scientific reports

Check for updates

OPEN

An intelligent ransomware based cyberthreat detection model using multi head attention-based recurrent neural networks with optimization algorithm in IoT environment

Sarah A. Alzakari[1], Mohammed Aljebreen[2], Nazir Ahmad[3], Asma A. Alhashmi[4]✉, Sultan Alahmari[5], Othman Alrusaini[6], Ali M. Al-Sharafi[7] & Wafa Sulaiman Almukadi[8]

The rapid growth of the Internet of Things (IoT) and its extensive use in many regions, such as smart homes, healthcare, and vehicles, have made IoT security increasingly critical. Ransomware is an advanced and adjustable threat influencing users globally, limiting admittance to their data or systems over models like file encryption or screen locking. Traditional ransomware detection methods frequently drop, deprived of the ability to combat these threats successfully. Therefore, an effective and reliable mechanism is needed for ransomware detection. Deep learning (DL) and machine learning (ML) methods are very efficient and enhance model efficacy, offering burgeoning research paths, mainly in the ransomware detection realm, and presenting advantageous possibilities for new solutions. This study proposes a novel Multi-head Attention-Based Recurrent Neural Network with Enhanced Gorilla Troops Optimization for Cybersecurity Ransomware Detection (MHARNN-EGTOCRD) approach. The main goal of the MHARNN-EGTOCRD approach is to detect and classify ransomware attacks using advanced hybrid and optimization models in IoT environments. In the data normalization stage, the min-max normalization transforms input data into a suitable format. The dung beetle optimization (DBO) model is employed for the feature selection procedure to eliminate irrelevant, redundant, or noisy features. In addition, the proposed MHARNN-EGTOCRD model also implements a multi-head attention mechanism hybrid with a long short-term memory (MHA-LSTM) model for detecting ransomware. Finally, the hyperparameter selection of the MHA-LSTM model is performed by utilizing the EGTO model. The experimental analysis of the MHARNN-EGTOCRD technique is established on a ransomware detection dataset. The experimental validation of the MHARNN-EGTOCRD technique portrayed a superior accuracy value of 98.53% over existing models.

Keywords Cybersecurity, Ransomware detection, Enhanced Gorilla troops optimization, Internet of things, Data normalization, Feature selection

[1]Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia. [2]Department of Computer Science, Community College, King Saud University, P.O. Box 28095, Riyadh 11437, Saudi Arabia. [3]Department of Computer Science, Applied College at Mahayil, King Khalid University, Abha, Saudi Arabia. [4]Department of Computer Science, College of Science, Northern Border University, Arar, Saudi Arabia. [5]King Abdul Aziz City for Science and Technology (KACST), Cybersecurity Institute, Riyadh, Kingdom of Saudi Arabia. [6]Department of Engineering and Applied Sciences, Applied College, Umm Al-Qura University Makkah, Mecca, Saudi Arabia. [7]Department of Computer Science and Artificial Intelligence, College of Computing and Information Technology, University of Bisha, 67714 Bisha, Saudi Arabia. [8]Department of Software Engineering, College of Engineering and Computer Science, University of Jeddah, Jeddah, Saudi Arabia. ✉email: Asma.alhashmi@nbu.edu.sa

The application of interconnected smart gadgets, generally termed the IoT, has had considerable development. IoT devices might be obtained from someplace, such as an office, vehicle, or home, to attain everyday activities[1]. Such smart gadgets are employed in healthcare services, smart cities, industries, vehicular networks, smart homes, and smart grids. In dual upsides and downsides, the gadgets related to the Internet are in danger of threats and digital attacks, inducing the administration's inability to transmit administrative refusals[2]. There are no established safety models that guarantee the digital security of such devices[3]. IoT has become a capitated platform for attackers since it can launch each kind of system threat on connected gadgets, usually leading to a few severe losses[4]. Applications or Malicious threats like malware and ransomware families continuously pose critical security concerns to cyber security and can cause catastrophic losses to data centres, computer systems, and the web through multiple industries and businesses[5].

Ransomware is primarily advanced to block and prevent victims from accessing system databases by utilizing a strong encoding model that attackers might decode[6]. Ransomware is a different and advanced attack that affects users throughout the globe and limits consumers from accessing the system or data by locking or encrypting the system screening and the consumer files until a ransom is contributed[7]. Traditional ransomware recognition models are unfit to oppose the threats. Consequently, artificial intelligence (AI) models have been deciding between cutting-edge and helpful models over recent years. Therefore, these models perform a substantial role in multiple domains, comprising information and cyber security[8]. AI-based DL and ML models were accepted to enhance their functionalities and flourished to recognize the diverse kinds of intrusions and threats, particularly unpredictable and unforeseen threats[9]. Compared with conventional ML techniques, DL can rapidly identify anomalies and assist in in-depth network data analysis[10].

This study proposes a novel Multi-head Attention-Based Recurrent Neural Network with Enhanced Gorilla Troops Optimization for Cybersecurity Ransomware Detection (MHARNN-EGTOCRD) approach. The main goal of the MHARNN-EGTOCRD approach is to detect and classify ransomware attacks using advanced hybrid and optimization models in IoT environments. In the data normalization stage, the min-max normalization transforms input data into a suitable format. The dung beetle optimization (DBO) model is employed for the feature selection procedure to eliminate irrelevant, redundant, or noisy features. In addition, the proposed MHARNN-EGTOCRD model also implements a multi-head attention mechanism hybrid with a long short-term memory (MHA-LSTM) model for detecting ransomware. Finally, the hyperparameter selection of the MHA-LSTM model is performed by utilizing the EGTO model. The experimental analysis of the MHARNN-EGTOCRD technique is established on a ransomware detection dataset. The major contribution of the MHARNN-EGTOCRD technique is listed below.

- The MHARNN-EGTOCRD model utilizes min-max normalization to standardize input data, improving the accuracy and stability of the detection process. This step ensures that all features are scaled appropriately, enhancing model performance. By employing this technique, the model can more effectually process and analyze data for ransomware detection.
- The MHARNN-EGTOCRD approach employs the DBO model for feature selection to detect the most relevant features for ransomware detection. This methodology enhances the model's performance by concentrating on the most critical variables, mitigating dimensionality. As a result, the model becomes more effective and accurate in detecting cyber threats.
- The MHARNN-EGTOCRD method improves ransomware detection by incorporating the MHA-LSTM model and effectively capturing intrinsic temporal dependencies. This integration allows the model to concentrate on significant patterns in data over time, significantly improving the technique's capability to detect evolving ransomware threats.
- The MHARNN-EGTOCRD methodology employs EGTO-based hyperparameter selection to fine-tune the model's parameters. This approach optimizes key settings, improving the technique's efficiency and overall predictive accuracy. By adjusting the hyperparameters, the model attains enhanced performance in ransomware detection.
- The novelty of the MHARNN-EGTOCRD model is in its unique integration of DBO for feature selection, MHA-LSTM for ransomware detection, and EGTO for hyperparameter optimization. This incorporation creates a robust and effective framework for detecting cyber threats. The model improves feature relevance and predictive performance by utilizing these advanced techniques. This novel approach significantly improves ransomware detection and cybersecurity resilience.

Related works

Hurley et al.[11] developed a novel recognition model named Adaptive Behavior Fingerprinting (ABF), which notably advanced to improve real-world recognition ability for ransomware by utilizing an adaptive learning structure concentrated on the behavioural study. ABF addresses current recognition gaps to offer an algorithmic framework that emphasizes behavioural signatures through conventional identifiers. This method presents a systematic technique to feature extractor that prioritizes and chooses ransomware-specific features, permitting the recognition method to continue either lightweight or efficient. The authors[12] introduce an Automated Android Malware Detection utilizing the Optimum Ensemble Learning Approach for Cyber-security (AAMD-OELAC) model. Then, the HPO method is leveraged for optimum parameter tuning of 3 DL techniques, which assists in performing enhanced malware recognition outcomes. Moritaka and Komuro[13] developed an innovative double-layered Random Forest method to increase ransomware recognition by utilizing a hierarchic study of opcode progressions, offering robustness and superior precision compared to classical techniques. The projected model contains a primary layer that takes overall opcode distribution designs, succeeded by an improved second layer that aimed at the most segregated aspects recognized over cutting-edge feature engineering models like TF-IDF transformations and n-gram techniques. In[14], a new structure is projected that synergizes the predictive

intensities of DL techniques with the dynamic decision-making abilities of Monte Carlo Tree Search (MCTS), offering an inclusive solution to the challenges modelled by developing ransomware alternatives. Over rigorous estimation, the hybrid structure established a substantial development in recognition precision, decreasing false positives and outperforming traditional ML techniques. The incorporation of MCTS permitted the exploration of several decision paths, improving the flexibility of innovative attacks in the real world.

The author[15] introduced a Rock Hyrax Swarm Optimize with DL-based AMD (RHSODL-AMD) technique. This method detects API calls and the essential privileges, which results in effectual differences between the malware and goodware applications. The authors[16] introduce an Optimum Graph CNN-based Ransomware Detection (OGCNN-RWD) method for cyber security in an IoT framework. The Learning Enthusiasm for TLBO (LETLBO) models for the FS method. In addition, the GCNN technique is utilized within this paper, and its hyper-parameters might be optimum selected by HSA. Sumathi and Rajesh[17] propose a hybrid IDS by utilizing Back Propagation Network (BPN), Self Organizing Map (SOM), and Grey Wolf Optimizer (GWO) for cloud computing, improving BPN performance. Feature selection is accomplished via a correlation-based approach with Stratified 10-fold cross-validation, and hyperparameters are fine-tuned by utilizing GWO. Dhande, Tiwari, and Rathod[18] develop a novel malware prediction model using Auto Encoders and Attention Mechanisms to improve malware pattern analysis and detection, overcoming the limitations of traditional methods in detecting growing threats and mitigating false positives. Sokkalingam and Ramakrishnan[19] present a hybrid ML IDS model with feature selection using 10-fold cross-validation. Support vector machine (SVM) parameters are fine-tuned by utilizing a hybrid Harris Hawks optimization (HHO) and particle swarm optimization (PSO) approach, with performance validated via a confusion matrix.

Berguiga, Harchay, and Massaoudi[20] present a hybrid DL-based IDS for IoMT networks (HIDS-IoMT), integrating CNN for feature extraction and LSTM for sequence prediction. The model is implemented on a Raspberry Pi using fog computing to improve responsiveness and reduce latency. Sumathi, Rajesh, and Lim[21] develop an efficient IDS for DDoS attack detection using an LSTM-based RNN and autoencoder-decoder DL strategy, with optimal parameter tuning through a hybrid HHO and PSO methods. Liu et al.[22] introduce SilentCatchR, an attack attribution framework that improves training data with a perturbation mechanism, utilizes a transformer-based model for stealth attack detection and combines a probabilistic graphical model for enhanced interpretability. Sumathi, Rajesh, and Karthikeyan[23] improve DDoS attack detection by incorporating C4.5 with SVM and KNN classifiers, utilizing 10-fold cross-validation. Aldossary, Alzamil, and Almutairi[24] introduce a Cross-Layer Convolutional Attention Network (CLCAN) methodology using multi-scale convolution, hierarchical attention, and dynamic feature fusion. Preprocessing techniques enhance data quality and mitigate class imbalances for efficient anomaly detection. Sumathi and Rajesh[25] implement BPN and multi-layer perceptron (MLP) approaches for intrusion detection. Min-max normalization is utilized to preprocess the data, and a hybrid HHO-PSO method selects and tunes significant features. Hwang et al.[26] propose ContextualGraph-LLM (CG-LLM), a framework integrating Graph Neural Networks (GNNs) and Large Language Models (LLMs) for multi-label intrusion detection in Darknet traffic.

The existing research in intrusion detection and malware prediction presents promising advancements, but there are various limitations and research gaps. Many methods, namely ABF and conventional classifiers, encounter issues detecting emerging malware strains like polymorphic and metamorphic variants, resulting in high false positives. Some approaches, like DL models, still face difficulty with adaptability to dynamic attack patterns. Furthermore, many systems do not effectually scale to real-time traffic or IoT networks, suffering from delays and limited interpretability. While some models enhance accuracy, integrating diverse methods (like hybrid models) is often limited, and performance in complex, large-scale environments like IoMT or Darknet is still underexplored. There is also a requirement for more robust, adaptive, and interpretable models that handle growing cyber threats effectively while mitigating computational complexity.

Materials and methods

This paper proposes a new MHARNN-EGTOCRD technique. The main goal of the proposed technique is to detect and classify ransomware attacks using advanced hybrid and optimization models in IoT environments. Figure 1 signifies the workflow of the MHARNN-EGTOCRD model.

Stage I: Min-max normalization

In the data normalization phase, the min-max normalization transforms input data into a suitable format[27]. This model is chosen for its capability to standardize input data within a specific range, usually between 0 and 1, improving ML methods' stability and performance. This technique ensures that all features contribute equally to the model by preventing larger-scale features from dominating the learning process. Compared to other normalization techniques, namely Z-score normalization, min-max normalization is simple to implement and works well when the data distribution is unknown or not Gaussian. Furthermore, it is specifically beneficial when the model depends on distance-based algorithms, ensuring all features are on the same scale. This results in faster convergence during training and improved accuracy.

Data normalization is necessary to remove inconsistent value ranges that may lead to bias in particular DL models and to speed up the optimizer procedure. It additionally develops the data for calculation and restricts the value ranges. This study uses the Min-Max Normalization approach that scales each data value from its unique range from (0,1), thus increasing accuracy and speed performance. The equation for Min-Max Normalization is established:

$$X_{new} = \frac{X - \min(X)}{\max(X) - \min(X)} \tag{1}$$

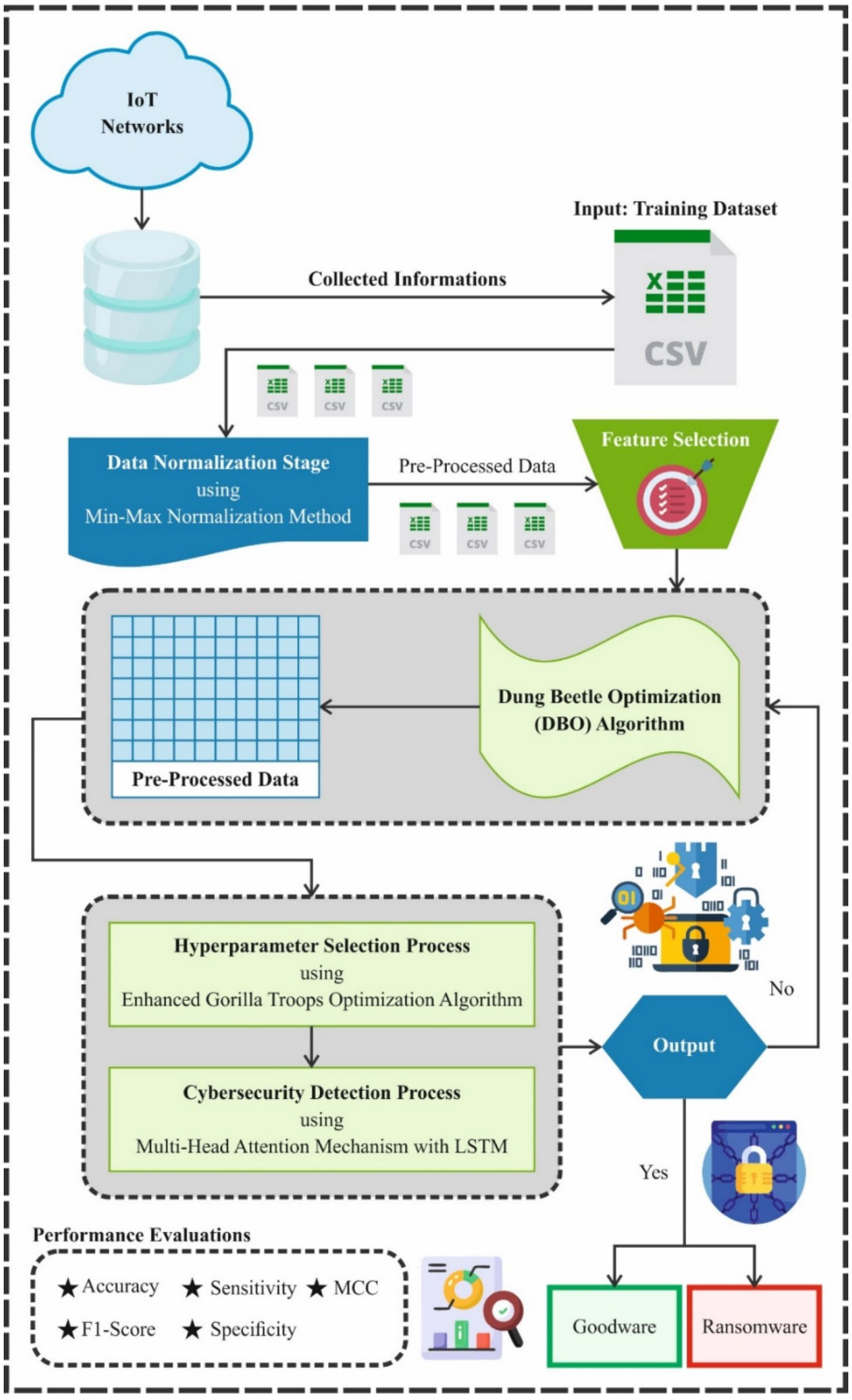

Fig. 1. Workflow of MHARNN-EGTOCRD model.

Whereas X_{new} denotes the new scaled value, X indicates the original value, $and\, Min\,(X)$ and $Max\,(X)$ suggest the data set's minimal and maximal values.

Stage II: DBO-based feature selection

For the feature selection procedure, the DBO model is utilized to eliminate irrelevant, redundant, or noisy features[28]. This method was chosen because of its capability to effectually detect the most relevant features in high-dimensional datasets, which is significant for enhancing the model's performance. This technique is

inspired by the natural foraging behaviour of dung beetles, allowing it to effectively navigate large search spaces and choose the most informative features. Unlike conventional methods like Recursive Feature Elimination (RFE) or mutual information, DBO does not depend on gradient-based approaches, making it appropriate for complex and non-linear relationships within the data. This method assists in mitigating overfitting by eliminating irrelevant or redundant features, enhancing computational efficiency. Additionally, the capability of the DBO model to avoid local optima ensures a more reliable and robust feature selection compared to simpler heuristics. Figure 2 illustrates the steps involved in the DBO methodology.

DBO is the swarm intelligence (SI) optimizer approach, which frequently looks for the optimum solution by mimicking the dancing, rolling, stealing, breeding, and foraging behaviours of DB. The parameter tuning stages according to DBO are usually as shown:

Stage 1: Initialize the DB population and determine the parameter vector being modified, whereas each DB individual symbolizes a vector. The vectors Lb and Ub establish the lower and upper limits of the parameters being modified. Randomly produce the primary location $x\,(0)$ for all DB in the solution area.

Stage 2: Establish the fitness function (FF). Using FF, compute and record the fitness of every DB individual.

Stage 3: DB carries out the behaviour of rolling navigate by sunlight. The location upgrade of rolling is stated as shown:

$$\begin{cases} x_i(t+1) = x_i\,(t) + \gamma\, k x_i(t-1) + b\Delta\, x \\ \Delta\, x = |x_i\,(t) - X^w| \end{cases} \tag{2}$$

Whereas $x_i\,(t)$ characterizes the location of the $i\,th$ DB at iteration t, X^w epitomizes the poorest global location of the iteration, k denotes the coefficient of the deflection, k indicates a constant, γ stands for the allocated natural coefficient of -1 or 1, b represents continuous characterized by the range $(0,1)$, and $\Delta\, x$ has been applied for simulating modifications of light intensity.

In samples, while a DB challenges problems hindering its forward motion, it requires a re-calibration of its route over the dancing-like behaviour performance, thus enabling the finding of another route. This behaviour is stated as follows:

$$x_i\,(t+1) = x_i\,(t) + \tan\,(\theta\,)\,|x_i\,(t) - x_i\,(t-1)| \tag{3}$$

Here, θ signifies the defection angle characterized by $[0,\ \pi\,]$.

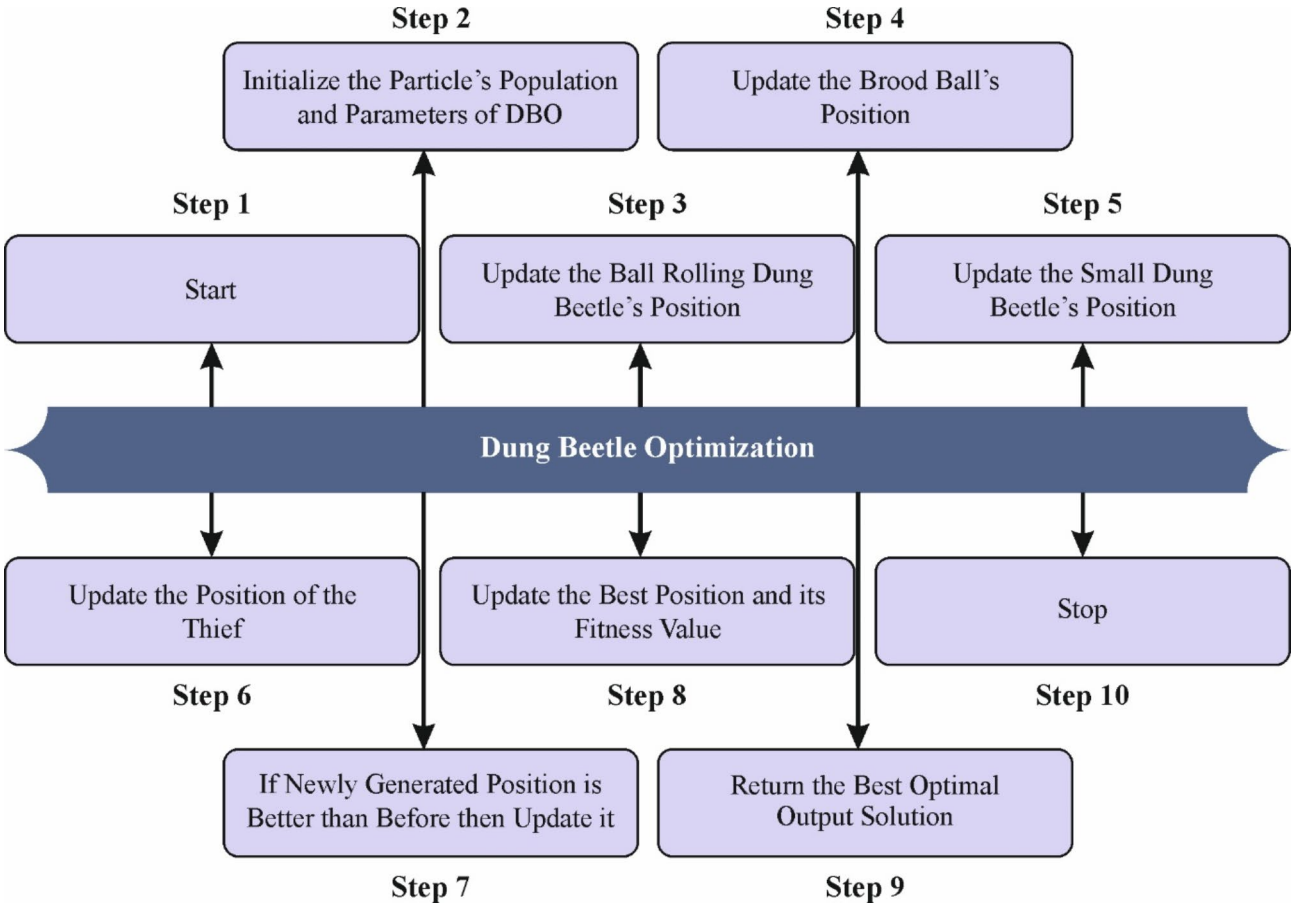

Fig. 2. Steps involved in the DBO model.

Stage 4: Selecting the updated area for the following generations of DBs over the boundary selection approach, mimicking the area choice for laying the egg, stated as:

$$\begin{cases} Lb^{*} = \max\left(X^{*}\left(1-R\right),\ Lb\right) \\ Ub^{*} = \min\left(X^{*}\left(1+R\right),\ Ub\right) \end{cases} \tag{4}$$

Whereas X^{*} characterizes the local optimum location in the iteration, Ub^{*} and Lb^{*} characterize the upper and lower bounds of the spawn region, and $R = 1 - t/T_{ite}$. At the same time, T_{ite} epitomizes the maximal iteration counts. Every DB puts only one egg in all iterations, and the spawning behaviour is stated as follows:

$$B_i\left(t+1\right) = X^{*} + b_1\left(B_i\left(t\right) - Lb^{*}\right) + b_2\left(B_i\left(t\right) - Ub^{*}\right) \tag{5}$$

$B_i\left(t\right)$ characterizes the location of the $i\,th$ brood ball throughout the $t\,th$ iteration, and b_1 and b_2 characterize self-governing arbitrary vectors of similar size as the vector parameters.

Stage 5: Choosing the optimum foraging region for smaller DBs directs their foraging behaviour. This region is designated over the following method:

$$\begin{cases} Lb^{b} = \max\left(X^{b}\left(1-R\right),\ Lb\right) \\ Ub^{b} = \min\left(X^{b}\left(1+R\right),\ Ub\right) \end{cases} \tag{6}$$

Here, X^{b} characterizes the global optimum location throughout the iteration, and Ub^{b} and Lb^{b} epitomize the upper and lower bounds of an optimum foraging region. The positional upgrade for smaller DBs is as demonstrated:

$$x_i\left(t+1\right) = x_i\left(t\right) + D_1\left(x_i\left(t\right) - Lb^{b}\right) + D_2\left(x_i\left(t\right) - Ub^{b}\right) \tag{7}$$

Now $x_i\left(t\right)$ characterizes the location information of $i\,th$ more minor DB at the $t\,th$ iteration, D_1 refers to a number generated at random that emulates normal standard distribution, and D_2 denotes a randomly formed vector appropriate to $(0,1)$.

Stage 6: Imitate the stealing behaviour of the DB. The location upgrade for the thief is defined as demonstrated:

$$x_i\left(t+1\right) = X^{b} + gH\left(\left|x_i\left(t\right) - X^{*}\right| + \left|x_i\left(t\right) - X^{b}\right|\right) \tag{8}$$

Now $x_i\left(t\right)$ signifies the location information of $i\,th$ thief at the $t\,th$ iteration, g represents a randomly generated vector of similar size as the vector parameter, and H symbolizes a continuous value.

Stage 7: Overall rounds of iteration, the FF was computed for every DB, concurrently upgrading the global optimum location X^{b} and the local best location X^{*}. Finally, afterwards, T_{ite} iterations, the most adjusted DB, similar to the most enhanced set of parameters, are recognized. The FF applied in the DBO approach is intended to have balances amongst the selected feature counts in all solutions (minimal), and the classification accuracy (maximal) gained by utilizing these designated characteristics; Eq. (9) characterizes the FF to evaluate solutions.

$$Fitness = \alpha\,\gamma_{R}\left(D\right) + \beta\,\frac{|R|}{|C|} \tag{9}$$

While $\gamma_{R}\left(D\right)$ characterizes the classifier rate of error of a specified classifier, $|R|$ refers to the cardinality of the designated subset, and $|C|$ means the total feature counts in the dataset. α and β represent dual parameters comparable to the importance of classifier excellence and the length of a subset.

Stage III: ransomware detection using MHA-LSTM

In addition, the proposed MHARNN-EGTOCRD approach implements a hybrid of the MHA-LSTM model for ransomware detection[29]. This approach is chosen for ransomware detection because it can capture both short-term and long-term dependencies in sequential data, which is critical for detecting evolving ransomware behaviours. The multi-head attention mechanism allows the model to concentrate on diverse parts of the input sequence, improving its capability to detect key patterns and anomalies related to ransomware activity. LSTM, on the contrary, efficiently handles the temporal nature of the data, allowing the model to remember and learn from previous states. This hybrid methodology outperforms conventional methods, such as CNNs or basic LSTMs, as it can adapt to intrinsic patterns and handle variable-length sequences. Integrating attention and LSTM ensures higher detection accuracy and robustness against advanced ransomware threats. Moreover, this model is computationally effectual and scalable, making it ideal for real-time detection in dynamic environments. Figure 3 represents the architecture of MHA-LSTM.

LSTM is a specific type of RNN that combines a $gate$ mechanism to control the data flow, successfully dealing with the problem of long-term dependencies. All LSTM components include a cell layer, input gate, output gate, and forget gate. These states and gates enable the acquisition of the model of longer-term dependence relations and allow it to disregard or remember input selectively. The particular computation equation is provided as shown:

$$f_t = \sigma\ \left(W_f \cdot\ \left[h_{t-1}, x_t\right] + b_f\right) \tag{10}$$

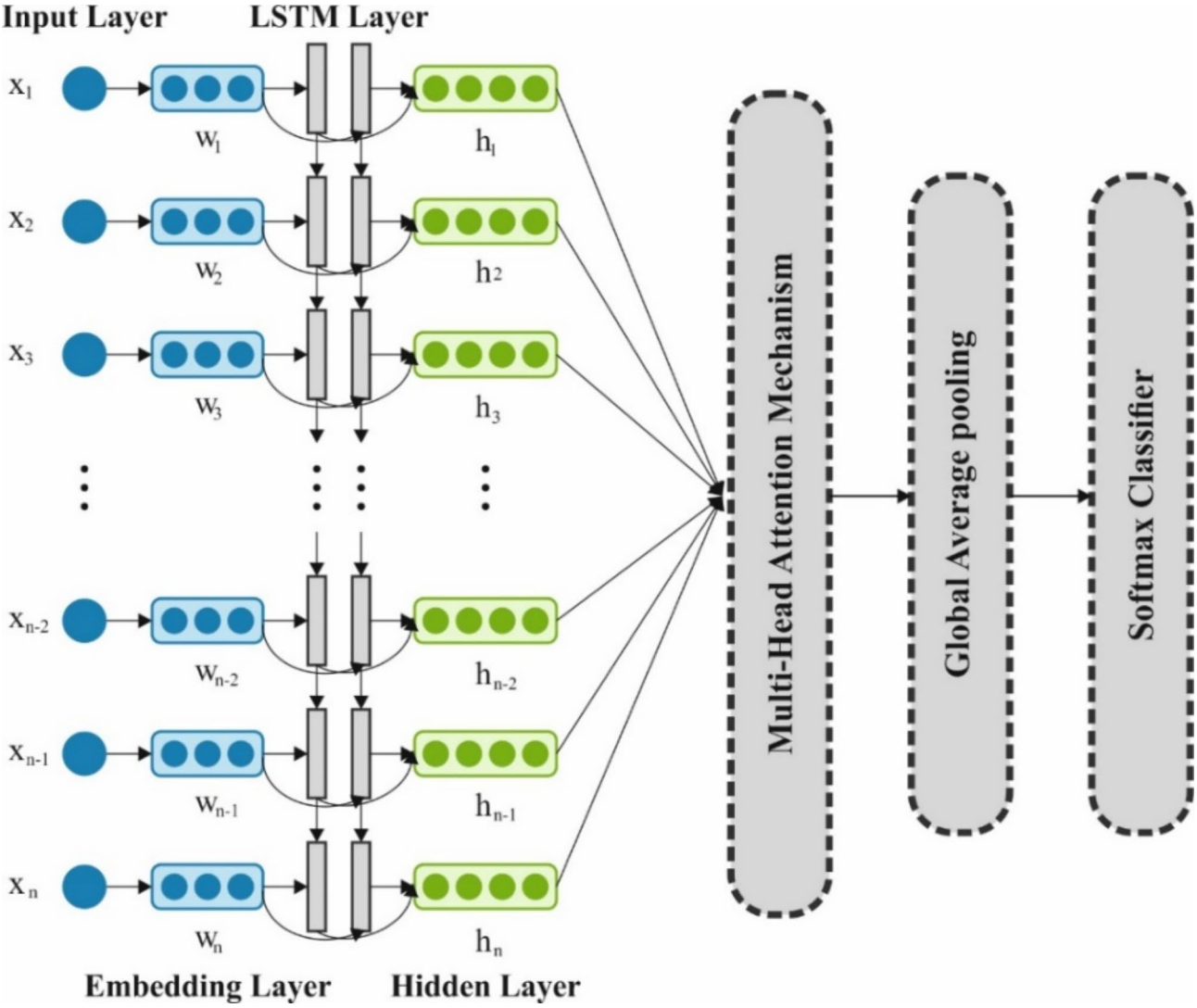

Fig. 3. Structure of MHA-LSTM.

$$i_t = \sigma\ (W_i \cdot\ [h_{t-1}, x_t] + b_i) \tag{11}$$

$$\widetilde{C} = \tanh\left(W_C \cdot\ [h_{t-1}, x_t] + b_C\right) \tag{12}$$

$$C_t = f_t \cdot\ C_{t-1} + i_t \cdot\ \widetilde{C}_t \tag{13}$$

$$o_t = \sigma\ (W_o \cdot\ [h_{t-1}, x_t] + b_o) \tag{14}$$

$$h_t = o_t \cdot\ \tanh\left(C_t\right) \tag{15}$$

For example, the equation above characterizes the computation equations for the update candidate value, cell layer update, input gate, output gate, and forget gate in sequence. i_t and $\widehat{C}_t$ signify the candidate cell state at the present step, σ and $tanh$ refer to the activation and hyperbolic tangent function. W_i and W_c correspondingly represent weighted matrices for the input gate and candidate cell state. W_o denotes the output weighted matrix; W_f signifies the forget weighted matrix, and $b_i,\ b_c,\ b_f$, and b_o denote offset vectors.

By capturing and extracting multi-dimensional features, the MHA mechanism allows the construction of more precise and effective predictive methods. All attention heads may focus on different features or time ranges. In this manner, MHA captures the multi-dimensional features and long short-term dependencies in prediction. The basic process of MHA is established on Scaled Dot-Product Attention, and it takes different feature representations over many independent attention heads. The multi-head attention mechanism supplements the model's representative capacity by calculating numerous attention heads simultaneously. The computation equation for the attention score and the multi-head attention of all heads is as shown:

$$Attention\,(Q,\ K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \tag{16}$$

$$Multihead\,(Q,K,\ V) = Concat\,(head_1,\cdots,head_n)\,W^O \tag{17}$$

In detail, Q characterizes the query matrix, K refers to the key matrix, V specifies the value matrix, d_k signifies the key vector dimension, $1/\sqrt{d_k}$ represents the scaling factor, h characterizes the head counts, and W_o denotes a weighted matrix.

The MHA-LSTM prediction model mainly includes numerous essential components, such as the fully connected (FC) layer, the MHA layer, an input layer, an output layer, and the LSTM layer. In detail, the input layer is responsible for acquiring the time-series data and its outside features and transforming them into a multi-dimensional tensor method appropriate for model processing and input. The multi-head attention mechanism layer can adaptively remove the main features directly associated with the prediction task and then transfer them to the LSTM layer. The LSTM layer handles the data according to the time-series sequence, deeply capturing and excavating the dynamic changing patterns in the time dimensions. Then, the FC layer manages additional in-depth handling of the gained features to make feature vectors with higher-dimensional representations.

Stage IV: hyperparameter tuning process

Finally, the parameter selection of the MHA-LSTM method is performed by utilizing the EGTO method. This method is chosen for its ability to effectually optimize hyperparameters by replicating the collaborative hunting strategy of gorilla troops. This methodology enables the model to explore a large search space for hyperparameter values, assisting in detecting the optimal configuration that improves performance. Unlike conventional techniques such as grid or random search, EGTO presents a more adaptive and intelligent search mechanism that averts local optima and converges faster. EGTO can handle complex and non-linear relationships between hyperparameters using a population-based nature. This method enhances the model's accuracy and robustness, specifically in ransomware detection tasks where fine-tuning is significant. Furthermore, the capability of the EGTO model to balance exploration and exploitation ensures a more reliable optimization process, resulting in improved predictive performance compared to simpler tuning techniques. Figure 4 demonstrates the EGTO model.

The presented method is improved to develop its efficiency and strike an improved balance between exploration and exploitation inside the search procedure[30]. The model is improved through a constriction component and a removal stage to increase its strength, quality of solution, and rate of convergence, which are applied by meta-heuristic models, such as the GTO model.

(A) Limitation component: The model uses a basic constriction component to control the speed of the search procedure. It permits the solutions or particles in the exploration area to converge near possible regions more quickly, whereas exploration of another region is also explored. The constriction component improves the coefficients of acceleration based on the swarm's optimum implementation, and it leads to stopping extreme actions and stimulating fast convergence. The constriction component is used for the random variables, such as $r_1,\ r_2,\ r_3$, and $rand$, using the succeeding equation:

$$r_1 = \frac{\theta}{1-\sqrt{\phi^{\,2}-4\phi}} \tag{18}$$

$$r_2 = \frac{\theta}{1-\sqrt{\phi^{\,2}-4\phi}}) \tag{19}$$

$$r_3 = \frac{\theta}{1-\sqrt{\phi^{\,2}-4\phi}} \tag{20}$$

$$rand = \frac{\theta}{1-\sqrt{\phi^{\,2}-4\phi}} \tag{21}$$

According to the research outcomes, using constraint components through each dimension is a promising model for every dimension. A previous study highlighted that this model provides superior results to related models. The present analysis offers an original development of the constraint element-based approach, which is discovered to be a validated model in scientific study. Rather than maintain their constant, the presented development proposes a slow, linear decrease in the variables $r_1,\ r_2,\ r_3$, and $rand$. To perform this model, the θ value is improved in an iterative method using the equation below:

$$\theta_{\,j} = \theta\ + \left(\overline{\theta}\ - \theta\ \right) \times\ \frac{L-j}{L-1} \tag{22}$$

Now, the variable L specifies the higher number of iterations. The jth variable establishes the present iteration inside the process. Detecting constriction elements is important to preserve the computing strength of the EGTO model.

Utilizing constriction elements to balance exploitation and exploration is a significant part of enhancing meta-heuristic methods, namely the GTO model. This development helps the method to converge more quickly toward an optimum solution. The constriction component improves the coefficients of acceleration depending

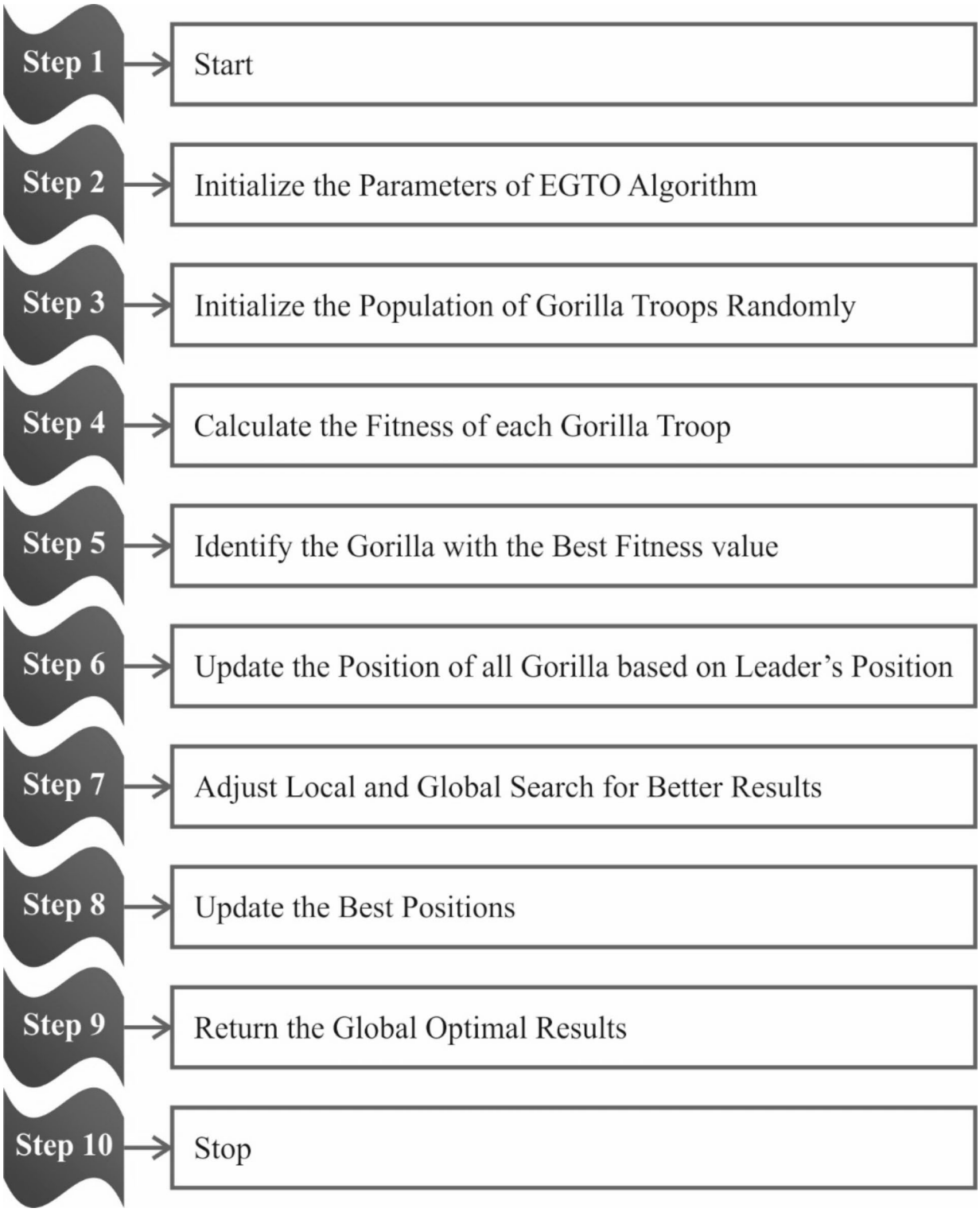

Fig. 4. Architecture of the EGTO technique.

upon the swarm's improved execution. It allows the particles or solutions to meet more effectively toward promising areas as they discover another region. Besides improving convergence, this development additionally increases the model's robustness.

(B) Elimination stage: After all iterations, a method recognized as the elimination stage is performed to eliminate the minimum efficient solution or candidate from the groups. This stage removes a part of the group according to specific selection conditions, such as fitness value.

After all iterations of this model, the elimination stage process is applied. This procedure involves classifying the population based on their fitness values and eliminating a particular solution counted by the lower effectiveness (NE), as described by the ER. Following this, added movements, like reproduction or replacement, are performed to keep the preferred size of the population, thus allowing the population to proceed towards better solutions in time. Incorporating the developments improves the model's optimization performance, convergence speed, and flexibility in composite problem settings. Strike a balance between exploration and exploitation. The constriction element is useful, whereas the elimination stage helps eliminate weak solutions to improve the development of greater individuals. The elimination stage further improves solution value by removing insufficient solutions that slowly enhance the overall population qualities. Furthermore, diversity maintenance is guaranteed by removing weak solutions, which prevents premature convergence and maintains diversity inside the population. Fitness selection is the major feature that manipulates performance in the EGTO model. The hyperparameter choice process comprises the solution encoder method to approximate the effectiveness of the candidate solutions.

$$Fitness = \max(P) \tag{23}$$

$$P = \frac{TP}{TP + FP} \tag{24}$$

Meanwhile, TP symbolizes the positive value of true, and FP indicates the positive value of false.

Experimental result and analysis

The performance analysis of MHARNN-EGTOCRD is studied under the ransomware detection dataset[31]. This dataset contains 840 records under dual-class labels such as Goodware and Ransomware, as portrayed in Table 1. The total number of features is 17, but only 12 features are selected.

Figure 5 displays the classifier results of the MHARNN-EGTOCRD technique. Figure 5a and b exemplifies the confusion matrices by precisely identifying and classifying distinct classes below 70%TRPH and 30%TSPH. Figure 5c demonstrates the PR outcome, which notified superior performance over all classes. Eventually, Fig. 5d represents the ROC outcome, which signifies skilful solutions with great ROC values for dissimilar class labels.

Table 2; Fig. 6 depict the cybersecurity detection of the MHARNN-EGTOCRD approach below 70%TRPH and 30%TSPH.

Using 70%TRPH, the MHARNN-EGTOCRD approach provides average $accu_y$, $sens_y$, $spec_y$, F_{score}, and MCC of 95.52%, 95.52%, 95.52%, 95.57%, and 91.19%, respectively. Simultaneously, using 30%TSPH, the MHARNN-EGTOCRD technique delivers average $accu_y$, $sens_y$, $spec_y$, F_{score}, and MCC of 98.53%, 98.53%, 98.53%, 98.41%, and 96.86%, correspondingly.

In Fig. 7, the training (TRA) $accu_y$ and validation (VAL) $accu_y$ performances of the MHARNN-EGTOCRD technique under 70%TRPH and 30%TSPH are showcased. The values of $accu_y$ are computed across a period of 0–25 epochs. The outcome highlighted that the values of TRA and VAL $accu_y$ present an increasing trend, indicating the capacity of the MHARNN-EGTOCRD technique through enhanced performance across numerous repetitions. Moreover, the TRA and VAL $accu_y$ values remain close through the epochs, notifying decreased overfitting and expressing the higher performance of the MHARNN-EGTOCRD model, which guarantees reliable calculation on unseen samples.

Figure 8 shows the TRA loss (TRALOS) and VAL loss (VALLOS) of the MHARNN-EGTOCRD model under 70%TRPH and 30%TSPH. The loss values are computed over a period of 0–25 epochs. The values of TRALOS and VALLOS demonstrate a declining tendency, which designates the proficiency of the MHARNN-EGTOCRD approach in corresponding a trade-off between generalization and data fitting. The succeeding dilution in loss values also ensures the superior performance of the MHARNN-EGTOCRD method and tunes the prediction outcomes gradually.

Figure 9 presents the classifier outcomes of the MHARNN-EGTOCRD method. Figure 9a and b illustrates the confusion matrices across specific classifications of dissimilar classes under 80%TRPH and 20%TSPH. Figure 9c depicts the PR examination, indicating a higher outcome through all classes. Finally, Fig. 9d demonstrates the ROC examination, signifying proficient solutions using great ROC values for dissimilar classes.

Table 3; Fig. 10 exemplify the cybersecurity detection of the MHARNN-EGTOCRD technique under 80%TRPH and 20%TSPH. The solutions imply that the MHARNN-EGTOCRD technique correctly acknowledged the samples. Using 80%TRPH, the MHARNN-EGTOCRD approach attained typical $accu_y$ of 97.34%, $sens_y$ of 97.34%, $spec_y$ of 97.34%, F_{score} of 97.32%, and MCC of 94.65%. Besides, based on 20%TSPH, the MHARNN-EGTOCRD approach attained typical $accu_y$ of 98.10%, $sens_y$ of 98.10%, $spec_y$ of 98.10%, F_{score} of 98.18%, and MCC of 96.37%.

Figure 11 shows the TRA $accu_y$ and VAL $accu_y$ solutions of the MHARNN-EGTOCRD technique below 80%TRPH and 20%TSPH. The $accu_y$ values are computed through an interlude of 0–25 epochs. The performances underscored that the TRA and VAL $accu_y$ values exhibit a cumulative trend, notifying the

Class labels	Records
"Goodware"	420
"Ransomware"	420
Total records	840

Table 1. Details of database.

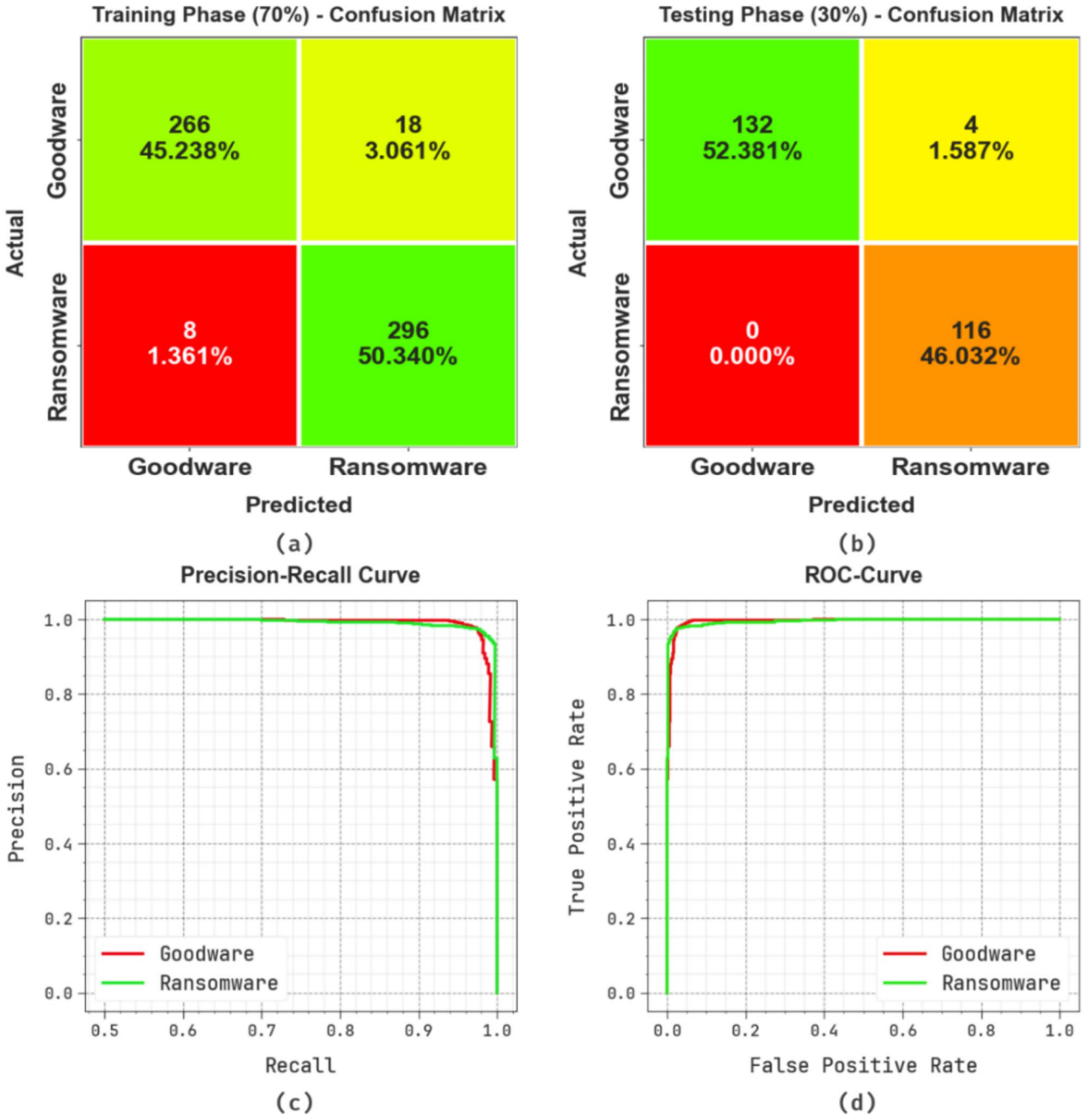

Fig. 5. (**a**,**b**) 70% and 30% confusion matrix and (**c**,**d**) curves of PR and ROC.

proficiency of the MHARNN-EGTOCRD method over superior performance through multiple iterations. In addition, TRA and VAL $accu_y$ values remain closer across the epochs, which notified diminished overfitting and states enhanced performance of the MHARNN-EGTOCRD method, assuring steady calculation on hidden samples.

In Fig. 12, the TRALOS and VALLOS of the MHARNN-EGTOCRD approach below 80%TRPH and 20%TSPH are exemplified. The loss values are computed through an interlude of 0–25 epochs. The TRALOS and VALLOS values represent a diminishing trend, notifying the competency of the MHARNN-EGTOCRD technique in equalizing a trade-off between data fitting and generalization. Moreover, the successive decrease in loss values secures the maximum outcome of the MHARNN-EGTOCRD technique and tunes the calculation solutions after a while.

Table 4 exemplifies the comparative results of the MHARNN-EGTOCRD method with existing methods under dissimilar metrics[16,32].

Figure 13 inspects the comparative $accu_y$ performances of the MHARNN-EGTOCRD approach. The solutions revealed that the MHARNN-EGTOCRD approach gains greater performance. According to $accu_y$

Class	$Accu_y$	$Sens_y$	$Spec_y$	F_{score}	MCC
TRPH (70%)					
Goodware	93.66	93.66	97.37	95.34	91.19
Ransomware	97.37	97.37	93.66	95.79	91.19
Average	95.52	95.52	95.52	95.57	91.19
TSPH (30%)					
Goodware	97.06	97.06	100.00	98.51	96.86
Ransomware	100.00	100.00	97.06	98.31	96.86
Average	98.53	98.53	98.53	98.41	96.86

Table 2. Cybersecurity detection of MHARNN-EGTOCRD model under 70%TRPH and 30%TSPH.

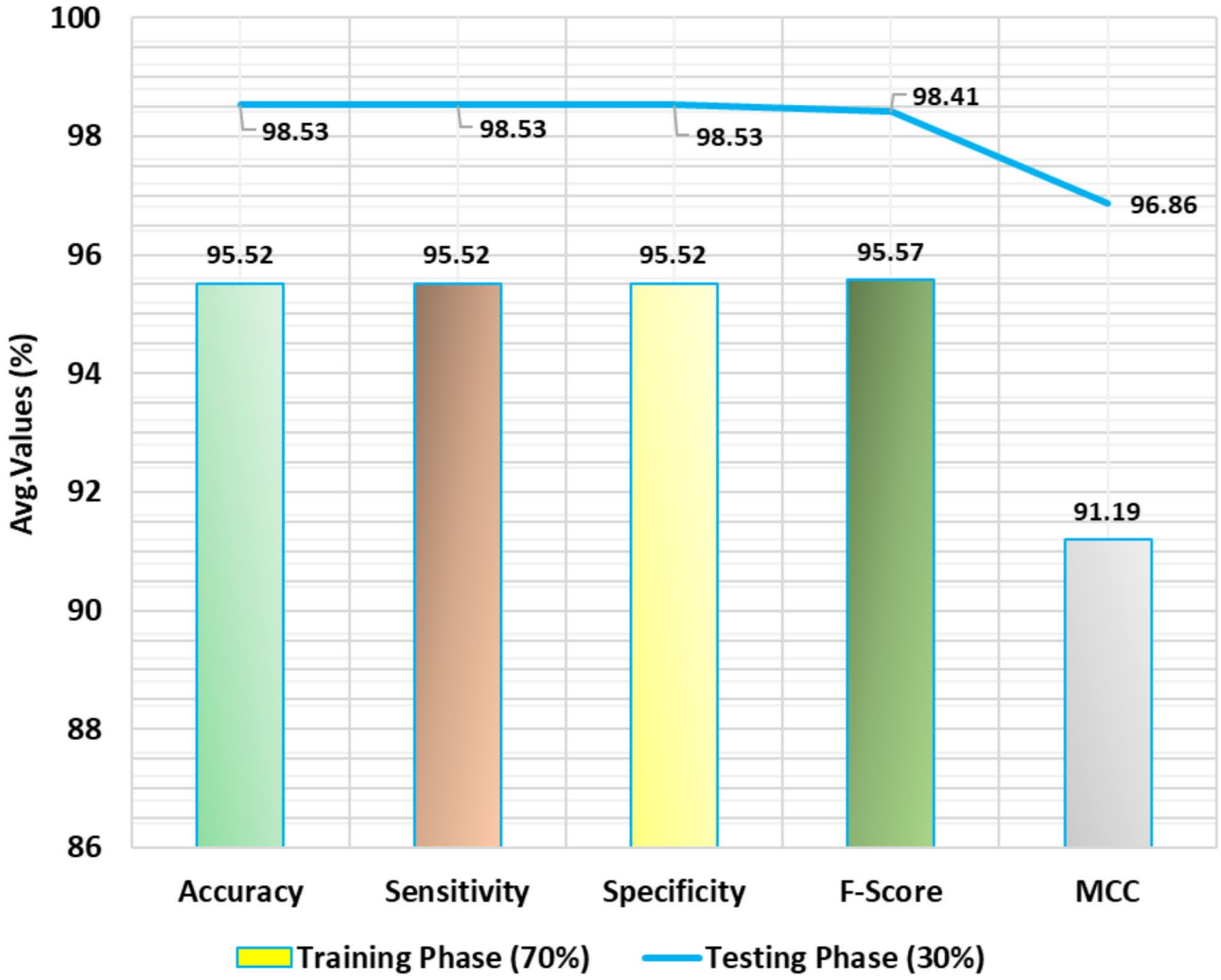

Fig. 6. Average of MHARNN-EGTOCRD model under 70%TRPH and 30%TSPH.

, the MHARNN-EGTOCRD method delivers a maximum $accu_y$ of 98.53%. In contrast, the OGCNN-RWD, DWOML, Bagging, AdaBoost-M1, Rotation Forest (ROF), DT, and RF models attain decrease $accu_y$ of 98.01%, 97.33%, 96.86%, 96.13%, 95.79%, 97.63%, and 97.25%, respectively.

In Fig. 14, a comparative $sens_y$ and $spec_y$ performance of the MHARNN-EGTOCRD technique is delivered. The performances suggest that the Bagging, AdaBoost-M1, and Rotation Forest techniques have exemplified poorer values of $sens_y$ and $spec_y$. Simultaneously, the RF and DT approaches have gained barely better $sens_y$ and $spec_y$. In the meantime, the DWOML and OGCNN-RWD techniques have depicted closer values of $sens_y$ and $spec_y$. However, the MHARNN-EGTOCRD model solutions have higher performance with $sens_y$ and $spec_y$ of 98.53% and 98.53%, respectively.

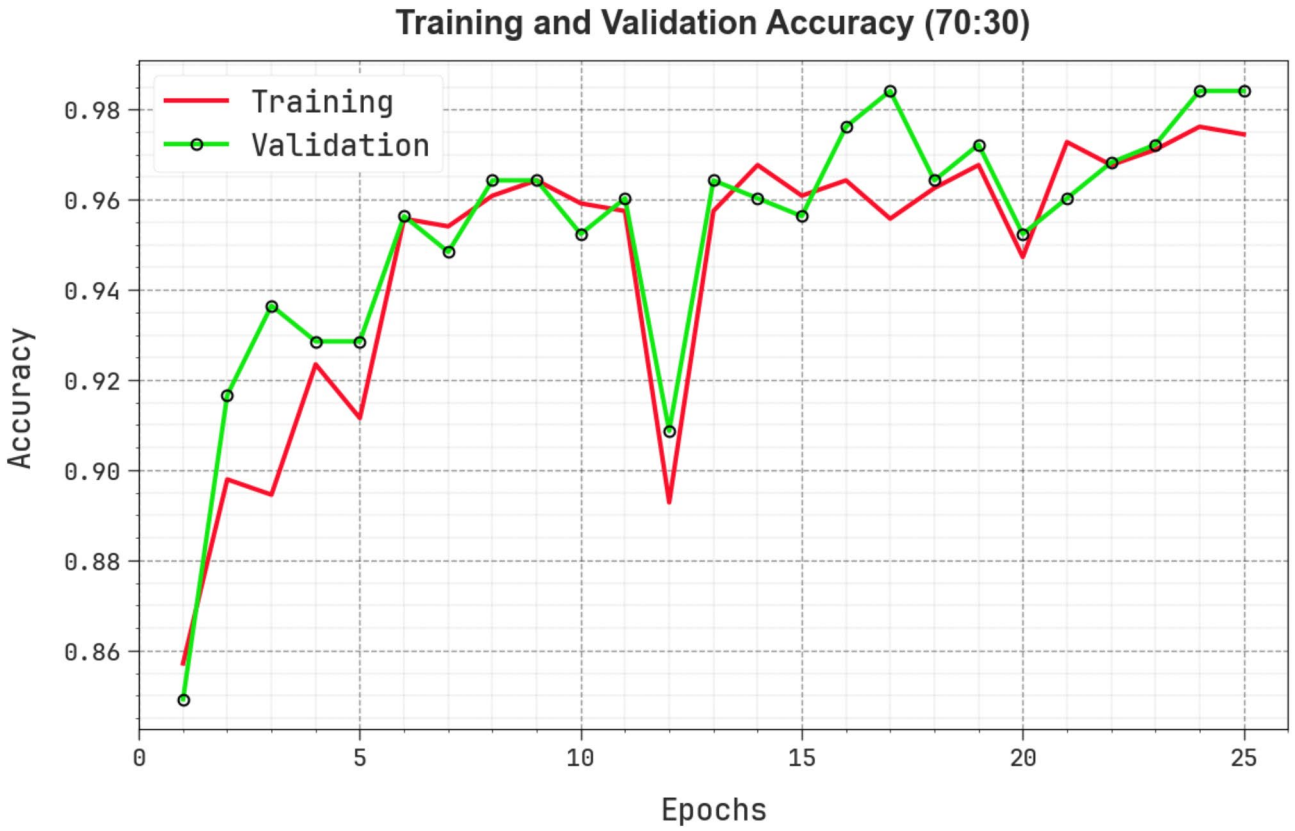

Fig. 7. $Accu_y$ curve of MHARNN-EGTOCRD model under 70%TRPH and 30%TSPH

Conclusion

In this article, a new MHARNN-EGTOCRD approach is proposed. The main goal of the MHARNN-EGTOCRD approach is to detect and classify ransomware attacks using advanced hybrid and optimization models in IoT environments. In the data normalization stage, the min-max normalization transforms input data into a suitable format. The DBO model eliminates irrelevant, redundant, or noisy features for the feature selection process. In addition, the proposed MHARNN-EGTOCRD model implements a hybrid of the MHA-LSTM model for ransomware detection. Eventually, the hyperparameter selection of the MHA-LSTM technique is employed by the design of the EGTO system. The experimental analysis of the MHARNN-EGTOCRD technique is established on a ransomware detection dataset. The experimental validation of the MHARNN-EGTOCRD technique portrayed a superior accuracy value of 98.53% over existing models.

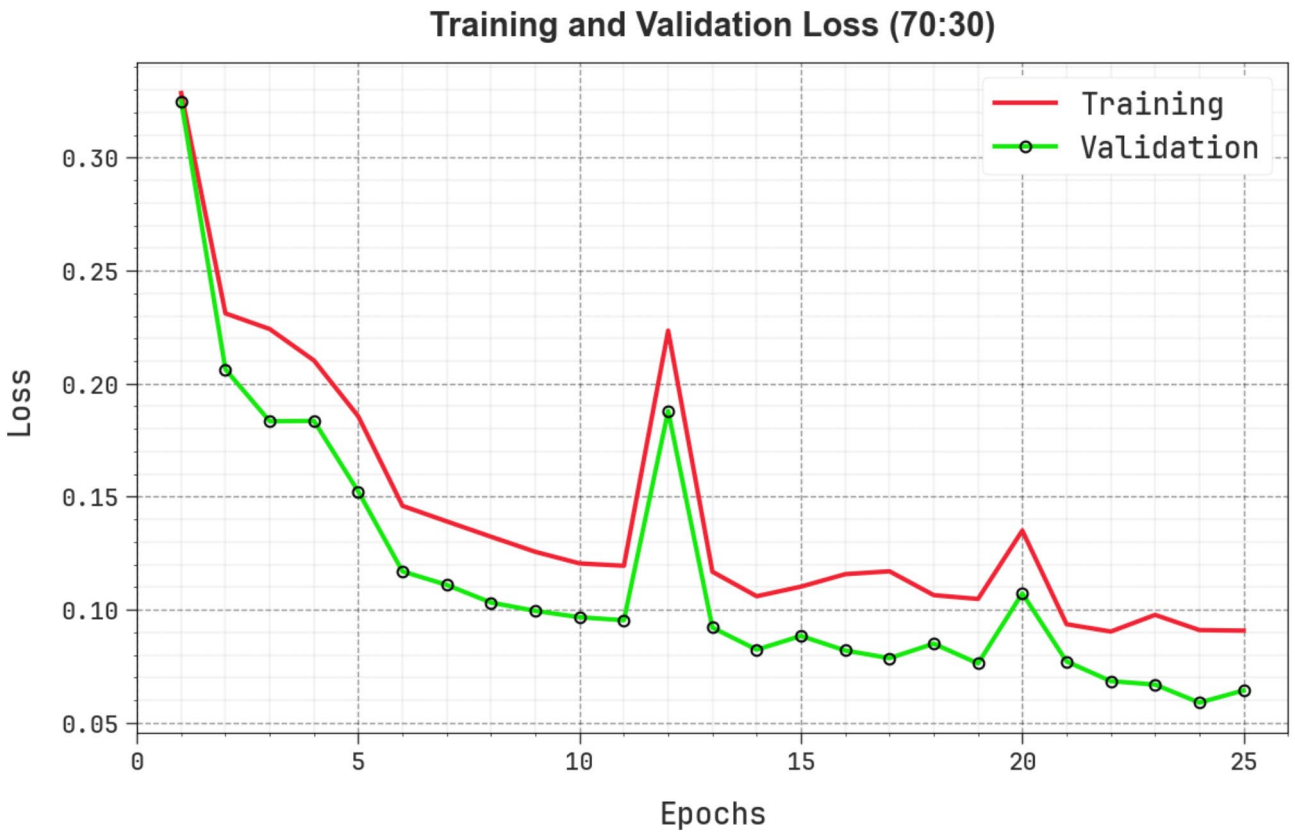

Fig. 8. Loss curve of MHARNN-EGTOCRD model under 70%TRPH and 30%TSPH.

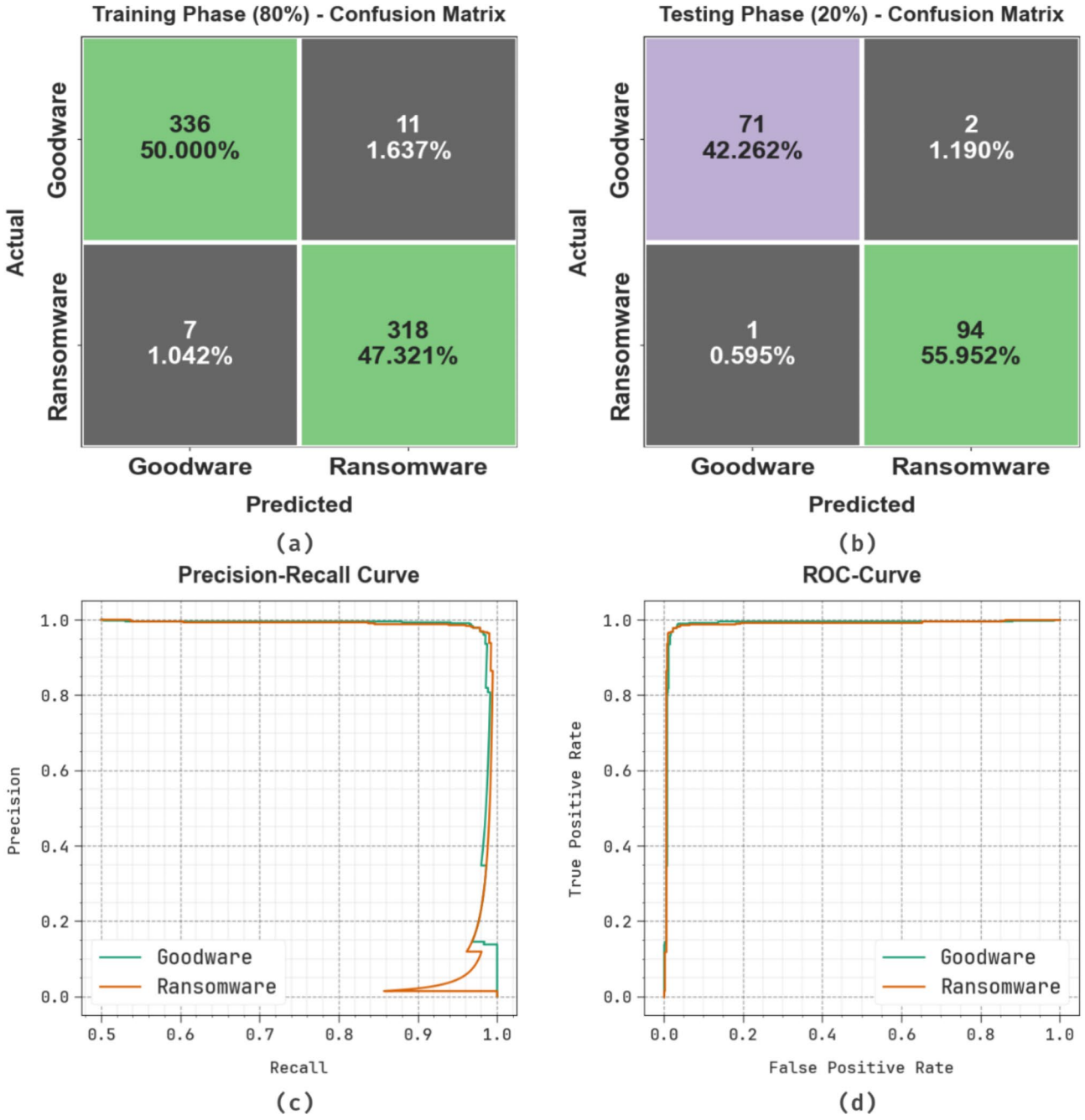

Fig. 9. (**a**,**b**) 80% and 20% confusion matrix and (**c**,**d**) curves of PR and ROC.

Classes	$Accu_y$	$Sens_y$	$Spec_y$	F_{score}	MCC
TRPH (80%)					
Goodware	96.83	96.83	97.85	97.39	94.65
Ransomware	97.85	97.85	96.83	97.25	94.65
Average	97.34	97.34	97.34	97.32	94.65
TSPH (20%)					
Goodware	97.26	97.26	98.95	97.93	96.37
Ransomware	98.95	98.95	97.26	98.43	96.37
Average	98.10	98.10	98.10	98.18	96.37

Table 3. Detection outcome of MHARNN-EGTOCRD model under 80%TRPH and 20%TSPH.

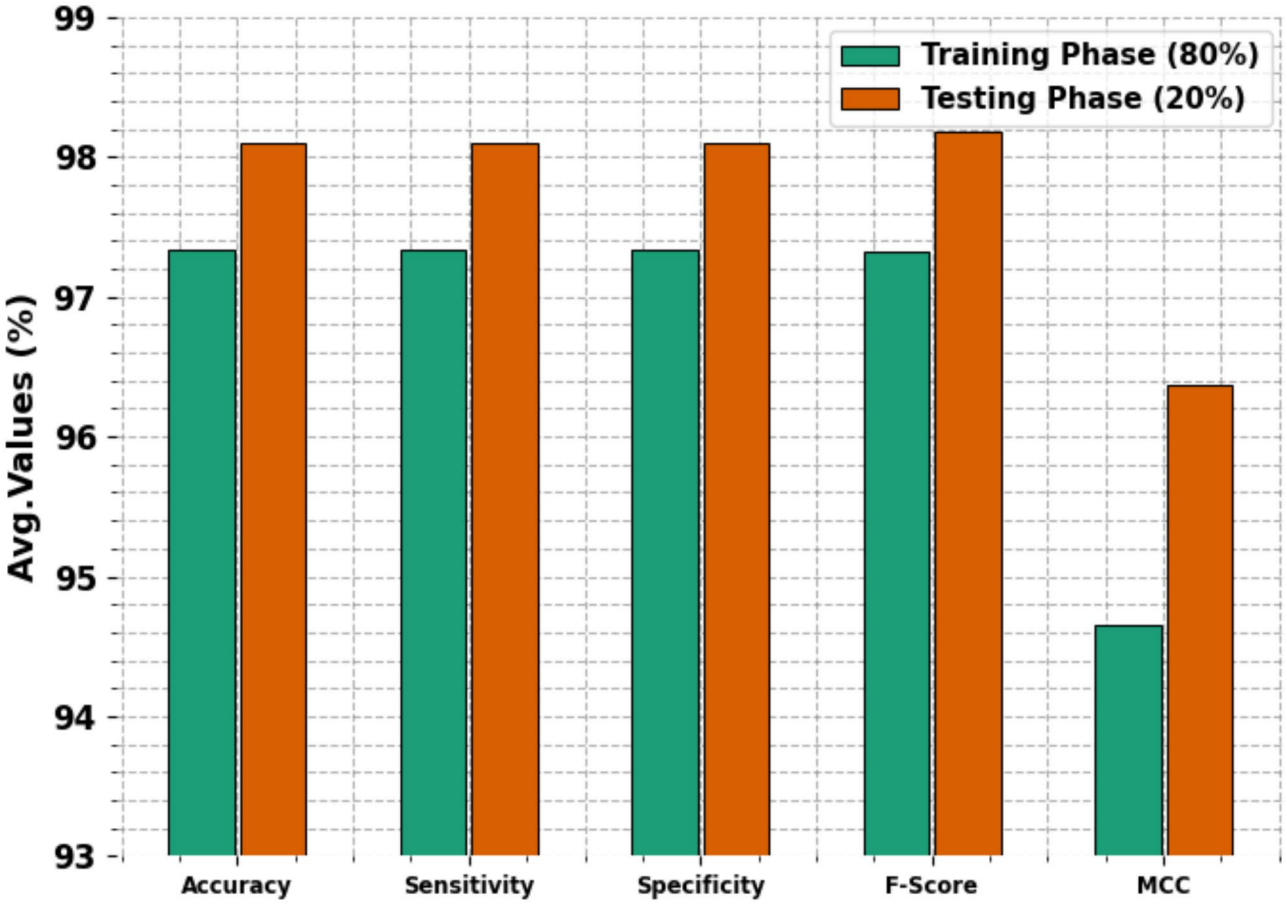

Fig. 10. Average of MHARNN-EGTOCRD model below 80%TRPH and 20%TSPH.

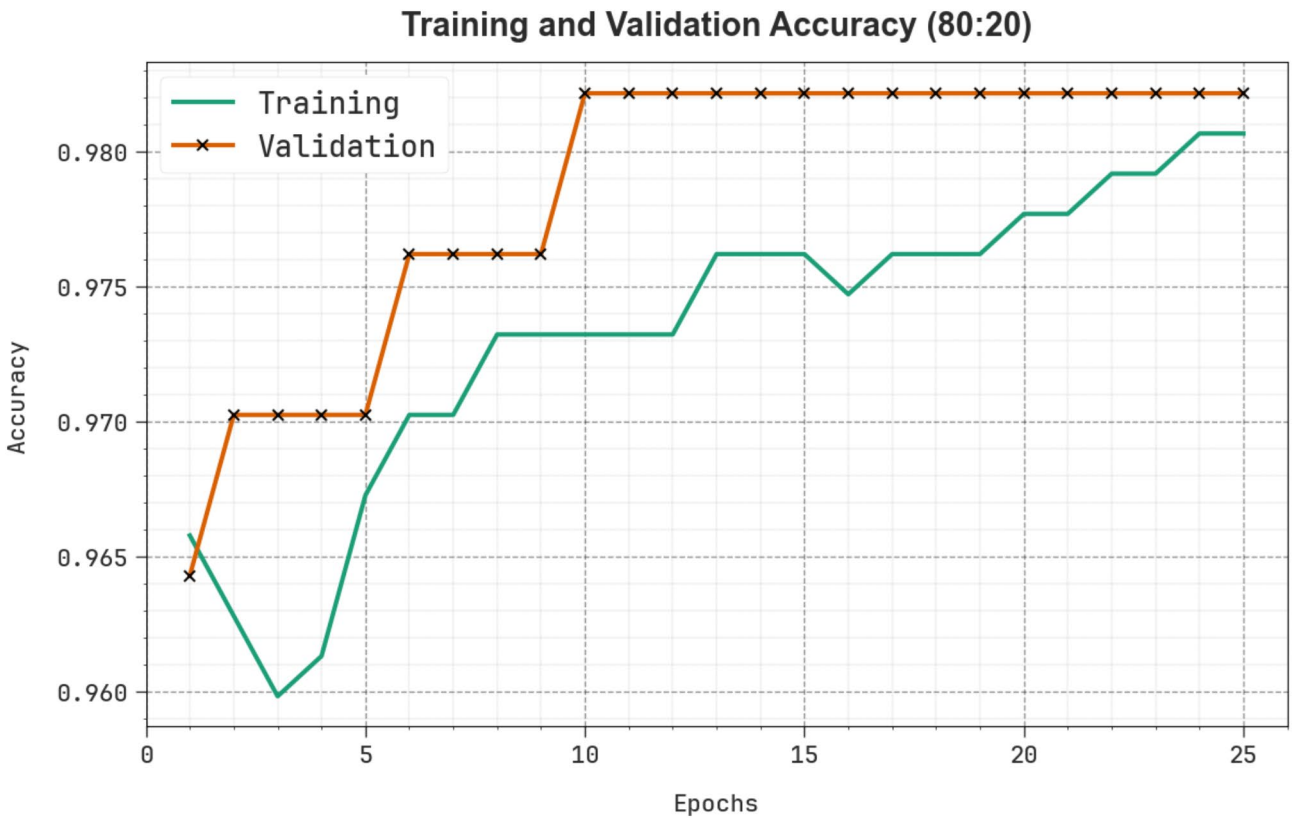

Fig. 11. $Accu_y$ curve of MHARNN-EGTOCRD method below 80%TRPH and 20%TSPH

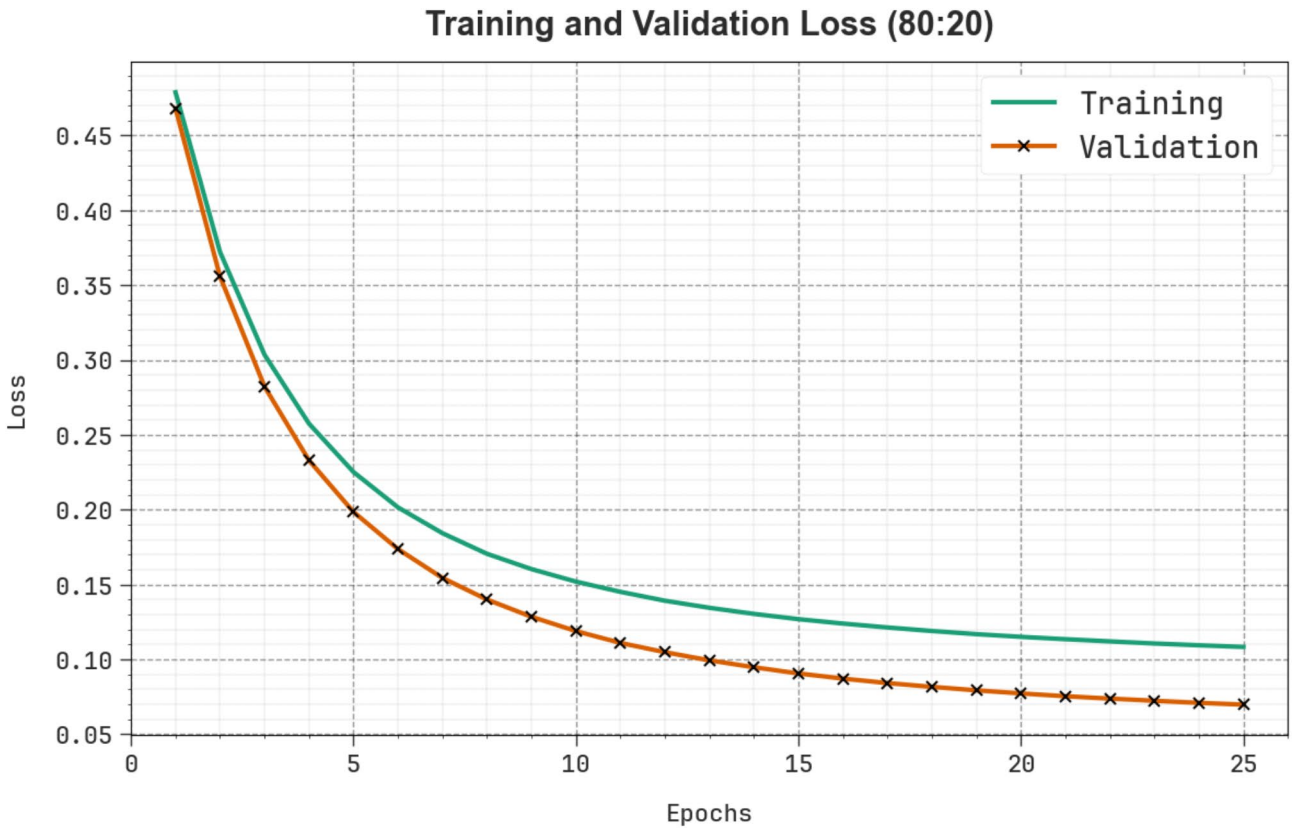

Fig. 12. Loss curve of MHARNN-EGTOCRD technique below 80%TRPH and 20%TSPH.

Methods	$Accu_y$	$Sens_y$	$Spec_y$
MHARNN-EGTOCRD	98.53	98.53	98.53
OGCNN-RWD	98.01	97.97	98.1
DWOML Method	97.33	97.92	97.39
Bagging Model	96.86	92.01	94.36
AdaBoost-M1Method	96.13	94.5	94.6
Rotation Forest (ROF) Model	95.79	96.77	97.38
DT Algorithm	97.63	97.82	98.12
RF Technique	97.25	97.03	96.54

Table 4. Comparative analysis of the MHARNN-EGTOCRD model with existing techniques.

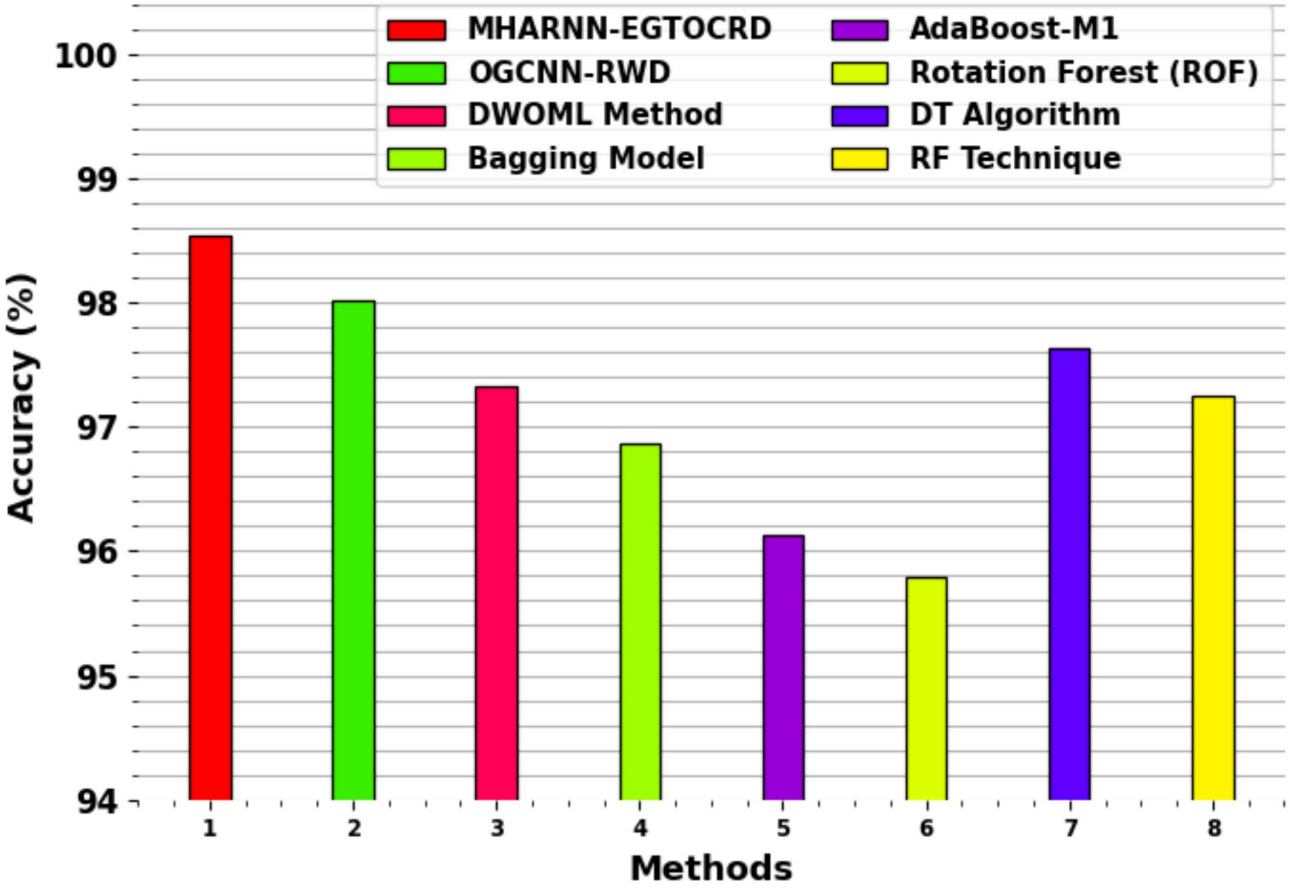

Fig. 13. $Accu_y$ analysis of MHARNN-EGTOCRD model with existing methods

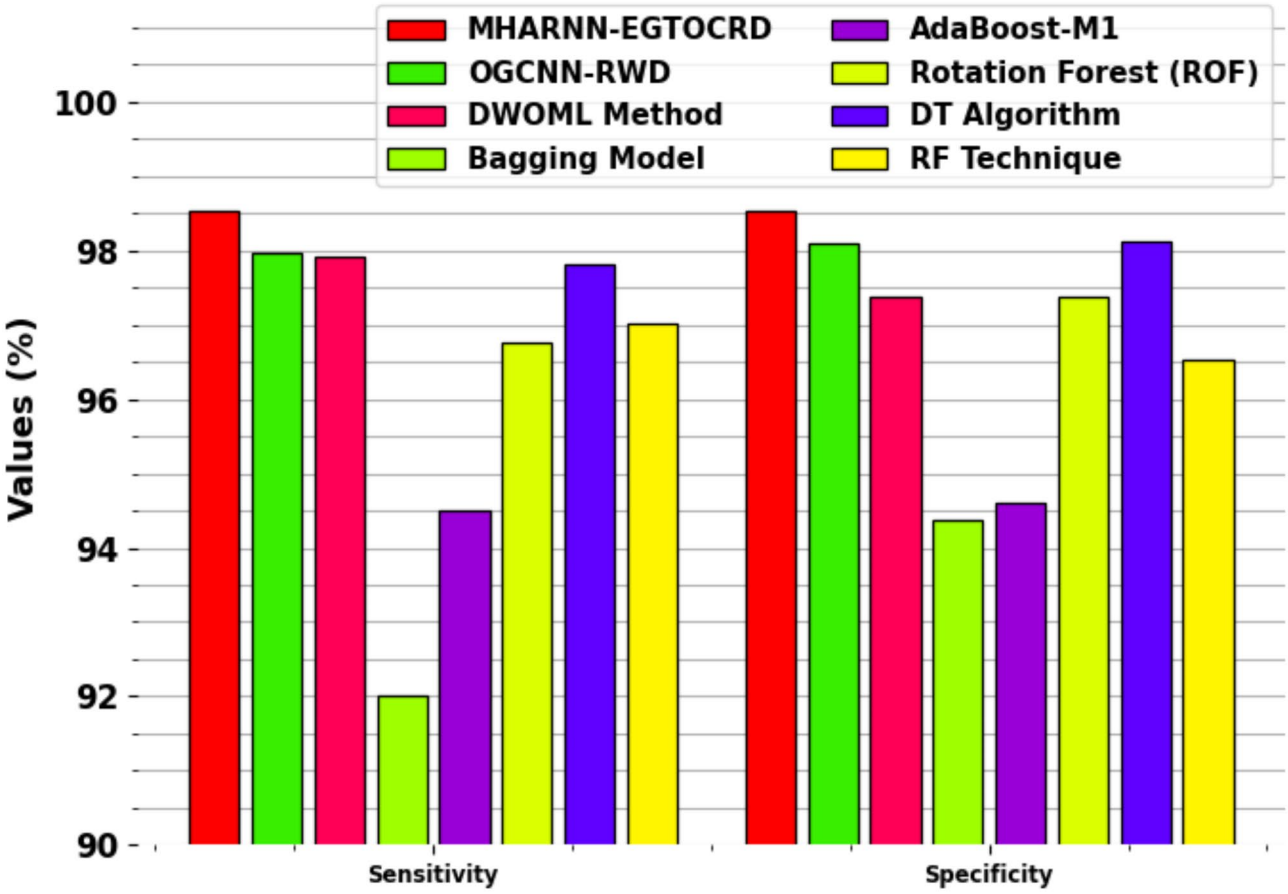

Fig. 14. $Sens_y$ and $Spec_y$ analysis of MHARNN-EGTOCRD model with existing methods

Data availability

The data that support the findings of this study are openly available in Kaggle repository at https://www.kaggle.com/datasets/amdj3dax/ransomware-detection-data-set, reference number[21].

Received: 1 February 2025; Accepted: 3 March 2025
Published online: 10 March 2025

References

1. Bae, S. I., Lee, G. B. & Im, E. G. Ransomware detection using machine learning algorithms. *Concurr Comput. Pract. Exp.* **31** e5422. (2020).
2. Rodriguez, E., Otero, B., Gutierrez, N. & Canal, R. A survey of deep learning techniques for cybersecurity in mobile networks. *IEEE Commun. Surv. Tutorials*. **23**(3), 1920–1955 (2021).
3. Sharma, S., Krishna, C. R. & Kumar, R. Androidransomware 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*, 1–6 (IEEE, 2020).
4. 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*, 1–6, 7. (IEEE, 2020).
5. Ankita, A. & Rani, S. July. Machine learning and deep learning for malware and ransomware attacks in 6G network. In *2021 Fourth International Conference on Computational intelligence and Communication Technologies (CCICT)*, 39–44 (IEEE, 2021).
6. Fernando, D. W., Komninos, N. & Chen, T. A study on the evolution of ransomware detection using machine learning and deep learning techniques. *IoT* **1**, 551–604 (2020).
7. 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.* **12**, 172 (2021).
8. Hijazi, A., Alhafez, N. & Al-khayat, I. An adaptive distributed intrusion detection system in local network: hybrid classification methods. *J. Intell. Syst. Internet Things*, **12**(1). (2024).
9. Dion, Y. & Brohi, S. N. An experimental study to evaluate the performance of machine learning alogrithms in ransomware detection. *J. Eng. Sci. Technol.* **15**, 967–981 (2020).
10. Mahindru, A. & Sangal, A. L. PARUDroid: validation of android malware detection dataset. *J. Cybersecur. Inform. Manag.*(2), 42 - 2 (2020).
11. Hurley, R., Kruger, P., Nascimento, H. & Keller, S. Real-time ransomware detection through adaptive behavior fingerprinting for improved cybersecurity resilience and defense. (2024).
12. Alamro, H. et al. Automated android malware detection using optimal ensemble learning approach for cybersecurity. *IEEE Access.* (2023).

13. Moritaka, H. & Komuro, D. Enhanced ransomware detection using dual-layer random forest on opcode sequences. (2024).
14. Li, G., Wang, S., Chen, Y., Zhou, J. & Zhao, Q. A hybrid framework for ransomware detection using deep learning and Monte Carlo tree search. (2024).
15. Albakri, A., Alhayan, F., Alturki, N., Ahamed, S. & Shamsudheen, S. Metaheuristics with deep learning model for cybersecurity and Android malware detection and classification. *Appl. Sci.*. **13**(4), 2172 (2023).
16. Khalid Alkahtani, H. et al. optimal graph convolutional neural network-based ransomware detection for cybersecurity in IoT environment. *Appli. Sci.* **13**(8), 5167. (2023).
17. Sumathi, S. & Rajesh, R. HybGBS: A hybrid neural network and grey Wolf optimizer for intrusion detection in a cloud computing environment. *Concurr. Comput. Pract. Exp.*. **36**(24), e8264 (2024).
18. Dhande, M. T., Tiwari, S. & Rathod, N. Design of an efficient malware prediction model using auto encoded & attention-based recurrent graph relationship analysis. *Int. Res. J. Multidiscip.Technov.*. **7**(1), 71–87 (2025).
19. Sokkalingam, S. & Ramakrishnan, R. An intelligent intrusion detection system for distributed denial of service attacks: A support vector machine with hybrid optimization algorithm based approach. *Concurrency Computation: Pract. Experience*. **34**(27), e7334 (2022).
20. Berguiga, A., Harchay, A. & Massaoudi, A. HIDS-IoMT: A deep Learning-Based intelligent intrusion detection system for the internet of medical things. *IEEE Access*. (2025).
21. Sumathi, S., Rajesh, R. & Lim, S. Recurrent and deep learning neural network models for DDoS attack detection. *J. Sens.* **2022**(1), 8530312 (2022).
22. Liu, W. et al. Attributing stealth cyberattacks via Temporal probabilistic graph neural networks. *J. Comput. Inform. Syst.* 1–15. (2025).
23. Sumathi, S., Rajesh, R. & Karthikeyan, N. DDoS attack detection using hybrid machine learning based IDS models. (2022).
24. Aldossary, M., Alzamil, I. & Almutairi, J. Enhanced intrusion detection in drone networks: a cross-layer convolutional attention approach for drone-to-drone and drone-to-base station communications. *Drones*. **9**(1), 46 (2025).
25. Sumathi, S. & Rajesh, R. A dynamic BPN-MLP neural network DDoS detection model using hybrid swarm intelligent framework. *Indian J. Sci. Technol.* **16**(43), 3890–3904 (2023).
26. Hwang, Y., Kurt, F., Curebal, F., Keskin, O. & Subasi, A. Contextualgraph-Llm: A Multimodal Framework for Enhanced Darknet Traffic Analysis. Available at SSRN 5099415.
27. Anargya, M. A. N., Ghozi, W. & Rafrastara, F. A. Random under sampling for performance improvement in attack detection on internet of vehicles using machine learning. *Jurnal Informatika: Jurnal Pengembangan IT*. **10**(1), 11–19 (2025).
28. Yang, H., Hu, S., Li, B., Gao, X. & Huang, H. Research on trajectory tracking of robotic fish based on DBO-backstepping control. *J. Mar. Sci. Eng.* **12**(12), 2364. (2024).
29. Ma, S. et al. Data-Driven Charging Load Prediction Based on Multi-Attention Mechanism and Long Short-Term Memory Networks of Electric Vehicles for Microgrid. Available at SSRN 5078734.
30. Li, F., Li, J. & Abza, F. Sentiment analysis of tweets employing convolutional neural network optimized by enhanced gorilla troops optimization algorithm. *Sci. Rep.* **15**(1), 795. (2025).
31. http://www.kaggle.com/datasets/amdj3dax/ransomware-detection-data-set
32. Alzahrani, I. R. & Allafi, R. Integrating Ebola optimization search algorithm for enhanced deep learning-based ransomware detection in internet of things security. *AIMS Math.* **9**(3), 6784–6802 (2024).

Acknowledgements

The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through Large Research Project under grant number RGP2/51/45. Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2025R716), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. Researchers Supporting Project number (RSP2025R459), King Saud University, Riyadh, Saudi Arabia. The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA for funding this research work through the project number "NBU-FFR-2025- 2913-01. The authors are thankful to the Deanship of Graduate Studies and Scientific Research at University of Bisha for supporting this work through the Fast-Track Research Support Program.

Author contributions

Sarah A. Alzakari: Conceptualization, methodology development, experiment, formal analysis, investigation, writing. Mohammed Aljebreen: Formal analysis, investigation, validation, visualization, writing. Nazir Ahmad: Formal analysis, review and editing. Sultan Alahmari : Methodology, investigation. Othman Alrusaini: Review and editing.Ali M. Al-Sharafi: Discussion, review and editing. Wafa Sulaiman Almukadi: Discussion, review and editing. Asma A. Alhashmi: Conceptualization, methodology development, investigation, supervision, review and editing.All authors have read and agreed to the published version of the manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to A.A.A.