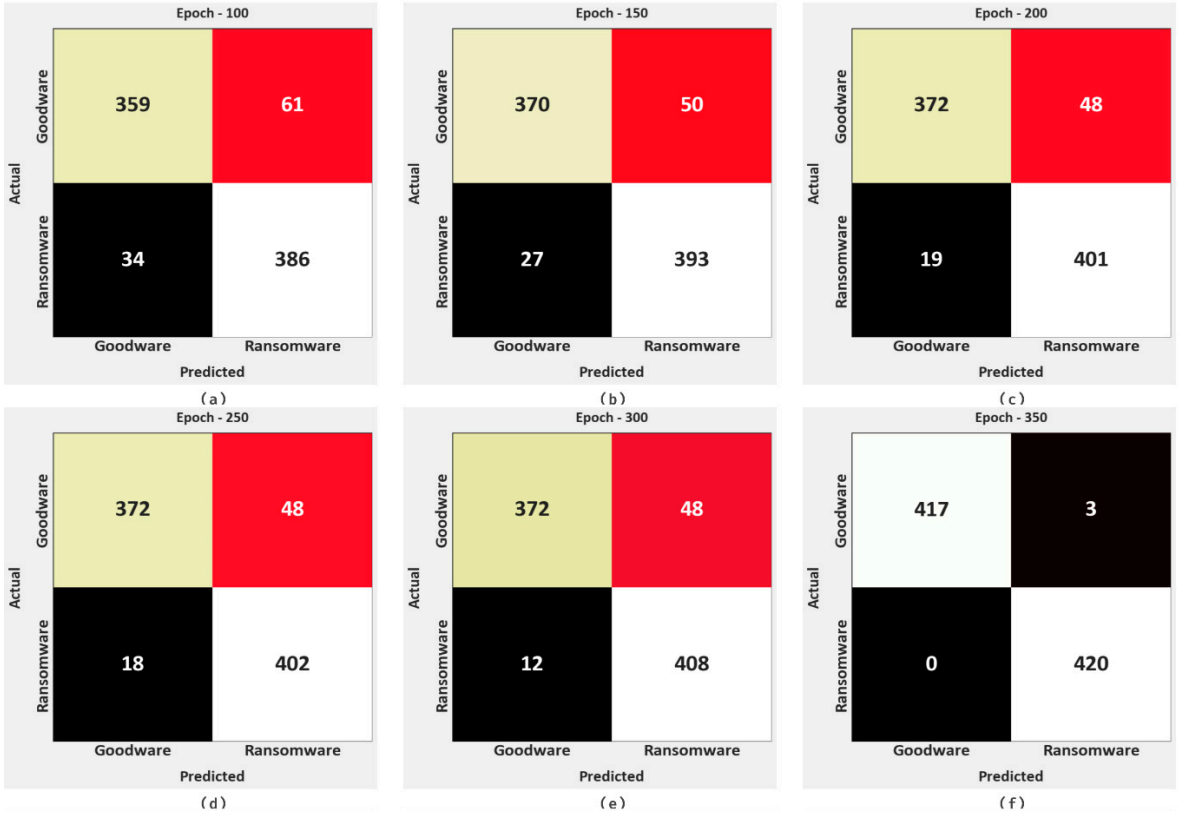

Figure 3. *Cont.*

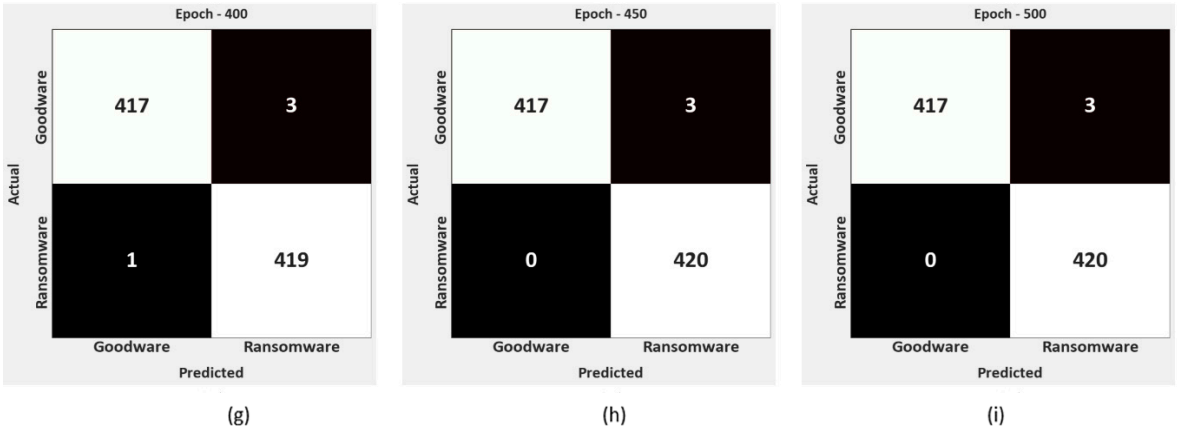

Figure 3. Confusion matrices of OGCNN-RWD system: (**a**–**i**) Epoch 100–500.

In Table 2, the overall ransomware classification outcomes of the OGCNN-RWD technique are inspected in several epochs. The OGCNN-RWD technique properly recognized goodware and ransomware. For the sample, with 100 epochs, the OGCNN-RWD methodology obtained *an* $accu_{bal}$ of 88.69%, $sens_y$ of 88.69%, $spec_y$ of 88.69%, F_{score} of 88.68%, and MCC of 77.54%. In the meantime, with 100 epochs, the OGCNN-RWD approach attained *an* $accu_{bal}$ of 92.02%, $sens_y$ of 92.02%, $spec_y$ of 92.02%, F_{score} of 92.01%, and MCC of 84.25%. Finally, with 100 epochs, the OGCNN-RWD method achieved *an* $accu_{bal}$ of 92.86%, $sens_y$ of 92.86%, $spec_y$ of 92.86%, F_{score} of 92.84%, and MCC of 86.03%. Also, with 100 epochs, the OGCNN-RWD method reached an $accu_{bal}$ of 99.52%, $sens_y$ of 99.52%, $spec_y$ of 99.52%, F_{score} of 99.52%, and MCC of 99.05%. At last, with 100 epochs, the OGCNN-RWD method attained an $accu_{bal}$ of 99.64%, $sens_y$ of 99.64%, $spec_y$ of 99.64%, F_{score} of 99.64%, and MCC of 99.29%.

Table 2. Ransomware classifier outcome of OGCNN-RWD algorithm with distinct epochs.

Class	$Accuracy_{bal}$	Sensitivity	Specificity	F-Score	MCC
Epoch—100					
Goodware	85.48	85.48	91.90	88.31	77.54
Ransomware	91.90	91.90	85.48	89.04	77.54
Average	**88.69**	**88.69**	**88.69**	**88.68**	**77.54**
Epoch—150					
Goodware	88.10	88.10	93.57	90.58	81.79
Ransomware	93.57	93.57	88.10	91.08	81.79
Average	**90.83**	**90.83**	**90.83**	**90.83**	**81.79**
Epoch—200					
Goodware	88.57	88.57	95.48	91.74	84.25
Ransomware	95.48	95.48	88.57	92.29	84.25
Average	**92.02**	**92.02**	**92.02**	**92.01**	**84.25**
Epoch—250					
Goodware	88.57	88.57	95.71	91.85	84.50
Ransomware	95.71	95.71	88.57	92.41	84.50
Average	**92.14**	**92.14**	**92.14**	**92.13**	**84.50**
Epoch—300					
Goodware	88.57	88.57	97.14	92.54	86.03
Ransomware	97.14	97.14	88.57	93.15	86.03
Average	**92.86**	**92.86**	**92.86**	**92.84**	**86.03**

Table 2. *Cont.*

Class	$Accuracy_{bal}$	Sensitivity	Specificity	F-Score	MCC
			Epoch—350		
Goodware	99.29	99.29	100.00	99.64	99.29
Ransomware	100.00	100.00	99.29	99.64	99.29
Average	**99.64**	**99.64**	**99.64**	**99.64**	**99.29**
			Epoch—400		
Goodware	99.29	99.29	99.76	99.52	99.05
Ransomware	99.76	99.76	99.29	99.52	99.05
Average	**99.52**	**99.52**	**99.52**	**99.52**	**99.05**
			Epoch—450		
Goodware	99.29	99.29	100.00	99.64	99.29
Ransomware	100.00	100.00	99.29	99.64	99.29
Average	**99.64**	**99.64**	**99.64**	**99.64**	**99.29**
			Epoch—500		
Goodware	99.29	99.29	100.00	99.64	99.29
Ransomware	100.00	100.00	99.29	99.64	99.29
Average	**99.64**	**99.64**	**99.64**	**99.64**	**99.29**

The TACY and VACY of the OGCNN-RWD method under distinct epochs are represented in Figure 4. The figure states that the OGCNN-RWD approach has shown higher performance with enhanced values of TACY and VACY. Notably, the OGCNN-RWD algorithm has achieved maximal TACY outcomes.

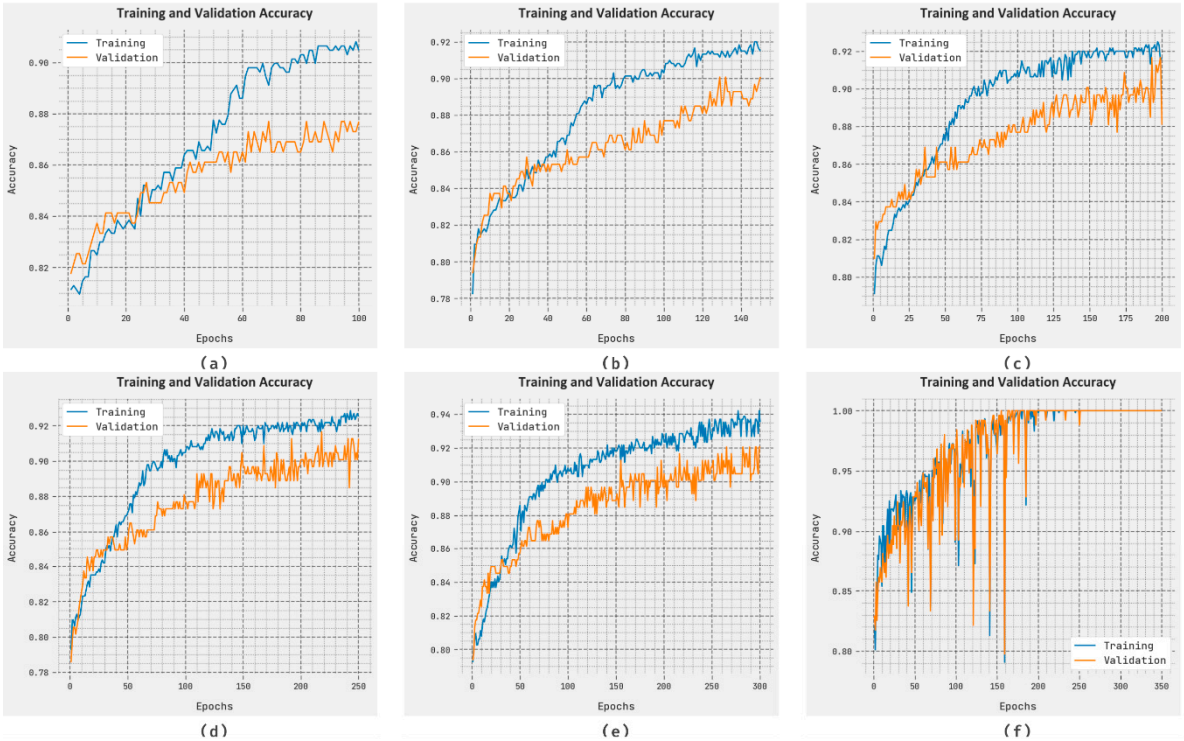

Figure 4. *Cont.*

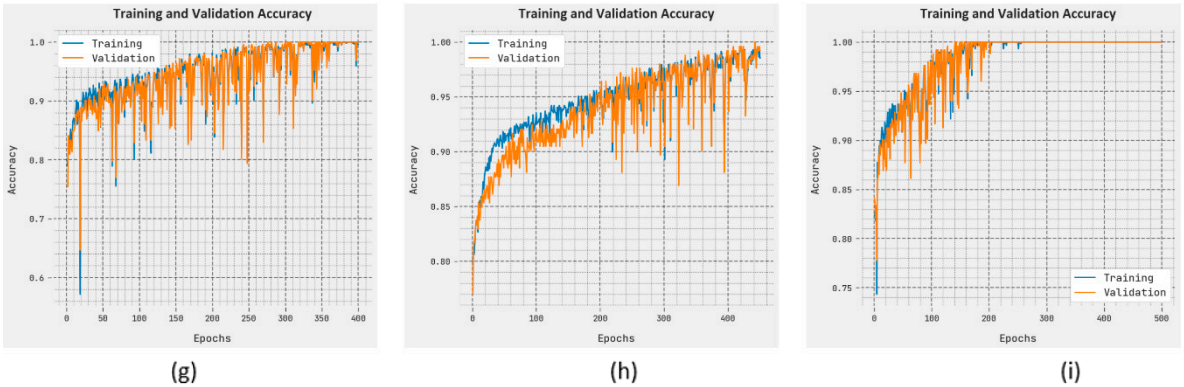

Figure 4. TACY and VACY outcome of OGCNN-RWD system: (**a**–**i**) Epoch 100–500.

The TLOS and VLOS of the OGCNN-RWD technique under distinct epochs are given in Figure 5. The figure infers that the OGCNN-RWD approach has demonstrated improved performance with the least values of TLOS and VLOS. Visibly, the OGCNN-RWD method has reduced VLOS outcomes. The lesser values indicate the effectual detection performance of the proposed model.

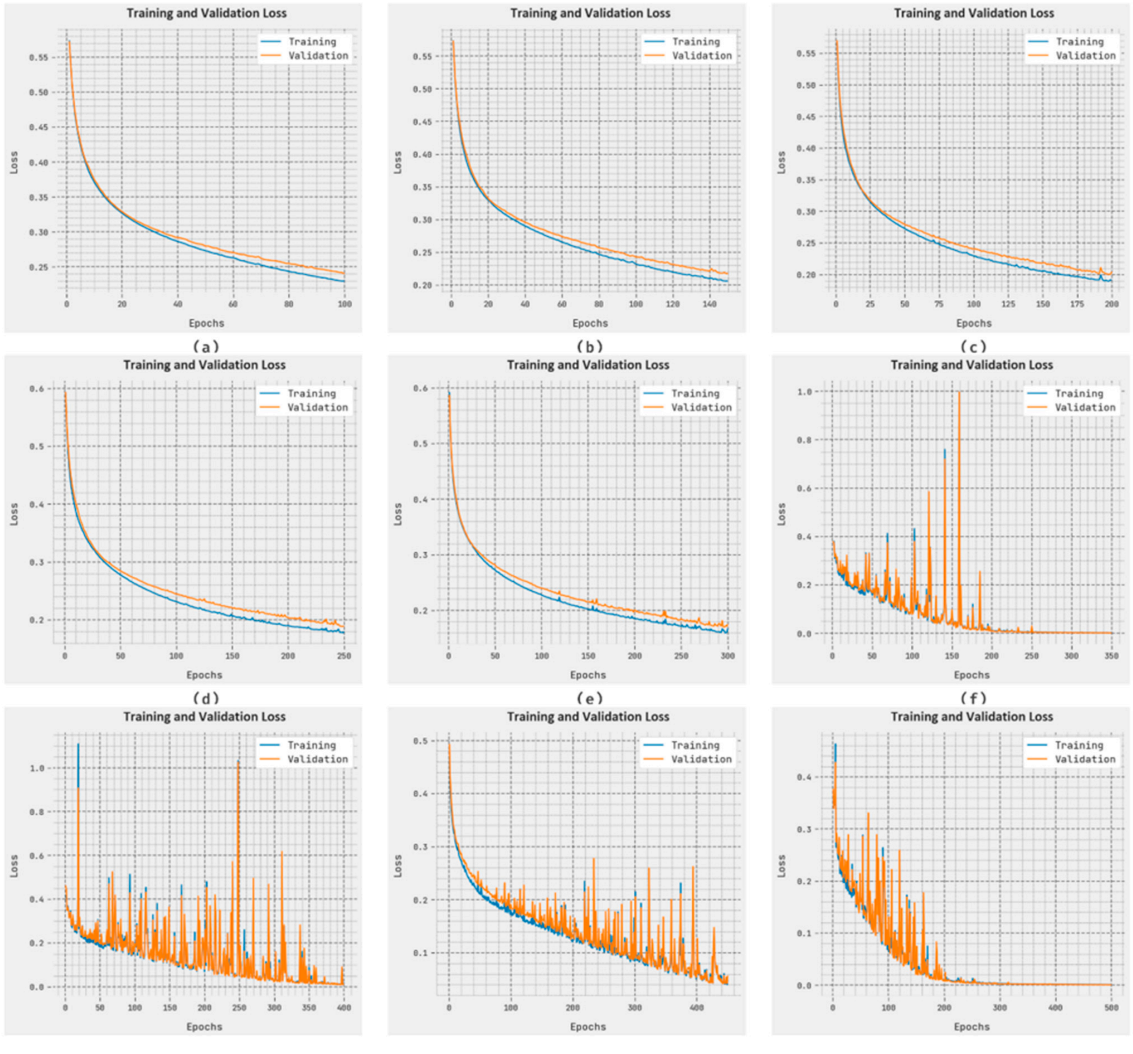

Figure 5. TLOS and VLOS outcomes of OGCNN-RWD system: (**a**–**i**) Epoch 100–500.

A brief precision–recall examination of the OGCNN-RWD method under distinct epochs is shown in Figure 6. The figure designates that the OGCNN-RWD algorithm has higher precision–recall values under two class labels.

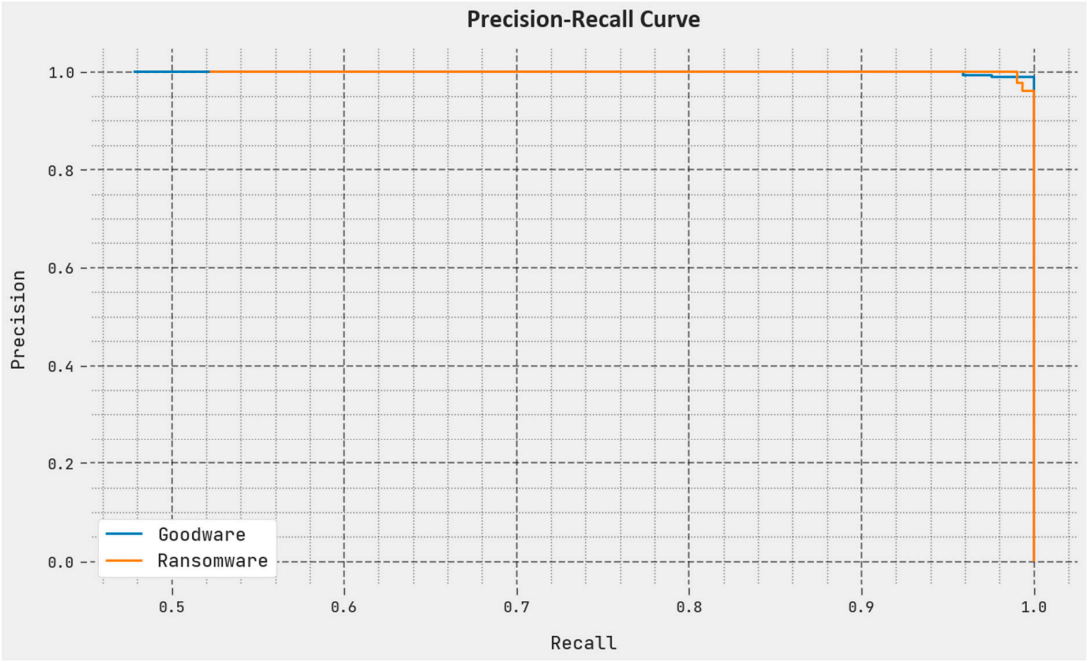

Figure 6. Precision–recall outcome of OGCNN-RWD system.

A clear ROC investigation of the OGCNN-RWD system under distinct epochs is portrayed in Figure 7. The results represent that the OGCNN-RWD algorithm has exhibited its capability in classifying different two-class labels.

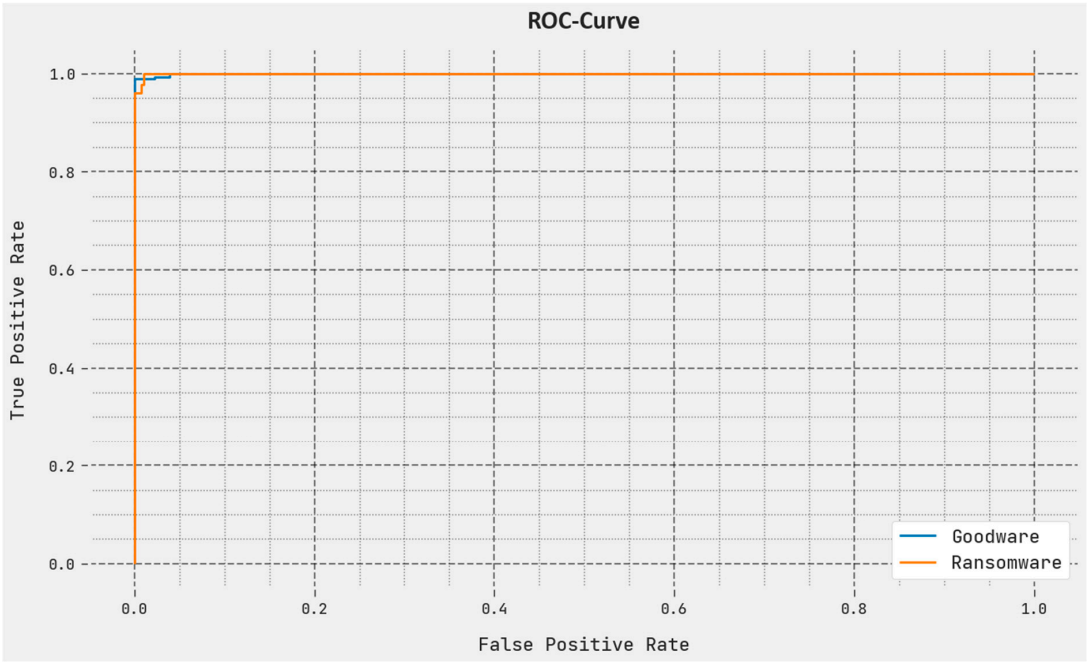

Figure 7. ROC outcome of OGCNN-RWD system.

To assure the improved outcomes of the OGCNN-RWD approach, a brief comparative investigation is made in Table 3 [21,22]. Figure 8 investigates the comparative examination

of the OGCNN-RWD technique in terms of $accu_y$. The experimental values indicate that the OGCNN-RWD technique reaches a maximum $accu_y$ of 99.64% while the DWOML, bagging, AdaBoost-M1, ROF, DT, and RF models result in a minimum $accu_y$ of 99.09%, 98.47%, 96.13%, 95.79%, 97.63%, and 98.83%, respectively.

Table 3. Comparative outcome of OGCNN-RWD approach with existing systems.

Methods	$Accu_y$	$Sens_y$	$Spec_y$
OGCNN-RWD	99.64	99.64	99.64
DWOML Model [21]	99.09	99.43	99.17
Bagging [22]	98.47	93.66	96.06
AdaBoost-M1 [22]	96.13	94.50	94.60
Rotation Forest (ROF) [22]	95.79	96.77	97.38
Decision Tree (DT) [22]	97.63	97.82	98.12
Random Forest (RF) [22]	98.83	98.79	98.26

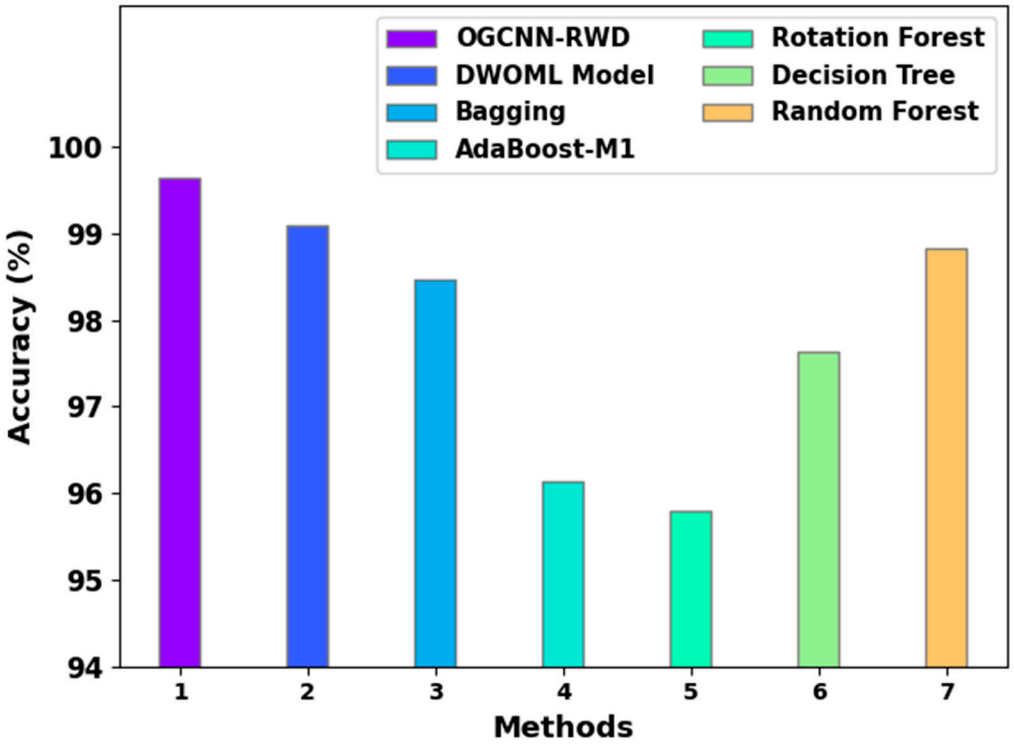

Figure 8. $Accu_y$ outcome of OGCNN-RWD system with existing systems.

Figure 9 inspects the comparative investigation of the OGCNN-RWD algorithm in terms of $sens_y$ and $spec_y$. Based on $sens_y$, the OGCNN-RWD technique reaches a maximum $accu_y$ of 99.64% while the DWOML, bagging, AdaBoost-M1, ROF, DT, and RF methods result in minimal $sens_y$ of 99.43%, 93.66%, 94.50%, 96.77%, 97.82% and 98.79%, respectively. Likewise, based on $spec_y$, the OGCNN-RWD technique reaches a maximum $spec_y$ of 99.64% while the DWOML, bagging, AdaBoost-M1, ROF, DT, and RF approaches result in minimum $spec_y$ of 99.17%, 96.06%, 94.60%, 97.38%, 98.12% and 98.26%, respectively.

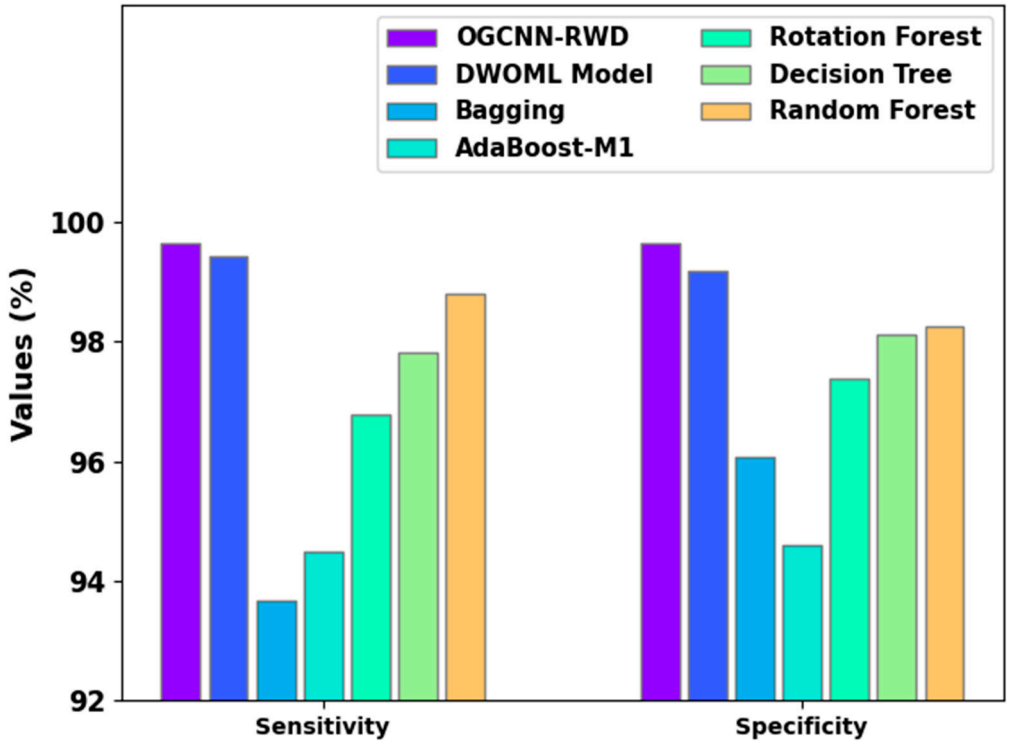

Figure 9. $Sens_y$ and $spec_y$ outcome of OGCNN-RWD algorithm with existing systems.

These results show the enhanced performance of the OGCNN-RWD technique over other models.

5. Conclusions

In this article, we established a novel OGCNN-RWD methodology for cybersecurity in an IoT environment. The OGCNN-RWD technique mainly intends to precisely distinguish ransomware from legitimate activities. In the presented OGCNN-RWD system, three subprocesses are involved, namely, the LETLBO approach-based feature subset selection, GCNN-based ransomware detection, and HSA based hyperparameter tuning. For exhibiting greater performance of the OGCNN-RWD algorithm, a series of simulations were made on the ransomware database. The simulation results portray the betterment of the OGCNN-RWD system over other existing systems with a maximum accuracy of 99.64%. Thus, the OGCNN-RWD methodology is employed for accurate ransomware detection in the IoT platform. In the future, we plan to extend the OGCNN-RWD technique by the design of an ensemble learning process.

Author Contributions: Conceptualization, H.K.A.; Methodology, K.M.; Software, M.O.; Validation, K.M., M.O., M.K. and M.A.D.; Formal analysis, A.A.A.; Investigation, A.E.O.; Data curation, M.A.D. and A.E.O.; Writing—original draft, H.K.A., K.M., M.A.D. and A.A.A.; Writing—review & editing, H.K.A., M.O., A.E.O., M.K., A.S.Z. and A.A.A.; Visualization, M.O. and A.E.O.; Supervision, H.K.A.; Project administration, M.A.D.; Funding acquisition, H.K.A., K.M. and M.A.D. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through Large Groups Project under grant number (112/44). Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R384), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. This study is supported via funding from Prince Sattam bin Abdulaziz University project number (PSAU/2023/R/1444).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable to this article, as no datasets were generated during the current study.

Conflicts of Interest: The authors declare that they have no conflict of interest. The manuscript was written through the contributions of all authors.

References

1. Fernando, D.W.; Komninos, N.; Chen, T. A study on the evolution of ransomware detection using machine learning and deep learning techniques. *IoT* **2020**, *1*, 551–604. [CrossRef]
2. Urooj, U.; Al-rimy, B.A.S.; Zainal, A.; Ghaleb, F.A.; Rassam, M.A. Ransomware detection using the dynamic analysis and machine learning: A survey and research directions. *Appl. Sci.* **2021**, *12*, 172. [CrossRef]
3. Damaševičius, R.; Venčkauskas, A.; Toldinas, J.; Grigaliūnas, Š. Ensemble-Based classification using neural networks and machine learning models for windows pe malware detection. *Electronics* **2021**, *10*, 485. [CrossRef]
4. Tien, C.W.; Chen, S.W.; Ban, T.; Kuo, S.Y. Machine learning framework to analyze iot malware using elf and opcode features. *Digit. Threat. Res. Pract.* **2020**, *1*, 1–19. [CrossRef]
5. Bae, S.I.; Lee, G.B.; Im, E.G. Ransomware detection using machine learning algorithms. *Concurr. Comput. Pract. Exp.* **2020**, *32*, e5422. [CrossRef]
6. Sharma, S.; Krishna, C.R.; Kumar, R. 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; pp. 1–6.
7. Dion, Y.; Brohi, S.N. An experimental study to evaluate the performance of machine learning alogrithms in ransomware detection. *J. Eng. Sci. Technol.* **2020**, *15*, 967–981.
8. Noorbehbahani, F.; Rasouli, F.; Saberi, M. 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; pp. 128–133.
9. Almaiah, M.A.; Almomani, O.; Alsaaidah, A.; Al-Otaibi, S.; Bani-Hani, N.; Hwaitat, A.K.A.; Al-Zahrani, A.; Lutfi, A.; Awad, A.B.; Aldhyani, T.H. Performance Investigation of Principal Component Analysis for Intrusion Detection System Using Different Support Vector Machine Kernels. *Electronics* **2022**, *11*, 3571. [CrossRef]
10. Mohammad, A.H.; Alwada'n, T.; Almomani, O.; Smadi, S.; ElOmari, N. Bio-Inspired Hybrid Feature Selection Model for Intrusion Detection. *Comput. Mater. Contin.* **2022**, *73*, 133–150. [CrossRef]
11. Al-Hawawreh, M.; Sitnikova, E. Leveraging deep learning models for ransomware detection in the industrial Internet of things environment. In Proceedings of the 2019 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 12–14 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
12. Al-Hawawreh, M.; Sitnikova, E. Industrial Internet of Things based ransomware detection using stacked variational neural network. In Proceedings of the 3rd International Conference on Big Data and Internet of Things, Melbourn, VIC, Australia, 22–24 August 2019; pp. 126–130.
13. Al-Hawawreh, M.; Sitnikova, E.; Aboutorab, N. Asynchronous Peer-to-Peer Federated Capability-Based Targeted Ransomware Detection Model for Industrial IoT. *IEEE Access* **2021**, *9*, 148738–148755. [CrossRef]
14. Basnet, M.; Poudyal, S.; Ali, M.H.; Dasgupta, D. Ransomware detection using deep learning in the SCADA system of electric vehicle charging station. In Proceedings of the 2021 IEEE PES Innovative Smart Grid Technologies Conference-Latin America (ISGT Latin America), Brisbane, Australia, 5–8 September 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–5.
15. Alrawashdeh, K.; Purdy, C. Ransomware detection using limited precision deep learning structure in fpga. In Proceedings of the NAECON 2018-IEEE National Aerospace and Electronics Conference, Dayton, OH, USA, 24–26 July 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 152–157.
16. Mathane, V.; Lakshmi, P.V. Predictive analysis of ransomware attacks using context-aware AI in IoT systems. *Int. J. Adv. Comput. Sci. Appl.* **2021**, *12*, 0120432. [CrossRef]
17. Ahmed, Y.A.; Huda, S.; Al-rimy, B.A.S.; Alharbi, N.; Saeed, F.; Ghaleb, F.A.; Ali, I.M. A Weighted Minimum Redundancy Maximum Relevance Technique for Ransomware Early Detection in Industrial IoT. *Sustainability* **2022**, *14*, 1231. [CrossRef]
18. Kaur, G.; Jyoti, K.; Mittal, N.; Mittal, V.; Salgotra, R. Optimized Approach for Localization of Sensor Nodes in 2D Wireless Sensor Networks Using Modified Learning Enthusiasm-Based Teaching–Learning-Based Optimization Algorithm. *Algorithms* **2023**, *16*, 11. [CrossRef]
19. Mezair, T.; Djenouri, Y.; Belhadi, A.; Srivastava, G.; Lin, J.C.W. A sustainable deep learning framework for fault detection in 6G Industry 4.0 heterogeneous data environments. *Comput. Commun.* **2022**, *187*, 164–171. [CrossRef]
20. Zhang, Y.; Li, J.; Li, L. A Reward Population-Based Differential Genetic Harmony Search Algorithm. *Algorithms* **2022**, *15*, 23. [CrossRef]

21. Alissa, K.A.; Elkamchouchi, D.H.; Tarmissi, K.; Yafoz, A.; Alsini, R.; Alghushairy, O.; Mohamed, A.; Al Duhayyim, M. Dwarf Mongoose Optimization with Machine-Learning-Driven Ransomware Detection in Internet of Things Environment. *Appl. Sci.* **2022**, *12*, 9513. [CrossRef]
22. Khammas, B.M. Ransomware detection using random forest technique. *ICT Express* **2020**, *6*, 325–331. [CrossRef]