IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 5 April 2024, accepted 25 April 2024, date of publication 30 April 2024, date of current version 8 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3395328

RESEARCH ARTICLE

Empowering Cybersecurity Using Enhanced Rat Swarm Optimization With Deep Stack-Based Ensemble Learning Approach

P. MANICKAM[1], M. GIRIJA[2], ASHIT KUMAR DUTTA[3], PALAMAKULA RAMESH BABU[4], KRISHAN ARORA[5], MUN KWEON JEONG[6], AND SRIJANA ACHARYA[7]

[1]Department of Computer Science, Thiagarajar College, Madurai, Tamil Nadu 625009, India
[2]Department of Computer Science, The American College, Madurai, Tamil Nadu 625002, India
[3]Department of Computer Science and Information Systems, College of Applied Sciences, AlMaarefa University, Diriyah, Riyadh 13713, Saudi Arabia
[4]Department of Information Technology, Chaitanya Bharathi Institute of Technology, Hyderabad, Telangana 500075, India
[5]School of Electronics and Electrical Engineering, Lovely Professional University, Phagwara, Punjab 144411, India
[6]Department of Research and Development, Information Technology Research Center, UMLogics Company Ltd., Daejeon 34012, Republic of Korea
[7]Department of Convergence Science, Kongju National University, Gongju-si 32588, South Korea

Corresponding authors: Mun Kweon Jeong (jmk@umlogics.com) and Srijana Acharya (srijana@kongju.ac.kr)

This work was supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by Korean Government (MSIT) (Development of Usable Security Technology for Improving Security Usability in a Non-Face-to-Face Environment) under Grant 2021-0-01003.

ABSTRACT Cybersecurity is a vital technology and measures intended to protect networks, computers, information, and programs from threats and illegal access, modification, or damage. A security model covers a network and a computer safety method. Each system has antivirus software, firewalls, and an intrusion detection system (IDS). IDS helps in discovering and identifying illegal system behavior such as usage, copying, alteration, and damage. By estimating network traffic anomalies and patterns, deep learning (DL) models can enhance the detection abilities of IDS when compared to traditional rule-based methods. These models learn complex representations from data, authorizing them to recognize subtle and developing attack patterns. Techniques like recurrent neural network (RNN) and convolutional neural network (CNN) can be applied to progress consecutive or spatial features in network data, correspondingly. This manuscript empowers Cybersecurity by utilizing an Enhanced Rat Swarm Optimizer with a Deep Stack-Based Ensemble Learning (ERSO-DSEL) model. The ERSO-DSEL approach leverages feature selection (FS) with EL strategies to boost cybersecurity. In the ERSO-DSEL system, Z-score normalization is employed to measure the input data. Besides, an improved equilibrium optimizer (IEO) based FS approach is applied to choose a set of features. For cyberattack recognition, the ERSO-DSBEL approach uses the DSEL approach comprising three models namely deep neural network (DNN), long short-term memory (LSTM), and bidirectional LSTM (Bi-LSTM). Furthermore, the hyperparameter selection of these DL models takes place using the ERSO system. The solution result of the ERSO-DSBEL model is executed on a benchmark IDS database. A wide-contrast study reported that the ERSO-DSBEL model accomplishes an enhanced accuracy outcome of 99.67% over other models of cybersecurity.

INDEX TERMS Cybersecurity, intrusion detection systems, ensemble learning, equilibrium optimizer, hyperparameter tuning.

I. INTRODUCTION

In the present scenario, the rise in the occurrence of the network has set severe issues linked to cybersecurity. The existence of novel smart network technology needs novel models of growth in cybersecurity [1]. In defense of crucial structures from threats and illegal access, cybersecurity is very significant. Cybersecurity contains numerous processes and technologies. Security of network, information, application, operational, end-user education, disaster recovery, etc. is a few classes of cybersecurity [2]. Cybersecurity threats pose a few of the most severe national and financial

The associate editor coordinating the review of this manuscript and approving it for publication was Tiago Cruz.

safety challenges. Cyberattacks are nothing but a war without any weapons but most terrible and malicious foremost to revealing business and private information, disturbing critical actions, nonstop exposures, and illegitimate access to mechanisms and software, so they impose high expenses on the country's financial [3]. Cybersecurity is a constant issue for many reputed businesses like retail stores, banks, and crucial organizations like SCADA, power grids, etc. While several attack detection methods are available currently the fast growth in attacks and the development of hacking abilities need more evolution of novel recognition methods very obligatory.

IDS are a safety line of a system [4]. IDS can be used alongside other safety methods like authentication tools, access control, and encryption models to protect the systems against cyberattacks in a better way. Utilizing patterns of specific rules define an exact attack; IDS can differentiate between malicious and normal activities [5]. Data mining is employed to define knowledge detection which aids in executing and using IDS with greater precision and strong behavior when equated to traditional IDS which may not be effective beside current sophisticated cyberattacks [6]. Furthermore, many researchers are struggling to discover complete and legal datasets to check and assess their projected models, and taking an appropriate dataset is an important challenge [7]. To check the efficacy of such tools, trustworthy datasets are required to (i) enclose both numerous and benign attacks, (ii) meet reality measures, and (iii) be openly accessible.

DL is a sub-field of machine learning (ML) that has increased great detection in numerous regions owing to its development in accuracy in difficult tasks and the current growth in software and hardware [8]. DL models enhance cybersecurity methods that prevent attacks by recognizing patterns that are dissimilar from usual behavior [9]. Cyberattacks share a general feature with image detection because more than 99 percent of the novel attacks are little mutants of present ones; in a similar technique modification in pictures can be recognized by little variations in their pixels. In IoT Fog methods, they are employed for identifying system attacks but the IoT network features (i.e., its spread nature and the restricted calculating abilities of the end-devices) need new solutions for IDS [10].

In [11], an ensemble-based DL technique is developed, uniting K-means with DL classification models like Gated Recurrent Unit (GRU), LSTM, CNN, RNN, and DNN. After pre-processing, an RF is executed for extraction. Later, an ensemble-based model is used. In [12], an effectual method is proposed. The authors [13] develop a novel ensemble learning (EL) model-based IDS. The effectual FS is achieved through a hybrid of Correlation FS joined with Forest Panelized Attributes (CFS–FPA). This involves using bagging and AdaBoosting EL techniques to adjust 4 ML classifiers. Hammood and Sadiq [14] proposed an intelligent network IDS (NIDS) utilizing ML models. A novel anomaly-based result for IoT systems using EML techniques consisting of LR, NB, DT, extra trees, RF, and gradient boosting is also presented.

In [15], an EL scheme based on an RF model is proposed. To decrease classification error, the SMOTE was proposed. In [16], a multi-layered behavior-based IDS utilizing EL models are offered for classification. Also, DT, RF, and Artificial Neural Networks (ANNs) models are selected to construct the ensemble. In [17], an ML-based NIDS with dual-phased hybrid EL and automatic FS with four ML classifiers is developed. This model contains binary learning stages where the 1st and 2nd stages are built utilizing classifiers from joining attack classes. Okey et al. [18] developed a BoostedEnML model. First, classifiers are trained to utilize the stacking, and a popular voting technique is acquired. The data balancing is executed with SMOTE and adaptive synthetic (ADASYN) models. Then, a stratified K-fold is applied.

Terbuch et al. [19] present a hybrid ML (HML) model, integrating key performance indicators (KPIs) with an unsupervised variational autoencoder (VAE) featuring LSTM layers. In [20], a fusion DL, consisting of Graph Convolutional LSTM (GC-LSTM) and a DNN model is proposed. Wang et al. [21] utilized a hierarchical model using wavelet transform and DL methods for extraction. In [22], a fusion ELM is proposed, utilizing unity normalization and PCA for pre-processing, optimization with Grey Wolf Optimizer (GWO) for classifiers, and also presents an effectual ELM selection model. Manokaran and Vairavel [23] introduce an optimized stacked ELM model, utilizing an improved GWO and stacking ensemble. In [24], an Anomaly Scoring ELM, a bagging EL framework is presented. The cited studies present several ensemble-based and fusion ML methods for intrusion and anomaly detection. These techniques incorporate various models namely DL, EL, and FS to improve detection accuracy and robustness against cyber threats.

This manuscript empowers Cybersecurity using Enhanced Rat Swarm Optimization with a Deep Stack-Based Ensemble Learning (ERSO-DSEL) Approach. In the ERSO-DSEL system, Z-score normalization is employed to measure the input data. Besides, an improved equilibrium optimizer (IEO) based FS approach is applied to choose a set of features. For cyberattack recognition, the ERSO-DSBEL model uses the DSEL system comprising 3 techniques long short-term memory (LSTM), deep neural network (DNN), and bidirectional LSTM (Bi-LSTM). Furthermore, the hyperparameter selection of these DL models takes place utilizing the ERSO approach. The performance validation of the ERSO-DSBEL system is executed on a benchmark database of IDS.

II. THE PROPOSED METHOD

In this research, cybersecurity using the ERSO-DSEL approach is empowered. The ERSO-DSEL technique leverages FS with EL strategies to boost cybersecurity. The ERSO-DSEL methodology involves four major procedures such as Z-score normalization, IEO-based FS, deep

stack-based EL, and ERSO-based hyperparameter tuning. Fig. 1 determines the complete development of the ERSO-DSEL model.

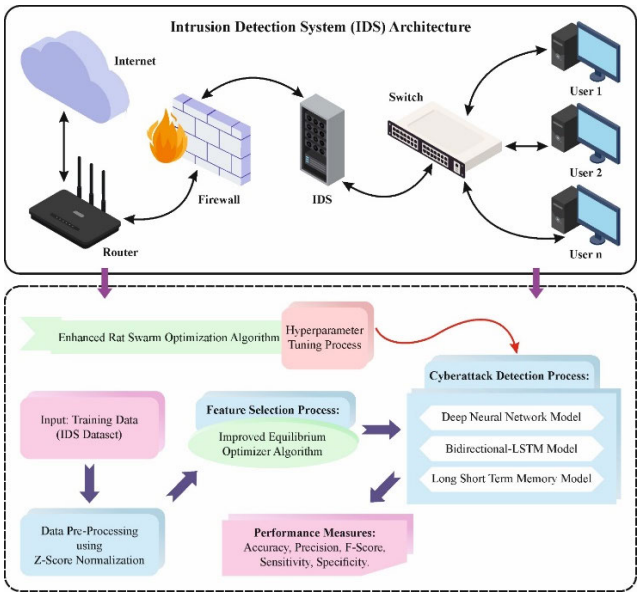

FIGURE 1. The overall process of the ERSO-DSEL algorithm.

A. Z-SCORE NORMALIZATION

At the first level, the ERSO-DSEL model uses Z-score normalization to measure the input data. Z-score normalization is also recognized as standardization. It is an arithmetical model utilized to convert a dataset by rescaling its values to consume a standard deviation (SD) of one and a mean of zero [25]. This procedure aids in transporting all the features of the dataset onto a general measure, making it simpler to equate and analyze them. It is attained by deducting the mean of every data point and separating it by the SD. Z-score standardization is generally used in ML and statistics to guarantee that dissimilar features donate similarly to the study, averting definite variables from controlling the method due to their higher measure.

B. IEO BASED FS

For the procedure of FS, the IEO-based FS approach is applied to elect a set of features. There is a need to evaluate $2^{141}-1$ times which is nearly not possible to acquire an optimum feature from MFCCs and pitch features. Thus, the IEO is used as a feature selector to attain optimal solutions within the time range [26]. EO is used to randomly initialize the population position and its updated position can be described by:

$$X_i(n+1) = X_{eq}(n) + \left(X_i(n) - X_{eq}(n)\right) F(n) + \frac{G(n)}{\lambda}\left(1F(n)\right) \quad (1)$$

In Eq. (1), X_{eq} shows the equilibrium pooling, and it is constructed by the position of the first four optimum solutions and their average value. The algorithm selects one randomly from X_{eq} for each run.

$$t(n) = \left(1 - \frac{n}{Max_{iter}}\right)^{\left(2\frac{n}{Max_{iter}}\right)} \quad (2)$$

$$F(n) = sign(r-0.5)\left[e^{-\lambda t(n)} - 1\right] \quad (3)$$

where F controls the balance between exploitation and exploration, Max_iter indicates the maximal iteration. λ and r are two randomly generated values within [0, 1]. *The sign* refers to the signum function of Matlab. G helps the algorithm to acquire superior performance, and it is computed by:

$$GCP = \begin{cases} 0.5r1 & if\,(r2 \geq GP) \\ 0 & else \end{cases} \quad (4)$$

$$G_0(n) = GCP * \left(X_{eq}(n) - X_i(n)\right) \quad (5)$$

$$G(n) = G_0(n) * F(n) \quad (6)$$

where $r1$ and $r2$ are two randomly generated numbers within [0, 1].

In EO, a solution is guided by the equilibrium pooling. However, the presence of four optimum solutions is positioned at local optima. On the other hand, if the solution is disseminated in the search region, it can slow down the convergence rate.

The two critical aspects, exploitation and exploration are used for assessing the efficacy of metaheuristic algorithms. The exploration improves the global search ability of EO and assists in escaping from local optima. Furthermore, exploration is used to empower the algorithm with stronger local search ability and promote it to comprehensively exploit promising regions and determine the optimum solution. Initially, the population is divided into three sub-swarms namely exploration ability, population diversity, and convergence and exploitation ability.

The transfer function has played an important role in the binary metaheuristic approach, and it converts continuous values into binary strings. The transfer function could ensure efficient exploration of search space, maintain population diversity, and avoid early convergence to perform binarization.

$$X_i^j(n+1) = \begin{cases} X_i^j(n) & if\ (S(value) < rand) \\ 1 - X_i^j(n) & else \end{cases} \quad (7)$$

$$value = \left(X_i(n) - X_{eq}(n)\right) F(n) + \frac{G(n)}{\lambda}\left(1 - F(n)\right) \quad (8)$$

whereas S signifies the transfer function, and $X_i^j(n)$ represents the location of i^{th} individuals in the j^{th} dimension at n^{th} iteration.

The equilibrium pooling is created in the first sub-swarm and S_1 is adopted as a transfer function where S_1 quickly shifts position. This sub-swarm is used to explore more space, and it has a better global search ability. S_2 is adopted as a transfer function in the second sub-swarm and it balances

local and global search and it has the benefits of *EO*. During the third sub-swarm, X_{eq} represents the global optimum solution, and S_3 is adopted as a transfer function where S_3 slowly shifts position. The sub-swarm exploits the optimum solution and has greater local search capability.

The fitness function (FF) reflects the accuracy of classification and the amount of nominated features. It increases the accuracy of classification and diminishes the set dimension of the chosen feature. So, the FF is employed to assess individual solutions as exposed in Eq. (17).

$$Fitness = \alpha * ErrorRate + (1 - \alpha) * \frac{\#SF}{\#All\_F} \tag{9}$$

Here ErrorRate directs the error rate of classification using the nominated feature. ErrorRate is calculated as the ratio of wrong order to the sum of classification prepared and has as a value among 0 and 1. (ErrorRate is the supplement of the accuracy of classification), $\#SF$ is the number of nominated features and $\#All\_F$ is the complete integer of aspects in the novel dataset. α is utilized to handle the prominence of classifier excellence and sub-set length., α is set to 0.9 in our experiments.

C. DEEP STACK-BASED EL

For cyberattack recognition, the ERSO-DSBEL technique applies the DSEL model comprising three models such as DNN, LSTM, and BiLSTM. The EL model integrates numerous base-learner procedure pattern systems when generating the best prediction method [27]. The proposed method executes much superior when compared to base learner systems alone. Additional EL model presented by Wolpert is stacked generalized, which has been widely functional in many areas ever since its inception. Stacking integrates the results of manifold-based learner methods needed to train a novel meta-learner technique for the output outcome. The stacking idea is constructed on dual phases of systems. The 1st phase contains numerous base-learner models, whereas the 2nd phase covers the meta-learner system.

1) DNN MODEL

An ANN contains frequent layers among the layers of output and input is recognized as a DNN. Data flows through NN in dual methods such as the multi-layer perceptron (MLP) technique is employed to forecast the output for the delivered data in forward propagation, and the method upgrades its parameters in backpropagation (BP) dependent upon the fault of prediction. Fig. 2 portrays the infrastructure of DNN. The first output and succeeding hidden layer (HL) are specified as below:

$$h_1 = \sigma\left[W_1.(x) + b_1\right] \tag{10}$$

$$h_i = \sigma\left[W_{1i}.(h_{i-1}) + b_i\right] \tag{11}$$

whereas i signifies the index layer, and σ denotes the activation function. The size of x is equivalent to 3636×9. Every column directs a neural network feature. The output of the

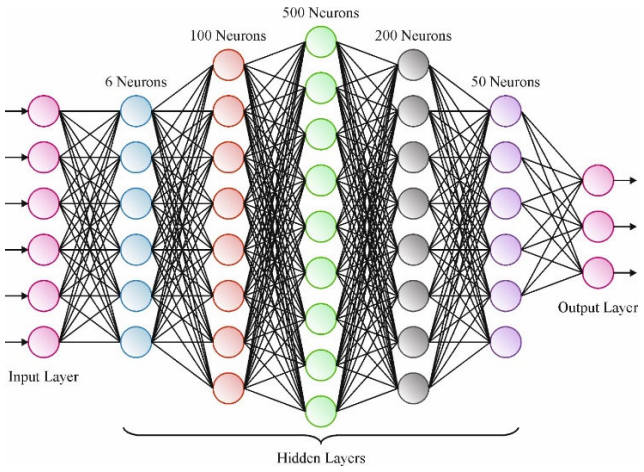

FIGURE 2. Structure of DNN.

MLP is as follows:

$$y = \sigma_{soft}\left(h_{out}\right) \tag{12}$$

Every layer is allotted an activation function dependent upon the *tanh* (hyperbolic tangent). The output layer uses the Softmax function to recognize the output.

2) LSTM MODEL

Hochreiter et al. presented an LSTM network and an upgraded RNN whose main attention is to take long-term dependence on data series to pledge the gradient vanishing problem. The local nearby activation adjusts a consistent RNNs HL but a few times it is stated to as short-term memory. Besides, the weights are altered by calculations executed during the prolonged data series. To preserve data reliability over broad arrays, LSTM is provided with a state of activation as a weight. The calculation formulations for LSTM neurons are expressed below:

$$I_t = \sigma\left[W_{xi}\left(x_t\right) + W_{hi}\left(h_{t-1}\right) + W_{ci}\left(c_{t-1}\right) + b_i\right] \tag{13}$$

$$F_t = \sigma\left[W_{xf}\left(x_t\right) + W_{hf}\left(h_{t-1}\right) + W_{cf}\left(c_{t-1}\right) + b_f\right] \tag{14}$$

$$O_t = \sigma\left[W_{xo}\left(x_t\right) + W_{ho}\left(h_{t-1}\right) + W_{co}\left(c_t\right) + b_0\right] \tag{15}$$

$$h_t = o_t \tanh\left(c_t\right) \tag{16}$$

From the abovementioned equations, f, I, c, and o denote the gate of forget, input, neuron memory cell, and output, correspondingly. The neuron's output and input are directed by h and x, respectively. While W, σ, and b represent the weight co-efficient matrix, excitation function, and bias matrix, correspondingly. x denotes to input feature and h specifies the predicted output.

3) BI-LSTM MODEL

Bi-LSTM is a precise RNN. The LSTM forecasts follow-up data. To increase the accuracy of prediction, the Bi-LSTM method acquires time-correlated data over both backward and forward ways at the same time. The backward and forward LSTM are signified by h_b and h_f, correspondingly.

The h_b and h_f are neutral and only transmit to their LSTM layer. The layer of forward displays the unidirectional movement from input to hidden to output layers. Linking the dual HL permits us to calculate the last Bi-LSTM prediction y_t. The method can be defined with the aid of Eqs. (17)- (19) as follows:

$$h_f = LSTM\left(x_t, h_{f(t-1)}\right) \tag{17}$$

$$h_b = LSTM\left(x_t, h_{b(t+1)}\right) \tag{18}$$

$$y_t = \sigma\left[W_{h_{Yf}}\left(h_f\right) + W_{hyb}\left(h_b\right) + b_y\right] \tag{19}$$

whereas, the weights of the backward and forward layers at time t are represented by W_{yb} and W_{hyf}, correspondingly; *LSTM* signifies network LSTM; The activation function is signified by σ; and b_y is represented by the output layer bias.

Algorithm 1 DSEL Method Pseudocod

Input: Training dataset $D = [(x_1, y_2), (x_1, y_2), \ldots, ((x_1, y_2))]$
Output: Base Classifier bc_1, bc_2, bc_3
Meta Classifier m
Deep Stacked Model DS_m
Base Model Training:
for $i = 1, 2, \ldots, K$
Train the base classifiers (bc_1, bc_2, bc_3)
Create new data: Met-classifier data

$$D_m = [p_i', y_i]$$

Where $p_i' = [bc_1(x_i), (bc_2(x_i), (bc_3(x_i)]$
End
Meta-Classifier Training:
Train meta-classifier on recently created data D_m
Performance measure valuation
Return stacked method

$$DS_m = [bc_1, bc_2, bc_3, m]$$

D. HYPERPARAMETER TUNING USING THE ERSO ALGORITHM

Eventually, the hyperparameter selection of these DL models takes place using the ERSO algorithm. The ERSO algorithm is employed to increase the accuracy of classification by changing the parameters [28]. A recent swarm intelligence (SI) method is the RSO. Method 1 offers the RSO algorithm, and the important RSO stages have been explained in a detailed manner.

Step 1: Creating the variables of RSOs: Numerous variables of RSO require initial values allocated. Three control parameters and two algorithmic variables achieved the RSO. The highest iterations count and the size of populace represented by N and $Tmax$, correspondingly. The random values have been signified as variables like R, C, and A, with C and R ranges of $C \in [0, 2]$ and $R \in [1, 5]$, correspondingly. Eq. (20) can be applied to initializing the value *ofA*. Parameters A and C achieve exploitation and exploration abilities in RSO.

$$A = R - t \times \left(\frac{R}{\tau_{\max}} t1, 2, \ldots\ldots, T_{max}\right) \tag{20}$$

Step 2: Primary RSO's population generation: The 2nd stage encloses solution vector or arbitrarily produced rat locations into the RSO's population (RP). N refers to several tasks that exist. Since proven in Eq. (21), the RP was represented by a 2D matrix with a size of $N \times d$.

$$RP = \begin{bmatrix} Y_{1,1} & Y_{1,2} & \cdots & Y_{1,d-1} & Y_{1,d} \\ Y_{2,1} & Y_{2,2} & \cdots & Y_{2,d-1} & Y_{2,d} \\ \vdots & \vdots & \ldots & \vdots & \vdots \\ Y_{N-1,1} & Y_{N-1,2} & \cdots & Y_{N-1,d-1} & Y_{N-1,d} \\ Y_{N,1} & Y_{N,2} & \cdots & Y_{N,\mathrm{d}-1} & Y_{N,\mathrm{d}} \end{bmatrix} \tag{21}$$

In RP, each row $Y_i = (Y_{i,1}, Y_{i,2}, \ldots, Y_{1,d-1}, Y_{i,d})$ defines *theith* location of the rate. A decision variable called Y_i is a value of random in the middle of Y_{jmin} and Y_{jmax}. The decision variables' higher and lower boundaries are Y_{jmax} and Y_{jmin}, correspondingly.

Step 3: Fitness calculation: The main function (*Yi*), with $\forall i = 1, 2, N$, is employed for the value of fitness. The rat was elected as *Ygbest* to be the finest location in *RP*.

Step 4: Position upgrade: applying 2 succeeding functions, hunting the prey and discussing all rat's locations with the prey in RP will be changed in such a stage.

1. Follow the prey: Rat in the perfect place, *Xgbest* refers to the position of the prey. The assignments of the alternative rats have been upgraded in RP according to the features of *Xgbest*, denoted in Eq. (22).

$$\overline{Y} = A \cdot \vec{Y}_i(t) + C \cdot \left(\vec{Y}^{\mathrm{gbest}} - \vec{Y}_i(t)\right) \tag{22}$$

whereas t represents the existing iteration, $\vec{Y}(t)$ refers to the i^{th} rat location, and variables like C and A were defined in the first stage and obtained values amongst 1 and $Tmax$.

2. Combating prey: Based on the place of the prey, as demonstrated in a formula, informing each rat's position in *RP*.

$$\vec{Y}(t+1) = \left|\vec{Y}^{\mathrm{gbest}} - \vec{Y}\right| \tag{23}$$

Step 5: This must be suggested that the optimum solution be upgraded. The upgrade of $(t+1)$ with Y_{gbest} is dependent on the previous fitness existence greater than the final in the stage. Specified contrarily, once the value of $(t+1)$ is lower than y_{gbest}, and after y_{gbest} can be upgraded to $(t+1)$, and i is in the range of one to N.

Step 6: Observe the state for stopping: The latter stage is to reiterate stages 4 and 5 till the stopping state is fulfilled. The termination of the iterations count is denoted by $T_{\max}$. Hence, the finest solution to the optimizer's difficulties is Y_{gbest}.

Kennedy and Eberhart first established RSO, an efficacious and well-known SI algorithm. A pool of particles will be primarily produced arbitrarily, where every particle's velocity and location signify its existing condition in the search space. The finest location is met by all the particles and the total

optimal place is determined thus, it is also tracked. The place of each particle can be upgraded by employing its velocity and its optimum position changed. Each particle's place will be computed by applying Eq. (24).

$$\vec{Y}_i\,(T+1) = \vec{Y}\,(T) + \vec{V}\,(T+1) \tag{24}$$

$$\vec{V}(t+1) = \vec{V}(t) + C_1 \cdot r_1 \cdot \left(\vec{Y}_i^{\,\text{lbest}}\,(t) - \vec{Y}\,(t)\right) + C_2 \cdot r_2 \cdot \left(\vec{Y}^{\,\text{gbest}}\,(t) - \vec{Y}\,(t)\right) \tag{25}$$

All rat's positions whereas following prey could be exhibited by employing Eq. (26) and is changed through both the rat's top location. Then, the best location has been achieved,

$$\vec{Y} = A \cdot \vec{Y}_i\,(t) + C \cdot \left(\vec{Y}^{\,\text{gbest}} - \vec{Y}_i\,(t)\right) + r_3 \cdot \left(\vec{Y}_i^{\,\text{lbest}}\,(t) - \vec{Y}_i\,(t)\right) \tag{26}$$

Here, r_3 refers to a random value in the middle of 0 and 1, T, $\vec{Y}^{\,\text{gbest}}$ is the best global position and $\vec{Y}^{\,\text{lbest}}$ is the greatest local location up to iteration accomplished by the RP up to iteration t.

The ERSO method improves an FF to achieve higher classifier performance. It states a positive numeral to indicate the upgraded candidate solution performance. In this study, the error rate of classifier minimization is measured as FF, as assumed in Eq. (27).

$$\begin{aligned} fitness\,(x_i) &= ClassifierErrorRate\,(x_i) \\ &= \frac{No.\ of\ misclassified\ instances}{Total\ no\ .of\ instances} * 100 \end{aligned} \tag{27}$$

III. EXPERIMENTAL VALIDATION

The performance evaluation of the ERSO-DSEL method is verified by employing the UNSW-NB15 dataset [29] with 10000 samples and 10 classes as represented in Table 1.

TABLE 1. Details of the dataset.

Classes	No. of Instances
Normal	1000
Generic	1000
Exploits	1000
Fuzzers	1000
DoS	1000
Reconnaissance	1000
Analysis	1000
Backdoor	1000
Shellcode	1000
Worms	1000
Total Instances	10000

Fig. 3 establishes the confusion matrices generated by the ERSO-DSEL system below 80:20 and 70:30 of TRAP/TESP. The outcomes specify the effective recognition and identification of all 10 classes precisely.

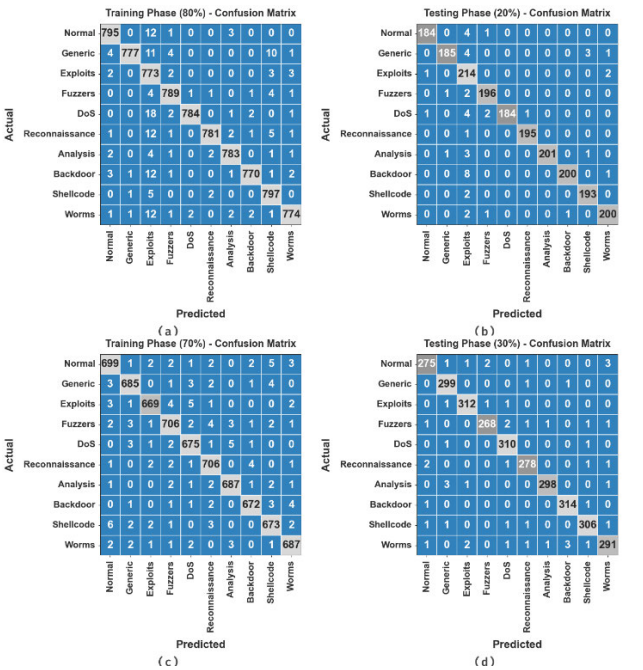

FIGURE 3. Confusion matrices (a-b) 80:20 of TRAP/TESPH and (c-d) 70:30 of TRAP/TESP.

The cyberattack detection results of the ERSO-DSEL model are examined with 80:20 of TRAP/TESP in Table 2

TABLE 2. Cyberattack detection outcome of ERSO-DSEL model under 80:20 of TRAP/TESP.

Classes	$Accu_y$	$Prec_n$	$Sens_y$	$Spec_y$	F_{Score}
TRAP (80%)					
Normal	99.64	98.39	98.03	99.82	98.21
Generic	99.59	99.62	96.28	99.96	97.92
Exploits	98.75	89.57	98.72	98.75	93.92
Fuzzers	99.69	98.38	98.50	99.82	98.44
DoS	99.66	99.62	97.03	99.96	98.31
Reconnaissance	99.65	99.36	97.14	99.93	98.24
Analysis	99.75	98.86	98.61	99.88	98.74
Backdoor	99.66	99.23	97.35	99.92	98.28
Shellcode	99.59	96.96	99.01	99.65	97.97
Worms	99.60	98.72	97.24	99.86	97.97
Average	99.56	97.87	97.79	99.75	97.80
TESP (20%)					
Normal	99.65	98.92	97.35	99.89	98.13
Generic	99.50	98.93	95.85	99.89	97.37
Exploits	98.35	87.70	98.62	98.32	92.84
Fuzzers	99.65	98.00	98.49	99.78	98.25
DoS	99.60	100.00	95.83	100.00	97.87
Reconnaissance	99.90	99.49	99.49	99.94	99.49
Analysis	99.75	100.00	97.57	100.00	98.77
Backdoor	99.50	99.50	95.69	99.94	97.56
Shellcode	99.70	97.97	98.97	99.78	98.47
Worms	99.60	98.04	98.04	99.78	98.04
Average	99.52	97.86	97.59	99.73	97.68

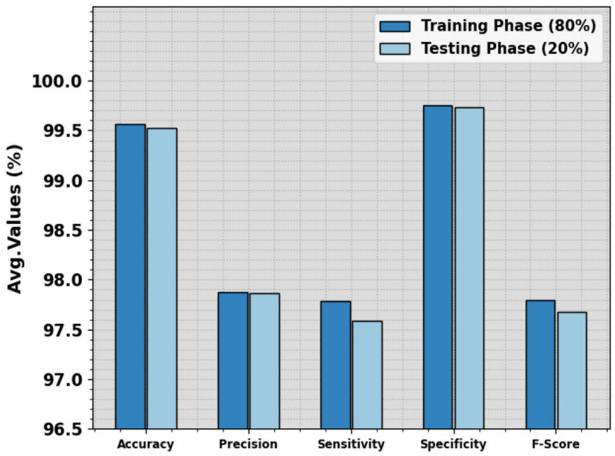

FIGURE 4. Average of ERSO-DSEL technique under 80:20 of TRAP/TESP.

and Fig. 4. The experimental results highlighted that the ERSO-DSEL method properly recognized different types of classes. With 80% of TRAP, the ERSO-DSEL approach offers an average $accu_y$ of 99.56%, $prec_n$ of 97.87%, $sens_y$ of 97.79%, $spec_y$ of 99.75%, and F_{score} of 97.80%. Additionally, with 20% of TESP, the ERSO-DSEL system offers an average $accu_y$ of 99.52%, $prec_n$ of 97.86%, $sens_y$ of 97.59%, $spec_y$ of 99.73%, and F_{score} of 97.65%.

The $accu_y$ curves for training (TRA) and validation (VL) revealed in Fig. 5 for the ERSO-DSEL system below 80:20 of TRAP/TESP offers respected visions into its performance below numerous epochs. Mainly, there is a stable growth in both TRA and TES $accu_y$ to rising epochs, establishing the model's capability to recognize and learn patterns from both TRA and TES data. The increasing trend in TES $accu_y$ highlights the model's flexibility to the TRA dataset and its aptitude to make exact predictions on hidden data, highlighting robust generalized aptitudes.

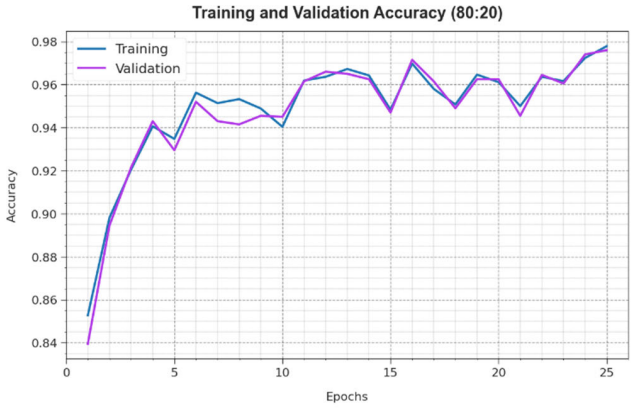

FIGURE 5. $Accu_y$ curve of ERSO-DSEL technique under 80:20 of TRAP/TESP.

Fig. 6 provides a comprehensive summary of the TRA and TES loss values for the ERSO-DSEL model below 80:20 of

FIGURE 6. Loss curve of ERSO-DSEL method under 80:20 of TRAP/TESP.

TRAP/TESP across different epochs. The TRA loss reliably diminishes as the technique enhances its weights to minimize identification faults on both datasets. The loss curves demonstrate the model's place with the TRA data, underlining its aptitude to take designs well in both datasets. Noteworthy is the nonstop modification of parameters in the ERSO-DSEL approach, proposed at diminishing differences amongst estimates and real TRA labels.

Regarding the PR curve offered in Fig. 7, the results confirm that the ERSO-DSEL approach under 80:20 of TRAP/TESP reliably attains enhanced PR values through every class. These outcomes underline the model's actual ability to discriminate among different classes, underlining its effectiveness in precisely spotting class labels.

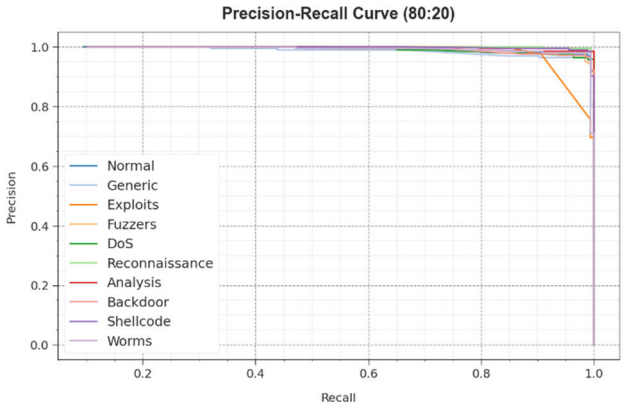

FIGURE 7. PR curve of ERSO-DSEL technique under 80:20 of TRAP/TESP.

Furthermore, in Fig. 8, we present ROC curves formed by the ERSO-DSEL model under 80:20 of TRAP/TESP, representing its aptitude in distinguishing between classes. These curves deliver valuable insights into how the trade-off amid TPR and FPR diverges across dissimilar classification epochs and thresholds. The results emphasize the model's accurate classification performance below several classes, highlighting its efficacy in addressing various classification challenges.

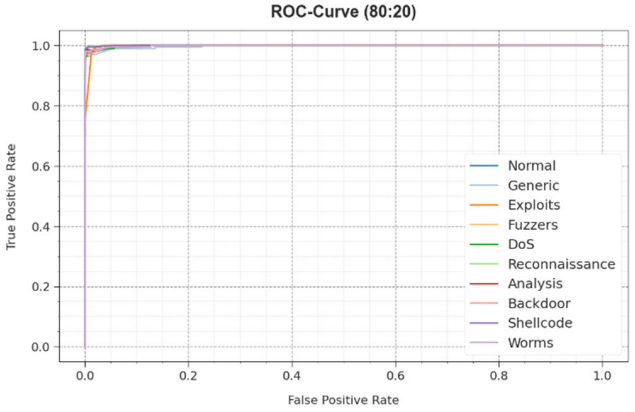

FIGURE 8. ROC curve of ERSO-DSEL technique under 80:20 of TRAP/TESP.

The cyberattack recognition outcomes of the ERSO-DSEL method are observed with 70:30 of TRAP/TESP in Table 3 and Fig. 9. The experimental outcomes underlined that the ERSO-DSEL method correctly recognized dissimilar kinds of classes. With 70% of TRAP, the ERSO-DSEL model provides an average $accu_y$ of 99.60%, $prec_n$ of 97.99%, $sens_y$ of 97.99%, $spec_y$ of 99.78%, and F_{score} of 97.99%. Moreover, with 30% of TESP, the ERSO-DSEL system delivers an average $accu_y$ of 99.67%, $prec_n$ of 98.37%, $sens_y$ of 98.34%, $spec_y$ of 99.82%, and F_{score} of 98.35%.

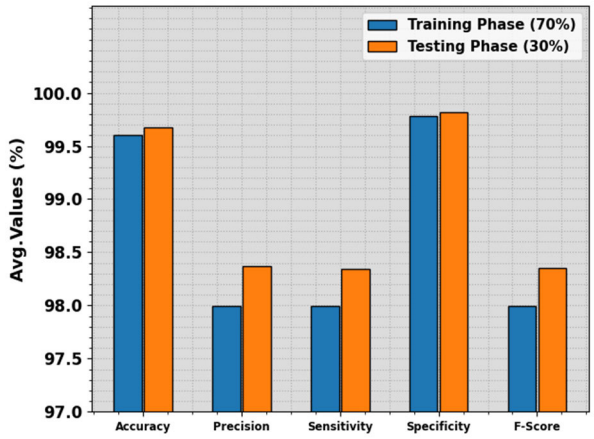

FIGURE 9. Average of ERSO-DSEL technique under 70:30 of TRAP/TESP.

The $accu_y$ curves for TRA and VL offered in Fig. 10 for the ERSO-DSEL system below 70:30 of TRAP/TESP deliver valuable visions into its performance under various epochs. Mainly, there is endless growth in both TRA and TES $accu_y$ to increasing epochs, specifying the model's aptitude in recognizing and learning patterns from both TRA and TES data. The rising trend in TS $accu_y$ emphasizes the model's flexibility to the TRA dataset and its aptitude to make exact predictions on unseen data, underlining strong generalized abilities.

TABLE 3. Cyberattack detection outcome of ERSO-DSEL technique under 70:30 of TRAP/TESP.

Classes	$Accu_y$	$Prec_n$	$Sens_y$	$Spec_y$	F_{Score}
TRAP (70%)					
Normal	99.49	97.49	97.49	99.71	97.49
Generic	99.61	98.14	98.00	99.79	98.07
Exploits	99.64	98.67	97.66	99.86	98.17
Fuzzers	99.50	97.78	97.38	99.75	97.58
DoS	99.59	97.68	98.11	99.75	97.90
Reconnaissance	99.60	97.65	98.47	99.73	98.06
Analysis	99.70	98.42	98.57	99.83	98.49
Backdoor	99.69	98.53	98.25	99.84	98.39
Shellcode	99.53	97.54	97.68	99.73	97.61
Worms	99.63	98.00	98.28	99.78	98.14
Average	99.60	97.99	97.99	99.78	97.99
TESP (30%)					
Normal	99.53	97.86	97.17	99.78	97.52
Generic	99.70	97.71	99.34	99.74	98.52
Exploits	99.77	98.73	99.05	99.85	98.89
Fuzzers	99.67	98.89	97.45	99.89	98.17
DoS	99.73	98.10	99.36	99.78	98.73
Reconnaissance	99.67	98.23	98.23	99.82	98.23
Analysis	99.77	99.33	98.35	99.93	98.84
Backdoor	99.80	98.74	99.37	99.85	99.05
Shellcode	99.67	98.39	98.39	99.81	98.39
Worms	99.43	97.65	96.68	99.74	97.16
Average	99.67	98.37	98.34	99.82	98.35

FIGURE 10. $Accu_y$ curve of ERSO-DSEL technique under 70:30 of TRAP/TESP.

Fig. 11 provides a comprehensive summary of the TRA and TES loss values for the ERSO-DSEL method under 70:30 of TRAP/TESP across several epochs. The TRA loss gradually declines as the method increases its weights to condense classification faults on both datasets. The loss curves prove the model's placement with the TRA data and, the prominence of its capacity to capture designs well in both datasets. Noteworthy is the continuous modification of parameters in the ERSO-DSEL methodology, projected at diminishing differences among predictions and real TRA labels.

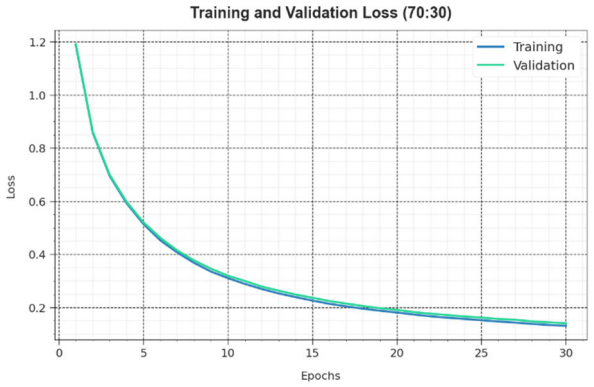

FIGURE 11. Loss curve of ERSO-DSEL technique under 70:30 of TRAP/TESP.

Regarding the PR curve offered in Fig. 12, the outcomes confirm that the ERSO-DSEL method under 70:30 of TRAP/TESP constantly attains upgraded PR values across each class. These results emphasize the model's real ability to discriminate amid different classes, emphasizing its efficiency in accurately recognizing classes.

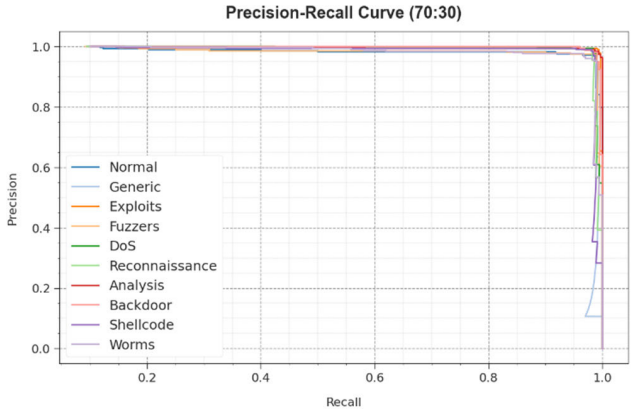

FIGURE 12. PR curve of ERSO-DSEL technique under 70:30 of TRAP/TESP.

Furthermore, in Fig. 13, we present ROC curves formed by the ERSO-DSEL procedure under 70:30 of TRAP/TESP, signifying its ability to distinguish between classes. These curves deliver valuable visions into how the trade-off between TPR and FPR differs across dissimilar classification epochs and thresholds. The results highlight the model's precise identification performance below several class labels, underscoring its efficiency in addressing various classification tasks.

In Table 4 and Fig. 14, a detailed comparative analysis of the ERSO-DSEL technique is provided [30]. The results stated that the SVM model gains poor performance whereas the KNN, DT, VLSTM, and SSA-CRNN methodologies obtain boosted results. Besides that, the MFSDL-ADIIoT method has managed to report moderate performance.

Also, the GJODL-CADC model has tried to accomplish near-optimal results with $prec_n$ of 97.23%, $reca_l$ of 97.09%, $accu_y$ of 99.40%, and F_{score} of 97.12%. But the ERSO-DSEL

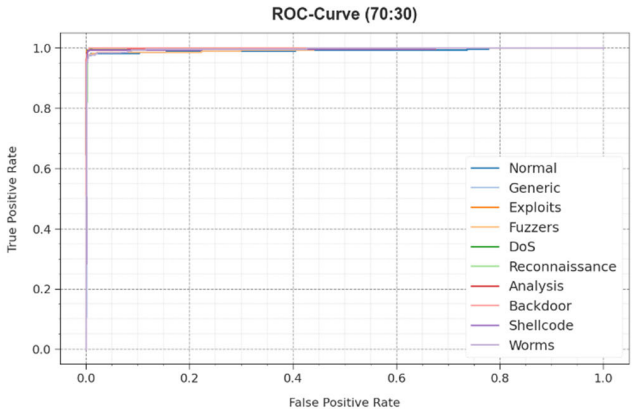

FIGURE 13. ROC curve of ERSO-DSEL technique under 70:30 of TRAP/TESP.

TABLE 4. Comparative analysis of the ERSO-DSEL technique with recent algorithms.

Methods	$Prec_n$	$Reca_l$	$Accu_y$	F_{Score}
kNN	62.80	53.03	70.99	52.72
SVM	50.60	53.18	64.24	53.24
DT	64.05	53.33	70.51	48.61
VLSTM	66.86	53.05	95.86	58.58
SSA-CRNN	66.91	58.94	98.60	59.69
MFSDL-ADIIoT	67.04	60.28	99.14	60.29
GJODL-CADC	97.23	97.09	99.40	97.12
ERSO-DSEL	98.37	98.34	99.67	99.35

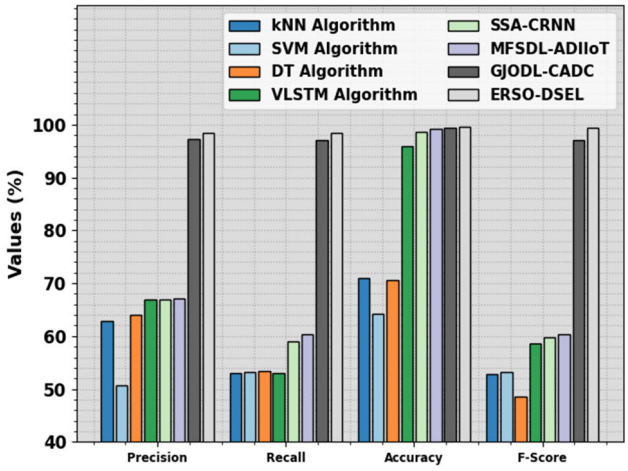

FIGURE 14. Comparative analysis of the ERSO-DSEL technique with recent algorithms.

system attains better performance with a maximum $prec_n$ of 98.37%, $reca_l$ of 98.34%, $accu_y$ of 99.67%, and F_{score} of 99.35%. Therefore, the ERSO-DSEL technique can be applied for an enhanced detection process.

IV. CONCLUSION

In this manuscript, cybersecurity using the ERSO-DSEL approach is empowered. The ERSO-DSEL methodology

leverages FS with EL strategies to boost cybersecurity. The ERSO-DSEL methodology involves four major procedures Z-score standardization, IEO-based FS, deep stack-based EL, and ERSO-based hyperparameter tuning. Initially, the ERSO-DSEL model applies Z-score standardization is employed to measure the input data. Besides, IEO based FS approach is applied to elect a set of features. For cyberattack recognition, the ERSO-DSBEL model uses the DSEL approach comprising three models namely DNN, LSTM, and BiLSTM. Furthermore, the hyperparameter selection of these DL models takes place using the ERSO approach. The performance validation of the ERSO-DSBEL model is implemented on a benchmark IDS dataset. A wide comparison research reported that the ERSO-DSBEL model accomplishes enhanced performance over other models of cybersecurity. The restriction of the ERSO-DSEL model includes additional analysis across diverse cybersecurity scenarios and potential threats in scalability and real-time utilization. Future studies may be on improving the adaptability to improve cyberthreats and optimizing its achievement in large-scale network environments.

REFERENCES

[1] K. Tomar, K. Bisht, K. Joshi, and R. Katarya, "Cyber attack detection in IoT using deep learning techniques," in *Proc. 6th Int. Conf. Inf. Syst. Comput. Netw. (ISCON)*, Mathura, India, Mar. 2023, pp. 1–6.

[2] R. Golchha, A. Joshi, and G. P. Gupta, "Voting-based ensemble learning approach for cyber attacks detection in Industrial Internet of Things," *Proc. Comput. Sci.*, vol. 218, pp. 1752–1759, Jan. 2023.

[3] A. Al-Abassi, H. Karimipour, A. Dehghantanha, and R. M. Parizi, "An ensemble deep learning-based cyber-attack detection in industrial control system," *IEEE Access*, vol. 8, pp. 83965–83973, 2020.

[4] F. Li, J. Lin, and H. Han, "FSL: Federated sequential learning-based cyberattack detection for Industrial Internet of Things," *Ind. Artif. Intell.*, vol. 1, no. 1, pp. 2–14, Mar. 2023.

[5] F. Khan, M. A. Jan, R. Alturki, M. D. Alshehri, S. T. Shah, and A. U. Rehman, "A secure ensemble learning-based fog-cloud approach for cyberattack detection in IoMT," *IEEE Trans. Ind. Informat.*, vol. 19, no. 10, pp. 10125–10132, Oct. 2023.

[6] A. Yazdinejad, M. Kazemi, R. M. Parizi, A. Dehghantanha, and H. Karimipour, "An ensemble deep learning model for cyber threat hunting in the Industrial Internet of Things," *Digit. Commun. Netw.*, vol. 9, pp. 101–110, Feb. 2023.

[7] H. Alkahtani and T. H. H. Aldhyani, "Intrusion detection system to advance Internet of Things infrastructure-based deep learning algorithms," *Complexity*, vol. 2021, Jul. 2021, Art. no. 5579851.

[8] T. Alatawi and A. Aljuhani, "Anomaly detection framework in fog-to-things communication for Industrial Internet of Things," *Comput., Mater. Continua*, vol. 73, no. 1, pp. 1067–1086, 2022.

[9] H. M. Rouzbahani, A. H. Bahrami, and H. Karimipour, "A snapshot ensemble deep neural network model for attack detection in Industrial Internet of Things," in *AI-Enabled Threat Detection and Security Analysis for Industrial IoT*. Cham, Switzerland: Springer, 2021, pp. 181–194.

[10] S. Nayak, N. Ahmed, and S. Misra, "Deep learning-based reliable routing attack detection mechanism for Industrial Internet of Things," *Ad Hoc Netw.*, vol. 123, Dec. 2021, Art. no. 102661.

[11] R. Bingu and S. Jothilakshmi, "Design of intrusion detection system using ensemble learning technique in cloud computing environment," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 5, pp. 751–764, 2023.

[12] S. Ennaji, N. E. Akkad, and K. Haddouch, "A powerful ensemble learning approach for improving network intrusion detection system (NIDS)," in *Proc. 5th Int. Conf. Intell. Comput. Data Sci. (ICDS)*, Fez, Morocco, Oct. 2021, pp. 1–6, doi: 10.1109/ICDS53782.2021.9626727.

[13] D. N. Mhawi, A. Aldallal, and S. Hassan, "Advanced feature-selection-based hybrid ensemble learning algorithms for network intrusion detection systems," *Symmetry*, vol. 14, no. 7, p. 1461, Jul. 2022.

[14] B. A. K. Hammood and A. T. Sadiq, "Ensemble machine learning approach for IoT intrusion detection systems," *Iraqi J. Comput. Informat.*, vol. 49, no. 2, pp. 93–99, Dec. 2023, doi: 10.25195/ijci.v49i2.458.

[15] H.-C. Lin, P. Wang, K.-M. Chao, W.-H. Lin, and Z.-Y. Yang, "Ensemble learning for threat classification in network intrusion detection on a security monitoring system for renewable energy," *Appl. Sci.*, vol. 11, no. 23, p. 11283, Nov. 2021.

[16] V. Agate, D. A. F. Maria, A. De Paola, P. Ferraro, G. L. Re, and M. Morana, "A behavior-based intrusion detection system using ensemble learning techniques," in *Proc. ITASEC*, 2022. [Online]. Available: https://ceur-ws.org/Vol-3260/paper15.pdf

[17] A. K. Mananayaka and S. S. Chung, "Network intrusion detection with two-phased hybrid ensemble learning and automatic feature selection," *IEEE Access*, vol. 11, pp. 45154–45167, 2023, doi: 10.1109/ACCESS.2023.3274474.

[18] O. D. Okey, S. S. Maidin, P. Adasme, R. L. Rosa, M. Saadi, D. C. Melgarejo, and D. Z. Rodríguez, "BoostedEnML: Efficient technique for detecting cyberattacks in IoT systems using boosted ensemble machine learning," *Sensors*, vol. 22, no. 19, p. 7409, Sep. 2022.

[19] A. Terbuch, P. O'Leary, N. Khalili-Motlagh-Kasmaei, P. Auer, A. Zöhrer, and V. Winter, "Detecting anomalous multivariate time-series via hybrid machine learning," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–11, 2023.

[20] A. Presekal, A. Stefanov, V. S. Rajkumar, and P. Palensky, "Attack graph model for cyber-physical power systems using hybrid deep learning," *IEEE Trans. Smart Grid*, vol. 14, no. 5, pp. 4007–4020, Jan. 2023.

[21] K. Wang, Y. Fu, X. Duan, T. Liu, and J. Xu, "Abnormal traffic detection system in SDN based on deep learning hybrid models," *Comput. Commun.*, vol. 216, pp. 183–194, Feb. 2024.

[22] Y. K. Saheed, O. H. Abdulganiyu, and T. A. Tchakoucht, "A novel hybrid ensemble learning for anomaly detection in industrial sensor networks and SCADA systems for smart city infrastructures," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 35, no. 5, May 2023, Art. no. 101532.

[23] J. Manokaran and G. Vairavel, "IGWO-SoE: Improved grey wolf optimization based stack of ensemble learning algorithm for anomaly detection in Internet of Things edge computing," *IEEE Access*, vol. 11, pp. 106934–106953, 2023.

[24] X. Liang, Y. Gao, and S. Xu, "ASE: Anomaly scoring based ensemble learning for highly imbalanced datasets," *Expert Syst. Appl.*, vol. 238, Mar. 2024, Art. no. 122049.

[25] H. A. Prihanditya, "The implementation of Z-score normalization and boosting techniques to increase accuracy of C4.5 algorithm in diagnosing chronic kidney disease," *J. Soft Comput. Exploration*, vol. 1, no. 1, pp. 63–69, Sep. 2020.

[26] L. Yue, P. Hu, S.-C. Chu, and J.-S. Pan, "Equilibrium optimizer for emotion classification from English speech signals," *IEEE Access*, vol. 11, pp. 134217–134229, 2023.

[27] E. Lodhi, F.-Y. Wang, G. Xiong, L. Zhu, T. S. Tamir, W. U. Rehman, and M. A. Khan, "A novel deep stack-based ensemble learning approach for fault detection and classification in photovoltaic arrays," *Remote Sens.*, vol. 15, no. 5, p. 1277, Feb. 2023.

[28] G. Dhiman, M. Garg, A. Nagar, V. Kumar, and M. Dehghani, "A novel algorithm for global optimization: Rat swarm optimizer," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 8, pp. 8457–8482, Aug. 2021.

[29] Accessed: Aug. 8, 2023. [Online]. Available: https:// research.unsw.edu.au/projects/unsw-nb15-dataset

[30] L. A. Maghrabi, I. R. Alzahrani, D. Alsalman, Z. M. AlKubaisy, D. Hamed, and M. Ragab, "Golden jackal optimization with a deep learning-based cybersecurity solution in Industrial Internet of Things systems," *Electronics*, vol. 12, no. 19, p. 4091, Sep. 2023, doi: 10.3390/electronics12194091.