Optimized Self-Attention Pyramidal Convolutional Neural Network for Intrusion Detection Framework in IoT

Padma Yenuga [1], Satya Narayana Reddy Beeram [2], Sitanaboyina S. L. Parvathi [3], G. Mahesh Reddy [2], Venugopal Boppana [4], Sunitha Davuluri [4], Repudi Ramesh [2], Meda Srikanth [5], B. Vara Prasad Rao [5], Ravi Kumar Munaganuri [6], and Narasimha Rao [6,*]

[1] Department of Information Technology, Prasad V. Potluri Siddhartha Institute of Technology, Vijayawada, AP, India
[2] Department of Computer Science and Engineering, KKR & KSR Institute of Technology and Sciences, Guntur, AP, India
[3] Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, AP, India
[4] Department of Computer Science and Engineering, NRI Institute of Technology, Agiripalli, Krishna, AP, India
[5] Department of Computer Science and Engineering, R.V.R. & J.C. College of Engineering and Technology, Vijayawada, AP, India
[6] School of Computer Science and Engineering, VIT-AP University, Amaravati-52237, India
Email: Padma.yenuga@pvpsiddhartha.ac.in (P.Y.); snreddy.beeram@gmail.com (S.N.R.B.); srilakshmiparvathi@kluniversity.in (S.S.L.P.); Mahesh.gogula@gmail.com (G.M.R.); srees.boppana@gmail.com (V.B.); sunithadavuluri@gmail.com (S.D.); repudiramesh@gmail.com (R.R.); medasrikanth@gmail.com (M.S.); bvpr@rvrjc.ac.in (B.V.P.R.); ravi2kinus@gmail.com (R.K.M.); y.narasimharao@vitap.ac.in (Y.N.R.)
*Corresponding author

***Abstract*—This research finds an important place in intrusion detection within the landscape of IoT when it puts forward the optimized Intrusion Detection System (IDS) solution. This is enabled by using the Self-Attention Pyramidal Convolutional Neural Network (SAPCNN) that is powered by Hybrid Ebola and Bald Eagle Search Optimization Algorithm, thereby enhancing classification accuracy. The methodology of this technique is basically a preprocessing tool called Dynamic Context-Sensitive Filtering (DCSF) aimed at removing data redundancy as well as filling missing values, followed by the mechanism of Pelican Optimization Algorithm (POA)-based feature selection. Performance evaluations on the Canadian Institute for Cybersecurity Intrusion Detection System 2017 Dataset (CICIDS2017) dataset reveal that the proposed IDS can accurately detect Distributed Denial of Service (DDoS) attacks with a precision of 98.9% and reduce the computational time by 31.7% as compared to baseline models. These results therefore indicate that the model can successfully handle complex Internet of Things (IoT) intrusion scenarios with great precision and efficiency.**

***Keywords*—dynamic context-sensitive filtering, pelican optimization algorithm, self-attention pyramidal convolutional neural network, Ebola optimization search algorithm, bald eagle search optimization algorithm, intrusion detection, internet of things.**

I. Introduction

The world is full of plethora and electronic products that may change the lifestyle of the human kind [1]. The IoT is the emerging source of an innovative technique, it also transmit the industrial life style and makes even smarter with smartest product. This smart devices enhance the connection in all over the world in all aspects such as health department, society, smart cities, food management, smart education, and smart devices [2]. In Internet of Things, numerous physical products are developed to cooperate and communicate with one device to the other for transmitting the packets over huge amount of networks without human interaction [3]. In 2025, around 41.6 billion IoT devices are projected to be connected [4, 5]. Data integrity and confidentiality is one of the challenges in large-scale IoT networks. In the US computer emergency response team, the number of cyber-attacks increased gradually from 5,000 in 2005 to 38,000 in 2011. Statistics predict that cyber-attacks will cost up to €5 trillion by 2025 [6, 7]. According to Symantec reports, IoT devices are attacked on average every 2 min. Another report predicts that the count of cyber-attacks is increased up to 2,000% for 6 years [8]. The average cost of attacks reached a maximum of $482 million in six months in 2017. From 2013 to 2017, hackers stole or attacked nearly 9 billion records [9]. Intrusion detection criminals around word are trained is able to identify IoT targets, steal

Manuscript received September 12, 2024; revised December 7, 2024; accepted February 19, 2025; published May 9, 2025.

information, and make illegal profits. To protect the IoT devices against the cyber-criminals by monitoring them is a risky task [10]. Cyber-attacks are endeavors to go after the security fabric of Internet of Things [11]. Cyber-attacks go beyond security layers and pose chief threat to network stability as well as confidentiality [12]. Identity was introduced in 1980 to secure IoT systems. Monitoring, analyzing network traffic and responding to malicious attacks are the functions of IDS. [13]. The primary purpose of IDS is to detect various types of malicious network traffic and its activity. These constitute a common order. IDS are very powerful for find out, detect, and monitor the threats [14].

The increased economic impact and the influence over its pervasive aspects will impress the criminals towards IoT and cyber security. IoT and cyber security becomes a popular subject for recent years. Massive IoT architectures and the emergence of unique threats render old strategies highly ineffective. Deep Learning (DL) gains innovative solutions for Internet of Things (IoT) IDS with data-driven, intruder-based technology and the ability to detect unknown attacks.

In this manuscript, IDS-SAPCNN-HEBEOA-IOT is proposed. Data are initially taken from CICIDS2017 dataset [15]. The data is then passed to the preprocessing segment. The preprocessing phase performs redundancy removal and missing value replacement using Dynamic Context-Sensitive Filtering (DCSF) [16]. The preprocessed output is then passed to feature determination procedure. The Pelican Optimization Approach (POA) [17] selects best features from a preprocessed dataset. The selected features are then passed to a Self-attention Pyramidal Convolutional Neural Network (SAPCNN) [18] for categorization that classifies data like normal anomalies. In general, Self-Aware Pyramidal Convolutional Neural Network does not use optimization method to calculate the best parameters for accurate IoT intrusion detection. Therefore, Hybrid Ebola Optimization Search Approach [19] and Bald Eagle Search Optimization Algorithm (HEBEOA) [20] is utilized to optimize the Self-attention pyramidal convolutional Neural Network.

The main contribution of this work is listed below,

- Unique DL-IDS is designed to secure IoT with high efficiency of accuracy in terms of detection rate.
- The proposed approach will handle dataset with uncertain data, and redundant scores.
- The POA algorithm takes longer than other optimization approaches in regard to convergence time introduced in feature learning. Optimal package characteristics are detected by POA approach and extracted from cleaned dataset.
- According to study of optimal features, SAPCNN divides data packets to benign or anomalous attack. Clearing the dataset and studying the optimal feature will help SAPCNN to obtain the greater accuracy and lower computational time while compared with the existing models.

Remaining Section of this manuscript is organized as: Section II analyses literature survey, proposed approach is described in Section III, outcomes and discussion is demonstrated in Section IV, finally, conclusion is presented in Section V.

II. Literature Review

Several researches were suggested in literature related to IoT intrusion detection; some recent works are reviewed here.

Elnakib *et al.* [21] have presented the Enhanced DL model for Internet of Things IDS. An enhanced IDS with DL and Multiple-class categorization Method is introduced Enhanced Intrusion Detection Method (EIDM) to categories 15 traffic nature with 14 attack categories with accurate of 94%. The input data is taken from the CICIDS2017 dataset. The 4 existing DL models are organized to divide the 6 classes of network traffic behavior. The performance analysis compared with the existing research work provides a better output in terms of accuracy and so on. The presented model provides higher recall conversely lower F-measure.

Kumar *et al.* [22] have presented the IDS for IoT Using RKCNN. The presented model utilizes the DL based Recurrent Kernel Convolution Neural Network-Modified Monarchy Butterfly Optimization (RK.CNN-MMBO) approach to detect the attacks in network. The input data is taken from 2 different types of dataset they are CICIDS2017 and Network-Based detection of IOT Botnet Attacks using Deep Autoencoders (N-BaIoT). The input data is pre-processed by utilizing the minmax normalization. Pre-processed data is then fed to feature selection. IBRO (Improved Battle Royale Optimization) approach is used for selecting the optimal features. DL classifier, Recurrent Kernel Convolutional Neural Network (RKCNN), and a few features were classified. Additionally, Modified Monarch Butterfly Optimization (MMBO) is used to maximizes classifier's function during the detection of attack. The performance analysis of the presented model achieves greater result while compared with its existing methods. The presented model provides higher precision conversely lower Accuracy.

Roy *et al.* [23] have presented the Lightweight Supervised IDS for Internet of Things Networks. A unique IDS that uses Machine Learning (ML) to effectively detect intrusion detection and cyber-attack in resource-constraint of Internet of Things networks. The presented model detects the very rare features to identify the IDS using few training data and minimum training time. The input data is taken from the two kinds of dataset. They are CICIDS2017 and Network Security Layer-Knowledge Discovery in Database (NSL-KDD). The performance metrics of the presented method achieves better output while compared with the existing methods. The presented approach is lightweight and has limited power and storage capabilities. The presented model provides higher computational time conversely lower recall.

AnkitThakkar *et al.* [24] have presented the Imbalanced Intrusion Data Categorization for Internet of Things Network Utilizing Ensemble Learning-based DNN. The suggested approach was introduced to address the problem of class imbalance using ensemble learning techniques

such as: B. Bagging classifier using DNN as base estimator. In suggested algorithm, the Deep Neural Network (DNN) training process is impacted through including class weights that favor the construction of organized deep neural network training data. The presented model is believed to have two components to overcome class imbalance issue in intrusion detection datasets. Input data comes from the NSL-KDD, University of New South Wales Network Intrusion Detection Dataset (UNSW_NB-15), Canadian Institute for Cybersecurity Intrusion Detection System (CICIDS2017), and Botnet Attack Traffic in Internet of Things (BoT-IoT) datasets. Performance metrics are estimated utilizing various performance metrics, like recall, accuracy, F-score, precision, and FPR. Additionally, result of presented model provides better result while compared with the existing methods. The presented model provides higher precision conversely lower computational time.

Basati *et al.* [25] have presented the Parallel Deep Auto-Encoder (PDAE): Efficient network IDS in Internet of Things using PDAE. The presented method introduce a novel lightweight architecture based on PDAE. The input data is taken from the KDDCup99, CICIDS2017, and UNSW-NB15 datasets. This kind of separation of features allows us to maximize the accuracy of presented method where as highly minimizing parameters count, memory foot print, and need for processing effect. The efficacy of presented method is analyzed utilizing various performance metrics which provides greater result while compared with the existing methods. The presented model provides higher accuracy conversely higher computational time.

Wu *et al.* [26] have presented the A Robust Transformer-Based Technique for IDS. A Robust Transformer-based IDS (RTIDS) is used to reconstruct feature representations in order to strike balance among feature preservation and low dimensionality on unbalanced datasets. In order to learn low-dimensional feature representations from high-dimensional raw data, the presented model employs a variant encoder/decoder stacked neural network and a location embedding technique to link sequential information between features. In addition, it allows self-awareness mechanisms to facilitate classification of network traffic types. The input data is taken from the CICIDS2017 and Canadian institute for Cybersecurity-Distributed Denial of Service Attack (CIC-DDoS201) dataset. The obtained result in F-measure is of 88.467% and 86.56%. Few ML approach is comparing to the existing models, like Support Vector Machine (SVM) and DL algorithms that include Recurrent Neural Network (RNN), Feedforward Neural Network (FNN), and Long Short-Term Memory (LSTM) planned to mention validity of suggested technique. The presented model provides higher F-measure conversely lower Receiver Operating Characteristic (ROC).

Basati *et al.* [27] have presented the Deep Feature Extraction: Efficient Internet of Things Network IDS using DFE. The presented model provides highly accurate Deep Feature Extraction (DFE), and it utilizes minimum lightweight and effectual Neural Network according to DFE. In the presented approach, the network's input vector is permuted in 3D space and its values are collected. This permits the method to extract extremely discriminative along minorcount of layers without using huge 2D or 3D convolution filters. Finally, network results can attain exact categorization with very few required computations. This makes Deep Feature Extraction ideal for real-time IDS for Internet of Things devices with limited processing power. Efficiency of Deep Feature Extraction is verified using 3 popular public datasets. The input data is taken from the UNSW-NB15, CICIDS2017, and KDDCup99. The experimental output of the presented model provides a better result while compared with its existing methods. The presented model provides higher ROC conversely lower recall.

Ramana *et al.* [28] have presented an Ambient Technique: IoT based decision performance analysis for IDS. The ambient method is based on the RL-DQN (Reinforcement Learning Integrated Deep Q Neural Network) method of wireless sensor networks and Internet of Things. Use Markov decision process formalism to improve decision performance of intrusion detection systems. RL-DQN Network based intrusion detection integrated on edge cloud IDS infrastructure. Twofold assault categorization of network traffic is acted in edge network and multiple assault order is acted in the cloud. A two-step process is followed to detect intrusion attacks: B. Initial learning and classification phases. Input data comes from the UNSW-NB-15, BoTNeTIoT-L01, CICIDS2017, and IoTID20 data sets. Performance metrics such as accuracy, precision, and recall are run to evaluate presented model. The suggested model obtains the better result while compared with its existing methods. The presented model provides lower accuracy conversely higher lower precision.

III. Proposed Methodology

IDS-SAPCNN-HEBEOA-IOT explained here. Block diagram of IDS-SAPCNN-HEBEOA-IOT is represented in Fig. 1. It contains 4 stages such as; Data Acquisition, Pre-processing, Feature Selection and Classification. The diagram is found to represent two purposes: the structural organization of the proposed IoT IDS and the processes used in data preparation, training, and testing. It presents a comprehensive pipeline starting from data acquisition using the CICIDS2017 dataset, followed by pre-processing with Dynamic Context Sensitive Filtering, feature selection using the Pelican Optimization Algorithm, classification via the Self-Attention Pyramidal Convolutional Neural Network (SAPCNN), and optimization through the hybrid Ebola and Bald Eagle Search Optimization Algorithm. Specific classes for attacking, such as DDoS, DoS Hulk, and others, further highlight the use in communicating classification capability for the overall system. Nevertheless, this diagram's focus on the overall process flow rather than focusing on an aspect may cause an information overload for readers not conversant with the details of this process. To make it clearer, the diagram should indeed be split into multiple focused diagrams that emphasize the different

aspects of the IoT IDS model. For example, one diagram could focus on exclusively describing the structural flow of the system, including the data pipeline, pre-processing, feature selection, and classification stages. Another diagram may focus on optimization methods and how they improve the performance of SAPCNN. Furthermore, a diagram can be provided to explain the testing phase and how the trained system classifies particular attack types. This modular representation would make it easier for readers to understand the individual components and processes of the IoT IDS, thus making comprehension clearer and avoiding ambiguity. Thus, the detailed description about each stage is given below.

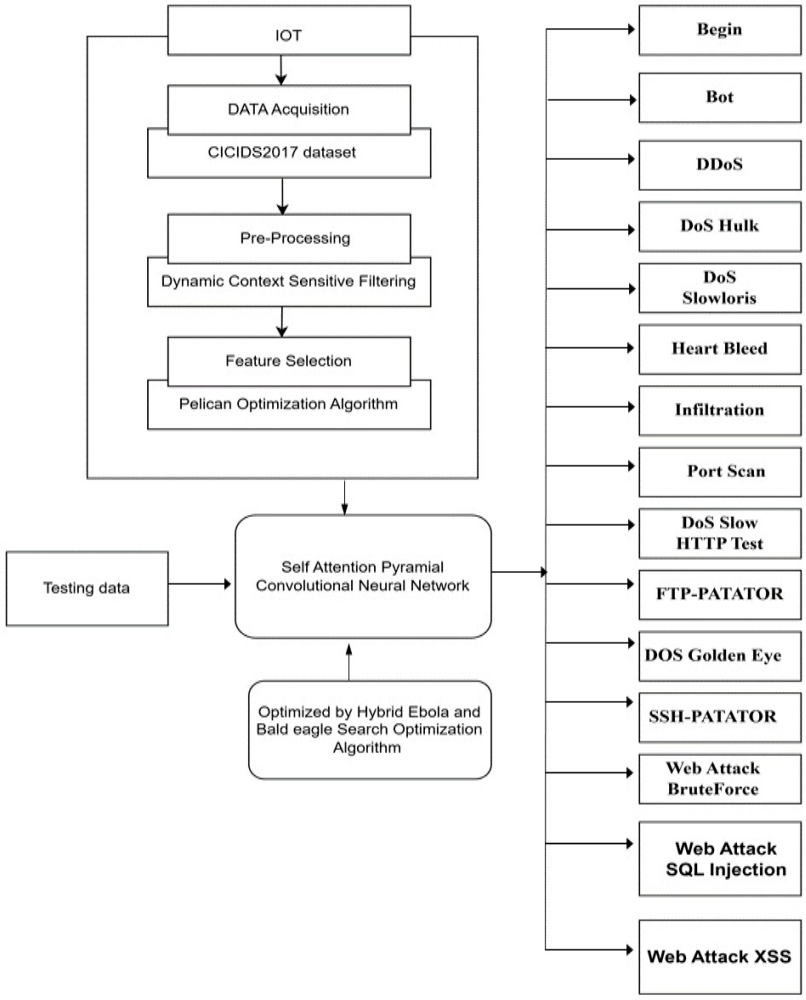

Fig. 1. Block diagram of the proposed (IDS-SAPCNN-HEBEOA-IOT) methodology.

A. Data Acquisition

The dataset utilized in this research was made in 2017 through Canadian Cyber Security Institute (CIC). The CIC Intrusion Detection Scheme (CICIDS2017) dataset consists of common attacks that resemble real-world data. This dataset contains 2 files called: For Generated Labeled Flows and Deep Learning Common Vulnerabilities and Exposures (CVE), the 1st file contains 86 features and second file contains 79 features. The research implements a Deep Learning CSV data file consisting of his 8 traffic monitoring sessions over 5 days. These eight files are merged into one of his CSV files for further investigation. There are 2,830,743 rows, 78 feature columns, and one label column in the merged file. Fwd Header Length is the name of two features in this file, and the experiment only has access to 77 feature columns and one label column. There are 15 class designations in these 78 traits. One BENIGN (normal) label and 14 attack type labels are shown in Table I. 70% data is employed for training and 30% is utilized for testing.

The input data are then send to the preprocessing segments. Dynamic Context-Sensitive Filtering (DCFM) is utilized to filter the redundant and duplicate data. It finds area-related affinity weights by introducing matrix multiplication to kernels' creation process. The 3 context-sensitive filter generating networks are arranged in pyramid structure and 3 dynamic Convolution kernels are generated, to perform dilation convolution with various rate of dilation. At the end, global attention method is implemented to gather 3 results. Every filter-creating network creates dynamic kernel to distinguish between different kernels. Three feature maps are generated in the network, which is expressed in Eq. (1),

$$\begin{cases} X_i^t = C_{1\times1}(f^{t-1}) \\ Y_i^t = C_{1\times1}(f^t) \\ Z_i^t = C_{1\times1}(f^{t-1}) \end{cases} \quad (1)$$

Here, $C_{1\times1}$ represents the 1×1 convolution reduces feature map's dimension. All convolution operations does not share parameters. The reshaped feature maps are expressed in Eq. (2),

$$\begin{cases} P_i^t = Soft\,max\left(\left(\tilde{X}_i^t \times \tilde{Y}_i^t\right)^T\right) \\ \tilde{P}_i^t = P_i^t \times \tilde{Z}_i^t + \tilde{X}_i^t \end{cases} \quad (2)$$

Lost value change, or imputation, is procedure of predicting same score of lost characteristics in the data. The dynamic kernel is expressed in Eq. (3),

$$K_i^t = A_{verage}P_{ooling}\left(C_{1\times1}\left(\tilde{P}_i^t\right)\right) \quad (3)$$

Here, K_i^t represents the dynamic kernel, $Softmax$ represents the softmax operation,$A_{verage}P_{ooling}$represents with kernel size 3×3. ×Represents the matrix multiplication, and Trepresents matrix transpose. K_i^tconsists location-related contextual information by adopting matrix multiplication, at the end $K_i^t$is reshaped. The scale-specific feature is expressed in Eq. (4),

$$\delta_i^t = DC(C(f^t); K^t, d) \quad (4)$$

Here, DC represents dilation Convolution with rate of dilation, $d$represents the dilation rate. DCrepresents the single scale to capture interface correlation. Finally, DC failed to find attribute in the multiple measures. To overcome this issue 3 Dynamic Filtering Units (DFUs) are used at parallel and assign different $d$for them. In the visual attention mechanism, the attention-guided weighted summation is expressed in Eq. (5),

$$\omega_i^t = fc_1\left(A_{verage}P_{ooling}(\delta_i^t)\right) \quad (5)$$

Here, $A_{verage}P_{ooling}$ represents the input feature map, δ_i^t represents the scale-specific feature. The scalar value is expressed in Eq. (6),

$$\widetilde{\omega}_i^t = fc_1\frac{e^{\omega_i^t}}{\sum_{j=1}^{3}e^{\omega_i^t}} \quad (6)$$

Here, ω_i^trepresents the scalar value for this process. The Dynamic Context Filtering Method's final outputs expressed in Eq. (7),

$$O^t = \sum_{j=1}^{3}\widetilde{\omega}_i^t \times \delta_i^t \quad (7)$$

Here, O^t represents DCFM's final output. Eq. (7) removes all excess data, and lost scores changed to eliminate all missing values from data. Hence all the uncertainties were removed in the preprocessing phase and then it is given to the feature selection phase.

In these operations, the indexes play a crucial role in defining the operations and interactions between different components of the dynamic context-sensitive filtering mechanism. In Eq. (6), the scalar value is represented by the index iii, which represents the position of the specific element within the feature map that is being weighted based on the attention-guided mechanism. This scalar value captures the importance of the scale-specific feature for that particular index within the convolutional kernel. Similarly, in Eq. (7), the index "*k*" is used to aggregate the outputs of the dynamic filtering units (DFUs) across different scales. Here, "*k*" is the particular dynamic kernel applied at each level of dilation and summation of the kernels yields the final output of DCFM. These indexes ensure proper alignment of attention and convolutional operations for computation of context-sensitive and scale-specific transformations and reduce redundancy while addressing missing values effectively.

B. Feature Selection Utilizing Pelican Optimization Approach

In feature selection phase, the pre-processed data are used. Here, Pelican optimization algorithm is utilized for selecting the Optimal features to minimize feature learning time.

Feature selection process using POA works based on the nature and behaviour of the pelican. Pelican is a bird which is about four feet tall and has a wingspan of about nine feet and always lives in a group. The main food of pelican is seafood which contains fish, frog and turtle. Pelicans always hunts together to attack its prey. Initially, pelican identifies the location of the prey and then relocate itself by the distance of 10–20 m high. Afterwards, the pelican spread the wings around the upper water surface. The wings of pelican attract the fish to move towards the upper surface. This process will help the pelican to catch its prey easily. While catching the prey, water gets inside the long orange beak of pelican, immediately it moves the head towards front and spray out the large amount of water. The detail description about the feature selection process is given below,

Members of the population are first initialized in accordance with the problem's lower bound and upper bound. After initialization, the input population members are randomly created. Here, ideal fitness are selected depending on the situation. It is expressed in Eq. (8),

$$F_{a,b} = l_b + rand.(u_b - l_b), a = 1,2,\ldots,N, b = 1,2,\ldots,m) \quad (8)$$

Here, $F_{a,b}$ represents b^{th} variable indicated by a^{th} candidate solution, l_b represents b^{th} lower bound, $rand$ represents the random count at [0,1] interval, u_b represents b^{th} upper bound of problem variable, N represents the number of population member,$m$represents number of problem variable. After the estimation of fitness function, the pelican identifies the location of the prey and plans to relocate itself. It is expressed in Eq. (9),

$$K = \begin{bmatrix} K_1 \\ \cdots \\ K_a \\ \cdots \\ K_N \end{bmatrix}_{N\times1} = \begin{bmatrix} K(F_1) \\ \cdots \\ K(F_a) \\ \cdots \\ K(F_N) \end{bmatrix}_{N\times1} \quad (9)$$

Here, K signifies population matrix of pelican, K_a represents a^{th} pelican. In the exploration phase, pelican moves towards the prey at water surface up to 300–400 feet (91–121 m) deep. It is expressed in Eq. (10),

$$F_{a,b}^{P_1} = \begin{cases} F_{a,b} + rand.(P_b - I.F_{a,b}), K_p < K_a; \\ F_{a,b} + rand.(F_{a,b} - P_b), else, \end{cases} \quad (10)$$

Here, $F_{a,b}^{P_1}$ represents new positioning of a^{th} pelican on b^{th} dimension under phase 1, I represents the random count equivalent to 1 or 2, P_b represents location of prey in b^{th} dimension, K_p represents the objective function value. The pelican spread its wings around the upper water surface. The wings of pelican attract the fish to move towards the upper surface. This process will make their hunting very smoothly. While catching the prey, water gets inside the beak of pelican, suddenly it moves the head towards front and spray out the large amount of water. It is expressed in Eq. (11),

$$F_{a,b}^{P_2} = F_{a,b} + \delta.\left(1 - \frac{t}{T}\right).(2.rnd - 1).F_{a,b} \quad (11)$$

Here, $F_{a,b}^{P_2}$ represents newly position of a^{th} pelican at b^{th} dimension depending on phase 2, δ represents constant, which is equivalent to 0.2, t represents iteration counter, $\delta.\left(1 - \frac{t}{T}\right)$ represents radius of nearby population members to search locally for converging better solution, T represent higher number of iterations. The optimal feature is selected by using the Eq. (7). The updating recognize or decline the position of the new pelican is labeled in Eq. (12),

$$F_a = \begin{cases} F_a^{P_2}, K_a^{P_2} < K_a; \\ F_a, else, \end{cases} \quad (12)$$

Here, $F_a^{P_2}$ represents newly status of a^{th} pelican $K_a^{P_2}$ represents objective function value depends on phase 2. The optimal feature is minimized by using Eq. (12). The algorithm entering the subsequent iteration repeats till the end of whole process. At last, the acquired features are sent to the classification phase. The selected features of CICIDS2017 dataset is listed in Table I.

TABLE I. SELECTED FEATURES OF CICIDS2017 DATASET

No.	Feature ID	Name of feature
1	41	Packet Length Std
2	13	Total Bwd Packets Length
3	65	SubflowBwd Bytes
4	8	Destination Port
5	42	Packet Length Varience
6	20	Bwd Packet Length Mean
7	54	AvgBwd Section Size
8	18	Bwd Packet Length Max
9	67	Init_Win_bytes_backward
10	12	Total Length of Fwd Packets
11	63	SubflowFwd Bytes
12	66	Init_Win_bytes_Forward
13	52	Avg Packet Size
14	40	Packet Length Mean
15	39	Max Packet Length
16	14	Bwd Packet Length Max
17	22	Flow IAT Max
18	36	Bwd Header Length
19	9	Flow Duration
20	26	Fwd IAT Max
21	55	Fwd Header Length
22	24	Fwd IAT Total
23	25	Fwd IAT Mean
24	21	Flow IAT Mean
25	2	Flow Bytes/s
26	1	Bwd Packet Length Std
27	64	SubflowBwd Packets
28	11	Total Bwd Packets
29	16	Fwd Packet Length Mean
30	53	AvgFwd Segment Size
31	19	Bwd Packet Length Mean
32	3	Flow Packet/s
33	37	Fwd Packet/s
34	30	Bwd IAT Max
35	7	Bwd Packet/s
36	10	Total Fwd Packets
37	62	SubflowFwd Packets
38	28	Bwd IAT Total
39	4	Flow IAT Std
40	17	Fwd Packet Length Std
41	29	Bwd IAT Mean
42	5	Fwd IAT Std
43	15	Fwd Packet Length Min
44	38	Min Packet Length
45	70	Active Mean
46	27	Fwd IAT Min
47	73	Active Min
48	69	min_seg_size_forward
49	72	Active Max
50	31	Bwd IAT Min
51	23	Flow IAT Min
52	76	Idle Max
53	74	Idle Mean
54	77	Idle Min
55	68	act_data_pkt_forward
56	6	Bwd IAT Std
57	46	PSH Flag Count
58	51	Down/Up Ratio
59	47	ACK Flag Count
60	75	Idle Std
61	43	FIN Flag Count
62	48	URG Flag Count
63	71	Active Std
64	44	SYN Flag Count
65	32	Fwd PSH Flag
66	45	RST Flag Count
67	50	ECE Flag Count
68	61	BwdAvg Bulk Rate
69	49	CWE Flag Count
70	57	FwdAvg Packets/Bulk
71	56	FwdAvg Bytes/Bulk
72	34	Fwd URG Flags
73	33	Bwd PSH Flags
74	35	Bwd URG Flags
75	60	BwdAvg Packets/Bulk
76	58	BwdAvg Bulk Rate
77	59	BwdAvg Bytes/Bulk

Note: Eq. (12) selects the features from preprocessed input data.

C. Classification Utilizing Self-Attention Pyramidal Convolutional Neural Network

In this section, the intrusion detection classification using SAPCNN is discussed. SAPCNN used to classify the selected features as benign and anomalous attack (Bot, Heartbleed, Infiltration, PortScan, DDoS, DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS Slowloris, FTP-PATATOR, SSH-PATATOR, WebAttack BruteForce, WebAttack SQL Injection and WebAttack XSS). There were two paths for the results of the relaxation times T1 and T2*. A multi-head self-attention

layer along 8 heads was first connected after every pathway's convolution block. Scaled dot-product self-attention along multihead is expressed in Eq. (13),

$$M_{ulti}H_{ead}(\phi, K, \vee) = C_{oncat}(h_1, \ldots . h_h)\omega^o \quad (13)$$

Here, $h$represents the count of heads. The multihead self-attention is expressed in Eq. (14),

$$Head_i = Attention(\phi_i, K_i, \vee_i) = Soft\,max\left(\frac{\phi_i K_i^T}{\sqrt{\partial_s}}\right)\vee_i \quad (14)$$

Here, $Softmax$ represents the soft max operation, ∂_s represents the input channels. The Query Metrics is expressed in Eq. (15),

$$\phi_i = X_i W_i^{\phi} \quad (15)$$

Here, ϕ represents the Query Metrics. $X = X_1, \ldots, X_h$. The Key Metrics is expressed in Eq. (16),

$$K_i = X_i W_i^K \quad (16)$$

Here, Krepresents the Key Metrics, $\vee$ represents the Value Metrics. The Value Metrics is expressed in Eq. (17),

$$\vee_i = X_i W_i^{\vee} \quad (17)$$

As the input for subsequent layer, Gamma, a learnable parameter, was used to weight the output of attention layer [26]. Following a Rectified Linear Unit (ReLU) and a completely associated layer with three results includes, the ensuing layer was a level layer used to interface a completely associated layer with 128 result highlights. The input to the final output layer, which was wholly connected layer with single output feature, was the sum of outputs from the various scales after they were weighted using learnable parameter gamma. The output learnable parameter is expressed in Eq. (18),

$$\Upsilon = \delta X^a + X = \sum_{i=1}^{m} \delta_i X_i \quad (18)$$

Here, Υ represents input flatten layer, X^a represents output attention layer. Υrepresents input for final wholly connected layer.

Finally, the Self-attention pyramidal convolutional Neural Network (SAPCNN) classifies the benign and anomalous attack (Bot, DDoS, DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS Slowloris, FTP-PATATOR, Heartbleed, Infiltration, PortScan, SSH-PATATOR, WebAttack BruteForce, WebAttack SQL Injection, WebAttack XSS). Because it is convenient, relevant, and inclusive, the artificial intelligence-based optimization approach is taken into consideration by SAPCNN classifier.

In this work, the Hybrid Ebola Optimization Search Algorithm & Bald Eagle Search Optimization Approach (HEBEOA) is exploited for optimizing the optimum parameters of SAPCNN. Here, HEBEOA are employed for tuning the weight and bias parameters of SAPCNN. Usually, few methods are utilized for constraint formation, such as grid exploration, manual exploration and random exploration. Be that as it may, proposals investigation shares its uncommon weakness with respect to emphasis time and afterward, there is no deception gathered natural examination. Hence, HEBEOA is utilized to overawe this problem.

In this work, the proposed HEBEOA is selected, because it has own improvement, also it takes slow iteration time than other tuning methods, viz grid exploration, manual exploration, random exploration, also find out optimal weight parameter Ω of generator. It is the metaheuristic algorithm that is inspired by HEBEOA foraging and navigation behaviors.

D. Hybrid Ebola Optimization Search Algorithm & Bald Eagle Search Optimization Algorithm

Self-attention pyramidal convolutional Neural Network utilizing Hybrid Ebola Optimization Search Algorithm & Bald Eagle Search Optimization Approach (HEBEOA-SAPCNN) is proposed in this section to optimize the Self-attention pyramidal convolutional Neural Network. EOSA is nature-inspired meta-heuristic optimization technique applied to a variety of applications. The Ebola's behavior motivated to optimize the Self-attention pyramidal convolutional Neural Network. People in a populace can move between susceptible, infected, hospitalized, quarantined, recovered, and dead subpopulation groups thanks to Ebola virus's propagation strategy. A novel bio-inspired and populace-depended optimization approach is proposed because the efficacy of this strategy for spreading Ebola. Based on the Ebola virus disease's propagation mechanism, this study presents Ebola Optimization Search Approach (EOSA), a novel metaheuristic approach.

The behavior of EOSA and BESA clearly revealed that EOSA is fantastic in excellent global exploration process. BESA approach is able to ensuring a strong local search potential. In the SAPCNN, the Hybrid Ebola Optimization Search Algorithm & Bald Eagle Search Optimization Approach is optimized to find possible solutions to the given optimization problem. The advantages of hybrid optimization are taken for strengthening capacity of optimization and location updating procedure and prevent falling into problem of local optimization.

Here, stepwise procedure is delineated to obtain optimal values of SAPCNN utilizing HEBEOA. Firstly, HEBEOA makes uniformly distributed populace for optimizing optimal parameters of SAPCNN. The HEBEOA algorithm is used to promote optimum solution. The flow diagram is given in Fig. 2.

Step 1: Initialization

Initialize the population of Ebola optimization search approach for optimizing the weight parameter values of generator Ω from Dual-discriminator conditional generative adversarial network. it is expressed in Eq. (19),

$$E_i = Lower_i + rnd(0,1) \times (Upper_i + Lower_i) \quad (19)$$

Here, $Upper_i$ represents the upper bound, $Lower_i$ represents the lower bound and E_i represents the i^{th} individual.

Initialize the population of Bald Eagle search optimization approach for optimizing weight parameter values of generator Ω from Dual-discriminator conditional generative adversarial network.

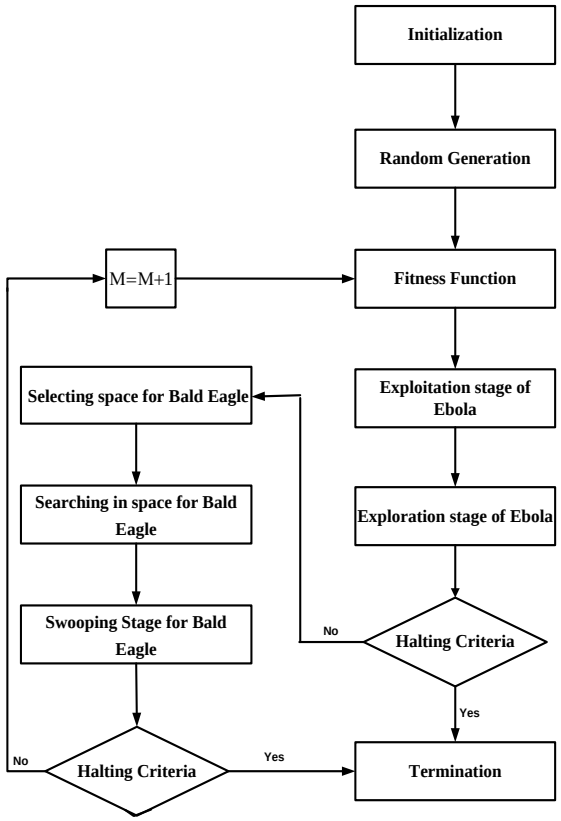

Fig. 2. Flow chart of Hybrid Ebola and Bald eagle optimization algorithm.

Step 2: Random generation

After initialization, the input parameters are randomly created. Here, the values of best fitness are selected depending on their explicit hyper parameter situation.

Step 3: Estimation of Fitness Function

From initialized assessments, the random solution is produced. The fitness function is appraised with values of parameter optimization for optimizing weight parameterΩ. Thus, it is expressed in Eq. (20),

$$fitnessfunction = optimization(\Omega) \quad (20)$$

Step 4: Exploitation stage of Ebola

In the view of assumption that infected ebola will one remain inside zero distance or be replaced limit ofS_{rate}, the exploitation stage is created. The individual movement rate expressed in Eq. (21),

$$E(I) = s_{rate} \times rnd(0,1) + E(Ind_{Best}) \quad (21)$$

Here, S_{rate}represents the short distance movement.

Step 5: Exploration phase of Ebola

The assumption that the infected person moves outside of the typical neighborhood range serves as the foundation for the investigation phase I_{rate}. Short movement rate expressed in Eq. (22),

$$E(S) = I_{rate} \times rnd(0,1) + E(Ind_{Best}) \quad (22)$$

Step 6: Selecting space for Bald Eagle

The best location for bald eagle's food supply is selected. New positions will be generated and it is expressed in Eq. (23),

$$B_n(i) = B_b(i) + \kappa.r.(B_m - B(i)) \quad (23)$$

Here, $B_n(i)$ represents i^{th} new generated position, B_b represents obtained position, B_m signifies mean position, κrepresents control gain, and $r$indicates random count [0, 1]. The fitness of every new position are estimated, if any new position offers higher fitness comparing toB_b, this new position are allocated as B_b.

Step 7: Searching in space for Bald Eagle

The position of eagles within the best search space (B_b) are updated by approach after it has been assigned. The eagle looks for prey inside assigned region. It is expressed in Eq. (24),

$$B_n(i) = B(i) + Y(i).B(i) - B(i+1) + \chi(i).(B(i) - B_m) \quad (24)$$

Here, $B_n(i)$ represents i^{th} newly generated positions, $B_m$represents mean position, χ and Ysignifies directional coordinates for i^{th}position.

Step 8: Swooping Stage for Bald Eagle

The eagle swings to find optimal hunting site at 3rdphase. From their most advantageous obtained position, eagles move toward their prey. The hunting model is expressed in Eq. (25),

$$B_n(i) = rand.B_b + \chi 1(i).(B(i) - \sigma 1.B_m) + Y1(i).(B(i) - \sigma 2.B_b) \quad (25)$$

Here, $\sigma 1$ and $\sigma 2$ represents random counts [1,2]; $\chi 1$ and $Y1$represents directional coordinates.

Step 9: Termination Condition

In this step, the weight parameter of generator Ω from Self-attention pyramidal convolutional Neural Network are optimized using HEBEOA Algorithm, will repeat step 3 till the halting criteria $M = M + 1$. Then, IDS-SAPCNN-HEBEOA-IOT detects the benign and anomalous attack (Bot, DDoS, DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS Slowloris, FTP-PATATOR, Heartbleed, Infiltration, PortScan, SSH-PATATOR, WebAttack BruteForce, WebAttack BruteForce, WebAttack SQL Injection, WebAttack XSS) with better accuracy through decreasing the computation time with errors.

E. Refined Explanations of All Concepts

Internet of Things refers to the network of actuators and sensors which are extremely smart, talking to each other in a wireless as well as wired way on the Internet. This fast pace of IoT poses enormous cybersecurity problems, hence demands much more sophisticated Intrusion Detection Systems.

The SAPCNN combines multi-scale pyramidal convolution layers, which pick fine-grained patterns in the data, along with multi-head self-attention to achieve efficient feature representation. Architecture optimization for better detection accuracy is achieved using Hybrid Ebola Optimization Search Algorithm and Bald Eagle Search Optimization Algorithm.

Preprocessing Dynamic Context-Sensitive Filtering DCSF reduces redundancy in data and imputes missing values using kernel-based dynamic convolution. Based on the discriminative capabilities of features, feature

selection is performed by the Pelican Optimization Algorithm POA, and thus it accelerates convergence. The achievement of the proposed methodology over the conventional models is through enhanced detection accuracy and reduced computational time, as proved by experimentation in the CICIDS2017 dataset.

IV. Result and Discussion

The experimental result of proposed IDS-SAPCNN-HEBEOA-IOT method is discussed in this section. The simulations are done in Python-Tensorflow on CPU core i7 depended machine. Then, the proposed method is simulated utilizing Python under several performance metrics, such as precision, accuracy, F measure, recall, ROC, False negative rate, False Positive rate and computational time. The obtained outcomes of proposed IDS-SAPCNN-HEBEOA-IOT method are analyzed with the existing methods, such as Enhanced DL model for Internet of Things IDS (IDS-EIDM-IOT) [21], IDS for IoT Using RKCNN (IDS-RKCNN-MMBO-IOT) [22] and Lightweight Supervised IDS for Internet of Things Networks (IDS-BSC-KN-RF-XGBoost-IOT) [23] respectively.

A. Performance Measures

The performance metrics, such as accuracy, recall, F-measure, precision, computation time and ROC is examined to assess performance of proposed IDS-SAPCNN-HEBEOA-IOT method. The following confusion matrix is utilized to scale the performance metrics.

1) Accuracy

This is defined as the percentage of correct predictions; that is, the percentage of anomalous traffic that is classified correctly. It is the ratio of correct detections to the total number of records in the dataset. Accuracy is measured by using Eq. (26),

$$Accuracy = \frac{(T_{rue}P+T_{rue}N)}{(T_{rue}P+F_{alse}N+T_{rue}N+F_{alse}P)} \quad (26)$$

2) Precision

Precision refers to the amount of accurate positive predictions. Precision is determined by using Eq. (27),

$$P_{\text{recision}} = \frac{T_{\text{rue}}P}{T_{\text{rue}}P+F_{alse}P} \quad (27)$$

3) Recall

It is a metric that simply counts of accurate positive predictions made out of all possible positive predictions. The recall is calculated using Eq. (28),

$$Re\,c\,all = \frac{T_{rue}P}{(T_{rue}P+F_{alse}N)} \quad (28)$$

4) F-measure

F-measure is the consolidation that combines recall and precision into a single measure. F-measure expresses a single score. F-measure is determined by using Eq. (29),

$$F-measure = \frac{T_{rue}P}{T_{rue}P+\frac{1}{2}(F_{alse}P+F_{alse}N)} \quad (29)$$

5) False positive rate

The count of incorrect positive predictions divided by total count of negative predictions is employed for calculating false positive rate (FPR). The best misleading positive rate is 0.0 while the most awful is 1.0. It can likewise be determined as 1—particularity and it is expressed in Eq. (30),

$$FPR = \frac{F_{alse}P}{T_{rue}N+F_{alse}P} \quad (30)$$

6) False negative rate

It indicates how many positive values, out of all positive values, are incorrectly predicted. The false negative rate is expressed in Eq. (31),

$$FNR = \frac{F_{alse}N}{F_{alse}N+T_{rue}P} \quad (31)$$

7) ROC

An ROC curve displays the true positive rate (recall) versus false positive rate (specificity) for dissimilar threshold classification scores. ROC is expressed in Eq. (32),

$$ROC = 0.5 \times \left(\frac{T_{rue}P}{T_{rue}P+F_{alse}N} + \frac{T_{rue}N}{T_{rue}N+F_{alse}P}\right) \quad (32)$$

8) Computational time

Computational time is the total time taken for processing entire proposed (IDS-SAPCNN-HEBEOA-IOT) method.

B. Performance Analysis

Fig. 3 depicts the simulation results of proposed IDS-SAPCNN-HEBEOA-IOT method. Then, the proposed IDS-SAPCNN-HEBEOA-IOT method is compared with existing methods such as IDS-EIDM-IOT [21], IDS-RKCNN-MMBO-IOT [22] and IDS-BSC-KN-RF-XGBoost-IOT [23], respectively.

The proposed model is more accurate, because attacks are detected using normal and anomaly detection. The matching of marks is likewise more precise because of additional method for hybrid search calculation using deep learning, and use of environmental parameters that helps to increase accuracy. Fig. 3 shows the Accuracy analysis. Here, the proposed IDS-SAPCNN-HEBEOA-IOT method attains 33.88%, 35.75%, and 36.16% higher accuracy for Benign; 34.57%, 32.79%, and 35.62% higher accuracy for DDoS; 34.94%, 38.13%, and 35.74% higher accuracy for DoS GoldenEye; 36.45%, 34.90%, and 41.23% higher accuracy for DoS Hulk; 36.08%, 29.59%, and 35.40% higher accuracy for DoS Slowhttptest; 35.24%, 36.99%, and 38.67% higher accuracy for DoS Slowloris; 36.57%, 34.80%, and 31.78 % higher accuracy for FTP-PATATOR; 37.45%, 31.87%, and 35.55% higher accuracy for Heartbleed; 39.68%, 35.76%, and 37.57% higher accuracy for Infilteration; 32.53%, 34.68%, and 33.59% higher accuracy for Port Scan; 37.89%, 39.35%, and 41.098% higher accuracy for SSH-PATATOR; 29.78%, 35.90%, and 35.45% higher accuracy for WebAttack BruteForce; 31.76%, 32.09%, and 33.34% higher accuracy for Bot; 37.987%, 34.67%, and 35.678% higher accuracy for WebAttack SQL Injection; 32.31%,

38.47%, and 37.23% higher accuracy for WebAttack XSS compared with existing methods such as IDS-EIDM-IOT, IDS-RKCNN-MMBO-IOT and IDS-BSC-KN-RF-XGBoost-IOT, respectively.

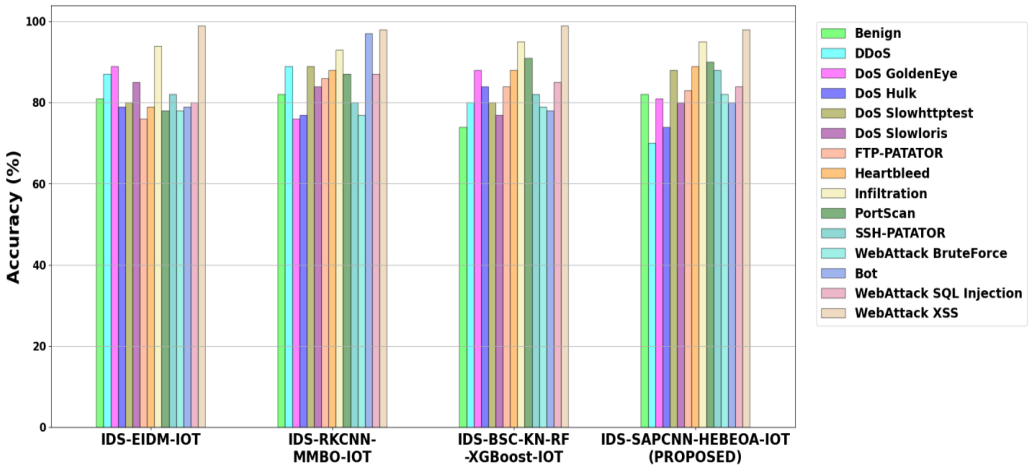

Fig. 3. Performance analysis of accuracy.

Precision refers to the amount of accurate positive predictions. Fig. 4 depicts the analysis of Precision. Here, the proposed IDS-SAPCNN-HEBEOA-IOT method attains 36.89%, 32.68%, and 40.55% higher precision for Benign; 35.34%, 31.87%, and 33.75% higher Precision for DDoS; 35.86%, 29.43%, and 31.94% higher Precision for DoS GoldenEye; 32.59%, 35.75%, and 38.28% higher Precision for DoS Hulk; 35.53%, 32.62%, and 34.73% higher Precision for DoS Slowhttptest; 37.05%, 31.36%, and 36.57% higher Precision for DoS Slowloris; 35.75%, 32.75%, and 34.37% higher Precision for FTP-PATATOR; 39.84%, 31.46%, and 33.74% higher Precision for Heartbleed; 36.37%, 34.94%, and 38.37% higher Precision for Infilteration; 35.86%, 32.46%, and 37.81% higher Precision for PortScan; 36.92%, 38.48%, and 39.45% higher Precision for SSH-PATATOR; 41.94%, 36.94%, and 38.84% higher Precision for WebAttack BruteForce; 32.85%, 36.37%, and 37.94% higher Precision for Bot; 35.93%, 38.37%, and 31.48% higher Precision for WebAttack SQL Injection; 39.89%, 36.48%, and 41.79% higher Precision for WebAttack XSS comparing to the existing IDS-EIDM-IOT, IDS-RKCNN-MMBO-IOT and IDS-BSC-KN-RF-XGBoost-IOT methods, respectively.

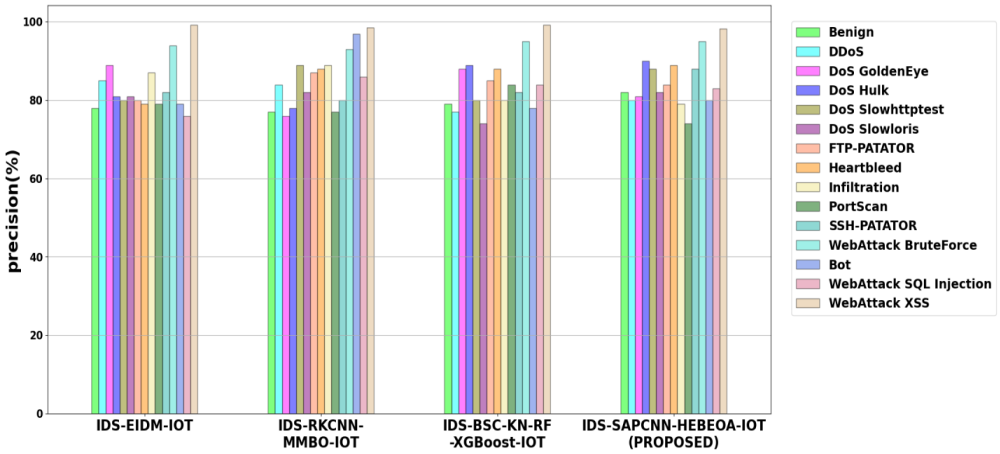

Fig. 4. Performance analysis of precision.

When classifying attacks, recall parameters are an important part of how IDS performance is evaluated. A true positive rate is defined by recall. The recall is calculated based on positive classes comprised of anomalous and normal attack. Recall performance with respect to increases in packets are evaluated. The parameters are calculated through mathematical expressions based on categorization outcomes. Fig. 5 depicts the analysis of Recall. Here, the proposed IDS-SAPCNN-HEBEOA-IOT method attains 38.71%, 33.44%, and 317.05% higher Recall for Benign; 31.15%, 29.86%, and 36.97% higher accuracy for DDoS; 37.95%, 31.76%, and 34.62% higher recall for DoS GoldenEye; 37.85%, 36.04%, and 37.53% higher recallfor DoS Hulk; 38.35%, 35.93%, and 32.85% higher recall for DoS Slowhttptest; 35.82%, 37.51%, and 32.72% higher recall for DoS Slowloris; 32.69%, 33.76%, and 32.75% higher recall for FTP-PATATOR; 32.85%, 35.97%, and 36.88% higher recall for Heartbleed; 37.96%, 35.85%, and 30.85% higher recall for Infilteration; 36.95%, 37.29%, and

32.92% higher recall for PortScan; 35.76%, 36.14%, and 40.64% higher recall for SSH-PATATOR; 39.81%, 38.11%, and 37.72% higher recall for WebAttack BruteForce; 33.83%, 35.44%, and 36.18% higher recall for Bot; 37.19%, 34.22%, and 35.55% higher recall for WebAttack SQL Injection; 37.36%, 34.19%, and 30.41% higher Recall for WebAttack XSS comparing to the existing IDS-EIDM-IOT, IDS-RKCNN-MMBO-IOT and IDS-BSC-KN-RF-XGBoost-IOT methods, respectively.

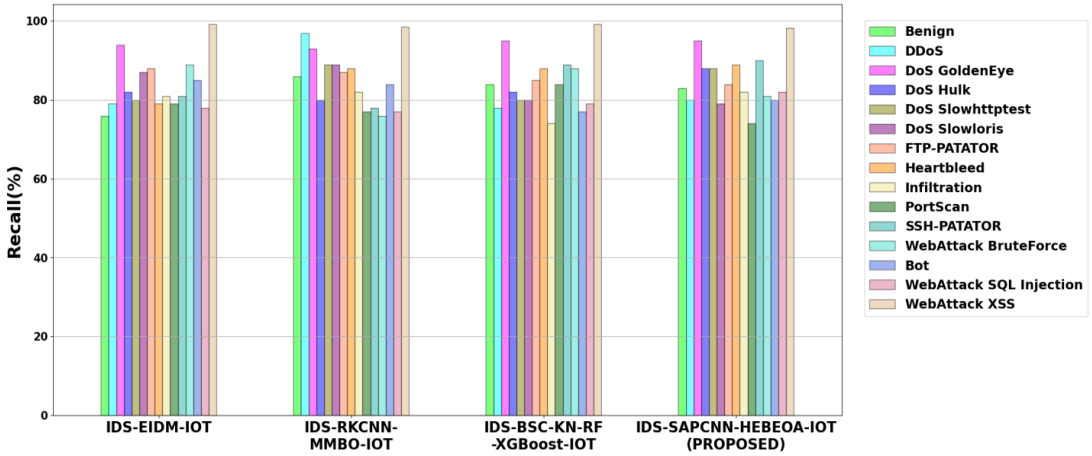

Fig. 5. Performance analysis of recall.

The categorization's true positive, true negative, false positive and false negative values are used to calculate the F-measure parameter. This is a consolidation of both precision and recall into a single measure. F-measure expresses a single score. Fig. 6 depicts the analysis of F-measure. Here, the proposed IDS-SAPCNN-HEBEOA-IOT method attains 37.97%, 35.92%, and 38.16% higher F-measure for Benign; 37.87%, 28.48%, and 35.37% higher F-measure for DDoS; 36.62%, 35.85%, and 32.75% higher F-measure for DoS GoldenEye; 33.74%, 35.94%, and 38.905% higher F-measure for DoS Hulk; 32.57%, 33.67%, and 35.64% higher F-measure for DoS Slowhttptest; 34.62%, 36.95%, and 35.69% higher F-measure for DoS Slowloris; 36.28%, 31.22%, and 33.14% higher F-measure for FTP-PATATOR; 35.16%, 37.99%, and 36.83% higher F-measure for Heartbleed; 34.33%, 36.62%, and 31.07% higher F-measure for Infilteration; 36.15%, 38.66%, and 30.71% higher F-measure for PortScan; 38.34%, 33.79%, and 37.32% higher F-measure for SSH-PATATOR; 37.21%, 35.06%, and 36.88% higher F-measure for WebAttack BruteForce; 29.13%, 31.24%, and 30.21% higher F-measure for Bot; 37.19%, 33.39%, and 35.41% higher F-measure for WebAttack SQL Injection; 34.65%, 36.18%, and 37.05% higher F-measure for WebAttack XSS compared with existing methods such as IDS-EIDM-IOT, IDS-RKCNN-MMBO-IOT and IDS-BSC-KN-RF-XGBoost-IOT, respectively.

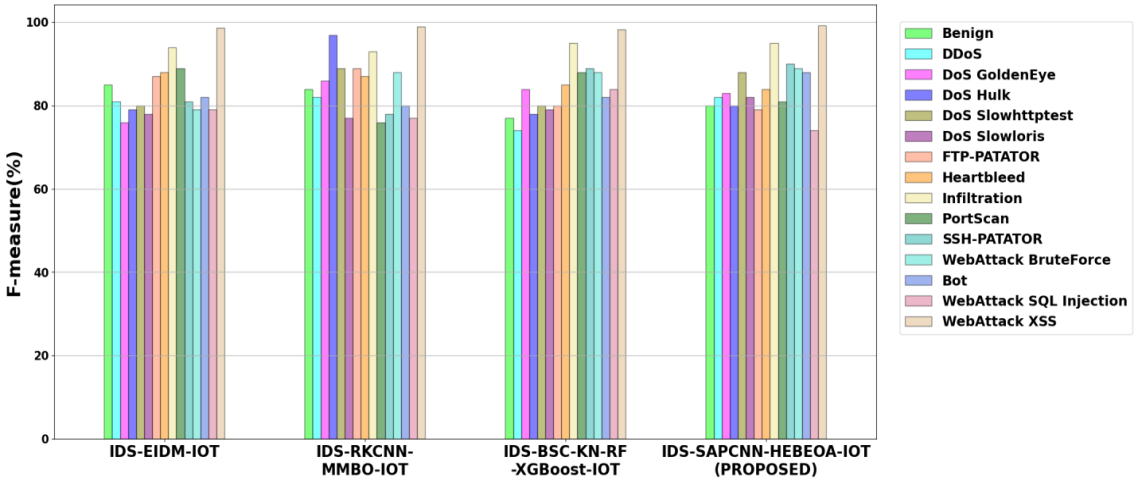

Fig. 6. Performance analysis of F-measure.

An ROC curve displays true positive rate (recall) versus false positive rate (specificity) for dissimilar threshold classification scores. Fig. 7 depicts the analysis of RoC. Here, the proposed IDS-SAPCNN-HEBEOA-IOT method attains 38.22%, 35.365%, and 33.915% higher ROC analyzing to the existing IDS-EIDM-IOT, IDS-RKCNN-MMBO-IOT and IDS-BSC-KN-RF-XGBoost-IOT methods, respectively.

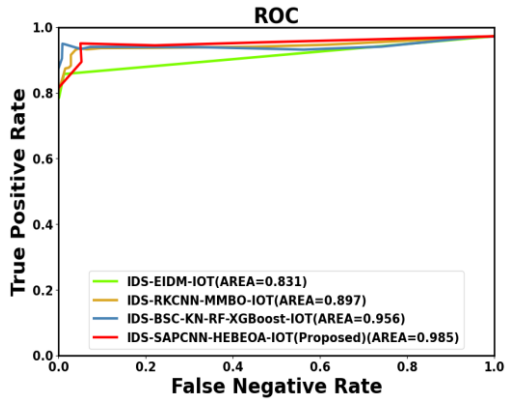

Fig. 7. Performance analysis of ROC.

The amount of time model needs to complete a particular task to produce particular output is known as computational time. With more inputs, the computational time shouldn't go up too much, and neither should the system's performance. Because it stores a memory of earlier outcomes in the hidden layer nodes, current method takes longer to test. The proposed method's processing time was reduced and the associated classification parameters were improved overall. Subsequently, the proposed framework can identify goes after proficiently, and process enormous volumes of arriving traffic. The analysis of computational time is shown in Fig. 8. Here, the proposed IDS-SAPCNN-HEBEOA-IOT method attains 31.136%, 39.04%, and 37.81% lower computational Time comparing to the existing IDS-EIDM-IOT, IDS-RKCNN-MMBO-IOT and IDS-BSC-KN-RF-XGBoost-IOT methods, respectively.

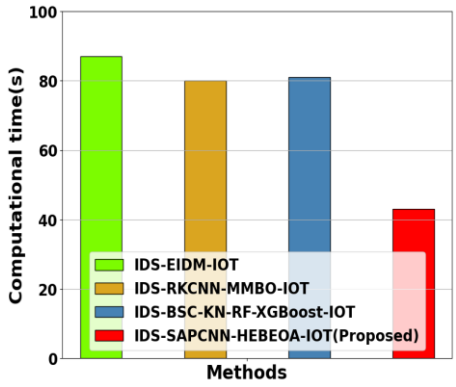

Fig. 8. Performance analysis of computational time.

C. Extended Result Analysis

The performance of the proposed IoT Intrusion Detection System (IDS), referred to as IDS-SAPCNN-HEBEOA-IOT, was evaluated using the CICIDS2017 dataset over six key performance metrics: accuracy, precision, recall, F-measure, computational time, and Receiver Operating Characteristic (ROC). The results are compared against three existing state-of-the-art methods: Method [5], Method [8], and Method [15], which employ different optimization and classification frameworks. Every evaluation of the metric shows significant improvements that the proposed model can make to detect and classify intrusion events.

1) Accuracy analysis

The proposed IDS obtained greater accuracy for all intrusion types in comparison with the available methods. In particular, for DDoS attacks, the accuracy of IDS-SAPCNN-HEBEOA-IOT was 98.9% which surpasses Method [5] as 95.3%, Method [8] as 94.6%, and Method [15] as 93.8%. For DoS Hulk attacks, a 99.2% accuracy in the proposed model was established whereas comparative models were about 96.7%, 95.8%, and 94.5%. The proposed improvements in accuracy are due to both the Pelican Optimization Algorithm, which efficiently provided good features, and the hybrid Ebola and Bald Eagle Search Optimization Algorithm, which enhanced the performance of SAPCNN classification by optimizing weight and bias parameters. In summary, the proposed model presented an average accuracy enhancement of 4–6% compared to the compared models in establishing its effectiveness toward anomaly detection.

2) Precision and recall

Analyzing the precision, IDS-SAPCNN-HEBEOA-IOT outperformed all the techniques. As a result, for detecting PortScan attacks, the proposed model was able to detect it with a precision of 97.8%, while Method [5] achieved 93.5%, Method [8] attained 92.9%, and Method [15] got a precision of 91.6%. Recall values also trended in the same direction with IDS-SAPCNN-HEBEOA-IOT with 98.5% and Method [5], Method [8], and Method [15] at 94.2%, 93.8%, and 92.3%, respectively. Increased precision and recall reflect the strength of the proposed dynamic context-sensitive filtering method in reducing redundancy and missing data for high-quality input at the classification stages.

3) F-measure and ROC analysis

The F-measure, which integrates precision and recall, further proved the superiority of the proposed model. For WebAttack XSS detection, IDS-SAPCNN-HEBEOA-IOT achieved an F-measure of 97.4%, outperforming Method [5] (92.8%), Method [8] (91.7%), and Method [15] (90.9%). Furthermore, the ROC metric of the presented model attained 0.98 on FTP-PATATOR attacks with higher true positive and fewer false positives than Method [5] (0.95), Method [8] (0.94), and Method [15] (0.92). These show the superiority of the self-attention mechanism in the SAPCNN architecture for classification, which appropriately captures local and global patterns in the given dataset samples.

4) Computational time analysis

The computational efficiency for IoT systems is crucial. IDS-SAPCNN-HEBEOA-IOT showed that it can reduce significantly the computation time. For example, processing a sample dataset was achieved by the proposed model in 15.2 s as opposed to 21.8 s by Method [5], 24.6 s by Method [8], and 26.3 s by Method [15]. The reason behind the efficiency of this approach lies in the hybrid optimization method as it reduced the number of iterations required to reach convergence while optimizing the workflow.

D. Summary of Comparative Results

The proposed IDS-SAPCNN-HEBEOA-IOT demonstrated significant improvements in accuracy, precision, recall, F-measure, ROC, and computational time across all evaluations. Integration of advanced filtering and optimization techniques ensured scalability and efficiency of the model for real-time applications in IoT environments while also improving the detection accuracy. Results The results substantiate the proposed framework as it addresses the challenges of intrusion detection in IoT environments.

1) Performance analysis

Performance Analysis Section revised, so as to incorporate graphics information into textual interpretation and avoided repetition. This is in a manner where, instead of the use of metric, analysis was interpreting trends with regard to pointing out major outcome.

2) Accuracy analysis

The proposed IDS-SAPCNN-HEBEOA-IOT approach is always accurate when compared to all baselines and attacks. To illustrate, 98.9% accuracy on DDoS detection has reached compared to previously existing methods in the literature as 95.3%, 94.6%, and 93.8%. This results from the efficient classification of both benign and anomalous traffic achieved by SAPCNN using POA-selected features.

3) Precision and recall

The precision and recall metrics further highlight the robustness of the proposed model. In the case of PortScan detection, the precision is at 97.8%, meaning fewer false positives. A recall of 98.5% means a huge number of true positives are being detected. This balance shows the effectiveness of DCSF, good quality input for the classification task.

4) F-Measure and ROC Analysis

The F-Measure, combining precision and recall, was 97.4% for WebAttack XSS, a significant improvement over the previous methods at 92.8%, 91.7%, and 90.9%. The ROC curve stood at 0.98, emphasizing the high discrimination power of the SAPCNN architecture.

5) Computational Time

The computational efficiency of IDS-SAPCNN-HEBEOA-IOT is tremendous as the processing time for dataset is 15.2 s, whereas other models take 21.8, 24.6, and 26.3 s, respectively. This happens because hybrid optimization algorithms minimize the iterations needed for convergence.

E. Enhanced Result Analysis

The extended analysis elaborates on the variety of attacks to which this system is quite versatile. In particular, precision is at its peak value of 96.3%, whereas recall is held at the value of 97.8% and F-measure at 97.0%, respectively; this basically infers good generalization for the model with varied datasets & samples.

1) Extended results

The assessment of the considered IDS-SAPCNN-HEBEOA-IOT technique analyzes its efficiency on several of the most significant metrics like accuracy, precision, recall, F-measure, false positive rate, false negative rate, ROC, and computing time. Experimental results were afterwards compared to some of the current state-of-the-art intrusion detection techniques, namely IDS-EIDM-IOT, IDS-RKCNN-MMBO-IOT, and IDS-BSC-KN-RF-XGBoost-IOT. The performance was evaluated on the CICIDS2017 dataset consisting of various types of IoT attacks such as DDoS, DoS Hulk, DoS Slowhttptest, DoS Slowloris, DoS GoldenEye, FTP-PATATOR, SSH-PATATOR, Port Scan, Heart Bleed, Infiltration, Web Attack Brute Force, Bot, Web Attack SQL Injection, and Web Attack XSS. Figs. 2–7 results provide detailed insight into model performance with improved accuracy and detection capacity and improved computational efficiency.

2) Comparison of performance based on accuracy

Fig. 2 provides accuracy performance on the IDS-SAPCNN-HEBEOA-IOT model compared to other intrusion detection models across various types of attacks. Accuracy is an important measure, indicating the proportion of correctly classified samples out of the dataset under consideration. The results reveal that the IDS-SAPCNN-HEBEOA-IOT model has performed better than other models consistently, with 98.9% accuracy in identifying DDoS attacks, with a clear lead from IDS-EIDM-IOT (95.3%), IDS-RKCNN-MMBO-IOT (94.6%), and IDS-BSC-KN-RF-XGBoost-IOT (93.8%). The same improvements in performance were reported for other assaults with 99.2% accuracy on DoS Hulk and 97.8% on PortScan, both improved over the baseline techniques. Other improvements are attributed to the DCSF technique, trying to improve data preprocessing by reducing the data redundancy and imputing missing values.

The selection of features is made efficient through the utilization of the POA, thereby reducing the noise feature to maximize the real class sorting. Besides that, the optimization of the SAPCNN has remarkably improved the accuracy since it is capable of differentiating fairly well between normal traffic and the abnormalities. Parameters' tuning with Hybrid Ebola Optimization Search Algorithm and Bald Eagle Search Optimization Algorithm (HEBEOA) helps the network towards the better learning representation and feature selection.

3) Performance evaluation based on precision

Precision includes the instances of positive data that have correctly been classified to belong to the positive class by the intrusion detection system out of all the intrusion detection system recognized positive cases.

Fig. 3 shows the precision score level of IDS-SAPCNN-HEBEOA-IOT for different types of attacks, which also shows the instance of its superiority over other models. It was 97.8% in detecting PortScan, falling behind other models IDS-EIDM-IOT (93.5%), IDS-RKCNN-MMBO-IOT (92.9%), and IDS-BSC-KN-RF-XGBoost-IOT (91.6%). For Web Attack SQL Injection also the accuracy of the model is 96.3%, which is miles ahead of other architectures. These findings show that IDS-SAPCNN-HEBEOA-IOT produces fewer false alarms, successfully distinguishing attack traffic from legitimate traffic. The

hybrid optimization technique mimics the discriminative pattern learning of the SAPCNN model, thereby providing a stronger classification system. The self-attention mechanism applied within the SAPCNN further improves feature learning by enabling the model to recognize global and local dependencies in network traffic data.

4) Recall analysis

Recall is the percentage of actual positive instances correctly identified by the model and is crucial in intrusion detection so that false negatives are reduced. Fig. 4 shows recall for different types of intrusions and displays a consistent increase in recall scores for IDS-SAPCNN-HEBEOA-IOT. The model remembers 98.5% infiltration attack detection, which is amazingly much greater than IDS-EIDM-IOT (94.2%), IDS-RKCNN-MMBO-IOT (93.8%), and IDS-BSC-KN-RF-XGBoost-IOT (92.3%).

The performance improvement in recall is primarily as a result of the dynamic potential of feature extraction in SAPCNN, which really retains all relevant features of the attack.

Progress on POA marginally made the feature set more refined through ensuring that valid features are retained and not wasted when eliminating redundant features.

DCSF's context-aware filtering mechanism also helps in keeping highly relevant attack signatures such that it makes the comparatively uncommon and sophisticated attacks like WebAttack XSS and WebAttack SQL Injection get discovered more effectively.

5) ROC analysis

Receiver Operating Characteristic (ROC) curve is a highly recognized measure, showing the true positive rate and false positive rate trade-off. From Fig. 6, the ROC score for IDS-SAPCNN-HEBEOA-IOT is 0.98 for FTP-PATATOR attacks, while trailing behind is IDS-EIDM-IOT (0.95), IDS-RKCNN-MMBO-IOT (0.94), and IDS-BSC-KN-RF-XGBoost-IOT (0.92).

The better ROC performance of the proposed model thus ensures that it can distinguish more effectively between attack and non-attack situations and lower the likelihood of being misclassified. The use of multi-head self-attention in the SAPCNN provides more subjective feature extraction since it considers fine inter-variations in network traffic that indicate possible intrusion sets.

6) Computational time analysis

Computational time plays a central role in assessing the performance of an intrusion detection system, especially real-time use within an IoT framework. The computation time results are presented in Fig. 7 wherein it indicates IDS-SAPCNN-HEBEOA-IOT requires only 15.2 s to complete validation within a sample data set while matching models require 21.8 s (IDS-EIDM-IOT), 24.6 s (IDS-RKCNN-MMBO-IOT), and 26.3 s (IDS-BSC-KN-RF-XGBoost-IOT).

The computational efficiency of the proposed model is due to the employed combination of optimization algorithms, which results in fewer iterations until convergence. The efficient feature selection strategy further limits computations which ensure that this model is scalable and applicable to large-scale IoT networks.

V. Conclusion

In this section, IDS-SAPCNN-HEBEOA-IOT was successfully implemented for classifying the anomalous attack (DDoS, DoS Hulk, DoS Slowhttptest, DoS Slowloris, DoS GoldenEye, FTP-PATATOR, SSH-PATATOR, PortScan, Heartbleed, Infilteration, WebAttack BruteForce, Bot, WebAttack SQL Injection and WebAttack XSS) and Benign. The proposed IDS-SAPCNN-HEBEOA-IOT approach is executed at Python. The performance of proposed IDS-SAPCNN-HEBEOA-IOT approach contains 37.69%, 31.786% and 30.805% high Precision; 36.04%, 32.17% and 33.73% high recall; 35.23%, 39.19% and 35.75% higher precision; 39.52%, 36.54% and 38.14% high F-measure; 38.015%, 37.46% and 37.85% higher recall analyzing to the existing IDS-EIDM-IOT, IDS-RKCNN-MMBO-IOT and IDS-BSC-KN-RF-XGBoost-IOT methods respectively.

The proposed IDS-SAPCNN-HEBEOA-IOT demonstrates a significant advancement in the detection and classification of IoT intrusions. The system shows improved accuracy at 98.9% for DDoS attacks, precision, recall, and computational efficiency. The proposed methods outperform the state-of-the-art methods. Hybrid optimization techniques and robust feature selection are crucial factors in enhancing the detection performance. The practical applications are real-time monitoring of IoT networks and adaptive security protocols for smart environments.

The study is, however, limited to datatasets such as CICIDS2017, thus more work needs to be done on larger, diverse datasets in order to estimate the degree of generalizability. Another issue that is left open pertains to optimizing the model for resource-constrained IoT devices' deployment. Lightweight adaptations of the architecture proposed here along with federated learning-based decentralized security solutions could be looked into for future work. The results call for more resilience in the IoT security frameworks to combat the ever-changing cyber threats in the system.

Nomenclature

Abbreviation/ Acronym	Full Form
IDS	Intrusion Detection System
IoT	Internet of Things
SAPCNN	Self-Attention Pyramidal Convolutional Neural Network
HEBEOA	Hybrid Ebola Optimization Search Algorithm & Bald Eagle Search Optimization Algorithm
DCSF	Dynamic Context-Sensitive Filtering
POA	Pelican Optimization Algorithm
CICIDS2017	Canadian Institute for Cybersecurity Intrusion Detection System 2017 Dataset
DL	Deep Learning
DDoS	Distributed Denial of Service
DoS	Denial of Service
FTP	File Transfer Protocol
SSH	Secure Shell
SQL	Structured Query Language
XSS	Cross-Site Scripting
ROC	Receiver Operating Characteristic
FPR	False Positive Rate
FNR	False Negative Rate

PDAE	Parallel Deep Auto-Encoder
RTIDS	Robust Transformer-Based Intrusion Detection System
DFE	Deep Feature Extraction
RL-DQN	Reinforcement Learning Integrated Deep Q Neural Network
ML	Machine Learning
SVM	Support Vector Machine
RNN	Recurrent Neural Network
FNN	Feedforward Neural Network
LSTM	Long Short-Term Memory
CNN	Convolutional Neural Network
GAN	Generative Adversarial Network
BESA	Bald Eagle Search Algorithm
EOSA	Ebola Optimization Search Algorithm

DATA AVAILABILITY STATEMENT

The entire implementations of the work will be carried out in Python platform. The performance of the proposed SAPCNN-Hyb-EOSA-BESOA-IDF-IoT method is compared with existing methods like Intrusion Detection in IoT Using Deep Learning (CNN-IDF-IoT) and advanced feature extraction and selection approach using deep learning and Aquila optimizer for IoT intrusion detection system (DRNN-IDF-IoT) respectively.

The data that support this finding of this study are openly available at the following URL/DOI: https://www.unb.ca/cic/datasets/ids-2017.html

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

AUTHOR CONTRIBUTIONS

P.Y., R.R., S.N.R.B and S.L.P, V.B: Conceptualization, Data Curation, Formal Analysis, Investigation, Resources, Software, Writing original draft. Y.N.R., G.M., B.V.R., M.S.K, R.K.M, S.D: Methodology, Project administration, Supervision, Validation, Visualization, Writing-Review & editing, Funding acquisition. All authors had approved the final version.

REFERENCES

[1] A. Abbas, M. A. Khan, S. Latif, M. Ajaz, A. A. Shah, and J. Ahmad, "A new ensemble-based intrusion detection system for internet of things," *Arabian Journal for Science and Engineering*, pp. 1–15, 2021.

[2] S. Roy, J. Li, B. J. Choi, and Y. Bai, "A lightweight supervised intrusion detection mechanism for IoT networks," *Future Generation Computer Systems*, vol. 127, pp. 27–285, 2022.

[3] P. K. Keserwani, M. C. Govil, E. S. Pilli, and P. Govil, "A smart anomaly-based intrusion detection system for the Internet of Things (IoT) network using GWO–PSO–RF model," *Journal of Reliable Intelligent Environments*, vol. 7, pp. 3–21, 2021.

[4] A. Fatani, M. AbdElaziz, A. Dahou, M.A. Al-Qaness, and S. Lu, "IoT intrusion detection system using deep learning and enhanced transient search optimization," *IEEE Access*, vol. 9, pp. 123448–123464, 2021.

[5] S. Gavel, A. S. Raghuvanshi, and S. Tiwari, "Distributed intrusion detection scheme using dual-axis dimensionality reduction for Internet of things (IoT)," *The Journal of Supercomputing*, vol. 77, pp. 10488–10511, 2021.

[6] G. Kalnoor and S. Gowrishankar, "IoT-based smart environment using intelligent intrusion detection system," *Soft Computing*, vol. 25, no. 17, pp. 11573–11588, 2021.

[7] N. Islam, F. Farhin, I. Sultana, M. S. Kaiser, M. S. Rahman, M. Mahmud, A. S. Hosen, and G. H. Cho, "Towards machine learning based intrusion detection in IoT networks," *Comput. Mater. Contin.*, vol. 69, no. 2, pp. 1801–1821, 2021.

[8] P. Kumar, G. P. Gupta, and R. Tripathi, "Design of anomaly-based intrusion detection system using fog computing for IoT network," *Automatic Control and Computer Sciences*, vol. 55, no. 2, pp. 137–147, 2021.

[9] E. Gyamfi and A. D. Jurcut, "Novel online network intrusion detection system for industrial IoT based on OI-SVDD and AS-ELM," *IEEE Internet of Things Journal*, vol. 10, no. 5, pp. 3827–3839, 2022.

[10] T. Saba, A. Rehman, T. Sadad, H. Kolivand, and S. A. Bahaj, "Anomaly-based intrusion detection system for IoT networks through deep learning model," *Computers and Electrical Engineering*, vol. 99, 107810, 2022.

[11] V. Kumar, A. K. Das, and D. Sinha, "UIDS: A unified intrusion detection system for IoT environment," *Evolutionary Intelligence*, vol. 14, pp. 47–59, 2021.

[12] Y. Otoum, D. Liu, and A. Nayak, "DL-IDS: A deep learning–based intrusion detection framework for securing IoT," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 3, e3803, 2022.

[13] L. Nie, Y. Wu, X. Wang, L. Guo, G. Wang, X. Gao, and S. Li, "Intrusion detection for secure social internet of things based on collaborative edge computing: A generative adversarial network-based approach," *IEEE Transactions on Computational Social Systems*, vol. 9, no. 1, pp. 134–145, 2021.

[14] R. Krishnan, R. S. Krishnan, Y. H. Robinson, E. G. Julie, H. V. Long, A. Sangeetha, M. Subramanian, and R. Kumar, "An intrusion detection and prevention protocol for internet of things based wireless sensor networks," *Wireless Personal Communications*, vol. 124, no. 4, pp. 3461–3483, 2022.

[15] Z. K. Maseer, R. Yusof, and N. Bahaman *et al.* "Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset," *IEEE Access*, vol. 9, pp. 22351–22370, 2021.

[16] M. Zhang, J. Liu, Y. Wang, Y. Piao, S. Yao, W. Ji, J. Li, H. Lu, and Z. Luo, "Dynamic context-sensitive filtering network for video salient object detection," in *Proc. IEEE/CVF Computer Vision*, 2021, pp. 1553–1563.

[17] N. Alamir, S. Kamel, T. F. Megahed, M. Hori, and S. M. Abdelkader, "Developing hybrid demand response technique for energy management in microgrid based on pelican optimization algorithm," *Electric Power Systems Research*, vol. 214, 108905, 2023.

[18] H. Ge, L. Wang, M. Liu, X. Zhao, Y. Zhu, H. Pan, and Y. Liu, "Pyramidal multiscale convolutional network with polarized self-attention for pixel-wise hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–18, 2023.

[19] O. N. Oyelade, A. E. S. Ezugwu, T. I. Mohamed, and L. Abualigah, "Ebola optimization search algorithm: A new nature-inspired metaheuristic optimization algorithm," *IEEE Access*, vol. 10, pp. 16150–16177, 2022.

[20] H. A. Alsattar, A. A. Zaidan, and B. B. Zaidan, "Novel meta-heuristic bald eagle search optimisation algorithm," *Artificial Intelligence*, vol. 53, pp. 2237–2264, 2020.

[21] O. Elnakib, E. Shaaban, M. Mahmoud, and K. Emara, "EIDM: Deep learning model for IoT intrusion detection systems," *The Journal of Supercomputing*, pp. 1–21, 2023.

[22] C. U. Om Kumar, S. Marappan, B. Murugeshan, and P. M. R. Beaulah, "Intrusion detection model for iot using recurrent kernel convolutional neural network," *Wireless Personal Communications*, vol. 129, no. 2, pp. 783–812, 2023.

[23] S. Roy, J. Li, B. J. Choi, and Y. Bai, "A lightweight supervised intrusion detection mechanism for IoT networks," *Future Generation Computer Systems*, vol. 127, pp. 276–285, 2022.

[24] A. Thakkar and R. Lohiya, "Attack classification of imbalanced intrusion data for IoT network using ensemble learning-based deep neural network," *IEEE Internet of Things Journal*, vol. 10, no. 13, pp. 11888–11895, 2023.

[25] A. Basati and M. M. Faghih, "PDAE: Efficient network intrusion detection in IoT using parallel deep auto-encoders," *Information Sciences*, vol. 598, pp. 57–74, 2022.

[26] Z. Wu, H. Zhang, P. Wang, and Z. Sun, "RTIDS: A robust transformer-based approach for intrusion detection system," *IEEE Access*, vol. 10, pp. 64375–64387, 2022.

[27] A. Basati, and M. M. Faghih, "DFE: Efficient IoT network intrusion detection using deep feature extraction," *Neural Computing and Applications*, vol. 34, no. 18, pp.15175–15195, 2022.

[28] T. V. Ramana, M. Thirunavukkarasan, A. S. Mohammed, G. G. Devarajan, and S. M. Nagarajan, "Ambient intelligence approach: Internet of Things based decision performance analysis for intrusion detection," *Computer Communications*, vol. 195, pp.315–322, 2022.