Computer Systems Science & Engineering
DOI: 10.32604/csse.2023.031063
Article

Tech Science Press

Hybrid Metaheuristics Feature Selection with Stacked Deep Learning-Enabled Cyber-Attack Detection Model

Mashael M Asiri[1], Heba G. Mohamed[2], Mohamed K Nour[3], Mesfer Al Duhayyim[4,*], Amira Sayed A. Aziz[5], Abdelwahed Motwakel[6], Abu Sarwar Zamani[6] and Mohamed I. Eldesouki[7]

[1]Department of Computer Science, College of Science & Art at Mahayil, King Khalid University, Muhayel Aseer, 62529, Saudi Arabia
[2]Department of Electrical Engineering, College of Engineering, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia
[3]Department of Computer Sciences, College of Computing and Information System, Umm Al-Qura University, Saudi Arabia
[4]Department of Computer Science, College of Sciences and Humanities- Aflaj, Prince Sattam bin Abdulaziz University, Saudi Arabia
[5]Department of Digital Media, Faculty of Computers and Information Technology, Future University in Egypt, New Cairo, 11835, Egypt
[6]Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, Al-Kharj, 16278, Saudi Arabia
[7]Department of Information System, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, AlKharj, Saudi Arabia
*Corresponding Author: Mesfer Al Duhayyim. Email: m.alduhayyim@psau.edu.sa

Received: 09 April 2022; Accepted: 08 June 2022

Abstract: Due to exponential increase in smart resource limited devices and high speed communication technologies, Internet of Things (IoT) have received significant attention in different application areas. However, IoT environment is highly susceptible to cyber-attacks because of memory, processing, and communication restrictions. Since traditional models are not adequate for accomplishing security in the IoT environment, the recent developments of deep learning (DL) models find beneficial. This study introduces novel hybrid metaheuristics feature selection with stacked deep learning enabled cyber-attack detection (HMFS-SDLCAD) model. The major intention of the HMFS-SDLCAD model is to recognize the occurrence of cyberattacks in the IoT environment. At the preliminary stage, data pre-processing is carried out to transform the input data into useful format. In addition, salp swarm optimization based on particle swarm optimization (SSOP-SO) algorithm is used for feature selection process. Besides, stacked bidirectional gated recurrent unit (SBiGRU) model is utilized for the identification and classification of cyberattacks. Finally, whale optimization algorithm (WOA) is employed for optimal hyperparameter optimization process. The experimental analysis of the HMFS-SDLCAD model is validated using benchmark dataset and the results are assessed under several aspects. The simulation outcomes pointed out the improvements of the HMFS-SDLCAD model over recent approaches.

Keywords: Cyberattacks; security; deep learning; internet of things; feature selection; data classification

1 Introduction

The Internet of things (IoT) consists of a compilation of heterogeneous resource-restrained objects interlinked through distinct network frameworks, namely wireless sensor networks (WSNs) [1]. These "things" or objects are generally made up of processors, sensors, and actuators with the capability to interact with one another for achieving a common objective or applications by unique identifiers in relation to the Internet protocol (IP) [2]. Recent IoT applications involve smart buildings, agriculture, industrial and manufacturing processes, aerospace and aviation, telecommunications, medical and pharmaceutical, and environmental phenomenon monitoring [3]. The fundamental IoT layered structure consists of 3 layers firstly the perception layer (comprising edge devices which interact with the environment for identifying specific external elements or other smart objects in the environment), secondly the network layer (made up of number of networking devices which finds and links devices beyond the IoT network for sending and receiving the sensed data), and lastly the application layer (made up of several IoT services or applications which is accountable for storage and data processing). Many cyber-attacks focus on the network and application layers of the IoT system [4]. After the IoT architecture is breached, attackers have the capability for sharing the IoT data with unapproved crews and may control consistency and preciseness of the IoT data over its whole life cycle [5]. Thus, these cyber-attacks must be addressed for utilization of safe IoT. Fig. 1 depicts the role of machine learning (ML) in cybersecurity.

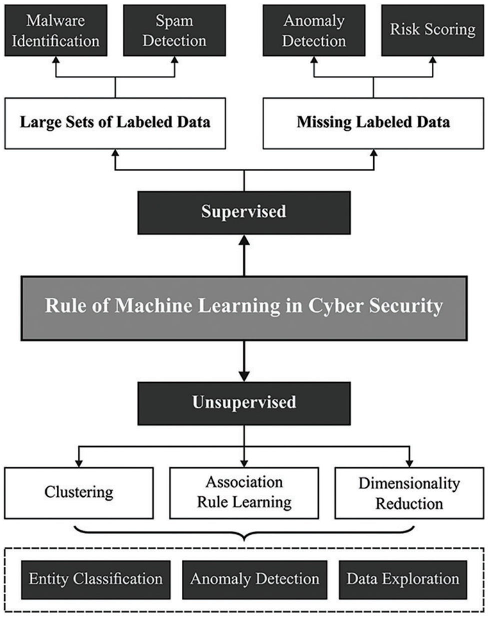


Figure 1: Role of ML in cybersecurity

Network intrusion identification approaches achieve progression from mechanisms lying on port inspection to methods making complete use of ML [6]. The normal port-related approaches are outdated since recent applications majorly depend on dynamic port allotment instead of registered port numbers [7]. The rise in the proportion of encrypted traffic drives the failure of payload-related methodologies. This guides the cybersecurity experts in the direction of using ML and network flow features. Current developments in ML methodologies for network anomaly identification were most welcomed [8,9]. Owing to the diverse and heterogeneous nature of cloud environments, ML offers responses to the difficulties impelled because of the availability of virtualized environments with its vast range of application workloads [10].

Panda et al. [11] utilized the University of New South Wales (UNSW)-NB15, a novel IoT-Botnet data (imbalanced and noisy dataset) to categorize cyberattacks. Scatter search-based feature engineering and K-Medoid sampling methods are utilized for obtaining representation data with optimum feature sets. Al-Haija [12] proposed an effectual and generic top-down structure for intrusion classification, along with recognition in IoT networks through non-conventional ML technique is presented. The presented method is personalized and utilized for intrusion classification/detection integrating IoT cyber-attack data, namely MeSSOge Queuing Telemetry Transport (MQTT) dataset, CICIDS Dataset, etc. Especially, the presented method is comprised of detection and classification (DC) subsystems, feature engineering (FE) subsystems, and feature learning (FL) subsystems. In [13], a hybrid deep random neural network (HDRaNN) for detecting cyber-attack in the IIoT is proposed. The presented method integrates a multilayer perceptron with dropout regularization and deep random neural network.

Amma [14] proposed a Vector Convolution Deep Autonomous Learning (VCDAL) classification for detecting cyberattacks in the network traffic dataset. The presented method classification extracts the feature through vector convolution neural network (CNN), automatically learns the feature via increment learning using distilled cross entropy, as well as classifies the developing network traffic dataset via softmax function. The presented classification has been by implementing experiments on standard network traffic data sets and it is clear that the presented classification could probably identify known and unknown cyberattacks. An et al. [15–18] presented an unsupervised ensemble autoencoder (AE) interconnected with the Gaussian mixture method (GMM) for adapting various fields nevertheless of the skewness of all the domains. In the hidden region of the ensemble AE, the attention-based latent representation and recreated feature of the minimal error are employed.

This study introduces novel hybrid metaheuristics feature selection with stacked deep learning enabled cyber-attack detection (HMFS-SDLCAD) model. The major intention of the HMFS-SDLCAD model is to design salp swarm optimization based on particle swarm optimization (SSOPSO) algorithm for feature selection process. Moreover, stacked bidirectional gated recurrent unit (SBiGRU) model is utilized for the identification and classification of cyberattacks. At last, whale optimization algorithm (WOA) is employed for optimal hyperparameter optimization process. The experimental analysis of the HMFS-SDLCAD model is validated using benchmark dataset and the results are assessed under several aspects.

2 The Proposed Cyber-Attack Detection Model

In this study, a new HMFS-SDLCAD model has been developed to recognize the occurrence of cyberattacks in the IoT environment. At the preliminary stage, data pre-processing is carried out to transform the input data into useful format. Then, the SSOPSO algorithm is utilized to elect features. In addition, the WOA with SBiGRU model is utilized for the identification and classification of cyberattacks. Fig. 2 demonstrates the block diagram of HMFS-SDLCAD technique.

2.1 Process Involved in SSOPSO Based FS Model

In this work, the SSOPSO algorithm is employed to choose an optimal subset of features from the preprocessed data. The framework of the presented technique is explained. It is named SSOPSO that integrates the SSO and PSO approaches. The fundamental infrastructure of SSO technique was improved by enhancing the upgrade step of population place. This alteration merges the upgrade process of PSO as to important infrastructure of SSO. This combination adds further flexibility to SSO in exploring the population and makes sure its diversity of it and attains the optimum value rapidly.

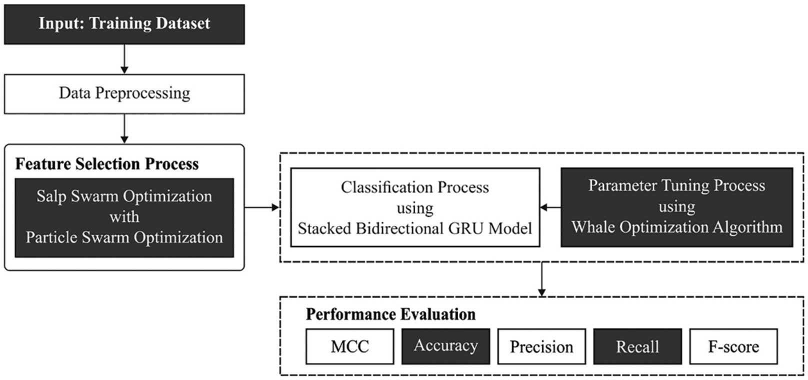

Input: Training Dataset
Data Preprocessing
Feature Selection Process
Salp Swarm Optimization with Particle Swarm Optimization
Classification Process using Stacked Bidirectional GRU Model
Parameter Tuning Process using Whale Optimization Algorithm
Performance Evaluation
MCC
Accuracy
Precision
Recall
F-score

Figure 2: Block diagram of HMFS-SDLCAD technique

In the primary stage, the presented SSOPSO is used for determining the parameter and creating the population that signifies the group of solutions to offered problem (feature selection) [19]. Next, the performance of all the solutions is measured by calculating the fitness function (FF) for everyone and defining the optimum of them. The next stage from the presented SSOPSO technique is for updating the existing population by utilizing also the SSO or PSO technique that depends upon the quality of FF (evaluated by their probability). When the probability of FF, to the present solution, is superior to 0.5 then SSO, else, the PSO was utilized. Next, the FF to all the solutions was calculated and optimum solution was defined then upgrades the population. The next stage is for checking when the end criteria are fulfilled before returning by optimum solution, then, repeating the preceding stages in calculating the probability to end.

The SSOPSO technique begins with determining the primary value of SSO and PSO techniques, next the SSO creates an arbitrary population X of size N in dimensional D, next SSO computes the food fitness to all the solutions $x_i,\ i = 1,\ 2,\ \ldots,\ N$. But, before calculating the objective function, all the solutions x_i was changed to binary vector (that comprises only 1's and $0's$) based on the value of an arbitrary threshold $e \in [O,\ 1]$ utilizing the subsequent formula:

$$x_i(t+1) = \begin{cases} 1 & if\ \dfrac{1}{1+e^{-x_i(t)}} > e \\ 0 & otherwise \end{cases} \tag{1}$$

Thus, only the x_j element which is equivalent to 1's were selected for representing the chosen features (moreover, the other elements were ignored later which can signify the irrelevant feature). The next stage is for computing the objective function for all x_i as in Eq. (2):

$$f(x_i(t)) = \xi E_{x_i(t)} + (1-\xi)\left(\frac{|x_i(t)|}{|C|}\right), \tag{2}$$

whereas $E_{x_i(t)}$ implies the error of classifier executed by the effectual classification, but the second term signifies the amount of chosen features. For balancing amongst the classifier error and the amount of chosen features, the parameter $\xi \in [0,\ 1]$ was utilized. The next stage is for computing the probability of all the FFs (Pro_i) as:

$$Pro_j = \frac{f_i}{\sum_{i=1}^{N} f_i} \tag{3}$$

Based on the Pro_j value, the present solution x_j is upgraded utilizing the SSO or PSO techniques. The FF was calculated for all upgrade solutions, and optimum solution was upgraded. This sequence was iterated still meeting the end criteria (the presented SSOPSO technique executes to the max iteration number as ending criteria).

2.2 SBiGRU Based Classification

Once the feature subsets are chosen, the next step is to identify the cyberattacks using the SBiGRU model. The SBiGRU is comprised of forwarding and backwarding layers stacked on top of the other. The input dataset is given to the initial forward and backward layers. The output is a sequence of latter forward and backward layers [20]. For time series t, the input series $\{e,\ e\ldots,\ e_t\}$ entered hidden layer in the forward direction $\{h_1^a,\ h_2^a,\ \ldots,\ h_t^a\}$ for obtaining comprehensive dataset from each historical time step and entering hidden layer in the reverse direction $\{h_1^c,\ h_2^c,\ \ldots,\ h_t^c\}$ for obtaining comprehensive data from each future time step. Next, the upper hidden layer takes the output from the low hidden layer as input for extracting features. Especially, the upper layer of the forwarding hidden layer is $\{h_1^b,\ h_2^b,\ \ldots,\ h_t^b\}$, and the upper layer of the backward hidden layer is $\{h_1^d,\ h_2^d,\ \ldots,\ h_t^d\}$. Lastly, the output layer integrates the hidden vector of two upper layers as output. For the initial forward layer, hidden layer h_t^a, is to attain the candidate value, update, and reset gates, correspondingly:

$$u_t^a = \sigma(W_u^a h_{t-1}^a + U_u^a e_t + b_u^a) \tag{4}$$

$$r_t^a = \sigma(W_r^a h_{t-1}^a + U_r^a e_t + b_r^a) \tag{5}$$

$$\tilde{C} = \tanh(W_c^a.\ [r_t^a * h_{t-1}^a] + U_c^a e_t + b_c^a) \tag{6}$$

$$h_t^a = u_t^a * \tilde{C}_{\mathrm{t}}^{\mathrm{a}} + (1 - u_t^a) * h_{t-1}^a \tag{7}$$

In the next forward layer, the hidden layer h_t^b, is to attain the candidate value, update, and reset gates, correspondingly:

$$u_t^b = \sigma(W_u^b h_{t-1}^b + U_u^b h_t^a + b_u^b) \tag{8}$$

$$r_t^b = \sigma(W_r^b h_{t-1}^b + U_r^b h_t^a + b_r^b) \tag{9}$$

$$\tilde{C} = \ \tanh\ (W_c^b.\ [r_t^b * h_{t-1}^b] + U_c^b h_t^a + b_c^b) \tag{10}$$

$$h_t^b = u_t^b * \tilde{C} + (1 - u_t^b) * h_{t-1}^b \tag{11}$$

In the initial backward layer, hidden layer h_t^c, is to attain the candidate value, update, and reset gates, correspondingly:

$$u_t^c = \sigma(W_u^c h_{t+1}^c + U_u^c e_t + b_u^c) \tag{12}$$

$$r_t^c = \sigma(W_r^c h_{t+1}^c + U_r^c e_t + b_r^c) \tag{13}$$

$$\tilde{C} = \ \tanh\ (W_c^c.\ [r_t^c * h_{t+1}^c] + U_c^c e_t + b_c^c) \tag{14}$$

$$h_t^c = u_t^c * \tilde{C}_{\mathrm{t}}^{\mathrm{c}} + (1 - u_t^c) * h_{t-1}^c \tag{15}$$

In the next backward layer, the hidden layer h_t^d, is to attain the candidate value, update, and reset gates, correspondingly:

$$u_t^d = \sigma(W_u^d h_{\imath+1}^d + U_u^d h_t^c + b_u^d) \tag{16}$$

$$r_t^d = \sigma(W_r^d h_{\imath+1}^d + U_r^d h_t^c + b_r^d) \tag{17}$$

$$\tilde{C} = tanh(W_c^d .[r_t^d * h_{\imath+1}^d] + U_c^d h_t^c + b_c^d) \tag{18}$$

$$h_t^d = u_t^d * \tilde{C}_{\mathrm{t}}^{\mathrm{d}} + (1 - u_t^d) * h_{t-1}^d \tag{19}$$

The output of next forward and backward layers is given in the following:

$$O_t = U^o h_t^b + W^o h_t^d + b^o \tag{20}$$

2.3 WOA Based Hyperparameter Optimization

Finally, the WOA is utilized for optimal hyperparameter optimization process. Mirjalili et al. [21] presented the WOA stimulated by the whale behavior. The foraging behavior is named bubble-net feeding technique. But, in WOA, the existing optimal candidate solution is to set the target prey or closer to the optimal. The other tries to upgrade the location towards the optimal one. Arithmetically, it can be expressed in the following:

$$D = |C \cdot X^*(t) - X(t)| \tag{21}$$

$$X(t + l) = X^*(t + 1) - A \cdot D \tag{22}$$

whereas t indicates the existing iteration, X denotes the location vector, X^* represent the location vector coincides with the optimal solution found, and A and C denote the coefficient vectors. A and C are determined as follows:

$$A = 2 \cdot a. r - a \tag{23}$$

$$C = 2 \cdot r \tag{24}$$

whereas r is positioned arbitrarily within $[0, 1]$ and a is reduced linearly from 2 to 0. This method has two stages: exploration and exploitation. The exploitation stage: is separated into; (1) shrinking encircling mechanism: This is attained by reducing a value. Noted that a indicates an arbitrary number within $[-a, a]$.

Spiral updating location: This technique estimates the distance among the whale and the prey. A spiral formula is utilized for mimicking the helix-shaped movement:

$$X(t + 1) = D^l e^{bl} \cdot \cos(2\pi l) + X^*(t) \tag{25}$$

whereas 1 denotes an arbitrary value within [1,1] and b denotes a constant. A possibility of 50% to choose among the shrinking encircling model or the spiral model. Therefore, the arithmetical method is expressed in the following:

$$X(t + 1) = \begin{cases} X^*(t) - .A \cdot D & ifp < 0.5 \\ D^1 \cdot e^{bl} \cos(2\pi l) + X^*(t) & ifp \geq 0.5 \end{cases} \tag{26}$$

In which p denotes an arbitrary value in a uniform distribution. The exploration stage:

While, in the exploration stage, A has utilized arbitrary value in $1 \prec A \prec -1$ to force the agent to move away from the position and arithmetically expressed in the following:

$$D = |C \cdot X_{rand} - X| \tag{27}$$

$$X(t+1) = X_{r\ and} - A \cdot D \tag{28}$$

3 Experimental Validation

In this section, the experimental validation of the HMFS-SDLCAD model is tested using a benchmark dataset, available at https://dataset.litnet.lt/. The dataset holds samples under 12 class labels and 84 features. The proposed model has chosen a set of 47 features. Tab. 1 provides the details related to the dataset.

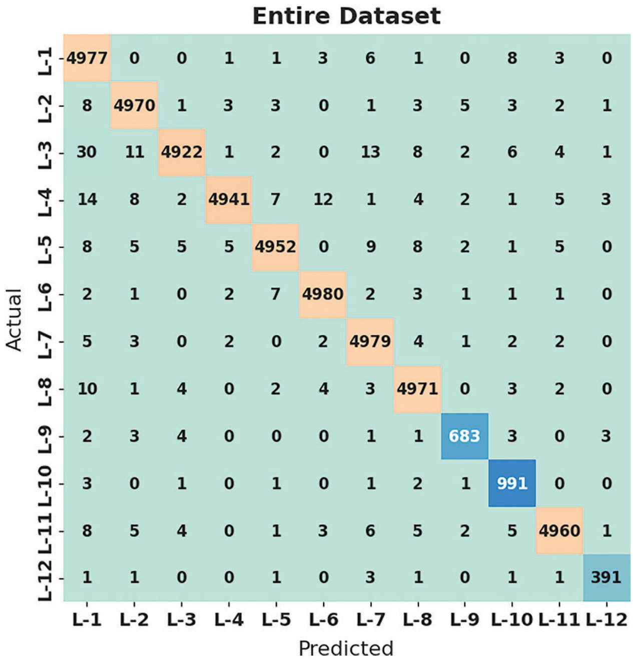
Table 1: Dataset details

Class labels	Attack type	Attacks	For experimentation
Label-1	Smurf	59479	5000
Label-2	ICMP-flood	11628	5000
Label-3	UDP-flood	59479	5000
Label-4	TCP SYN-flood	3725838	5000
Label-5	HTTP-flood	22959	5000
Label-6	LANDattack	52417	5000
Label-7	Blaster worm	24291	5000
Label-8	Code red worm	1255702	5000
Label-9	Spam bot's detection	747	700
Label-10	Reaper worm	1176	1000
Label-11	Scanning/spread	6232	5000
Label-12	Packet fragmentation attack	477	400
Total No. of Samples		5220425	47100

Fig. 3 indicates the confusion matrix created by the HMFS-SDLCAD model on the classification of cyberattacks under entire dataset. The figure indicated that the HMFS-SDLCAD model has identified all the 12 classes effectively.

Tab. 2 and Fig. 4 highlight the overall classification outcomes of the HMFS-SDLCAD model on entire dataset. The experimental values indicated that the HMFS-SDLCAD model has gained maximum classifier results under all class labels. For instance, with label-1, the HMFS-SDLCAD model has offered $accu_y$, $prec_n$, $reca_l$, F_{score}, and Mathew Correlation Coefficient (MCC) of 99.76%, 98.20%, 99.54%, 98.87%, and 98.74% respectively. Also, with label-5, the HMFS-SDLCAD approach has offered $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC of 99.85%, 99.50%, 99.04%, 99.27%, and 99.18% correspondingly. Moreover, with label-10, the HMFS-SDLCAD algorithm has offered $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC of 99.91%, 96.68%, 99.10%, 97.88%, and 97.84% correspondingly. Furthermore, with label-12, the HMFS-SDLCAD system has obtainable $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC of 99.96%, 97.75%, 97.75%, 97.75%, and 97.73% correspondingly.

Fig. 5 designates the confusion matrix created by the HMFS-SDLCAD approach on the classification of cyberattacks under 70% of training set (TRS) dataset. The figure represented that the HMFS-SDLCAD methodology has identified all the 12 classes effectively.



	L-1	L-2	L-3	L-4	L-5	L-6	L-7	L-8	L-9	L-10	L-11	L-12
L-1	4977	0	0	1	1	3	6	1	0	8	3	0
L-2	8	4970	1	3	3	0	1	3	5	3	2	1
L-3	30	11	4922	1	2	0	13	8	2	6	4	1
L-4	14	8	2	4941	7	12	1	4	2	1	5	3
L-5	8	5	5	5	4952	0	9	8	2	1	5	0
L-6	2	1	0	2	7	4980	2	3	1	1	1	0
L-7	5	3	0	2	0	2	4979	4	1	2	2	0
L-8	10	1	4	0	2	4	3	4971	0	3	2	0
L-9	2	3	4	0	0	0	1	1	683	3	0	3
L-10	3	0	1	0	1	0	1	2	1	991	0	0
L-11	8	5	4	0	1	3	6	5	2	5	4960	1
L-12	1	1	0	0	1	0	3	1	0	1	1	391

Figure 3: Confusion matrix of HMFS-SDLCAD technique under entire dataset

Table 2: Result analysis of HMFS-SDLCAD technique with various measures on entire dataset

Entire dataset					
Class labels	Accuracy	Precision	Recall	F-Score	MCC
Label 1	99.76	98.20	99.54	98.87	98.74
Label 2	99.86	99.24	99.40	99.32	99.24
Label 3	99.79	99.58	98.44	99.00	98.89
Label 4	99.85	99.72	98.82	99.27	99.18
Label 5	99.85	99.50	99.04	99.27	99.18
Label 6	99.91	99.52	99.60	99.56	99.51
Label 7	99.86	99.08	99.58	99.33	99.25
Label 8	99.85	99.20	99.42	99.31	99.23
Label 9	99.93	97.71	97.57	97.64	97.61
Label 10	99.91	96.68	99.10	97.88	97.84
Label 11	99.86	99.50	99.20	99.35	99.27
Label 12	99.96	97.75	97.75	97.75	97.73
Average	99.86	98.81	98.96	98.88	98.81

Tab. 3 and Fig. 6 demonstrate the overall classification outcome of the HMFS-SDLCAD technique on 70% of TRS dataset. The experimental values represented that the HMFS-SDLCAD approach has reached maximum classifier results under all class labels. For instance, with label-1, the HMFS-SDLCAD approach has offered $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC of 99.77%, 98.29%, 99.51%, 98.89%, and 98.77% correspondingly. In addition, with label-5, the HMFS-SDLCAD model has offered $accu_y$, $prec_n$, $reca_l$,

F_{score}, and MCC of 99.83%, 99.51%, 98.92%, 99.22%, and 99.12% correspondingly. Furthermore, with label-10, the HMFS-SDLCAD system has accessible $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC of 99.92%, 97.07%, 99.29%, 98.16%, and 98.13% respectively. Besides, with label-12, the HMFS-SDLCAD model has obtainable $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC of 99.95%, 97.51%, 97.16%, 97.34%, and 97.31% respectively.

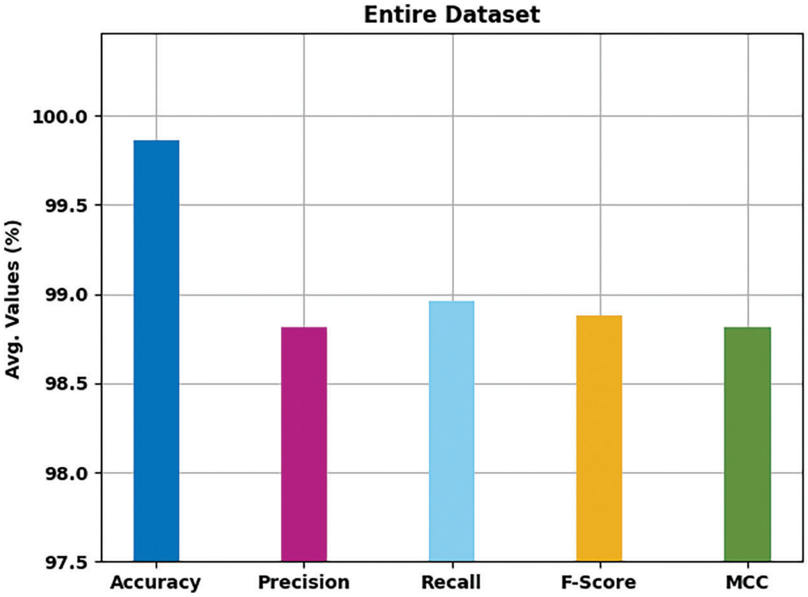
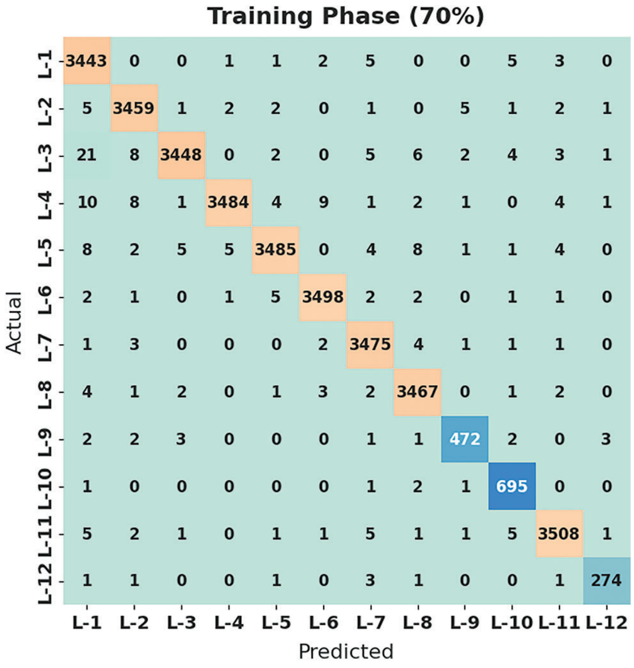


Figure 4: Result analysis of HMFS-SDLCAD technique on entire dataset



	L-1	L-2	L-3	L-4	L-5	L-6	L-7	L-8	L-9	L-10	L-11	L-12
L-1	3443	0	0	1	1	2	5	0	0	5	3	0
L-2	5	3459	1	2	2	0	1	0	5	1	2	1
L-3	21	8	3448	0	2	0	5	6	2	4	3	1
L-4	10	8	1	3484	4	9	1	2	1	0	4	1
L-5	8	2	5	5	3485	0	4	8	1	1	4	0
L-6	2	1	0	1	5	3498	2	2	0	1	1	0
L-7	1	3	0	0	0	2	3475	4	1	1	1	0
L-8	4	1	2	0	1	3	2	3467	0	1	2	0
L-9	2	2	3	0	0	0	1	1	472	2	0	3
L-10	1	0	0	0	0	0	1	2	1	695	0	0
L-11	5	2	1	0	1	1	5	1	1	5	3508	1
L-12	1	1	0	0	1	0	3	1	0	0	1	274

Figure 5: Confusion matrix of HMFS-SDLCAD technique under 70% of TRS dataset

Table 3: Result analysis of HMFS-SDLCAD technique with various measures on 70% of TRS dataset

Training phase (70%)					
Class labels	Accuracy	Precision	Recall	F-Score	MCC
Label 1	99.77	98.29	99.51	98.89	98.77
Label 2	99.85	99.20	99.43	99.31	99.23
Label 3	99.80	99.62	98.51	99.07	98.96
Label 4	99.85	99.74	98.84	99.29	99.20
Label 5	99.83	99.51	98.92	99.22	99.12
Label 6	99.90	99.52	99.57	99.54	99.49
Label 7	99.87	99.14	99.63	99.39	99.31
Label 8	99.87	99.23	99.54	99.38	99.31
Label 9	99.92	97.52	97.12	97.32	97.28
Label 10	99.92	97.07	99.29	98.16	98.13
Label 11	99.87	99.40	99.35	99.38	99.30
Label 12	99.95	97.51	97.16	97.34	97.31
Average	99.87	98.81	98.91	98.86	98.79

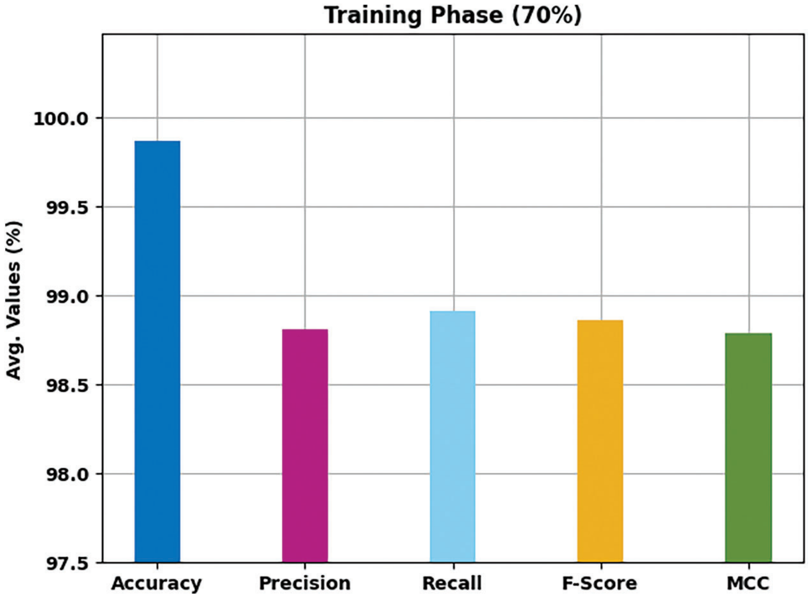


Figure 6: Result analysis of HMFS-SDLCAD technique on 70% of TRS dataset

Fig. 7 demonstrates the confusion matrix created by the HMFS-SDLCAD algorithm on the classification of cyberattacks under 30% of testing set (TSS) dataset. The figure outperformed that the HMFS-SDLCAD approach has identified all the 12 classes effectively.

Tab. 4 and Fig. 8 examine the overall classification outcome of the HMFS-SDLCAD technique on 30% of TSS dataset. The experimental values revealed that the HMFS-SDLCAD algorithm has gained maximal classifier results under all class labels. For instance, with label-1, the HMFS-SDLCAD model has offered

$accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC of 99.74%, 98.02%, 99.61%, 98.81%, and 98.67% respectively. Likewise, with label-5, the HMFS-SDLCAD model has offered $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC of 99.87%, 99.46%, 99.32%, 99.39%, and 99.32% correspondingly. Similarly, with label-10, the HMFS-SDLCAD model has obtainable $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC of 99.88%, 95.79%, 98.67%, 97.21%, and 97.16% respectively. Eventually, with label-12, the HMFS-SDLCAD technique has offered $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC of 99.98%, 98.32%, 99.15%, 98.73%, and 98.72% respectively.

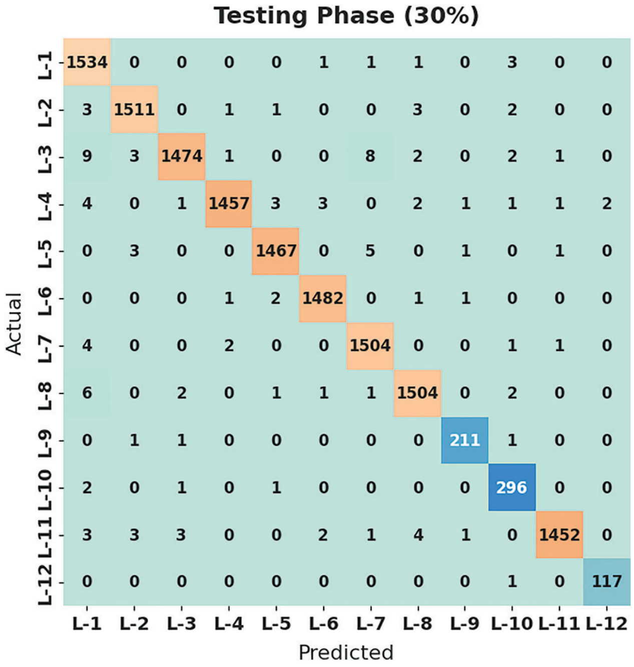


Figure 7: Confusion matrix of HMFS-SDLCAD technique under 30% of TSS dataset

Table 4: Result analysis of HMFS-SDLCAD technique with various measures on 30% of TSS dataset

Testing phase (30%)					
Class labels	Accuracy	Precision	Recall	F-Score	MCC
Label 1	99.74	98.02	99.61	98.81	98.67
Label 2	99.86	99.34	99.34	99.34	99.26
Label 3	99.76	99.46	98.27	98.86	98.73
Label 4	99.84	99.66	98.78	99.22	99.13
Label 5	99.87	99.46	99.32	99.39	99.32
Label 6	99.92	99.53	99.66	99.60	99.55
Label 7	99.83	98.95	99.47	99.21	99.11
Label 8	99.82	99.14	99.14	99.14	99.04
Label 9	99.95	98.14	98.60	98.37	98.34
Label 10	99.88	95.79	98.67	97.21	97.16
Label 11	99.85	99.73	98.84	99.28	99.20
Label 12	99.98	98.32	99.15	98.73	98.72
Average	99.86	98.79	99.07	98.93	98.85

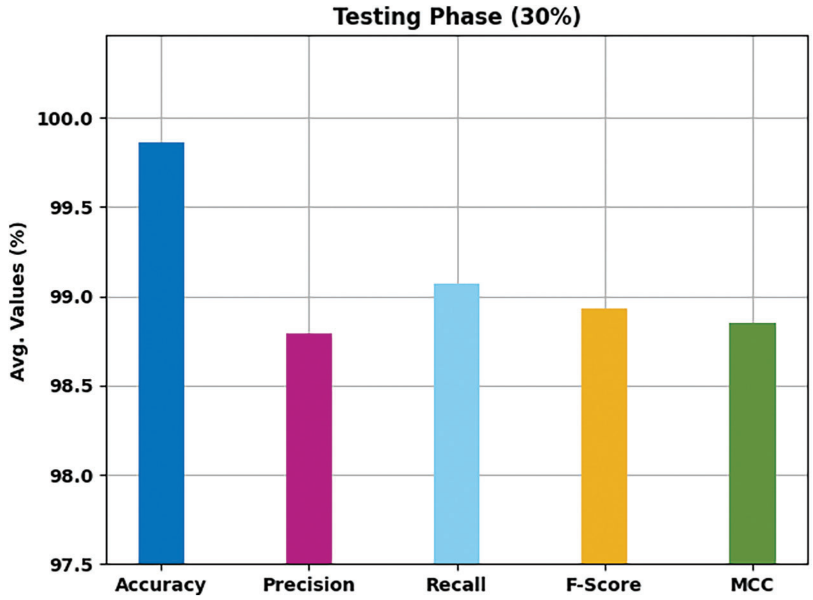


Figure 8: Result analysis of HMFS-SDLCAD technique on 30% of TSS dataset

The training accuracy (TA) and validation accuracy (VA) attained by the HMFS-SDLCAD method on test dataset is demonstrated in Fig. 9. The experimental outcome implied that the HMFS-SDLCAD model has gained maximum values of TA and VA. In specific, the VA seemed that superior to TA.

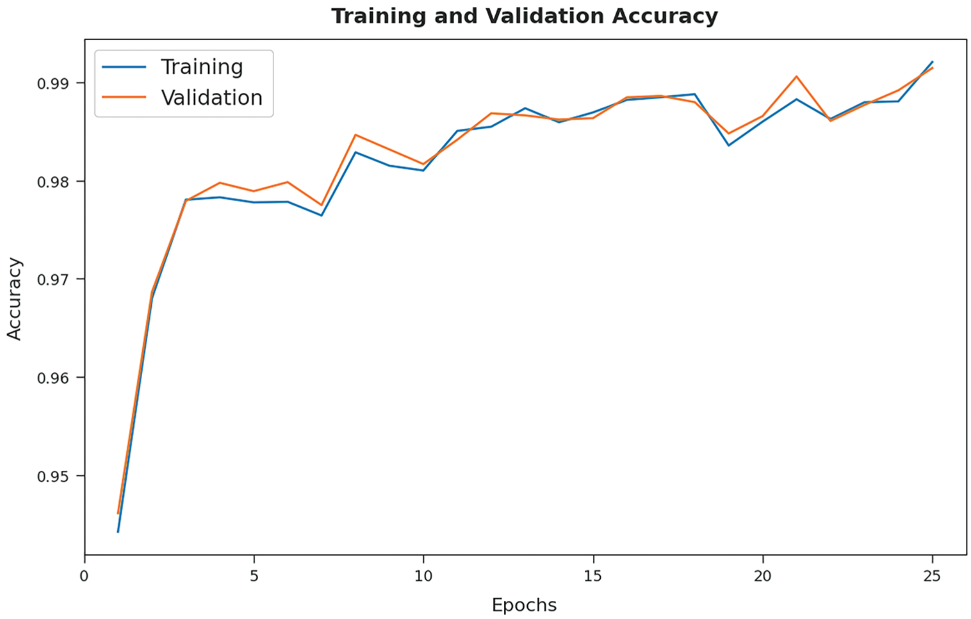


Figure 9: TA and VA analysis of HMFS-SDLCAD technique

The training loss (TL) and validation loss (VL) achieved by the HMFS-SDLCAD system on test dataset are established in Fig. 10. The experimental outcomes inferred that the HMFS-SDLCAD approach has able least values of TL and VL. In specific, the VL appeared to be lower than TL.

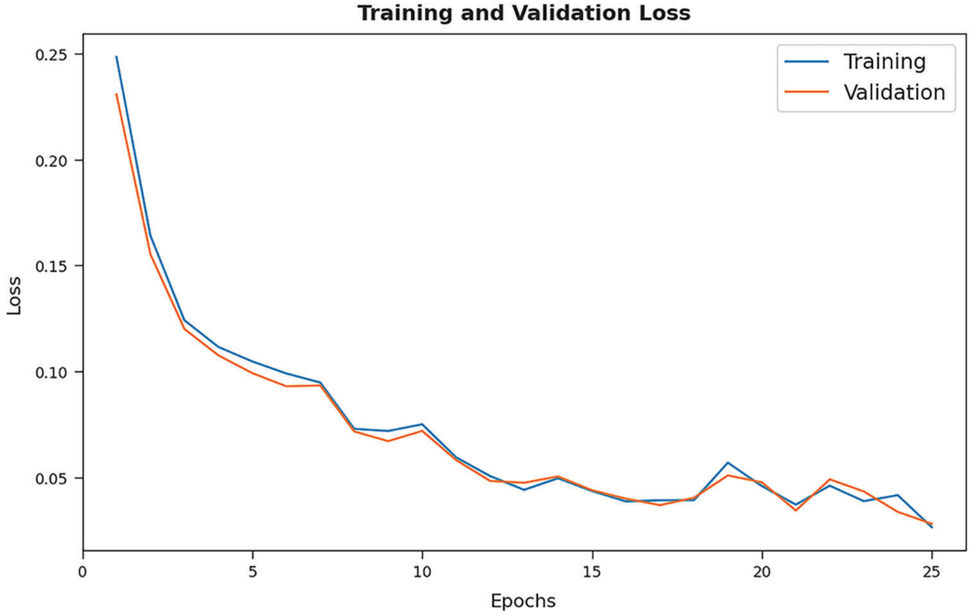


Figure 10: TL and VL analysis of HMFS-SDLCAD technique

A detailed precision-recall examination of the HMFS-SDLCAD approach on test dataset is exhibited in Fig. 11. By observing the figure, it can be noticed that the HMFS-SDLCAD model has accomplished maximal precision-recall performance under all labels.

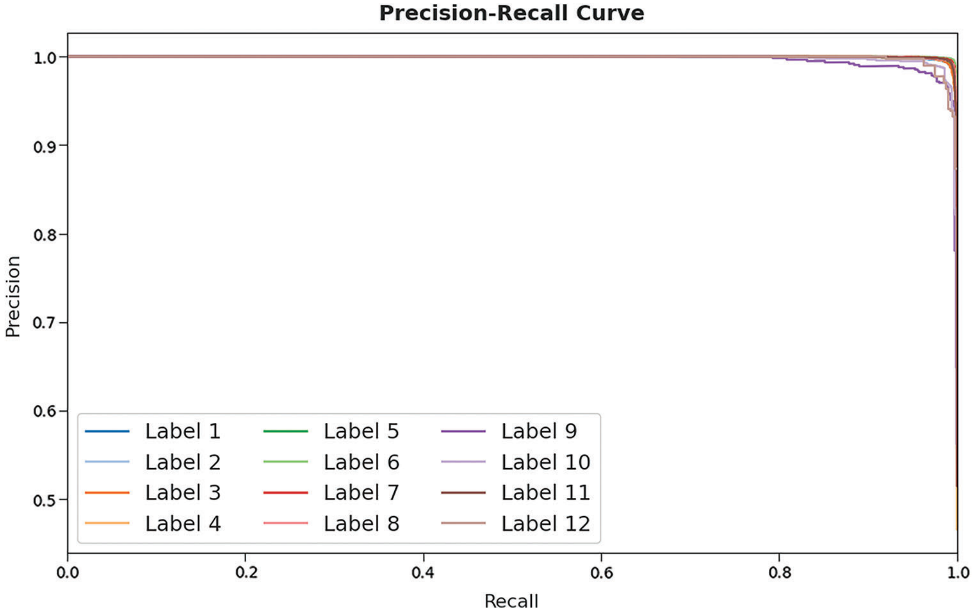


Figure 11: Precision-recall curve analysis of HMFS-SDLCAD technique

Fig. 12 depicts the comparative acc_y, $prec_n$, and $reca_l$ analysis of the HMFS-SDLCAD approach with existing methods [22–24]. The figure represents that the random forest (RF), support vector machine (SVM), multilayer perceptron (MLP), and deep neural network (DNN) techniques have demonstrated worse performance with lower values of acc_y, $prec_n$, and $reca_l$. Next, the long short term memory (LSTM) algorithm has tried to exhibit moderate performance with acc_y, $prec_n$, and $reca_l$ of 99.42%, 97.85%, and 98.90% respectively. In addition, the RF-NML model has resulted in reasonable outcomes with acc_y, $prec_n$, and $reca_l$ of 98.50%, 97.26%, and 98.12% correspondingly. But, the HMFS-SDLCAD model has outperformed other methods with maximum acc_y, $prec_n$, and $reca_l$ of 98.86%, 98.79%, and 99.07% correspondingly.

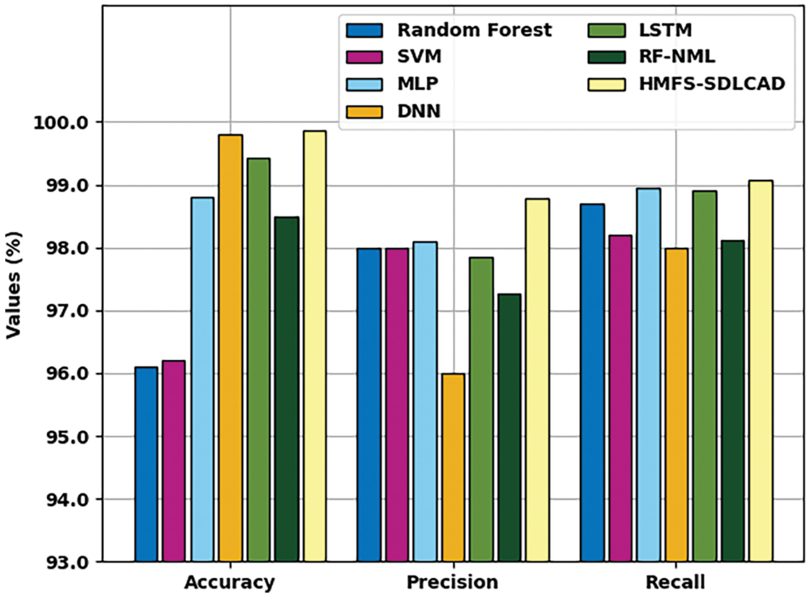


Figure 12: Acc_y, $prec_n$, and $reca_l$ analysis of HMFS-SDLCAD technique with existing algorithms

Fig. 13 demonstrates a comparative F_{score} and MCC examination of the HMFS-SDLCAD model with existing models. The figure indicated that the RF, SVM, MLP, and DNN models have shown poor performance with lower values of F_{score} and MCC. Next, the LSTM model has tried to exhibit moderate performance with F_{score} and MCC of 98.34% and 98.26% respectively. Then, the RF-NML model has resulted in reasonable outcomes with F_{score} and MCC of 98.59% and 98.11% respectively. However, the HMFS-SDLCAD model has outperformed other methods with maximum F_{score} and MCC of 98.93% and 98.85% respectively. The above mentioned results and discussion reported that the HMFS-SDLCAD model has accomplished effectual outcomes over other methods.

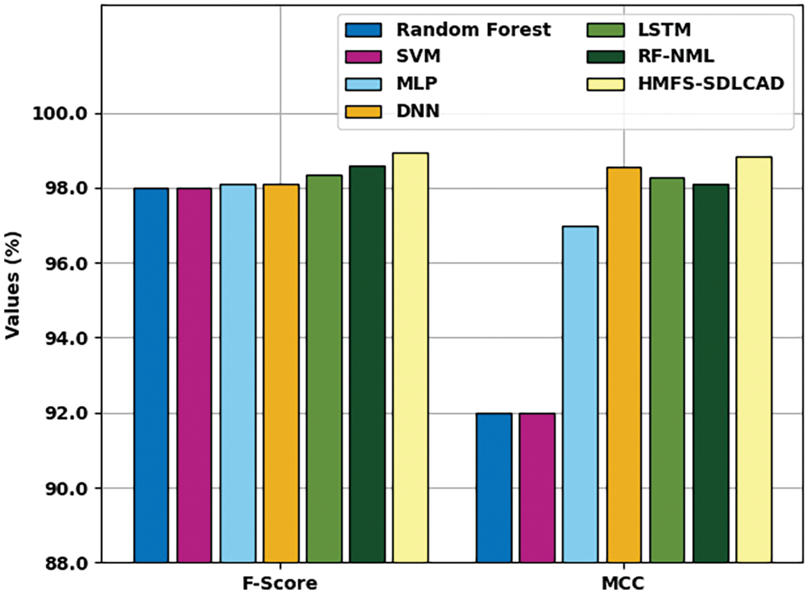


Figure 13: F_{score} and MCC analysis of HMFS-SDLCAD technique with existing algorithms

4 Conclusion

In this study, a new HMFS-SDLCAD model has been developed to recognize the occurrence of cyberattacks in the IoT environment. At the preliminary stage, data pre-processing is carried out to transform the input data into useful format. Then, the SSOPSO algorithm is utilized to elect features. In addition, the WOA with SBiGRU model is utilized for the identification and classification of cyberattacks. The experimental analysis of the HMFS-SDLCAD model is validated using benchmark dataset and the results are assessed under several aspects. The simulation outcomes pointed out the improvements of the HMFS-SDLCAD model over recent approaches. Thus, the HMFS-SDLCAD model can be employed for effectual identification of cyberattacks in the IoT environment. In future, feature reduction and outlier removal approaches can be included to enhance the classifier outcomes.

Funding Statement: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through Large Groups Project under grant number (45/43). Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R140), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: (22UQU4310373DSR16).

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

References

[1] I. Ullah and Q. H. Mahmoud, “Design and development of a deep learning-based model for anomaly detection in IoT networks,” *IEEE Access*, vol. 9, pp. 103906–103926, 2021.

[2] M. Wazid, A. K. Das, V. Bhat K and A. V. Vasilakos, “LAM-CIoT: Lightweight authentication mechanism in cloud-based IoT environment,” *Journal of Network and Computer Applications*, vol. 150, pp. 102496, 2020.

[3] X. Gao, C. Shan, C. Hu, Z. Niu and Z. Liu, “An adaptive ensemble machine learning model for intrusion detection,” *IEEE Access*, vol. 7, pp. 82512–82521, 2019.

[4] S. Sapre, P. Ahmadi and K. Islam, “A robust comparison of the kddcup99 and nsl-kdd iot network intrusion detection datasets through various machine learning algorithms,” arXiv preprint, arXiv:1912.13204, 2019.

[5] S. U. Jan, S. Ahmed, V. Shakhov and I. Koo, “Toward a lightweight intrusion detection system for the internet of things,” *IEEE Access*, vol. 7, pp. 42450–42471, 2019.

[6] O. Brun, Y. Yin and E. Gelenbe, “Deep learning with dense random neural network for detecting attacks against iot-connected home environments,” *Procedia Computer Science*, vol. 134, pp. 458–463, 2018.

[7] R. Damasevicius, A. Venckauskas, S. Grigaliunas, J. Toldinas, N. Morkevicius *et al.,* “LITNET- 2020: “An annotated real-world network flow dataset for network intrusion detection,” *Electronics*, vol. 9, no. 5, pp. 800, 2020.

[8] S. Garg, K. Kaur, N. Kumar, G. Kaddoum, A. Y. Zomaya *et al.,* “A hybrid deep learning-based model for anomaly detection in cloud datacenter networks,” *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 924–935, 2019.

[9] M. Pawlicki, M. Choraś and R. Kozik, “Defending network intrusion detection systems against adversarial evasion attacks,” *Future Generation Computer Systems*, vol. 110, pp. 148–154, 2020.

[10] D. Tiwari, B. S. Bhati, B. Nagpal, S. Sankhwar and F. Al-Turjman, “An enhanced intelligent model: To protect marine IoT sensor environment using ensemble machine learning approach,” *Ocean Engineering*, vol. 242, pp. 110180, 2021.

[11] M. Panda, A. A. A. Mousa and A. E. HaSSOnien, “Developing an efficient feature engineering and machine learning model for detecting iot-botnet cyber attacks,” *IEEE Access*, vol. 9, pp. 91038–91052, 2021.

[12] Q. A. A. Haija, “Top-down machine learning-based architecture for cyberattacks identification and classification in iot communication networks,” *Frontiers in Big Data*, vol. 4, pp. 782902, 2022.

[13] Z. E. Huma, S. Latif, J. Ahmad, Z. Idrees, A. Ibrar *et al.,* “A hybrid deep random neural network for cyberattack detection in the industrial internet of things,” *IEEE Access*, vol. 9, pp. 55595–55605, 2021.

[14] N. Amma, “A vector convolutional deep autonomous learning classifier for detection of cyber attacks,” *Cluster Computing*, 2022, https://doi.org/10.1007/s10586-022-03577-4.

[15] P. An, Z. Wang and C. Zhang, “Ensemble unsupervised autoencoders and Gaussian mixture model for cyberattack detection,” *Information Processing & Management*, vol. 59, no. 2, pp. 102844, 2022.

[16] M. A. Hamza, S. B. Haj Hassine, I. Abunadi, F. N. Al-Wesabi, H. Alsolai *et al.,* “Feature selection with optimal stacked sparse autoencoder for data mining,” *Computers*,” *Materials & Continua*, vol. 72, no. 2, pp. 2581–2596, 2022.

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

[18] A. M. Hilal, J. S. Alzahrani, I. Abunadi, N. Nemri, F. N. Al-Wesabi *et al.,* “Intelligent deep learning model for privacy preserving IIoT on 6 g environment,” *Computers, Materials & Continua*, vol. 72, no. 1, pp. 333–348, 2022.

[19] R. A. Ibrahim, A. A. Ewees, D. Oliva, M. A. Elaziz and S. Lu, “Improved salp swarm algorithm based on particle swarm optimization for feature selection,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 8, pp. 3155–3169, 2019.

[20] A. A. Wazrah and S. Alhumoud, “Sentiment analysis using stacked gated recurrent unit for arabic tweets,” *IEEE Access*, vol. 9, pp. 137176–137187, 2021.

[21] S. Mirjalili and A. Lewis, “The whale optimization algorithm,” *Advances in Engineering Software*, vol. 95, pp. 51–67, 2015.

[22] V. Dutta, M. Choraś, M. Pawlicki and R. Kozik, “A deep learning ensemble for network anomaly and cyber-attack detection,” *Sensors*, vol. 20, no. 16, pp. 4583, 2020.

[23] T. Gopalakrishnan, D. Ruby, F. A. Turjman, D. Gupta, I. V. Pustokhina *et al.,* “Deep learning enabled data offloading with cyber attack detection model in mobile edge computing systems,” *IEEE Access*, vol. 8, pp. 185938–185949, 2020.

[24] M. H. Saracevic, S. Z. Adamović, V. A. Miškovic, M. Elhoseny, N. D. Maček *et al.,* “Data encryption for internet of things applications based on Catalan objects and two combinatorial structures,” *IEEE Transactions on Reliability*, vol. 70, no. 2, pp. 819–830, 2021.