Enhancing Cybersecurity through Ransomware Detection using Hybridization of Heuristic Feature Selection with Deep Representation Learning Model

Maha Farouk Sabir[1,*]

[1]Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

Email: msaber@kau.edu.sa

Abstract

Network security has become vulnerable to hacker threats owing to its advancement and easily accessible to computer and internet technology. Ransomware is the most commonly used malware in cyberattacks to mislead the victim user to expose private and sensitive data to hackers. Ransomware is malicious software that encodes the entire system or consumer's files, creating it impossible, and later demands a payment fee from the victim's computer in exchange for the decryption key. Ransomware attacks become highly popular and overwhelming for both individuals and organizations. Recently, deep learning (DL) and machine learning (ML) models are established to identify ransomware attacks in real-time and categorize them into various types. The system will be considered to examine the behaviors of malicious software and detect the particular kind of ransomware being utilized. This data will enhance the system's accuracy and deliver appropriate data to cybersecurity professionals and victims. Therefore, this study proposes an accurate Ransomware Detection and classification using the Hybrid Metaheuristic Feature Selection with Deep Learning (RDC-HMFSDL) technique. The aim is in effectually detecting and classifying the ransomware attacks. Initially, the RDC-HMFSDL technique utilizes min-max model to transform the input data into a standard setting. Furthermore, the hybrid red deer sparrow search optimization (HRDSO) approach is used for the feature selection (FS). For ransomware attack detection, the long short-term memory autoencoder (LSTM-AE) approach is employed. Finally, the sine cosine algorithm (SCA) is used to optimally choose the parameter values of the LSTM-AE approach. The RDC-HMFSDL approach was tested on a benchmark dataset, achieving a superior accuracy of 99.88% compared to existing methods.

Keywords: Ransomware Detection; Metaheuristic; Cybersecurity; Deep Learning; Sine Cosine Algorithm

1. Introduction

Accessibility of encoded information kept in the cloud and safeguarding privacy have become a major issue because of the development of cloud services and the constantly improving amount of online information [1]. Recently, cyberattacks like ransomware attacks have become more popular and critical: "malicious software sneaks onto the computer which encodes significant files, and after requires a ransom for decrypting them". Such attacks on organizations, persons, and even government organizations have caused serious data breaches and financial losses. The two main techniques for identifying ransomware—signature-based algorithms and rule-based methods are not capable of maintaining the quickly developing feature of ransomware attacks [2]. Frequently upgrading the signatures and rules to recognize novel modifications may reduce response time and exit systems susceptible to new attacks. The encrypted feature of cloud information confines analysis and accessibility; this creates hard-to-identity ransomware attacks employing traditional methodologies. With the continuous development of ransomware attacks, there is a crucial requirement for medical specialists and researchers to explore deeper into the difficulties and recognize efficient approaches for mitigation and prevention. Several

DOI: https://doi.org/10.54216/JCIM.170204
Received: February 05, 2025 Revised: May 26, 2025 Accepted: July 02, 2025

researchers have introduced solutions for detecting ransomware [3]. Generally, antivirus systems refer to signature-based files to a predetermined list of malware signatures. Alternatively, these models cannot be efficient against zero-day malware detection. Different dynamic and static analysis-based identification method was introduced in this manuscript to tackle the limitations of signature-based recognition. The static analysis-based recognition method evaluates an inbound file with no implementation [4].

These systems are faster but fail to detect polymorphic attacks. In contrast, dynamic analysis examines files in a virtual environment. Conversely, it takes a more time-consuming method to perform every inbound file on a virtual platform [5]. Alternative techniques will be analysed only in ad hoc measures to save time. Specific occurrences-based techniques cannot be appropriated as they cannot be activated or can occur while the process is permanent. Accordingly, these approaches endure a higher false alarm rate (FAR) and reduced recognition rate [6]. Intrusion detection system (IDS) refers to a highly sensitive to identify susceptible malware. For example, ransomware is the major vulnerable malware. The expansion of the DL ensemble model for ransomware detection represents an alternative significant step forward [7]. With a huge amount of DL methods for collecting various features, the ensemble model increases robustness and detection efficiency [8]. This innovative method will support the development of superior real-time detection techniques for ransomware [9]. A recent study trend is normally to employ DL abilities for generalizing methods and automated feature extractor from raw information. Owing to such distinct features, the DL has demonstrated its efficiency in numerous domains like recommendation systems, image recognition, speech recognition, and sentiment analysis [10]. Existent DL-based detection models were dependent upon the assumption that basic data allocation of the latent type of attack must be the same as the seen type of attack.

This study proposes an accurate Ransomware Detection and classification using the Hybrid Metaheuristic Feature Selection with Deep Learning (RDC-HMFSDL) technique. The aim is in effectually detecting and classifying the ransomware attacks. Initially, the RDC-HMFSDL technique utilizes min-max model to transform the input data into a standard setting. Furthermore, the hybrid red deer sparrow search optimization (HRDSO) approach is used for the feature selection (FS). For ransomware attack detection, the long short-term memory autoencoder (LSTM-AE) approach is employed. Finally, the sine cosine algorithm (SCA) is used to optimally choose the parameter values of the LSTM-AE approach. The RDC-HMFSDL approach was tested on a benchmark dataset

2. Related Works

Al Duhayyim et al. [11] introduced an Artificial Algae Optimizer Algorithm with Optimal Deep Belief Network (AAA-ODBN) model. Initially, the AAA-FS approach is employed to choose features sub-categorises. Next, DBN architecture is utilized for ransomware recognition. In [12], a cost-sensitive pareto ensemble system (CSPE-R) is introduced. Primarily, this model uses an unsupervised deep CAE approach. Heterogeneous base estimators were also trained through these removed subspaces. Subsequently, an innovative Pareto Ensemble-based estimator selection approach was employed for accomplishing cost-sensitive cooperation among false negatives and false positives. Lastly, the result of chosen estimators must be combined to increase the identification. Lu et al. [13] developed an effective malware detection architecture dependent upon the deep neural networks (DNN) named DLAMD and deals with large-scale instances. A hybrid identification model integrates quick pre-screening with deep analysis, using random forest (RF) for key feature selection and CNN to extract hidden patterns from APK files for accurate detection. Jemal and Lo [14] projected a multi-variant classification model. The DL methods applied in the developed technique is CNN and BiLSTM. The DL methods could be compared against a traditional ML technique like RF, SVM, and LR.

In [15], a Siamese neural network (SNN) model is presented. Such entropy features are utilized more for training and optimizing this architecture utilizing a pre-trained network (for example, VGG16) in a meta-learning manner. This technique produces highly correct weight factors, related to feature images that is employed. Lavanya and Sekhar [16] implement an innovative technique called Wavelet Deep CNN (WDCNN) for categorizing cyber-attacks. This technique was developed for identifying attacks in large-scale data as well as decreasing the difficulties of identification with the highest detection accuracy. The developed technique was employed in Python. In [17], an innovative DL approach employing the BC network is proposed. The sequence-based statistical feature extractor was executed, and further, the TF-IDF was determined for every feature. Similarly, the feature fusion should be developed with the help of a fractional notion. Lastly, the categorization of Ransomware was achieved by employing Deep stacked-AE (Deep-SAE) in which the developed Water wave-based Moth Flame optimizer (WMFO) was modified to produce the optimum weights. Kavitha et al. [18] presented an Intelligent Cybersecurity Classification utilizing a Chaos Game Optimizer with DL (ICC-CGODL) method. The model initially utilizes min-max for normalizing the information. Then, the Bidirectional Gated Recurrent Unit (BiGRU) approach was employed to identify and categorize cyberattacks. Besides, the CGO method was further employed for adopting the hyperparameters associated with the BiGRU technique.

DOI: https://doi.org/10.54216/JCIM.170204
Received: February 05, 2025 Revised: May 26, 2025 Accepted: July 02, 2025

3. The Proposed Method

The paper presents the RDC-HMFSDL model. The RDC-HMFSDL model intends to distinguish and classify the occurrence of ransomware attacks accurately. To accomplish this, the RDC-HMFSDL method incorporates min-max data normalization, HRDSO-based FS, LSTM-AE-based classification, and SCA-based tuning. Fig. 1 describes the flow of the RDC-HMFSDL method.

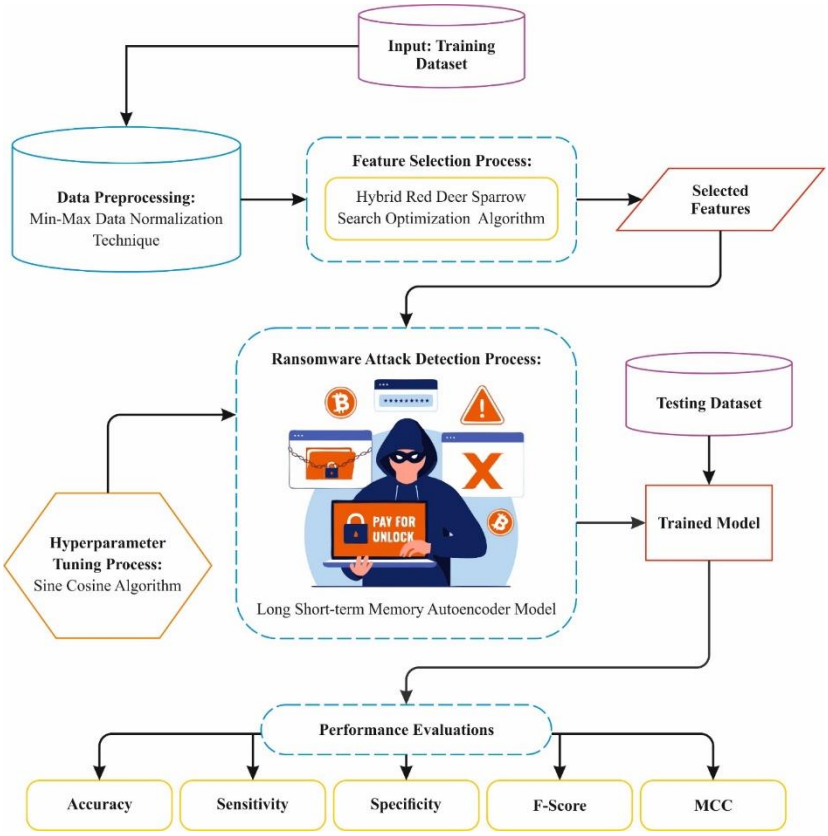

Figure 1. Workflow of RDC-HMFSDL technique

A. Pre-processing

Initially, the min-max data normalization is utilized to convert the input data into a uniform format. Min-max normalization, otherwise called feature scaling, is a pre-processing approach that converts arithmetical data into a normalized range [0,1]. The objective is to confirm that each feature equally contributes to the modeling processes and analysis [19], which prevents the dominance of specific features based on the original scale. In min-max normalization, the data point x_i in features is transformed by Eq. (1):

$$x_i' = \frac{x_i - \min(X)}{\max(X) - \min(X)} \tag{1}$$

Where x_i' refers to the normalization value, x_i specifies the original value, and $\min(X)$ and $\max(X)$ are the higher and lesser outcomes in the feature. This process is used to scale the values proportionally in [0,1], which maintains a similar relationship between data points.

B. HRDSO-based FS

The RDC-HMFSDL technique employs the HRDSO model for the FS process. HRDSO model is a novel hybrid optimizer method that integrates the red deer algorithm (RDA) and sparrow search algorithm (SSA), which is used to recognize the useful features sub-set for the classification of BC [20]. SSA was stimulated by the sparrow's behavior of foraging and highlights exploration, whereas RDA is dependent upon the red deer behavior of herding, and concentrates on exploitation. By uniting these dual processes, the hybrid technique can attack stability among exploitation and exploration, permitting effectual search through the performance space. The hybrid optimizer termed HRDSO which contains the subsequent steps,

DOI: https://doi.org/10.54216/JCIM.170204
Received: February 05, 2025 Revised: May 26, 2025 Accepted: July 02, 2025

Step1: Initialize.

Thus, the below-mentioned matrix is employed to portray the search agent's position utilizing Eq. (2):

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N\times d} = \begin{bmatrix} X_{1,1} & \cdots & X_{1,\mathrm{j}} & \cdots & X_{1,\mathrm{d}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ X_{i,1} & \cdots & X_{i,\mathrm{j}} & \cdots & X_{i,1} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ X_{N,1} & \cdots & X_{N,\mathrm{j}} & \cdots & X_{N,\mathrm{d}} \end{bmatrix}_{N\times d} \quad (2)$$

Whereas n depicts the search agent count, and d portrays the variable size that wants to be adjusted. Tent chaotic (TC) sequence is utilized to modify the search agent population.

Step2: Population upgrading utilizing TC sequence.

In deterministic non-linear methods, chaos is a different occurrence that rises without extra arbitrary variables. The variables of chaos always seek and navigate the entire space and disclose rules as an outcome of their characteristic randomness. To avoid being caught in local optima, chaos can often be used to enhance searching problems. A typical chaotic model contains a logistic map. It is conveyed in Eq. (3) as follows:

$$x_{i+1} = \begin{cases} 2x_i, 0 \le x \le 1/2 \\ 2\,(1-x_i), 21 < x \le 1 \end{cases} \quad (3)$$

The Eq. (4) is the TC map that follows the Bernoulli shift transformation:

$$x = (2x) mod 1 \quad (4)$$

The TC sequence is initiated to take unbalanced periodic covers and transient breaks, averting it from being a temporary or unbalanced stage fact. The enhanced formulation comprises the random variable rand $(0,1) * 1/NA$ to preliminary Tent mapping expression:

$$x_{i+1} = \begin{cases} 2x_i + rand(0,1) * 1/NA, 0 \le x \le 1/2 \\ 2(1-x_i) + rand(0,1) * 1/NA, \frac{1}{2} < x \le 1 \end{cases} \quad (5)$$

The Eq. (6) follows the Bernoulli transformation,

$$x_{i+1} = (2xi) mod 1 + rand(0,1) * \frac{1}{NA} \quad (6)$$

During this situation, NA denotes chaotic sequence particle count, and $rand\ (0,1)$ refers to the arbitrary number selected at a range value of [0 and 1]. Once producing a chaotic sequence in the appropriate area utilizing the improved chaotic map expression.

During the range value of (0 and 1), generate the initial value x_0 randomly by the index equivalent to 0.

Initiate an iterative calculation utilizing Eq. (6) to create the χ sequence, with i enhancing by one each time.

Once the selected iteration amount is finished, stop securing the produced x sequence.

Step3: Fitness calculation.

During the HRDSO, the objective function (OF) signifies the fitness performance. Fitness computation includes estimating the OF for every better result. It offers an assessable measure of how well a specific solution fulfils the optimizer objectives. It aids in differentiating among better and worse solutions. The fitness calculation is executed utilizing Eq. (7).

$$Fit = \min(E) \quad (7)$$

Whereas E denotes the value of error. For fitness function (FF) calculation, the worst and best fitness values are intended.

Step4: Position upgrading.

Male RD (MRD) is busy during this stage to boost their grace. While it happens naturally, the roaring procedure may succeed or fail. MRDs are better results for this method. It excels at discovering the adjacent performances and replacing them with high-OFs from preceding iterations. Each MRD is permitted to shift locations. The discoverer (male), who is on duty to discover the food, invented the 1st five better results, but the participants (female) are the next 5 optimum performers and are assumed to be the protectors. This method is upgraded by utilizing Eq. (8):

DOI: https://doi.org/10.54216/JCIM.170204
Received: February 05, 2025 Revised: May 26, 2025 Accepted: July 02, 2025

$$m^{t+1} = \begin{cases} m^t + a_1 * ((Ub - Lb) * a_2 + Lb, & if a_3 \geq 0.5 \\ m^t - a_1 * ((Ub - Lb) * a_2 + Lb, & if a_3 < 0.5 \end{cases} \quad (8)$$

where m^{t+1} denotes the upgraded male location. Each female search agent upgrade (f^{t+1}) their positions in unity with the formulation in Eq. (9) as follows:

$$f^{t+1} = \begin{cases} Q.exp\left(\frac{f_{worst}^t - f^t}{i^2}\right), if\ i > n/2 \\ f_{best}^{t+1} + |f^t - f_{best}^{t+1}|.A^t.L, Otherwise \end{cases} \quad (9)$$

where f_{worst}^t signifies the present worst location, f_{besi}^{t+1} implies the next better result for the discovery from the present state, A denotes the matrix with size $1 \times d$.

Step5: Fighting among male pioneers.

Each leader has specified an arbitrary pair of stags to fight with. To improve efficiency, the commander is to be swapped with an extra optimum result. This includes measuring 4 selections such as the stag, commander, and dual newly attained results. It attains binary novel selections and substitutes the commander with the optimum result. Dual expressions in mathematics is presented to determine the fighting procedure in Eq. (10):

$$New\ 1 = \frac{(m^{t+1} + f^{t+1})}{2} + b_1 \times \left((Ub - Lb) * b_2\right) + Lb \quad (10)$$

$$New\ 2 = \frac{(m^{t+1} + f^{t+1})}{2} - b_1 \times \left((Ub - Lb) * b_2\right) + Lb \quad (11)$$

where New 1 and New 2 signify the dual new solutions produced by the battle procedure. The Com and $Stag$ words imply the commanders and stags, correspondingly. Ub and Lb represents the upper and lower limits, separately. These boundaries limit the feasibility of novel results. Owing to the arbitrariness of the battle procedure, b_1 and b_2 are formed by an even distribution function amid 0 and 1. Only the finest choice with respect to OF is selected among 4 options, such as m^{t+1}, f^{t+1}, New 1, and New 2.

Step6: Guards upgrade.

Once danger is known, the searching agent group can use anti-predation performances and its position can upgraded in Eq. (12) as:

$$f^{t+1} = \begin{cases} f_{best}^t + \beta.|f^t - f_{best}^t|\ if\ Fit_i > Fit_g \\ f^{t+1} + K.\left(\frac{f^t - f_{worst}^t}{(Fit_i - Fit_w) + e}\right)\ if\ Fit_i = Fit_g \end{cases} \quad (12)$$

From the above-mentioned expression, the length of step directing parameter is signified by the β, where the mean value is equivalent to 0 and the variance value of 1. $K \in [-1,1]$ is an arbitrary value. To avoid 0 denomination, e is fixed to a minimum constant; Pit_i implies the fitness value; Fit_g and Pit_w are said to be search agents. When $Fit_i > Fit_g$; it denotes that the middle-class searching agents are alert to the danger offered by the hunters and function to decrease their danger of being consumed.

Step7: Mating.

Mutation, Crossover, and gender grouping take place at the time of mating in LA. Male and female generate cubs over this mutation and crossover procedure that helps to remove male and female essential portions. Estimation of fertility has helped in the changeover far from the local optimum performance. Once the upgraded female is signified as f^{t+1}, the mating procedure develops according to the improvement. Once there is no f^{t+1} to substitute the female, f^t is supposed that productive to generate better cubs through the upgrade process.

$$f^{t+1}(k) = \begin{cases} f^{t+1}, if\ l = k \\ f^t, Otherwise \end{cases} \quad (13)$$

$$f^{t+1}(l) = \min\left[f_k^{\max}, \max\left(f_k^{\min}, \nabla_k\right)\right] \quad (14)$$

$$\nabla_k = [f^t + (0.1r_2 - 0.05)(m^t - r1f^t)] \quad (15)$$

where, l^{th} and k^{th} denote the vector elements of $f^{t+1}(l)$ and $f^{t+1}(k)$, correspondingly. The arbitrary number in [0 and 1] is represented as r_1 and r_2, the female upgrade location is presented as r, and [1, L] is denoted as r_1 and r_2.

Step8: Protect the so far attained solution.

DOI: https://doi.org/10.54216/JCIM.170204
Received: February 05, 2025 Revised: May 26, 2025 Accepted: July 02, 2025

Step9: Return the optimum solution.

The FF considers the classification performance and the chosen features. It decreases the fixed size of chosen features and enhances the accuracy. Accordingly, the FF is utilized for assessing each solution.

$$Fitness = \alpha * ErrorRate + (1 - \alpha) * \frac{\#SF}{\#All\_F} \quad (16)$$

In Eq. (16), $ErrorRate$ depicts the classifier error rate and is evaluated as amount of incorrect classification to the amount of classification made, considered to be 0 and 1. ($ErrorRate$ denotes the complement of classifier accuracy). α controls the importance of classification quality, subset length and the value of α is fixed as 0.9.$\#SF$ indicates the number of chosen features and $\#All\_F$ shows the entire feature counts in the new data.

C. Attack detection using LSTM-AE Technique

For ransomware attack detection, the LSTM-AE is applied to detect and classify ransomware attacks. Recently, DL is attained substantial progressions in processing non-linear and higher-dimension data [21]. By creating multi-layer neural networks (NNs), DL could separately remove high-order aspects from data and identify intricate patterns. This capability make it to carry out in complicated tasks. Nevertheless, in time-series study, a major challenge for DL is effectually acquiring longer-term dependency in data. The LSTM developed as an effective solution. The LSTM network is a NN framework intended to process longer sequence data, tackling the problem of gradient explosion or vanishing in conventional RNN. Within the LSTM, internal recurring loop making it to recollect prior data and determine temporal dependency among consecutive data points. The cell state is an essential variable which allows transferring data from previous steps through the overall system. The procedure contains 3 key steps: input, output and forget that is recognized by 3 gates. Particularly, the gate framework of LSTM contains input, output and forget gates.

1. The input, output, and forget gates i_t, o_t, and f_t are computed using the current input x_t and the previous output h_{t-1}, with a sigmoid activation function that outputs values in [0, 1]. The differences among these gates are in their respective weight and bias parameters.

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \quad (17)$$

$$f_t = sigmoid\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \quad (18)$$

$$i_t = sigmoid(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (19)$$

$$o_t = sigmoid(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (20)$$

Now, h_{t-1} and x_t indicates the output and input at moment t-1 and t and W_f, b_f, W_i, b_i, W_o, and b_o refers to input weights and biases of forgetting, inputting, and outputting gates, correspondingly.

2. c_t depicts unit cell state. It permits the LSTM technique for learning the longer-term dependency well.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (21)$$

$$c_t = f_t * c_{t-1} + i_t^* \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (22)$$

Now f_t indicates the output of forget gate, b_c and W_c refers to bias and weight values and c_{t-1} signifies the unit cell state at t-1, it is the outputs of inputting gate.

3. h_t refers to final output of LSTM method at time step t.

$$h_t = \tanh(c_t) * o_t \quad (23)$$

Now o_t and c_t depicts the output gate and unit cell output at time t.

AE is broadly employed unsupervised learning model intended to accomplish reducing dimensionalities and extracting features of data by learning an effective encode of unlabelled data. It comprises encoding and decoding. The objective of encoding is modifying the input data x into a lower-dimension depiction. The aim is to map the lower-dimension representations back to unique state and rebuilds data of input. This kind of methodology improves performance by altering hidden layers in several manners. While NN is deeper, the vanishing gradient concern is tackled by stacking hidden layers. This model employs LSTM components for constructing the encoding and decoding, therefore allowing the decoder and encoder for processing time-dependent data or time-series data.

DOI: https://doi.org/10.54216/JCIM.170204
Received: February 05, 2025 Revised: May 26, 2025 Accepted: July 02, 2025

D. SCA-based Hyperparameter Tuning

Finally, the SCA is used for optimally choosing the parameter values of LSTM-AE. SCA utilizes the mathematical model of sine and cosine functions and randomly creates several candidates results to update the location in the search space [22]. In it, P_i^t is the present optimum individual at $t^{he\ tth}$ iteration, X_i^t refers to the location in the i^{th} parameter at t^{th} iteration, according to Eq. (24), the computation process is performed:

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \times \sin(r_2) \times |r_3 P_i^t - X_i^t|, r_4 < 0.5 \\ X_i^t + r_1 \times \cos(r_2) \times |r_3 P_i^t - X_i^t|, r_4 \geq 0.5 \end{cases} \quad (24)$$

Where $r_1 = a\left(1 - \frac{t}{T}\right)$, the search step size r_1 controls the search procedure, a is fixed to 2. t and T are the existing and maximum iteration counter. $r2 \in [0,2\pi], r3 \in [0,2], r4 \in [0,1]$, r_2, r_3 and r_4 are arbitrary parameters.

In SCA, with the increase of t, r_1 decreased from 2 to 0. Moreover, r_2 refers to a random integer within the range of $[0,2\pi]$. It is easy to see that if $T/2 \leq t \leq T$, then $r1$ decreased from 1 to 0 and $r_1 \sin(r_2)$ sine function value or $r_1 \cos(r_2)$ cosine function value is in [1,1]. If $0 \leq t \leq T/2$, then $r1$ decreased from 2 to 1 and $r_1 \sin(r_2)$ value is within [1,2] or $r_1 \cos(r)$ value is within $[-2,-1]$. According to Eq. (24), the position updating of the solution changes dramatically, which is widely explored to search for further possible solutions. Now, these fluctuations of the performance are very light, and the result can easily fall in local optima. Fig. 2 depicts the SCA architecture.

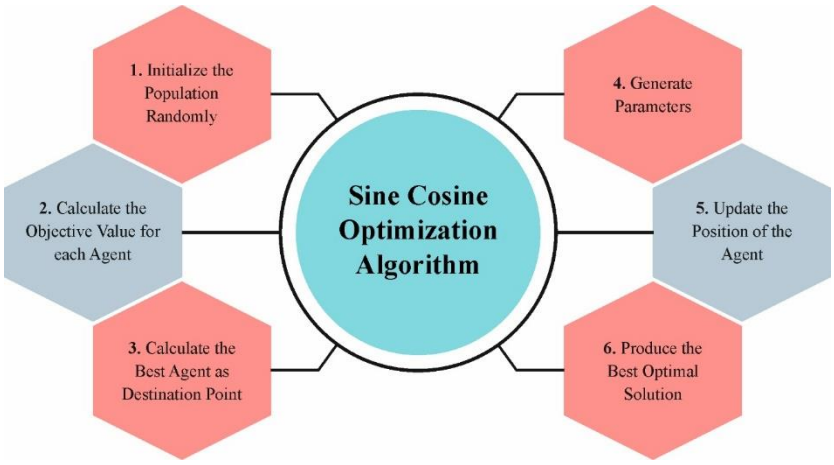

Figure 2. SCA structure

In SCA, with the increase of iterations t, the r_1 step size factor linearly reduces to 0. The update change of solution is slight at the next iteration, and the possible solution is found in the local range.

$$r1 = a \sin\left(\frac{\pi}{2}\left(1 - \frac{t}{T}\right)\right) + b \; a = 2, b = 0.5 \quad (25)$$

It is not difficult to see that the r_1 value is within [0.5, 2.5] from and the $r_1 \sin(r_2)$ value or $r_1 \cos(r_2)$ value is within $[-2.5,2.5]$. The solution range is wide. Once the search step size factor r_1 reduces, then it is delayed linear reduction that is also contributing to global improvement.

The fitness function efficiently determines the SCA method, with hyperparameter selection using the encrypted result to assess solution efficiency. The SCA methodology defines accuracy as a major condition for developing the FF.

$$Fitness = \max(P) \quad (26)$$

$$P = \frac{TP}{TP + FP} \quad (27)$$

Where, TP and FP indicates the true and false positive values.

DOI: https://doi.org/10.54216/JCIM.170204
Received: February 05, 2025 Revised: May 26, 2025 Accepted: July 02, 2025

4. Result analysis

The simulation results of the RDC-HMFSDL model is tested under a benchmark dataset [23] containing 840 samples and dual classes as definite in Table 1.

Table 1: Dataset specification

Classes	Instance Numbers
Goodware (GW)	420
Ransomware (RW)	420
Overall Instances	840

Fig. 3 exhibits the confusion matrices of the RDC-HMFSDL approach with diverse epochs. These accomplished outcomes specify the effective good ware and ransomware sample recognition with each class.

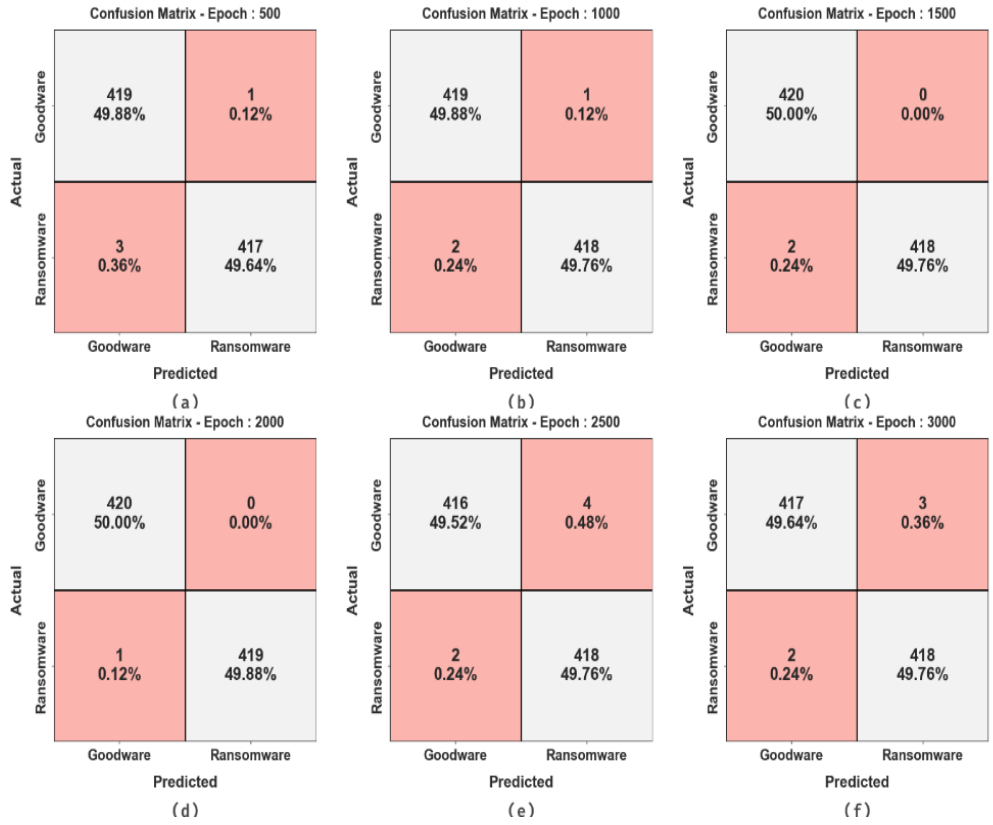

Figure 3. Confusion matrices of RDC-HMFSDL technique (a-f) Epochs 500-3000

The ransomware recognition solutions of the RDC-HMFSDL model are demonstrated under varying epochs in Table 2 and Fig. 4. The solution indicates the effectual detection of the samples. The RDC-HMFSDL method attained an average $accu_y$, $sens_y$, $spec_y$, and F_{score} of 99.52% with 500 epochs, improving to 99.64% at 1000 epochs and reaching 99.76% at 1500 epochs. The MCC also increased from 99.05% to 99.52% across these stages, illustrating consistent performance gains with more training. Meanwhile, on 2000 and 2500 epochs, the RDC-HMFSDL method attains slightly reduced values. Finally, on 3000 epochs, the RDC-HMFSDL model attains lower values.

DOI: https://doi.org/10.54216/JCIM.170204
Received: February 05, 2025 Revised: May 26, 2025 Accepted: July 02, 2025

Table 2: Ransomware recognition analysis of RDC-HMFSDL technique on varying epochs

Classes	$Accu_y$	$Sens_y$	$Spec_y$	F_{Score}	MCC
Epoch500					
GW	99.76	99.76	99.29	99.52	99.05
RW	99.29	99.29	99.76	99.52	99.05
Average	99.52	99.52	99.52	99.52	99.05
Epoch1000					
GW	99.76	99.76	99.52	99.64	99.29
RW	99.52	99.52	99.76	99.64	99.29
Average	99.64	99.64	99.64	99.64	99.29
Epoch1500					
GW	100.00	100.00	99.52	99.76	99.52
RW	99.52	99.52	100.00	99.76	99.52
Average	99.76	99.76	99.76	99.76	99.52
Epoch2000					
GW	100.00	100.00	99.76	99.88	99.76
RW	99.76	99.76	100.00	99.88	99.76
Average	99.88	99.88	99.88	99.88	99.76
Epoch2500					
GW	99.05	99.05	99.52	99.28	98.57
RW	99.52	99.52	99.05	99.29	98.57
Average	99.29	99.29	99.29	99.29	98.57
Epoch3000					
GW	99.29	99.29	99.52	99.40	98.81
RW	99.52	99.52	99.29	99.41	98.81
Average	99.40	99.40	99.40	99.40	98.81

DOI: https://doi.org/10.54216/JCIM.170204
Received: February 05, 2025 Revised: May 26, 2025 Accepted: July 02, 2025

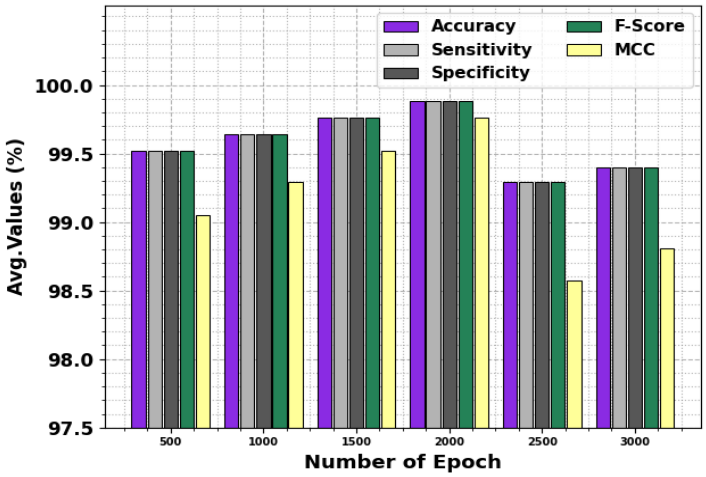

Figure 4. Average of RDC-HMFSDL technique under various epochs

Fig. 5 depicts the $accu_y$ curves for training (TR) and validation (VL) of the RDC-HMFSDL approach under diverse epochs. Initially, both TR/TS $accu_y$ consistently increased with more epochs, illustrating the robust learning ability and pattern recognition of the model. The upward trend in testing accuracy highlights its generalization strength, effectively anticipating unseen data.

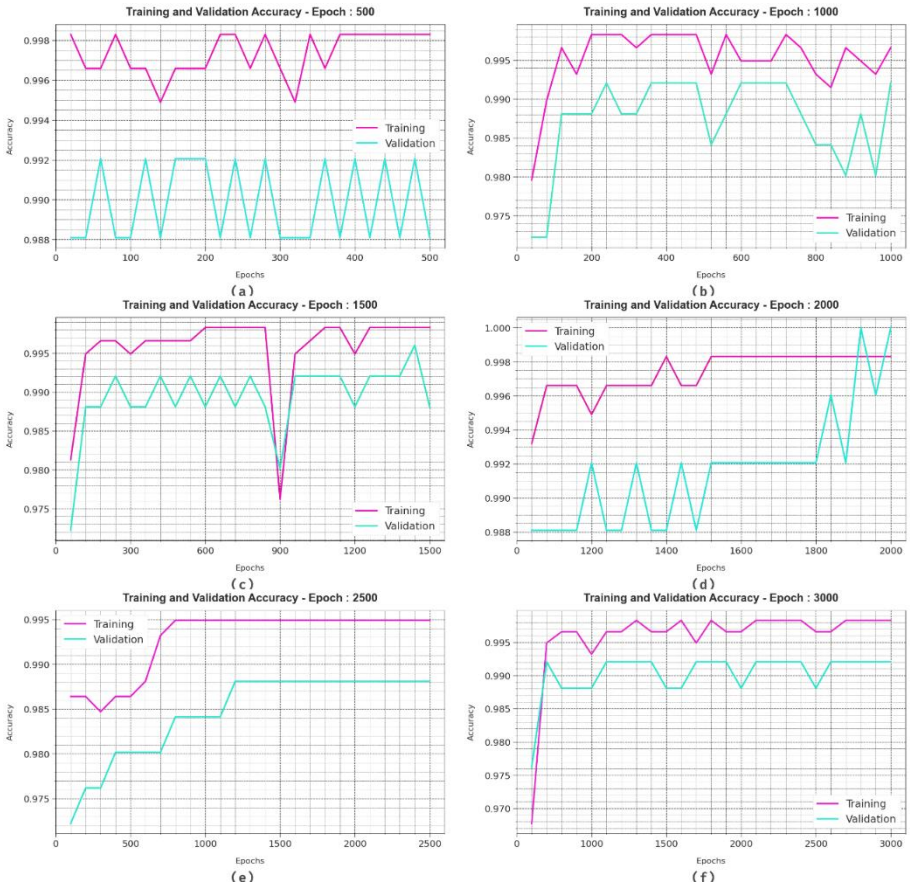

Figure 5. $Accu_y$ Curves of RDC-HMFSDL technique (a-f) Epochs 500-3000

Fig. 6 depicts the loss values of the RDC-HMFSDL approach under diverse epochs. The TR loss steadily reduces as the method optimizes weights for reducing errors. This trend indicates robust alignment with TR data and effective pattern learning. The continuous parameter updates in the RDC-HMFSDL model assists in mitigating discrepancies between actual and predicted labels, improving overall performance.

DOI: https://doi.org/10.54216/JCIM.170204
Received: February 05, 2025 Revised: May 26, 2025 Accepted: July 02, 2025

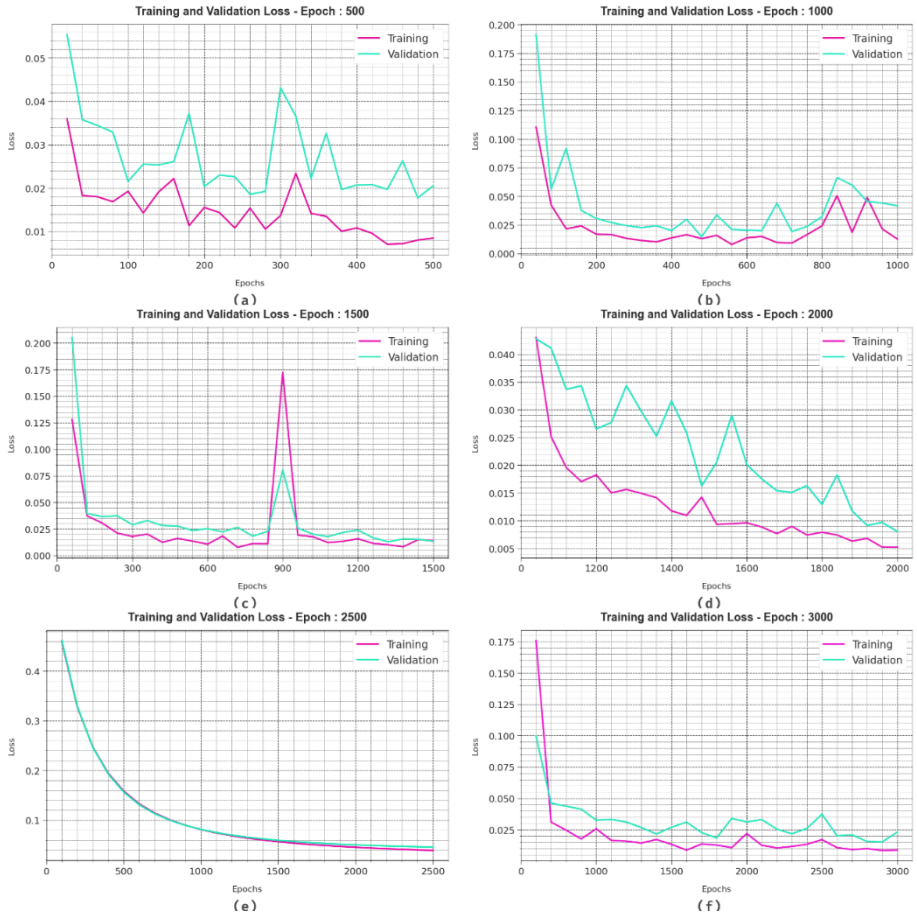

Figure 6. Loss curves of RDC-HMFSDL technique (a-f) Epochs 500-3000

Fig. 7 represents the PR of the RDC-HMFSDL model under epoch 2000. These solutions emphasize the effectual discrimination capability of the technique among diverse classes, underlining its efficacy in appropriately identifying classes.

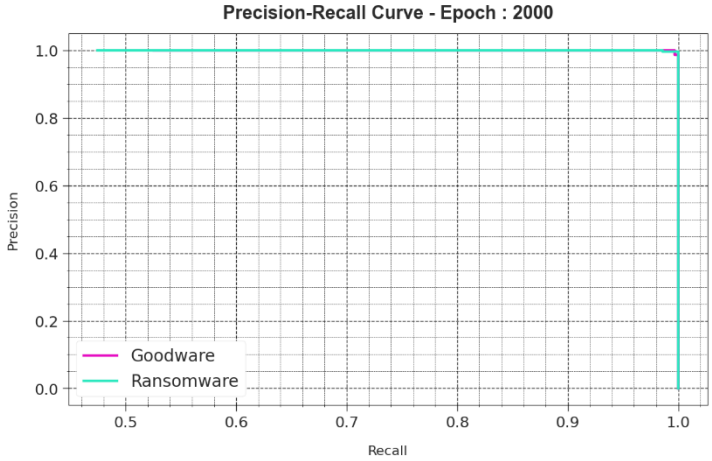

Figure 7. PR curve of RDC-HMFSDL method under Epoch 2000

The ROC curves produced by the RDC-HMFSDL approach with epoch 2000 is represented in Fig. 8, signifying its capability in discriminating among classes. These curves exhibit the balance between FPR and TPR over diverse epochs and thresholds, reflecting the robust accuracy and efficiency of the model in managing complex, multi-class classification tasks.

DOI: https://doi.org/10.54216/JCIM.170204
Received: February 05, 2025 Revised: May 26, 2025 Accepted: July 02, 2025

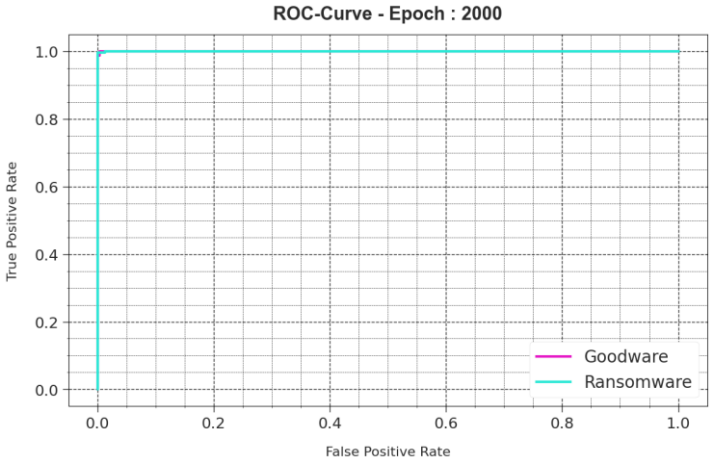

Figure 8. ROC curve of RDC-HMFSDL method under Epoch 2000

Table 3 portrays the comprehensive comparison study of the RDC-HMFSDL methodology [24]. In Fig. 9, the comparison result of the RDC-HMFSDL methodology is provided in terms of $accu_y$. the results specify the better solution of the RDC-HMFSDL technique. Based on $accu_y$, the RDC-HMFSDL technique presents enhanced $accu_y$ of 99.88% but, the OGCNN-RWD, DWOML, Bagging, Adaboost-M1, rotation forest (ROF), decision tree (DT), and RF techniques obtain decreased $accu_y$ of 99.64%, 99.09%, 98.47%, 96.13%, 95.79%, 97.63%, and 98.83%, subsequently.

Table 3: Comparison assessment of RDC-HMFSDL model with existing techniques

Methods	$Accu_y$	$Sens_y$	$Spec_y$
RDC-HMFSDL	99.88	99.88	99.88
OGCNN-RWD	99.64	99.64	99.64
DWOML	99.09	99.43	99.17
Bagging	98.47	93.66	96.06
AdaBoost-M1	96.13	94.50	94.60
ROF	95.79	96.77	97.38
DT	97.63	97.82	98.12
RF	98.83	98.79	98.26

A comparison analysis of the RDC-HMFSDL technique is described with respect to $sens_y$ and $spec_y$ in Fig. 10. These accomplished findings specify the higher performance of the RDC-HMFSDL technique. According to $sens_y$, the RDC-HMFSDL technique provides enriched $sens_y$ of 99.88% and based on $spec_y$, the RDC-HMFSDL model provides an improved $spec_y$ of 99.88%.

DOI: https://doi.org/10.54216/JCIM.170204
Received: February 05, 2025 Revised: May 26, 2025 Accepted: July 02, 2025

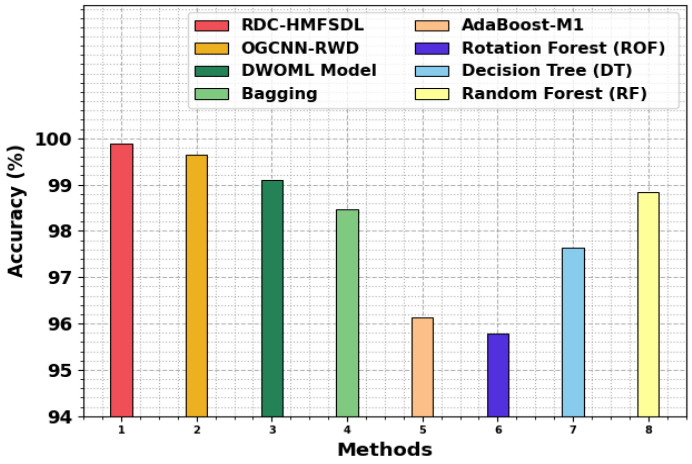

Figure 9. $Accu_y$ Analysis of RDC-HMFSDL model with existing techniques

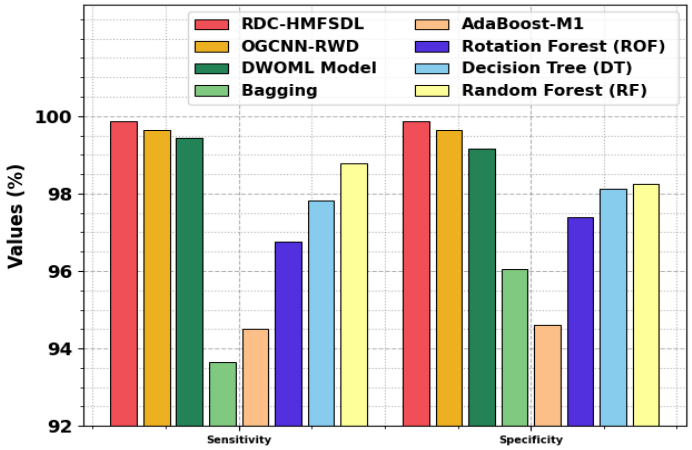

Figure 10. $Sens_y$ and $Spec_y$ analysis of RDC-HMFSDL model with existing techniques

Therefore, the RDC-HMFSDL methodology is applied for an enhanced and automated detection process.

5. Conclusion

This study proposed the RDC-HMFSDL technique, which aimed to categorize and find the existence of ransomware attacks accurately. To achieve this, the RDC-HMFSDL technique encompasses min-max data normalization, HRDSO-based FS, LSTM-AE-based classification, and SCA-based tuning. Initially, min-max data normalization is used. Furthermore, the RDC-HMFSDL model employs the HRDSO method for the FS procedure. For ransomware attack detection, the LSTM-AE method is applied to detect and classify ransomware attacks. Finally, SCA method is used to optimally choose the parameter values of the LSTM-AE. The simulation analysis of the RDC-HMFSDL model is examined under a benchmark dataset. The comparison study of the RDC-HMFSDL model portrayed a superior accuracy value of 99.88% over existing methods.

Funding: "This research received no external funding"

Conflicts of Interest: "The authors declare no conflict of interest."

References

[1] Singh *et al.*, "Enhancing ransomware attack detection using transfer learning and deep learning ensemble models on cloud-encrypted data," *Electronics*, vol. 12, no. 18, p. 3899, 2023.

[2] H. M. Alhassan, A. A. Alzahrani, and M. A. Alghamdi, "A hybrid deep learning model for ransomware detection in cloud environments," *J. Cloud Comput.*, vol. 12, no. 1, p. 1, 2023, doi: 10.1186/s13677-023-00356-4.

DOI: https://doi.org/10.54216/JCIM.170204
Received: February 05, 2025 Revised: May 26, 2025 Accepted: July 02, 2025

[3] P. Sharma, K. Chaudhary, and M. G. Khan, "The Art-of-Hyper-Parameter Optimization with Desirable Feature Selection: Optimizing for multiple objectives: ransomware anomaly detection," in *Proc. Int. Conf. Med. Imag. Comput.-Aided Diagnosis (MICAD)*, 2021, pp. 218-227.

[4] R. M. Al-Sabahi, H. A. Alharbi, and M. S. Alhussain, "An enhanced machine learning approach for ransomware detection using feature extraction," *J. Ambient Intell. Humaniz. Comput*, vol. 14, no. 3, pp. 1533-1545, 2023, doi: 10.1007/s12652-022-03796-2.

[5] P. Qi, Z. Zhang, W. Wang, and C. Yao, "Malware detection by exploiting deep learning over binary programs," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 9068-9075.

[6] R. A. Alsaidi *et al.*, "Ransomware detection using machine and deep learning approaches," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 11, 2022.

[7] K. S. Sangher, A. Singh, and H. M. Pandey, "Signature-based ransomware detection based on optimizations approaches using RandomClassifier and CNN algorithms," *arXiv preprint arXiv: 2303.05725*, 2023.

[8] S. I. Imtiaz *et al.*, "DeepAMD: Detection and identification of Android malware using high-efficient Deep Artificial Neural Network," *Future Gener. Comput. Syst.*, vol. 115, pp. 844-856, 2021.

[9] U. Zahoora *et al.*, "Zero-day ransomware attack detection using deep contractive autoencoder and voting based ensemble classifier," *Appl. Intell.*, vol. 52, no. 12, pp. 13941-13960, 2022.

[10] Almomani, A. Alkhayer, and W. El-Shafai, "E2E-RDS: Efficient End-to-End Ransomware Detection System Based on Static-Based ML and Vision-Based DL Approaches," *Sensors*, vol. 23, no. 9, p. 4467, 2023.

[11] M. Al Duhayyim *et al.*, "Artificial Algae Optimization with Deep Belief Network enabled ransomware detection in IoT environment," *Comput. Syst. Sci. Eng.*, vol. 46, no. 2, 2023.

[12] U. Zahoora *et al.*, "Ransomware detection using deep learning based unsupervised feature extraction and a cost sensitive Pareto Ensemble classifier," *Sci. Rep.*, vol. 12, no. 1, p. 15647, 2022.

[13] N. Lu *et al.*, "An efficient combined deep neural network based malware detection framework in 5G environment," *Comput. Netw.*, vol. 189, p. 107932, 2021.

[14] M. Jemal and D. C. T. Lo, "Detection of Ransomware Attack Using Deep Learning," in *Proc. IEEE Conf. Dependable Secure Comput. (DSC)*, Nov. 2023, pp. 1-9.

[15] Zhu *et al.*, "A few-shot meta-learning based siamese neural network using entropy features for ransomware classification," *Comput. Secur*, vol. 117, p. 102691, 2022.

[16] V. Lavanya and P. C. Sekhar, "Efficient Cybersecurity Model Using Wavelet Deep CNN and Enhanced Rain Optimization Algorithm," *Int. J. Image Graph*, p. 2450048, 2023.

[17] G. Nalinipriya *et al.*, "Optimized deep stacked autoencoder for ransomware detection using blockchain network," *Int. J. Wavelets, Multiresolution, Inf. Process.*, vol. 19, no. 6, p. 2150022, 2021.

[18] S. Kavitha, N. U. Maheswari, and R. Venkatesh, "Intelligent Intrusion Detection System using Enhanced Arithmetic Optimization Algorithm with Deep Learning Model," *Teh. Vjesn*, vol. 30, no. 4, pp. 1217-1224, 2023.

[19] M. Shantal, Z. Othman, and A. A. Bakar, "A Novel Approach for Data Feature Weighting Using Correlation Coefficients and Min-Max Normalization," *Symmetry*, vol. 15, no. 12, p. 2185, 2023.

[20] M. Alshehri, "Breast Cancer Detection and Classification Using Hybrid Feature Selection and DenseXtNet Approach," *Mathematics*, vol. 11, no. 23, p. 4725, 2023.

[21] Li and X. Zhang, "Prediction of stress-strain behavior of rock materials under biaxial compression using a deep learning approach," *PLoS ONE*, vol. 20, no. 4, p. e0321478, 2025.

[22] X. Guo *et al.*, "WSN Clustering Routing Algorithm Combining Sine Cosine Algorithm and Lévy Mutation," *IEEE Access*, vol. 11, pp. 22654-22663, 2023.

[23] L. T. L. Tan, S. K. Shahrin, and R. A. Rahman, "A novel approach for ransomware detection using a hybrid model of deep learning and ensemble methods," *Comput. Secur*, vol. 120, p. 102817, 2023, doi: 10.1016/j.cose.2022.102817.

[24] H. K. Alkahtani *et al.*, "Optimal Graph Convolutional Neural Network-Based Ransomware Detection for Cybersecurity in IoT Environment," *Appl. Sci.*, vol. 13, no. 8, p. 5167, 2023.

DOI: https://doi.org/10.54216/JCIM.170204
Received: February 05, 2025 Revised: May 26, 2025 Accepted: July 02, 2025