INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN ENGINEERING AND TECHNOLOGY (IJARET)

ISSN Print: 0976-6480 ISSN Online: 0976-6499

https://iaeme.com/Home/journal/IJARET

High Quality Peer Reviewed Referred Scientific, Engineering & Technology, Medicine and Management International Journals

PUBLISHED BY

IAEME Publication

Plot: 03, Flat- S 1, Poomalai Santosh Pearls Apartment,
Plot No. 10, Vaiko Salai 6th Street, Jai Shankar Nagar, Palavakkam,
Chennai - 600 041, Tamilnadu, India

Email : editor@iaeme.com, iaemedu@gmail.com

www.iaeme.com

International Journal of Advanced Research in Engineering and Technology (IJARET)
Volume 16, Issue 4, July-August 2025, pp. 10-50, Article ID: IJARET_16_04_002
Available online at https://iaeme.com/Home/issue/IJARET?Volume=16&Issue=4
ISSN Print: 0976-6480 and ISSN Online: 0976-6499; Journal ID: 1793-4637
Impact Factor (2025): 28.80 (Based on Google Scholar Citation)
DOI: https://doi.org/10.34218/IJARET_16_04_002

AN INTELLIGENT INTRUSION DETECTION SYSTEM FOR CLOUD COMPUTING ENVIRONMENTS USING SYMMETRY-PRESERVING DUAL-STREAM GRAPH NEURAL NETWORKS

Vineshraj S[1], Dr P Ebby Darney[2]

[1]Research Scholar – LIPS Research - European International University Paris.
[2]Research Supervisor - LIPS Research Advanced R & D European international University Paris.
[2]Professor – Raja Rajeswari College of Engineering, Bangalore, Karnataka, India.

ABSTRACT

With the development of cloud computing, data transmission security continues to grow daily. The widespread use of cloud computing has made the necessity of strong data transmission security more apparent. Although these benefits, the risks related to cloud services have also increased as their usage has accelerated. Consequently, one of the most important defenses for identifying attacks in the cloud computing platform is an intrusion detection system (IDS). The huge volume of traffic present in the cloud environment presents certain difficulties for current IDSs to handle and analyze at the same time, which reduces the accuracy of detecting attacks. Therefore, a smart

Intrusion Detection System for Cloud Computing Environments Using Symmetry-Preserving Dual-Stream Graph Neural Networks (IDS-CC-SPDGNN) is developed in this study. Firstly, the input data are sourced from two well-established datasets such as UNSW-NB15 and CIC-IDS2017. To ensure data quality, a Non-uniform Weighted Guided Filtering (NUWGF) technique is employed for cleaning and normalization. Following pre-processing, the Hiking Optimization Algorithm (HOA) selects the most relevant features. The proposed Symmetry-Preserving Dual-Stream Graph Neural Networks (SPDGNN) model is then utilized to distinguish network traffic as benign or malicious behavior by exploiting graph-based structural representations. To further enhance detection accuracy, the Starfish Optimization Algorithm (SOA) is applied to optimize the weight parameters of the SPDGNN. The proposed IDS-CC-SPDGNN technique attains 7.12%, 3.84% and 5.69% higher accuracy, 4.89%, 6.45% and 3.92% higher precision when compared with existing techniques. Analyses have shown that the proposed approach has proven successful in securing cloud servers from a range of possible threats.

Keywords: cloud computing, Hiking Optimization Algorithm, Non-uniform Weighted Guided Filtering, Symmetry-Preserving Dual-Stream Graph Neural Networks, Starfish Optimization Algorithm

Cite this Article: Vineshraj S, P Ebby Darney. (2025). An Intelligent Intrusion Detection System for Cloud Computing Environments Using Symmetry-Preserving Dual-Stream Graph Neural Networks. *International Journal of Advanced Research in Engineering and Technology (IJARET)*, 16(4), 10-50.
DOI: https://doi.org/10.34218/IJARET_16_04_002

1. Introduction

One of the most recent developments in the domain of information technology (IT) is cloud computing. The primary advantage of cloud computing is that it enables access possible without regard to time or place restrictions [1]. Customers can access online storage and a range of computational capabilities through cloud computing. Cloud-based services are now essential for both people and enterprises [2]. It provides lower costs and flexible management of storage functions and support for mobile and collaborative apps and services [3]. Additionally, because cloud services are multisource, end customers can choose from a variety of service providers

according to their needs [4]. Utilizing cloud computing also lowers the need for service and physical facilities for on-site storage, as well as startup cost and power usage. As cloud-based computing solutions proliferate, many businesses, banks and governments have embraced this technology [5]. Strong security protocols are crucial as this change has made these systems susceptible to different forms of cyberattacks. A variety of security services are offered as apps by numerous cloud providers [6]. Consider the Amazon Web Services (AWS) marketplace, where services are available with different validity periods and expiration dates based on licensing duration [7]. The necessity for constant backup and updating is clear given the growing amounts of data, especially crucial medical information [8]. Because it is private, hackers find it easy to modify or utilize healthcare data for illegal reasons like political or financial gain. Individual patient histories, medication details, medical device data and other sensitive health information can all be found in health data and medical records [9]. The most important thing is that this information be kept private. Therefore, since they result in unauthorized access to the banking sector, social interactions, financial information and online attacks such as identity theft should be avoided [10]. Because customer and operating data are confidential, cybersecurity must be a top issue for healthcare providers when implementing cloud computing technology [11]. Current IDS use abnormality or signature identification as its technique. Operational and patient data may be taken if the detection method is not appropriate [12]. In addition to strengthening an organization's protection against hackers, a cybersecurity measures assist in discovering and prevention of malicious activities [13, 14]. Studies that have already been conducted in international hospitals have only focused on user knowledge of the significance of cybersecurity, such as implementing strong passwords, deleting or filtering spam emails, encrypting information, securely managing login details, accessing data carefully and quickly reporting security breaches. [15]. Health industry also has IDS in place; however there may be hazards in their local network. In addition, hospitals and other organizations can manage their activities with the help of the cloud, which offers a very helpful and secure service [16]. Eventually, all government IT operations will be performed on the cloud. At the moment, Bahrain's Department of Health is moving its systems to the cloud environment. The types of security offered by cloud providers vary based on their services [17, 18]. While it is possible to store some operational data in the cloud, certain highly confidential information must remain off-cloud due to security concerns [19]. Data kept on internal servers requires protection that matches the standards of cloud security. Ensuring robust security for

cloud resources demands considerable effort, which is essential for establishing trust when handling sensitive and private medical data in cloud environments [20].

Traditional IDS models fail to capture the intricate relationships and dependencies between different traffic flows. Many existing systems employ manual feature selection techniques that either rely heavily on domain expertise or use conventional statistical methods. This can result in the exclusion of subtle but important features. Unlike traditional GNNs, the proposed SPDGNN architecture introduces a dual-stream structure that separately processes both topological and feature-based information of the network traffic graph. The introduction of two nature-inspired optimization algorithms such as HOA for feature selection and the SOA for hyper parameter tuning improve both data and model optimization.

Major contribution of this manuscript is brief as below,

- A novel IDS tailored for cloud computing environments, using a SPDGNN to effectively classify network traffic based on graph-structured representations.
- NUWGF method is employed during the preprocessing stage to ensure high-quality input data. NUWGF dynamically adjusts filtering weights based on the local relevance of features.
- Deployment of the HOA to automatically select the most relevant features, enhancing model efficiency and reducing dimensionality. HOA effectively selects the most informative features by balancing exploration and exploitation.
- Development of a SPDGNN architecture that preserves input data symmetry and captures both local and global structural relationships within network traffic graphs.
- Integration of the SOA to fine-tune the SPDGNN's hyper parameters. This automated tuning improves detection accuracy and stabilizes training of proposed model.

The following sections outline the structure of the paper: Section 2 specifies a literature survey; Section 3 depicts a proposed approach; Section 4 presents results with discussion; and Section 5 explains the conclusion.

2. Literature review

A number of research papers given in the literatures were based on intrusion detection in cloud computing; some of these were discussed here.

In 2024, Jansi Sophia Mary, C. and Mahalakshmi, K., [21] presented modeling of malware detection utilizing sea horse optimization in a cloud platform. The seahorse

optimization with deep echo state network based attack detection approach was presented in the suggested paper for use in cloud environments. The purpose of the suggested technique was to use an intrusion detection procedure to achieve cloud security. This was accomplished by pre-processing the presented technology using a min–max normalization strategy. Furthermore, the deep echo state network model was used by the suggested technique to identify and categorize incursions into several types. It attains high precision and high computation time.

In 2024, Nagamani, S., et.al [22] presented horse herds optimization (HHO) in a cloud computing setting using detection of attacks based on deep learning (DL) methods. By utilizing these, the presented technique aims to secure the cloud service. The input data is mostly scaled using min-max scalar in the presented approach. The invasive weed optimization (IWO) method is used in the presented technique to choose the characteristics. The attention based bidirectional LSTM (A-BiLSTM) approach is then used to detect invasions. Finally, the HHO strategy has been implemented to enhance the hyperparameter tuning of the A-BiLSTM technique. It attains great accuracy and less recall.

In 2024, Long, Z., et.al [23] presented a transformer based threat detection method for safe cloud computing. The suggested research provided a novel IDS technique based on the Transformer model and carefully tuned for cloud settings. The method combined the fundamentals about network breach detection via the transformer model's advanced attention system, allowing for a more in-depth examination of the connections between the feature inputs and various intrusion categories, increasing the detection rate. It attains high recall and high false positive rate.

In 2024, Sharma, H.S. and Singh, K.J., [24] presented a deep feedforward neural network method enhanced with advanced features selection techniques for detecting intrusions in cloud environments. A filtering based selecting features approach was used in the research to present a feedforward deep neural network (FFDNN) approach based on DL technique. From the feature importance score, the method of attribute selection sought to identify and choose the most highly important subset of characteristics for the innovation of DL models. This method provides great specificity and attains less accuracy.

In 2024, Sarkar, N., et.al [25] presented cloud intrusion detection enhanced by an improved squirrel search algorithm (ISSA) and modification of deep belief network (MDBN) model. The presented study used the UNSW-NB15 dataset using ISSA and MDBN to present faster and better IDS for detecting anomalies in the cloud setting. To manage highly dimensional communication data, ISSA was utilized to extract the pertinent characteristics

from a set of features. It chose the best and most pertinent feature subset to be used in the testing and training procedures. In the meanwhile, an MDBN was presented, which discovered irregularities in classification tasks involving two or multiple classes using the features that were extracted. It attains low computation time and low recall.

In 2023, Samunnisa, K., et.al [26] presented hybrid clustering and detection approaches for IDS in distributed cloud environment. In order to classify malicious attack categories such as normal, DoS, Probe, U2R, and R2L using thresholds-based functions, the suggested article developed effective hybrid K means with random forest algorithms for constructing an anomaly-based IDS. The outcomes have been evaluated with two distinct threshold levels. It attains high accuracy and low specificity.

In 2023, Jain, D.K., et.al [27] established fuzzy deep neural network (FDNN) for cloud settings that combines the Honey Badger Algorithm (HBA) with FDNN to identify intrusions while protecting privacy. A blockchain-enabled secure privacy technique and an intrusion detection approach served as the foundation for the suggested system. To detect and classify intrusions, a successful training approach using the FDNN model was used. It attains low false positive rate and high computation time.

Table 1: Summary of recent intrusion detection techniques in cloud computing

Author(s)	**Objective**	**Methods used**	**Advantages**	**Disadvantages**
Jansi Sophia Mary, C. & Mahalakshmi, K. (2024)	To secure cloud environments using intrusion detection.	Seahorse Optimization, Deep Echo State Network	High precision	High computation time
Nagamani, S., et al. (2024)	Secure cloud services using deep learning-based attack detection.	HHO, IWO, A-BiLSTM, Min–Max Scaling	High accuracy	Low recall
Long, Z., et al. (2024)	Develop an IDS using Transformer model for cloud computing.	Transformer model with attention mechanism	High recall	High false positive rate
Sharma, H.S. & Singh, K.J. (2024)	Detect intrusions in cloud using optimized DNN model.	Feedforward Deep Neural Network, Filtering-based Feature Selection	High specificity	Low accuracy

Sarkar, N., et al. (2024)	Improve IDS in cloud with optimized feature selection and classification.	ISSA, MDBN	Low computation time	Low recall
Samunnisa, K., et al. (2023)	Classify attacks using hybrid clustering and classification in distributed clouds.	Hybrid K-means with Random Forest, Threshold-based classification	High accuracy	Low specificity
Jain, D.K., et al. (2023)	Identify intrusions while ensuring privacy in cloud environments.	FDNN, HBA, Blockchain privacy model	Low false positive rate	High computation time

Table 1 presents summary of recent intrusion detection techniques in cloud computing. Several studies, including those by Jansi Sophia Mary, C. & Mahalakshmi, K., and Jain, D.K., et al., show that the presented methods achieve high precision or low false positive rates but are burdened by high computation time, indicating a need for more efficient algorithms. Techniques like those presented by Nagamani, S., et al., and Sarkar, N., et al., demonstrated high accuracy but struggled with low recall. Samunnisa, K., et al.'s hybrid K-means and Random Forest method achieved high accuracy but had low specificity. To overcome these limitations, IDS-CC-SPDGNN is proposed using SPDGNN and SOA.

3. Proposed methodology

This section primarily presents the fundamentals, a thorough implementation of the proposed algorithm. The block diagram of proposed IDS-CC-SPDGNN model is presented in Figure 1.

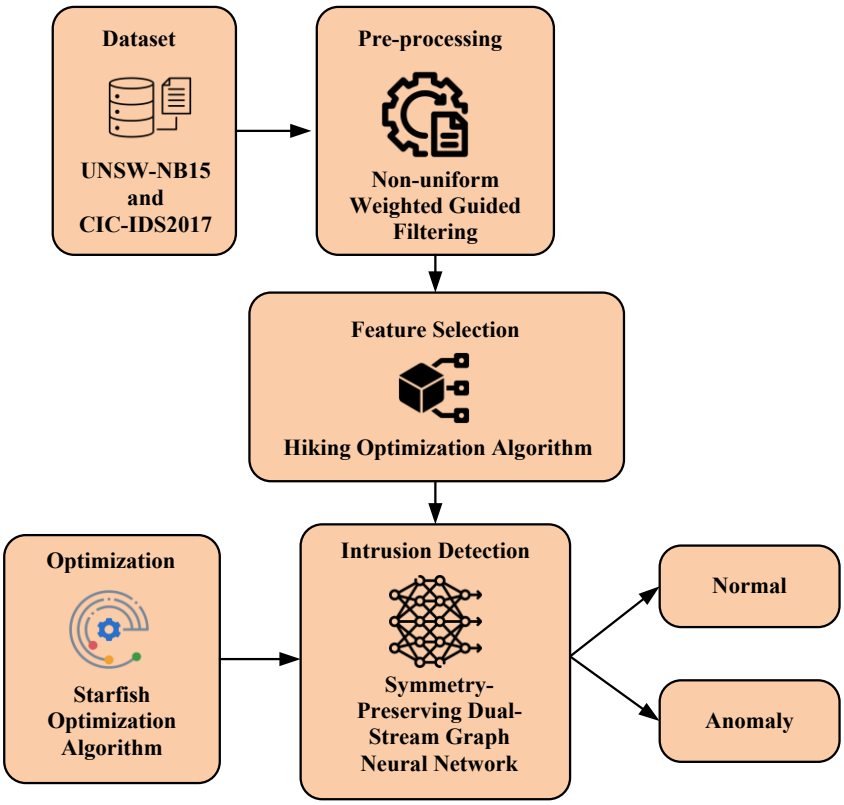

Figure 1: Block diagram of proposed IDS-CC-SPDGNN method

3.1 Dataset description

In this section, two datasets such as UNSW-NB15 and CIC-IDS2017 are discussed to used for train the proposed model.

3.1.1 UNSW-NB15

The UNSW-NB15 dataset [28] was developed by researchers at the Australian Center for Cyber Security (ACCS). It comprises 100 GB of raw network traffic data captured with the TCP-Dump tool, resulting in a total of 2,540,044 realistic traffic records. This dataset encompasses a diverse range of network protocols, including TCP, UDP, ICMP and HTTP. Additionally, it contains detailed metadata such as duration, destination addresses and timestamps. For training and testing purposes, the dataset provides 56,000 and 37,000 normal traffic samples, respectively. Table 2 presents UNSW-NB15 dataset features.

Table 2: Features of UNSW-NB15 dataset

Seq.No	Feature Label	Seq.No	Feature Label	Seq.No	Feature Label
1	Service	18	Djit	34	is_sm_ips_ports
2	Dstip	19	Dur	35	dsport
3	Sloss	20	ct_flw_http_mthd	36	ct_dst_src_ltm
4	Proto	21	Sttl	37	Stime
5	Ackdat	22	Srcip	38	attack_cat
6	Dbytes	23	Swin	39	Dwin
7	Spkts	24	Dmeansz	40	ct_dst_ltm
8	Sport	25	Dload	41	Sbytes
9	ct_ftp_cmd	26	ct_dst_sport_ltm	42	is_ftp_login
10	res_bdy_len	27	State	43	ct_state_ttl
11	Dintpkt	28	Synack	44	Dloss
12	Dttl	29	Sjit	45	Sintpkt
13	ct_src_ ltm	30	Dpkts	46	Smeansz
14	Dtcpb	31	ct_src_dport_ltm	47	trans_depth
15	ct_srv_src	32	Label	48	ct_srv_dst
16	Sload	33	Stcpb	49	Tcprtt
17	Ltime				

3.1.2 CIC-IDS2017

The CIC-IDS2017 dataset [29] has 2830745 cases and 80 features, divided into two classes: normal and anomalous. The normal class has 2273098 instances, while the abnormal class has 557647 instances. Table 3 shows the features of the CIC-IDS2017 dataset.

Table 3: CIC-IDS2017 feature Set

Seq.No	Feature Label	Seq.No	Feature Label	Seq.No	Feature Label
1	Active Std	28	Init_Win_bytes_fwd	55	Bwd_Packets/s
2	AvgFwd_SegmentSize	29	act_data_pkt_fwd	56	Average Packet Size
3	FwdIATTotal	30	SYNFlag_Count	57	FINFlag_Count
4	BwdIAT Std	31	FwdPacket Length_Min	58	BwdIATMean

5	Subflow Bwd Byte	32	PSH_FlagCount	59	MinPacket_Length
6	TotalFwdPackets	33	BwdAvg_Packets/Bulk	60	Flow IATMax
7	Source Port	34	TotalLength_FwdPackets	61	Active Mean
8	FlowIAT_Mean	35	Total_BackwardPackets	62	Fwd URGFlags
9	Fwd IATMean	36	Subflow Bwd Packets	63	Bwd IAT Total
10	Active Max	37	Max Packet Length	64	Active Min
11	BwdPacket Length_Min	38	FwdPacket Length_Mean	65	BwdPacket Length_Mean
12	Idle Mean	39	Total_Length_BwdPackets	66	FwdIATMin
13	FwdAvg_Bytes/Bulk	40	Subflow Fwd Bytes	67	Flow Duration
14	min_segsize_fwd	41	Packet Length_Mean	68	Idle Max
15	BwdIATMax	42	FwdPacket Length_Std	69	Flow Packets/s
16	Flow Bytes/s	43	Bwd Header Length	70	Packet Len. Std
17	CWE Flag Count	44	BwdURG Flags	71	Flow _IAT_Min
18	BwdAvg Bytes/Bulk	45	Idle Std	72	Fwd_IAT_Max
19	Fwd CWR Flags	46	Down/Up Ratio	73	Fwd ECE Flags
20	ACK Flag Count	47	Fwd Avg Packets/Bulk	74	Protocol
21	Init_Win_bytes_bwd	48	Fwd PSH Flags	75	Destination Port
22	RST Flag Count	49	BwdPacket Length_Std	76	URG Flag Count
23	Bwd IAT Min	50	BwdPacket Length_Max	77	Idle Min
24	FwdHeader Length	51	Packet_Length_Variance	78	Fwd IAT Std
25	BwdAvg_BulkRate	52	AvgBwd_SegmentSize	79	FlowIATStd
26	ECE Flag Count	53	FwdPacket Length_Max	80	Fwd Avg Bulk Rate
27	Label	54	Subflow Fwd Packets		

3.2 Pre-processing using Non-uniform Weighted Guided Filtering

In this section, pre-processing using NUWGF [30] for data cleaning and normalization is discussed. Data cleaning involves identifying and fixing errors or inconsistencies in a dataset to enhance its overall quality. Data normalization refers to rescaling numerical values to standardized range. NUWGF adjusts its behavior based on local characteristics of the data. It uses local linear models that adapt to variation in each neighborhood, making it suitable for heterogeneous datasets. NUWGF ensures that more relevant or trustworthy data points have

stronger influence by using non-uniform weights. Filtered output for data point is calculated using equation (1)

$$P_u = \bar{x}_u U_u + \bar{y}_u, \quad \forall u \in d_l, \tag{1}$$

Where, P_u indicates the filtered output for data point u, $\bar{x}_u$ and $\bar{y}_u$ is a locally averaged coefficients for feature transformation, U_u represents input data value and d_l is a neighborhood or similarity-based set of data points around point l. These coefficients reflect the local behavior of the data and allow for data cleaning. The neighborhood is not spatial but determined by data similarity. The learning step in the filtering process is given in equation (2).

$$F(x_l, y_l) = \sum_{u \in d_l} \left[(x_l U_u + y_l - Q_u)^2 + \frac{\delta}{\mathrm{K}_u} x_l^2 \right] \tag{2}$$

Where, $F(x_l, y_l)$ represents the objective function for finding best coefficients for a local model around point l, Q_u is a target output value for point u, δ represents regularization constant and K_u indicates normalization factor. It ensures the proposed model respects local structure and is regularized to avoid fitting to noise. The normalization factor is given in equation (3).

$$\mathrm{K}_u = \frac{1}{N} \sum_{u=1}^{N} \frac{\alpha(u')^2 + \gamma}{\alpha(u)^2 + \gamma} \tag{3}$$

Where, $\alpha(u)$ indicates the weight of data point u, γ represents the small stability constant, N is the maximum number of neighboring data and u' indicates the mean data point. The value of K_u influences how much regularization is applied during model fitting. The consensus model coefficients $\bar{x}_u, \bar{y}_u$ for each data point are calculated by equation (4).

$$\{\bar{x}_u, \bar{y}_u\} = \frac{1}{D_F} \sum_{u \in d_l} D_F \{x_u, y_u\} \tag{4}$$

Where, x_u, y_u are the coefficients for neighbor data point u and D_F is the normalization weight for each data point. Equation (4) ensures consistency and reduces sensitivity to outlier neighbors in the final filtering output. The normalized influence weight for each pair of data point u and l within a neighborhood is calculated by equation (5).

$$D_F = \frac{D_F(a_u, a_v) L(a_u, a_v)}{\sum_{v \in d_l} D_F(a_u, a_v) L(a_u, a_v)} \tag{5}$$

Where, a_u, a_v are the features vectors of data points u and v, L represents the kernel function. The normalization ensures that all weights across the neighborhood sum to 1. It ensures that more similar data points have higher influence in equation (4). Finally using NUWGF, the data is cleaned and the structured data is normalized. Then the output of the pre-processing stage is fed to next stage.

3.3 Feature selection using Hiking Optimization Algorithm

Here, feature selection process using HOA is discussed [31] to select the most important features from input dataset for intrusion detection. HOA makes sure that only the most essential features are kept for the classification phase. The popular recreational pastime of hiking served as the inspiration for HOA, which compares the feature selection landscape to the rugged terrain hikers traverse. Finding the most pertinent elements in this situation is similar to navigating tough terrain where choices must be made based on different degrees of difficulty. The mathematical basis of HOA rests on Tobler's Hiking Function (THF), which in this adaption calculates the feature agents' selection velocity by taking into account the feature's relevance gradient and the distance (variation) between feature subsets. The initializing of feature agents is an essential step in metaheuristic-based feature selection, particularly when employing HOA, since it influences convergence speed and the quality of the solution. Prior information or randomization can be used to initialize the position. The upper limit η_v^2 and the lower limit η_v^1 determine the initialization process of features positions $\delta_{u,s}$, using equation (6).

$$\delta_{u,s} = \eta_v^1 + \beta_v\left(\eta_v^2 - \eta_v^1\right) \tag{6}$$

Where, β_v is a random variable uniformly distributed over [0,1]. It is the key to balancing exploration and exploitation in the feature selection process. Next, each solution's fitness score and probabilities are calculated by equation (7).

$$Fitness\,Function = [Select\ optimal\ \ features] \tag{7}$$

In HOA, THF is a mathematical framework designed to inform the choosing of the most relevant and non-redundant subset of features from a high-dimensional dataset. Then, THF is given in equation (8).

$$\lambda_{u,s} = 6e^{-2.5|T_{u,s}+0.05|} \tag{8}$$

Where, $\lambda_{u,s}$ represents the selection velocity at iteration, and $T_{u,s}$ is the relevance slope of the feature at iteration s. THF governs the selection velocity of each feature based on its relevance and relative position in the feature space. Then, the slope $T_{u,s}$ calculated using equation (9)

$$T_{u,s} = \frac{dg}{da} = \tan\theta_{u,s} \tag{9}$$

Where, dg and da represents the change in relevance score and feature index covered by the selector, respectively. The advantages of the social behavior of feature selectors and the individual selection power of particular traits are capitalized upon by the heuristic optimization algorithm (HOA). A feature's beginning relevance, the position of the most relevant feature, and its own current position is adjusted by a sweep factor. As a result, the feature's selection velocity is determined by equation (10).

$$\lambda_{u,s} = \lambda_{u,s-1} + \upsilon_{u,s}\left(\delta_{lead} - \phi_{u,s}\delta_{u,s}\right) \tag{10}$$

Where, $\upsilon_{u,s}$ is a random value distributed within the range [0,1], δ_{lead} represent the lead feature velocities, and $\phi_{u,s}$ is the sweep factor (SF) for the feature, adjusting how aggressively it follows the lead feature. A high SF allows more deviation, enabling exploration; a lower SF ensures close following of the best features. By combining the above with Equation (8), the updated position of the feature is measured by equation (11)

$$\delta_{u,s+1} = \delta_{u,s} + \lambda_{u,s} \tag{11}$$

When the optimum number of repetitions has been reached and the desired better result is achieved, the HOA method is completed. The features selected by HOA from UNSW-NB15 and CIC-IDS2017 datasets are tabulated in table 4 and 5.

Table 4: Selected features from UNSW-NB15

Seq.No	Feature Label	Seq.No	Feature Label	Seq.No	Feature Label
1	Service	7	ct_dst_ltm	13	ackdat
2	ct_srv_src	8	ct_state_ttl	14	ct_ftp_cmd
3	Sintpkt	9	attack_cat	15	Label
4	Dintpkt	10	Sjit	16	dwin
5	Tcprtt	11	Stcpb	17	Djit
6	Stime	12	ct_src_ ltm	18	Sload

Table 5: Selected features from CIC-IDS2017

Seq.No	Feature Label	Seq.No	Feature Label	Seq.No	Feature Label
1	BwdIATMean	11	Subflow Bwd Packets	21	CWE Flag Count
2	FlowIAT_Mean	12	Fwd CWR Flags	22	FwdIATTotal
3	RST Flag Count	13	Label	23	SYNFlag_Count
4	Active Mean	14	TotalFwdPackets	24	Fwd ECE Flags
5	Down/Up Ratio	15	Init_Win_bytes_bwd	25	Active Max
6	Fwd PSH Flags	16	TotalLength_FwdPackets	26	Packet Len. Std
7	Protocol	17	Flow Duration	27	Idle Std
8	PSH_FlagCount	18	Fwd Avg Bulk Rate	28	Flow IATMax

9	Idle Mean	19	BwdAvg Bytes/Bulk	29	BwdIAT Std
10	Flow _IAT_Min	20	Source Port	30	Fwd IATMean

3.4 Classification using Symmetry-Preserving Dual-Stream Graph Neural Network

In this section, SPDGNN [32] is used to differentiate normal from malicious behaviors. The SPDGNN offers significant advantages for cloud intrusion detection by effectively modeling the elaborate and adaptable nature of cloud environments. By preserving the symmetry inherent in network communication patterns, SPDGNN improves the proposed method's capacity to detect subtle deviations demonstrative of malicious behavior. The dual-stream architecture processes both structural and semantic graph information independently, allowing the model to capture topological relationships between nodes as well as contextual information like service types or access logs. Figure 2 shows architecture diagram of SPDGNN.

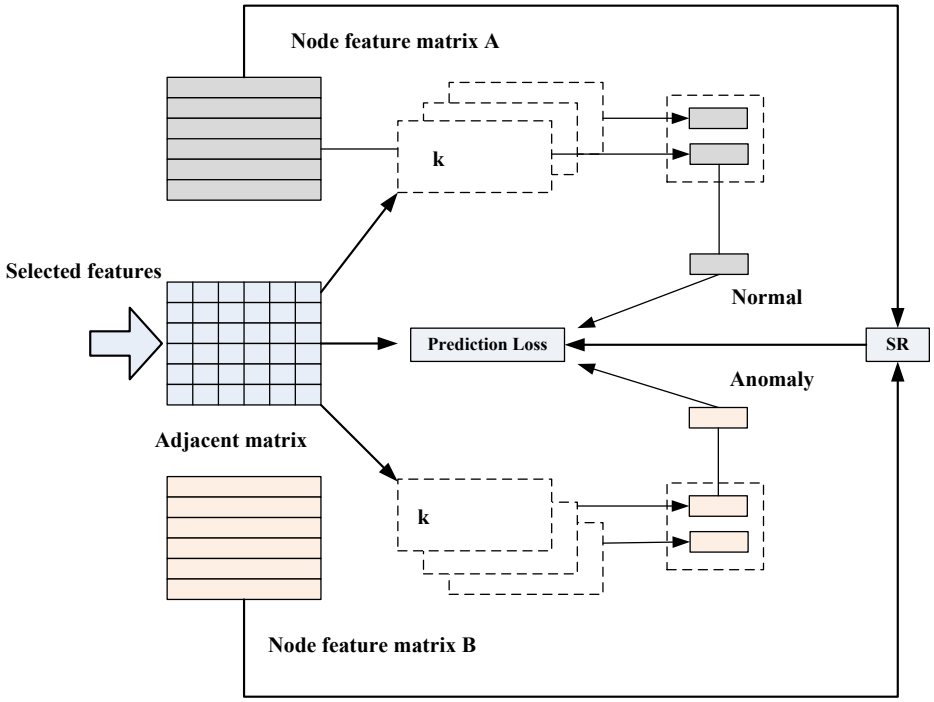

Figure 2: Architecture diagram of SPDGNN

The node embedding propagation is given in equation (12).

$$q_u^{(k)} = \sum_{v \in M(u)} \frac{1}{|M(u)|} q_v^{(k-1)}, \quad p_d^{(k)} = \sum_{c \in M(d)} \frac{1}{|M(d)|} q_c^{(k-1)} \tag{12}$$

Where, $q_u^{(k)}$ represents embedding of node u at layer k, $p_d^{(k)}$ is an embedding of node d at layer k, $M(u)$ indicates neighborhood of node u, $|M(u)|$ defines size of neighborhood $M(u)$, v,c are the neighboring nodes of u and d, respectively $q_v^{(k-1)}$ and $q_c^{(k-1)}$ represents embeddings of neighbors from the previous layer. This is the core propagation step of a Graph Neural Network (GNN). At each layer k, the representation of a node is updated by averaging the features of its neighbors. Both local and global structural patterns are combined using equation (13).

$$q_u = \sum_{k=0}^{K} \mu_k q_u^{(k)}, \quad p_d = \sum_{k=0}^{K} \mu_k p_d^{(k)} \tag{13}$$

Where, q_u, p_d are the final embeddings of node u and d, respectively, μ_k represents weight coefficient for layer k and K denotes the number of layers in the GNN. It aggregates embeddings from multiple GNN layers using weights μ_k, forming the final embedding for each node. The pairwise similarity prediction is calculated by equation (14).

$$\hat{x}_{u,d} = \langle q_u, p_d \rangle \tag{14}$$

Where, $\hat{x}_{u,d}$ indicates similarity score between node u and d. This is the actual intrusion classification output step. It computes a similarity score between node u and d, which interpreted as the predicted label. The regularization for initial embeddings is measured by equation (15).

$$\left\| q_u^{(0)} - p_d^{(0)} \right\|^2 = \| a_u - b_u \|^2 \to 0 \tag{15}$$

Where, $q_u^{(0)}, p_d^{(0)}$ are the initial embeddings of nodes u,d and a_u, b_u are the feature representations. It helps guide learning by enforcing symmetry and acts as a stabilizer in training. A loss function evaluates performance and enforces regularization, which is given in equation (16).

$$\vartheta = \sum_{x_{u,d} \in \omega} \left(x_{u,d} - \hat{x}_{u,d} \right)^2 + \frac{\sigma}{2} \left\| A - B \right\|^2 \tag{16}$$

Where, ϑ indicates overall loss function, $x_{u,d}$ indicates ground truth label for pair (u,d), $\hat{x}_{u,d}$ is a predicted score for (u,d), ω indicates known weights, σ is a regularization coefficient and A, B represents possibly matrices of initial node embeddings. The first term is the prediction loss between ground truth and predicted score. The second term is a regularization term ensuring smoothness. Finally, the input data is classified as attacks and normal using SPDGNN. To improve the threat classification accuracy, the bias factor σ from SPDGNN is optimized using SCA.

3.5 Optimization using Starfish Optimization Algorithm

Here, SOA [33] is utilized to optimize the bias factor σ from SPDGNN classifier. Once SOA determines the best SPDGNN configuration, the malicious behaviors are identified with high accuracy. SOA simulates starfish behaviors such as discovery, feeding, and renewal. It has two phases: exploration and exploitation. During the exploration phase, a hybrid scheme merges five dimensional (5D) and one dimensional (1D) search patterns, boosting computational efficiency and search power. In the exploitation phase, the algorithm mimics the starfish's feeding and regeneration behaviors.

Step 1: Initialization

The bias parameter values are randomly initialized within the predetermined constraints during the SOA initialization stage, creating a matrix that shows the starting locations of potential solutions. It is calculated by equation (17).

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1C} \\ A_{21} & A_{122} & \cdots & A_{2C} \\ \vdots & \vdots & \ddots & \vdots \\ A_{M1} & A_{M2} & \cdots & A_{MC} \end{bmatrix}_{M \times C} \tag{17}$$

Where A is the matrix representing the initialized bias parameter values for each candidate solution, with dimensions $M \times C$, where M indicates the size of population and C represents the number of bias parameters. During the initialization phase, each candidate bias configuration (i.e., each "starfish") in A is evaluated using Equation (18),

$$A_{u,v} = k_v + rand\left(j_u - k_v\right),\ u = 1,2,...M,\ v = 1,2,...C \tag{18}$$

Where $A_{u,v}$ represents the value of the v^{th} bias parameter in the u^{th} candidate solution (starfish), $rand$ indicates a random value in the range (0, 1), and j_u and k_v denote the maximum and minimum limits of the v^{th} bias parameter, respectively.

Step 2: Fitness Function Calculation

In order to optimize weight parameter σ of SPDGNN, the optimization parameter value is employed to compute the fitness function. Fitness function is calculated using equation (19),

$$Fitness\ Function = Optimize(\sigma) \tag{19}$$

Step 3: Exploration phase

A unique search strategy that combines the 5D searching pattern for $C > 5$ and the one-dimensional search pattern for $C \le 5$ in various optimization challenges is put forth during the SOA investigation phase. The five eyes of starfish are used to determine the threshold of dimensions. The search space for an optimization problem is large if its dimension is more than 5 $(C > 5)$, necessitating that starfish move all five limbs in order to investigate their surroundings. Furthermore, in order to direct their movement, starfish arms need to know where the search agents are most effective. The exploration phase is given in equation (20).

$$\begin{cases} B_{u,q}^S = A_{u,q}^S + x_1\left(A_{best,q}^S - A_{u,q}^S\right)\cos\theta, & z \le 0.5 \\ B_{u,q}^S = A_{u,q}^S - x_1\left(A_{best,q}^S - A_{u,q}^S\right)\sin\theta, & z > 0.5 \end{cases} \tag{20}$$

Where $B_{u,q}^S$ and $A_{u,q}^S$ represent the updated and current values of the selected bias parameters in the u^{th} candidate solution, respectively and $A_{best,q}^S$ indicates the values of the most performing solution in the q selected dimensions. In order to guarantee search capacity and increase the effectiveness of searches, Equation (20) use 5D search pattern regarding $C > 5$ in an optimization challenges for updating only 5D of the locations. The eyes then prefer to remain in the original position instead of relocating when the revised position is outside the bounds of the parameters, which is expressed in equation (21).

$$A_{u,q}^{S+1} = \begin{cases} B_{u,q}^{S} & k_{v,q} \le B_{u,q}^{S} \le i_{v,q} \\ A_{u,q}^{S} & otherwise \end{cases} \tag{21}$$

Where, q refers to five randomly chosen parameter dimensions. Each update respects the predefined parameter bounds, with $k_{v,q}$ and $i_{v,q}$ representing the lower and upper limits for the q^{th} parameter.

Step 4: Exploitation phase

Two update techniques are created in the exploitation phase of SOA, which takes into account the preying and regeneration behaviors in order to find global solutions. A distance can be computed as equation (22).

$$c_n = \left(A_{best}^{S} - A_{n}^{S}\right), \quad n = 1,....,5 \tag{22}$$

Here, c_n indicates the distances among the global the most effective and five selected at random starfish A_n^S. The predatory behavior is guided by these distances, and each starfish modifies its posture accordingly, which is given in equation (23).

$$B_u^S = A_u^S + rand_1 c_{n1} + rand_2 c_{m2} \tag{23}$$

Where, c_{n1} and c_{n2} are randomly chosen inside c_n, and $rand_1$ and $rand_2$ are random numbers between (0,1). If the value computed from Equation (23) falls outside the allowable bounds of the design variables, the position is adjusted to remain within those bounds. The new position is calculated by equation (24).

$$A_u^{S+1} = \begin{cases} B_u^S & k_v \le B_u^S \le j_v \\ k_v & B_u^S < k_u \\ j_v & B_u^S > j_u \end{cases} \tag{24}$$

Step 5: Termination

The weight parameter value σ from SPDGNN has been optimized using SOA, and step 2 will be reiterated continuously up to the stopping criteria $A = A + 1$ are fulfilled. Pseudo code of SOA is given in Algorithm 1.

Algorithm 1: Pseudo code of SOA

```
Input: M (number of starfish), Smax,
Output: Best solution found
1. Initialize parameters and problem boundaries
2. Generate initial positions for all starfish
3. Evaluate fitness of each starfish
4. Set iteration counter S = 0
5. While S < Smax do
6.     For each starfish:
7.         If random() > C
8.             Compute angle and energy
9.             If dimension > 5:
10.                - Select 5 random dimensions
11.                - Update each selected dimension
12.            Else:
13.                - Select 1 random dimension
14.                - Update it using random values
15.        Check and fix boundaries
16.        Else:
17.            - Update position based on best starfish
18.            - If last starfish, regenerate its position
19.        - Check and fix boundaries
20.    End For
21. Evaluate fitness of all starfish
22. Update global best solution
23. A=A+1
24. End While
25. Return global best solution
```

4. Results and discussion

Here, a thorough analysis of the proposed approach's final result with existing techniques was carried out utilizing a number of performance indicators. Primarily the experimental setup is presented, along with the simulation parameters and experimental environment configuration; next, a comparative analysis of various outcomes of the experiment confirms the effectiveness of the proposed method.

4.1. Experimental settings

Python was the programming language employed for the experiment and every component of the model's design had been finished using the Pytorch deep learning development platform. Using the Python 3.10 tool on a PC with an AMD Ryzen 9 7900X processor, 1 TB NVMe SSD, GeForce RTX 4080 16 GB, 64 GB RAM, and 4 TB HDD, the proposed method is simulated. Table 6 displays the experiment's parameter setting.

Table 6: Simulation parameters

Parameters	**Values**	**Parameters**	**Values**
Number of Layers	2	Epochs	200
Dropout value	0.5	Batch Size	64
Population size	100	Learning rate	0.001
Hidden units	64	Maximum Iterations	2000
Activation Function	ReLU	Optimizer	SOA

4.2. Performance measures

Performance metrics include precision, recall, accuracy, specificity, F1-score, False Positive Rate (FPR) and computation time are used to evaluate the effectiveness of proposed model.

4.2.1 Accuracy

Accuracy indicates the proportion of correctly identified intrusions and normal behaviors in a cloud environment. It is quantified by the equation (25)

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \tag{25}$$

Where, TP indicates true positive, FP represents false positive, TN is a true negative, and FN is a false negative.

4.2.2 Precision

Precision measures how many of the cloud intrusions flagged by the system are actually true threats. It is measured by using equation (26)

$$\Pr ecision = \frac{TP}{(TP + FP)} \tag{26}$$

4.2.3 Recall

Recall reflects the method's capacity to detect actual intrusion attempts occurring in the cloud environment. Recall is measured using equation (27)

$$Recall = \frac{TP}{(TP + FN)} \tag{27}$$

4.2.4 Specificity

Specificity assesses how effectively the system identifies legitimate, non-malicious cloud activities. It is measured by using equation (28).

$$Specificity = \frac{TN}{TN + FP} \tag{28}$$

4.2.5 F1 Score

The F1 Score provides a balanced assessment of the cloud IDS's precision and recall. It is determined using equation (29).

$$F1score = 2 \times \left(\frac{Precision \times Recall}{Precision + Recall} \right) \tag{29}$$

4.2.6 False Positive Rate

FPR represents the proportion of normal cloud activities that are incorrectly classified as intrusions. It is measured by equation (30)

$$FPR = \frac{FP}{FP + TN} \tag{30}$$

4.3 Performance Comparision using UNSW-NB15 dataset

Figure 3 to 9 shows the simulation outcomes of proposed IDS-CC-SPDGNN technique using UNSW-NB15 dataset. The proposed IDS-CC-SPDGNN techniques are compared with existing IDS-SHO-ML-CE [21], HHO-DL-IDS-CCE [22] and TRAN-NIDS-CS [23] techniques.

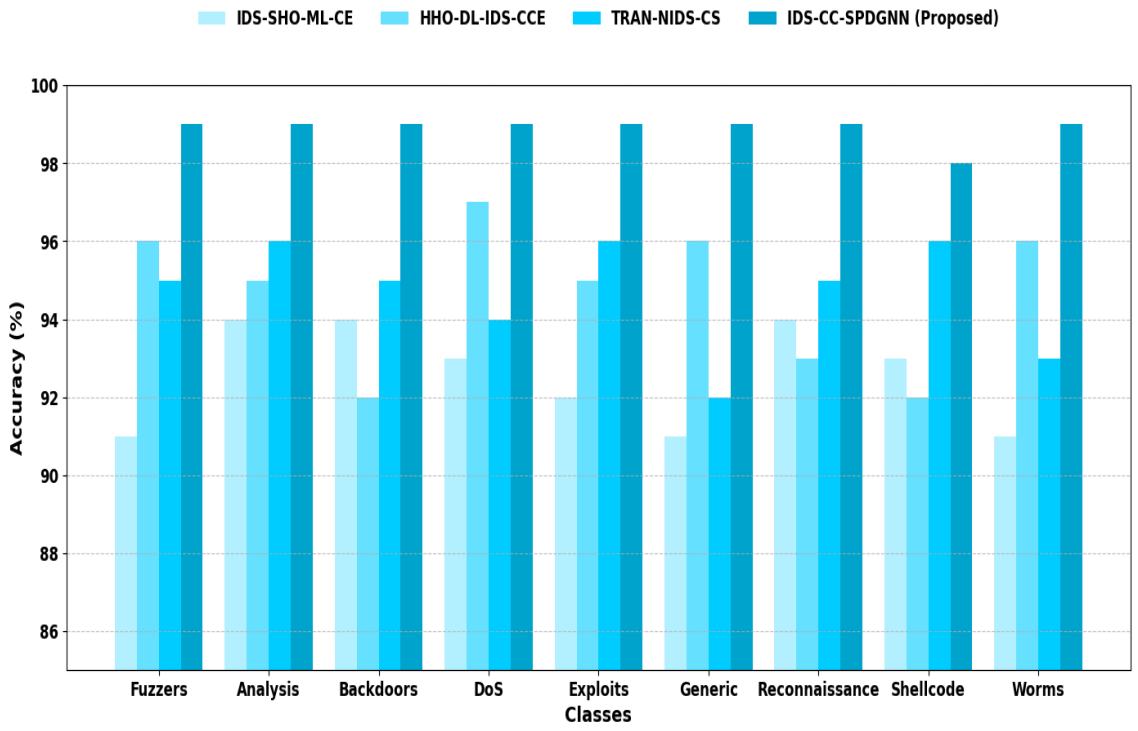

Figure 3: Accuracy analysis on the dataset of UNSW-NB15

Figure 3 presents accuracy analysis on the dataset of UNSW-NB15. The IDS-CC-SPDGNN technique attains 7.12%, 3.84% and 5.69% higher accuracy for fuzzers; 4.51%, 6.37% and 3.29% higher accuracy for analysis; 7.94%, 4.08% and 6.51% higher accuracy for Backdoors; 5.17%, 7.86% and 4.43% higher accuracy for DoS; 3.72%, 6.94% and 5.13% higher accuracy for exploits; 6.18%, 4.96% and 7.41% higher accuracy for generic; 5.74%, 3.67% and 6.28% higher accuracy for reconnaissance; 4.65%, 7.33% and 3.96% higher accuracy for shellcode; 6.73%, 5.38% and 4.26% higher accuracy for worms when compared with existing techniques like IDS-SHO-ML-CE, HHO-DL-IDS-CCE and TRAN-NIDS-CS respectively.

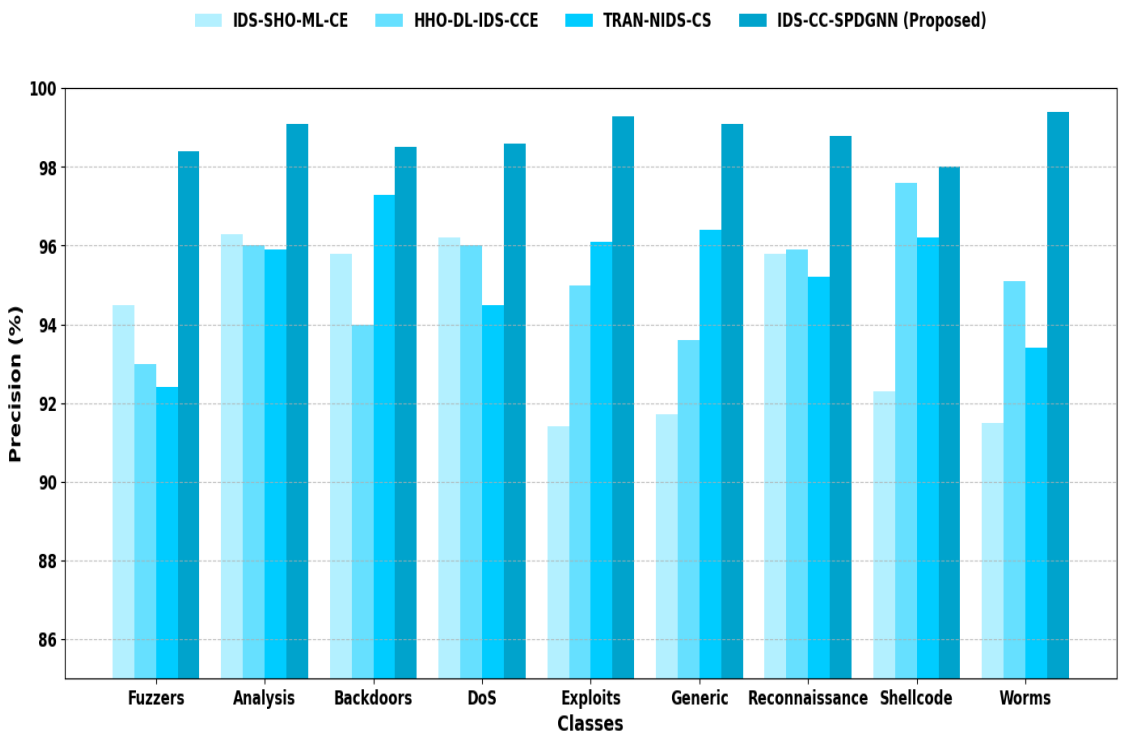

Figure 4: Precision analysis on the dataset of UNSW-NB15

Figure 4 presents precision analysis on the dataset of UNSW-NB15. The proposed IDS-CC-SPDGNN technique attains 4.89%, 6.45% and 3.92% higher precision for fuzzers; 7.28%, 4.36% and 6.13% higher precision for analysis; 3.57%, 7.91% and 5.64% higher precision for Backdoors; 6.02%, 3.75% and 7.49% higher precision for DoS; 5.86%, 4.08% and 6.97% higher precision for exploits; 3.41%, 7.63% and 4.78% higher precision for generic; 6.84%, 5.19% and 3.68% higher precision for reconnaissance; 4.22%, 6.79% and 5.91% higher precision for shellcode; 7.34%, 3.85% and 6.51% higher precision for worms when compared with existing techniques like IDS-SHO-ML-CE, HHO-DL-IDS-CCE and TRAN-NIDS-CS respectively.

Figure 5 presents recall analysis on the dataset of UNSW-NB15. The IDS-CC-SPDGNN technique attains 6.81%, 3.77% and 5.94% higher recall for fuzzers; 4.28%, 7.36%, and 6.02% higher recall for analysis; 3.91%, 6.58% and 7.13% higher recall for Backdoors; 5.22%, 7.85%, and 4.39% higher recall for DoS; 6.74%, 4.87% and 3.66% higher recall for exploits; 7.21%, 5.46% and 3.58% higher recall for generic; 4.93%, 6.67% and 7.48% higher recall for reconnaissance; 3.45%, 5.71% and 6.34% higher recall for shellcode; 7.69%, 4.13% and 5.36% higher recall for worms when compared with existing techniques like IDS-SHO-ML-CE, HHO-DL-IDS-CCE, and TRAN-NIDS-CS respectively.

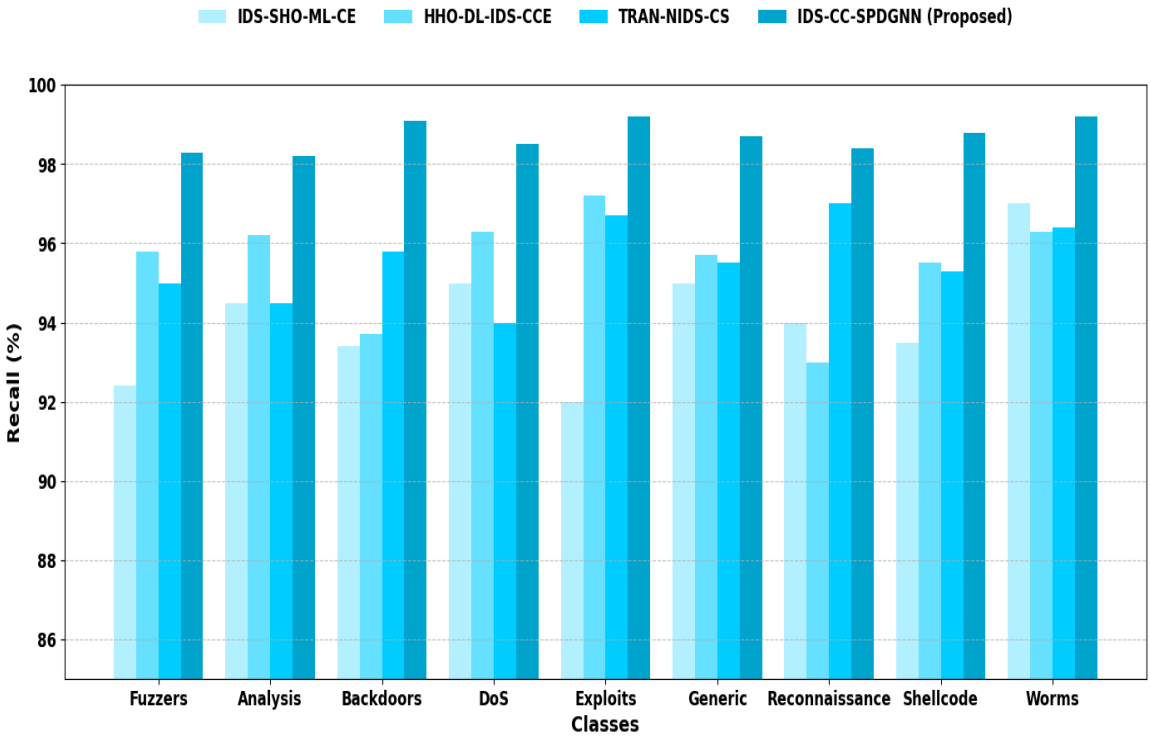

Figure 5: Recall analysis on the dataset of UNSW-NB15

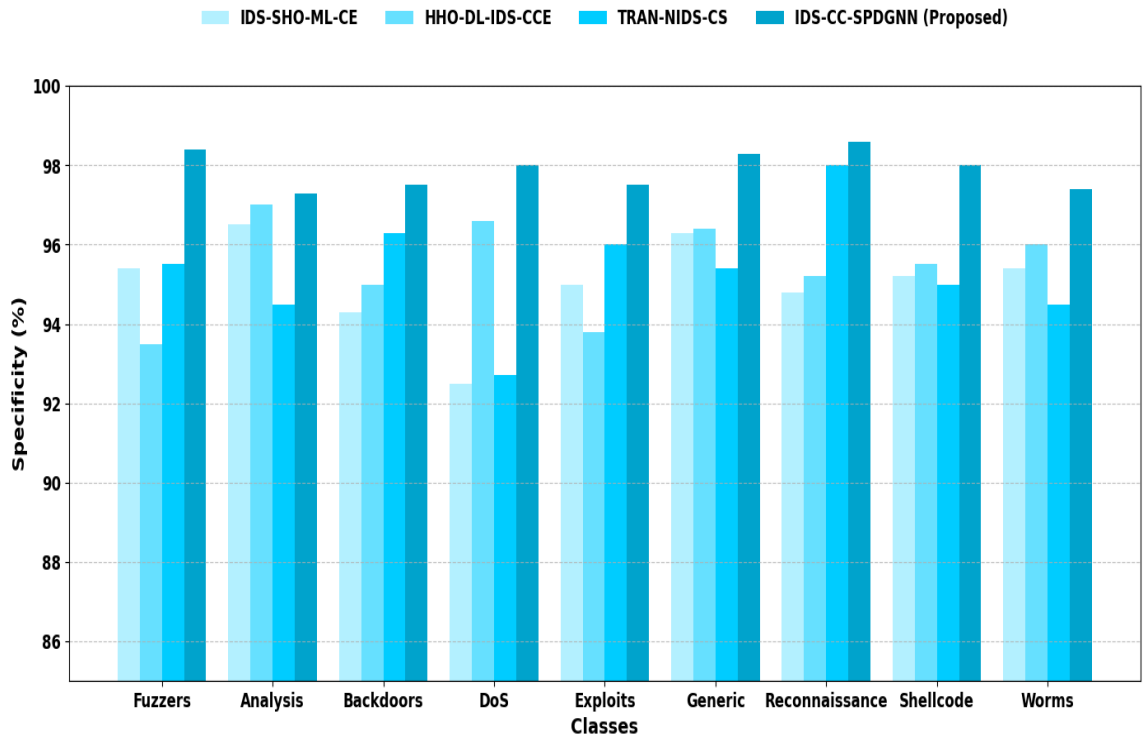

Figure 6: Specificity analysis on the dataset of UNSW-NB15

Figure 6 show specificity comparison on the dataset of UNSW-NB15. The IDS-CC-SPDGNN technique attains 5.73%, 7.42% and 3.86% higher specificity for fuzzers; 6.25%, 4.91% and 7.58% higher specificity for analysis; 3.77%, 6.83% and 5.12% higher specificity for Backdoors; 4.68%, 7.95% and 6.04% higher specificity for DoS; 5.39%, 3.62% and 7.71% higher specificity for exploits; 6.98%, 4.37% and 5.84% higher specificity for generic; 3.54%, 6.46% and 7.36% higher specificity for reconnaissance; 5.91%, 7.19% and 4.28% higher specificity for shellcode; 6.13%, 3.41% and 7.87% higher specificity for worms when compared with existing techniques like IDS-SHO-ML-CE, HHO-DL-IDS-CCE and TRAN-NIDS-CS respectively.

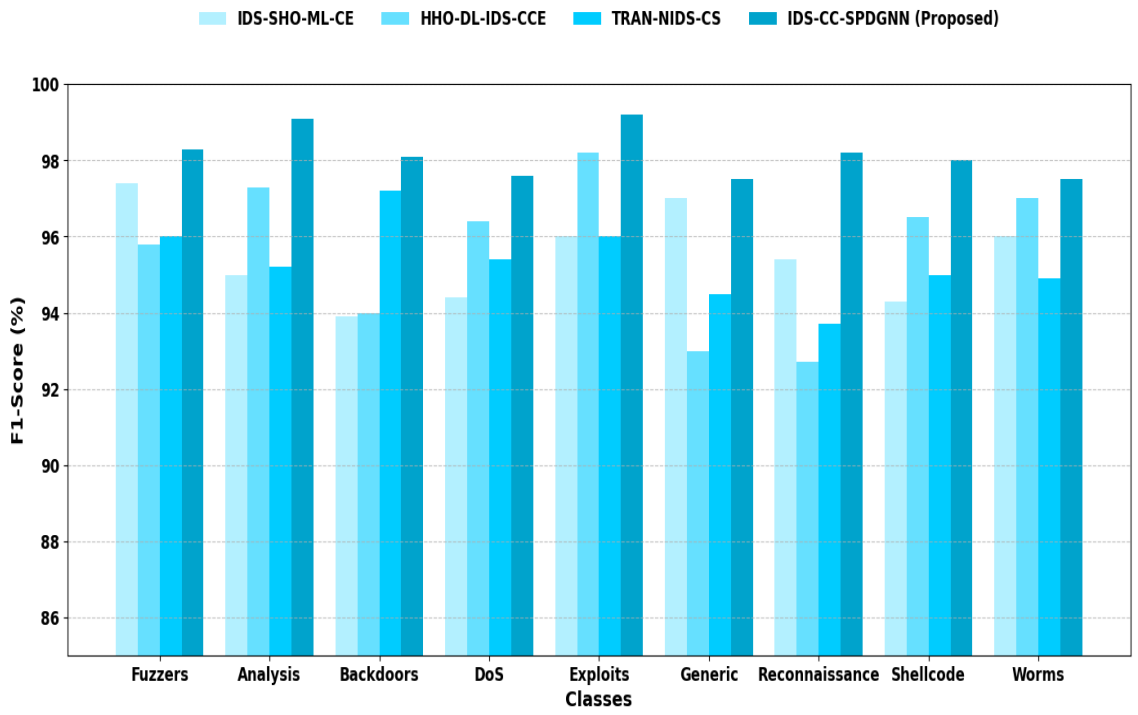

Figure 7: F1-score analysis on the dataset of UNSW-NB15

Figure 7 presents F1-score analysis on the dataset of UNSW-NB15. The IDS-CC-SPDGNN technique attains 6.17%, 3.49% and 7.23% higher F1-score for fuzzers; 5.78%, 6.91%, and 4.15% higher F1-score for analysis; 7.63%, 5.24% and 3.68% higher F1-score for Backdoors; 6.45%, 4.92% and 7.86% higher F1-score for DoS; 3.84%, 6.32% and 5.97% higher F1-score for exploits; 7.38%, 4.56% and 6.09% higher F1-score for generic; 5.11%, 3.73% and 6.87% higher F1-score for reconnaissance; 4.79%, 7.52% and 5.36% higher F1-score for shellcode; 6.73%, 3.94% and 7.12% higher F1-score for worms when compared with

existing techniques like IDS-SHO-ML-CE, HHO-DL-IDS-CCE and TRAN-NIDS-CS respectively.

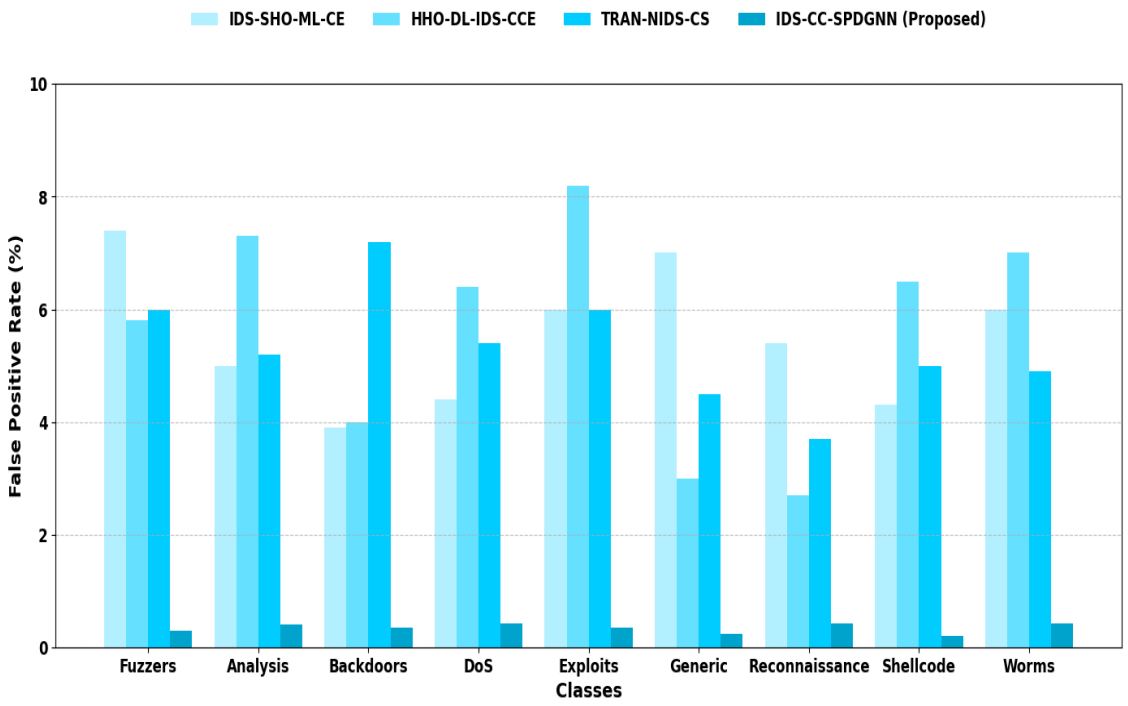

Figure 8: FPR analysis on the dataset of UNSW-NB15

Figure 8 presents FPR analysis on the dataset of UNSW-NB15. The IDS-CC-SPDGNN technique attains 3.87%, 6.45% and 4.29% lower FPR for fuzzers; 7.34%, 5.63% and 3.78% lower FPR for analysis; 6.02%, 4.91% and 7.12% lower FPR for Backdoors; 5.36%, 3.94% and 6.89% lower FPR for DoS; 7.81%, 4.58% and 3.62% lower FPR for exploits; 6.14%, 5.49% and 7.53% lower FPR for generic; 4.75%, 6.73% and 3.55% lower FPR for reconnaissance; 7.28%, 5.22% and 4.17% lower FPR for shellcode; 3.91%, 6.87% and 5.84% lower FPR for worms when compared with existing techniques like IDS-SHO-ML-CE, HHO-DL-IDS-CCE and TRAN-NIDS-CS respectively.

Figure 9 presents computation time analysis on UNSW-NB15 dataset. The proposed IDS-CC-SPDGNN technique attains 20.59%, 18.76% and 15.04% lower computational time when compared with existing techniques likes IDS-SHO-ML-CE, HHO-DL-IDS-CCE and TRAN-NIDS-CS respectively.

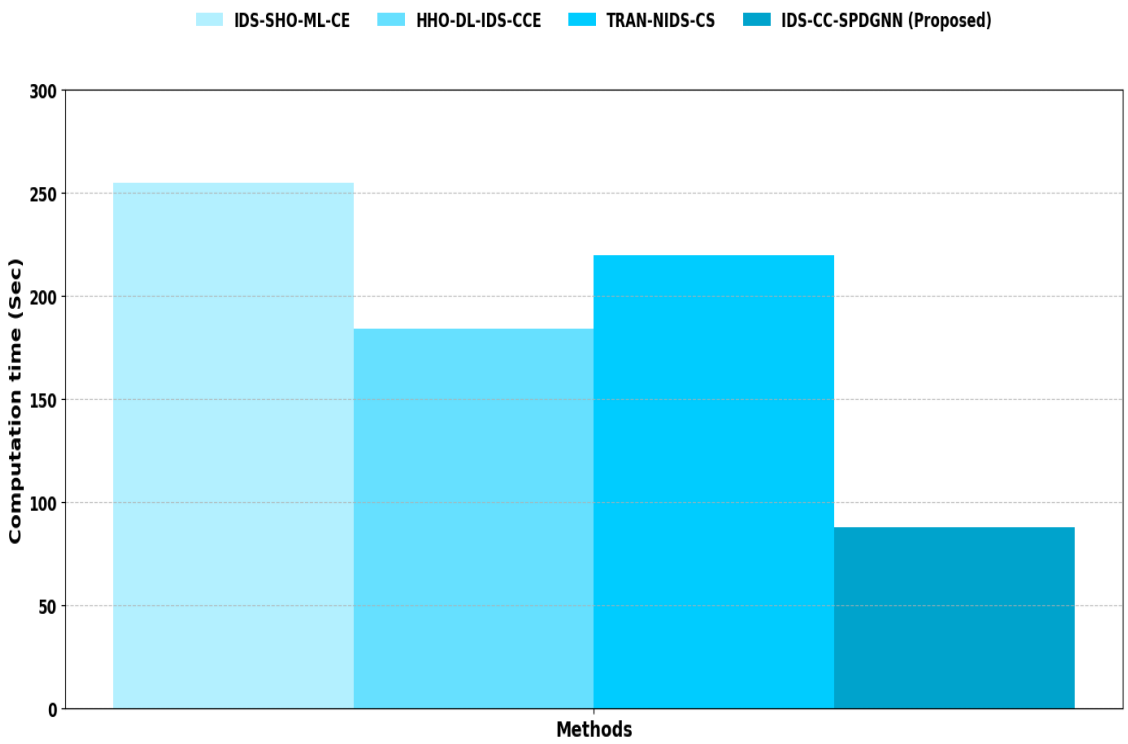

Figure 9: Computation time analysis on the dataset of UNSW-NB15

4.4 Performance Comparison using CIC-IDS2017 dataset

Table 7 to 12 shows the simulation outcomes of proposed IDS-CC-SPDGNN technique using UNSW-NB15 dataset. The proposed IDS-CC-SPDGNN techniques are compared with existing IDS-SHO-ML-CE [21], HHO-DL-IDS-CCE [22] and TRAN-NIDS-CS [23] techniques.

Table 7: Accuracy analysis on CIC-IDS2017 dataset

Methods	**Accuracy (%)**	
	Normal	**Anomaly**
IDS-SHO-ML-CE	93.54	94.65
HHO-DL-IDS-CCE	90.23	89.43
TRAN-NIDS-CS	95.07	94.32
IDS-CC-SPDGNN (Proposed)	99.28	99.34

Table 7 presents accuracy analysis on CIC-IDS2017 dataset. The proposed IDS-CC-SPDGNN technique attains 20.59%, 18.76% and 15.04% higher accuracy for normal; 20.59%,

18.76% and 15.04% higher accuracy for anomaly when compared with existing techniques likes IDS-SHO-ML-CE, HHO-DL-IDS-CCE and TRAN-NIDS-CS respectively.

Table 8: Precision analysis on CIC-IDS2017 dataset

Methods	**Precision (%)**	
	Normal	**Anomaly**
IDS-SHO-ML-CE	95.21	93.86
HHO-DL-IDS-CCE	90.54	89.94
TRAN-NIDS-CS	93.91	94.58
IDS-CC-SPDGNN (Proposed)	98.64	99.05

Table 8 presents precision analysis on CIC-IDS2017 dataset. The proposed IDS-CC-SPDGNN technique attains 5.85%, 10.78% and 7.29% higher precision for normal; 7.32%, 11.58% and 5.98% higher precision for anomaly when compared with existing techniques likes IDS-SHO-ML-CE, HHO-DL-IDS-CCE and TRAN-NIDS-CS respectively.

Table 9: Recall analysis on CIC-IDS2017 dataset

Methods	**Recall (%)**	
	Normal	**Anomaly**
IDS-SHO-ML-CE	88.76	89.05
HHO-DL-IDS-CCE	91.12	92.63
TRAN-NIDS-CS	90.07	91.83
IDS-CC-SPDGNN (Proposed)	98.23	98.51

Table 9 presents recall analysis on CIC-IDS2017 dataset. The proposed IDS-CC-SPDGNN technique attains 12.24%, 9.07% and 10.04% higher recall for normal; 11.12%, 8.21% and 9.16% higher recall for anomaly when compared with existing techniques likes IDS-SHO-ML-CE, HHO-DL-IDS-CCE and TRAN-NIDS-CS respectively.

Table 10: Specificity analysis on CIC-IDS2017 dataset

Methods	Specificity (%)	
	Normal	Anomaly
IDS-SHO-ML-CE	89.64	90.25
HHO-DL-IDS-CCE	91.34	92.56
TRAN-NIDS-CS	93.29	94.08
IDS-CC-SPDGNN (Proposed)	98.4	97.85

Table 10 presents specificity analysis on CIC-IDS2017 dataset. The proposed IDS-CC-SPDGNN technique attains 11.59%, 9.94% and 6.82% higher specificity for normal; 10.21%, 8.02% and 5.79% higher specificity for anomaly when compared with existing techniques likes IDS-SHO-ML-CE, HHO-DL-IDS-CCE and TRAN-NIDS-CS respectively.

Table 11: F1-score analysis on CIC-IDS2017 dataset

Methods	F1-score (%)	
	Normal	Anomaly
IDS-SHO-ML-CE	93.39	94.46
HHO-DL-IDS-CCE	92.03	95.78
TRAN-NIDS-CS	90.86	89.23
IDS-CC-SPDGNN (Proposed)	98.05	98.32

Table 11 presents f1-score analysis on CIC-IDS2017 dataset. The proposed IDS-CC-SPDGNN technique attains 7.34%, 8.43% and 11.18% greater F1-score for normal; 5.78%, 3.63% and 12.92% greater F1-score for anomaly when evaluated with existing techniques likes IDS-SHO-ML-CE, HHO-DL-IDS-CCE and TRAN-NIDS-CS respectively.

Table 12: FPR analysis on CIC-IDS2017 dataset

Methods	FPR (%)	
	Normal	Anomaly
IDS-SHO-ML-CE	7.54	6.83
HHO-DL-IDS-CCE	4.27	4.32

TRAN-NIDS-CS	5.52	6.21
IDS-CC-SPDGNN (Proposed)	0.34	0.53

Table 12 presents FPR analysis on CIC-IDS2017 dataset. The proposed IDS-CC-SPDGNN technique attains 12.65%, 7.32% and 8.9% lower FPR for normal; 7.25%, 3.42% and 7.15% lower FPR for anomaly when compared with existing techniques likes IDS-SHO-ML-CE, HHO-DL-IDS-CCE and TRAN-NIDS-CS respectively.

4.5 Comparisons with existing strategies

Table 13: Comparison analysis using state-of-the-art strategies

Authors	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-score (%)**	**Specificity (%)**	**Computation time (s)**
Jansi Sophia Mary, C. & Mahalakshmi, K. (2024) [21]	93.16	89.59	91.54	92.85	94.84	212
Nagamani, S., et al. (2024) [22]	94.81	91.56	92.04	93.11	92.21	274
Long, Z., et al. (2024) [23]	92.67	92.86	93.29	92.83	90.32	180
Sharma, H.S. & Singh, K.J. (2024) [24]	95.73	94.62	96.36	94.75	95.77	164
Sarkar, N., et al. (2024) [25]	90.98	91.48	92.38	90.96	89.79	231
Samunnisa, K., et al. (2023) [26]	94.51	95.95	92.56	96.23	95.05	185
Jain, D.K., et al. (2023) [27]	92.65	93.18	94.32	-	91.07	-
IDS-CC-SPDGNN (proposed)	99.28	98.84	98.19	99.04	98.64	89

Table 13 presents comparison analysis using state-of-the-art strategies. The proposed IDS-CC-SPDGNN model outperforms previous cutting-edge techniques, with the maximum accuracy of 99.28%, precision of 98.84%, recall of 98.19%, F1-score of 99.04%, specificity of

98.64%, and minimal computational time of 89 seconds. These findings show that the proposed approach is very effective, accurate, and has quick computational power, making it an excellent solution for advanced cloud intrusion detection and prevention.

4.6 Discussion

In this paper, experiments were conducted on two widely recognized datasets to assess the effectiveness of the proposed IDS-CC-SPDGNN method. Experimental results demonstrate that the proposed IDS-CC-SPDGNN system achieves high detection accuracy and reliability. On the UNSW-NB15 dataset, IDS-CC-SPDGNN achieved 99.28% accuracy, 98.19% recall and 99.04% f1-score. On the CIC-IDS2017 dataset, it obtained a best accuracy of 99.05%, high precision of 98.54%, and 98.85% f1-score. The high performance of IDS-CC-SPDGNN results from its capability to identify both local and global structural patterns in network traffic data through graph-based representations. The dual-stream architecture ensures balanced learning across different aspects of the data.

5. Conclusion

This paper presented IDS-CC-SPDGNN using SPDGNN and SOA for develop advanced cloud IDS. By employing the NUWGF technique, we ensured high-quality, normalized input data. The HOA was used to select the most important features, enhancing the proposed method's focus on significant traffic patterns. The novel SPDGNN efficiently captured complex relationships in network data, and the SOA was used to optimize its weight parameter for enhanced classification performance. The proposed IDS-CC-SPDGNN technique attains 11.59%, 9.94% and 6.82% higher specificity, 12.65%, 7.32% and 8.9% lower FPR when compared with existing techniques likes IDS-SHO-ML-CE, HHO-DL-IDS-CCE and TRAN-NIDS-CS respectively. Future research can focus on extending the proposed IDS-CC-SPDGNN approach to handle real time detection in cloud environments.

References:

[1] Al-Ghuwairi, A.R., Sharrab, Y., Al-Fraihat, D., AlElaimat, M., Alsarhan, A. and Algarni, A., 2023. Intrusion detection in cloud computing based on time series anomalies utilizing machine learning. Journal of Cloud Computing, 12(1), p.127.

[2] Mohamed, D. and Ismael, O., 2023. Enhancement of an IoT hybrid intrusion detection system based on fog-to-cloud computing. Journal of Cloud Computing, 12(1), p.41.

[3] Lin, H., Xue, Q., Feng, J. and Bai, D., 2023. Internet of things intrusion detection model and algorithm based on cloud computing and multi-feature extraction extreme learning machine. Digital Communications and Networks, 9(1), pp.111-124.

[4] Abid, A., Jemili, F. and Korbaa, O., 2024. Real-time data fusion for intrusion detection in industrial control systems based on cloud computing and big data techniques. Cluster Computing, 27(2), pp.2217-2238.

[5] Kasongo, S.M., 2023. A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework. Computer Communications, 199, pp.113-125.

[6] Bakro, M., Kumar, R.R., Alabrah, A., Ashraf, Z., Ahmed, M.N., Shameem, M. and Abdelsalam, A., 2023. An improved design for a cloud intrusion detection system using hybrid features selection approach with ML classifier. IEEE Access, 11, pp.64228-64247.

[7] Gulia, N., Solanki, K., Dalal, S., Dhankhar, A., Dahiya, O. and Salmaan, N.U., 2023. Intrusion Detection System Using the G-ABC with Deep Neural Network in Cloud Environment. Scientific Programming, 2023(1), p.7210034.

[8] Tariq, N., Alsirhani, A., Humayun, M., Alserhani, F. and Shaheen, M., 2024. A fog-edge-enabled intrusion detection system for smart grids. Journal of Cloud Computing, 13(1), p.43.

[9] Javadpour, A., Pinto, P., Ja'fari, F. and Zhang, W., 2023. DMAIDPS: A distributed multi-agent intrusion detection and prevention system for cloud IoT environments. Cluster Computing, 26(1), pp.367-384.

[10] Pandey, B.K., Veeramanickam, M.R.M., Ahmad, S., Rodriguez, C. and Esenarro, D., 2023. ExpSSOA-Deep maxout: Exponential Shuffled shepherd optimization based Deep maxout network for intrusion detection using big data in cloud computing framework. Computers & Security, 124, p.102975.

[11] Vashishtha, L.K., Singh, A.P. and Chatterjee, K., 2023. HIDM: A hybrid intrusion detection model for cloud based systems. Wireless Personal Communications, 128(4), pp.2637-2666.

[12] El-Kosairy, A. and Abdelbaki, N., 2023. Deception as a service: intrusion and ransomware detection system for cloud computing (IRDS4C). Advances in Computational Intelligence, 3(3), p.9.

[13] Raj, M.G. and Pani, S.K., 2023. Hybrid feature selection and BWTDO enabled DeepCNN-TL for intrusion detection in fuzzy cloud computing. Soft Computing, pp.1-20.

[14] Chithanya, K.V.K. and Reddy, L., 2025. Automatic intrusion detection model with secure data storage on cloud using adaptive cyclic shift transposition with enhanced ANFIS classifier. Cyber Security and Applications, 3, p.100073.

[15] Sivamohan, S. and Sridhar, S.S., 2023. An optimized model for network intrusion detection systems in industry 4.0 using XAI based Bi-LSTM framework. Neural Computing and Applications, 35(15), pp.11459-11475.

[16] Salvakkam, D.B., Saravanan, V., Jain, P.K. and Pamula, R., 2023. Enhanced quantum-secure ensemble intrusion detection techniques for cloud based on deep learning. Cognitive Computation, 15(5), pp.1593-1612.

[17] Sangaiah, A.K., Javadpour, A., Ja'fari, F., Pinto, P., Zhang, W. and Balasubramanian, S., 2023. A hybrid heuristics artificial intelligence feature selection for intrusion detection classifiers in cloud of things. Cluster Computing, 26(1), pp.599-612.

[18] Vadigi, S., Sethi, K., Mohanty, D., Das, S.P. and Bera, P., 2023. Federated reinforcement learning based intrusion detection system using dynamic attention mechanism. Journal of Information Security and Applications, 78, p.103608.

[19] Sah, G., Banerjee, S. and Singh, S., 2023. Intrusion detection system over real-time data traffic using machine learning methods with feature selection approaches. International Journal of Information Security, 22(1), pp.1-27.

[20] Maheswari, K.G., Siva, C. and Nalinipriya, G., 2023. Optimal cluster based feature selection for intrusion detection system in web and cloud computing environment using hybrid teacher learning optimization enables deep recurrent neural network. Computer Communications, 202, pp.145-153.

[21] Jansi Sophia Mary, C. and Mahalakshmi, K., 2024. Modelling of intrusion detection using sea horse optimization with machine learning model on cloud environment. *International Journal of Information Technology*, *16*(3), pp.1981-1988.

[22] Nagamani, S., Arivalagan, S., Senthil, M. and Sudhakar, P., 2024. Horse Herd optimization with deep learning based intrusion detection in cloud computing environment. *International Journal of Information Technology*, pp.1-7.

[23] Long, Z., Yan, H., Shen, G., Zhang, X., He, H. and Cheng, L., 2024. A Transformer-based network intrusion detection approach for cloud security. *Journal of Cloud Computing*, *13*(1), p.5.

[24] Sharma, H.S. and Singh, K.J., 2024. A feed forward deep neural network model using feature selection for cloud intrusion detection system. *Concurrency and Computation: Practice and Experience*, *36*(9), p.e8001.

[25] Sarkar, N., Keserwani, P.K. and Govil, M.C., 2024. A better and fast cloud intrusion detection system using improved squirrel search algorithm and modified deep belief network. *Cluster Computing*, *27*(2), pp.1699-1718.

[26] Samunnisa, K., Kumar, G.S.V. and Madhavi, K., 2023. Intrusion detection system in distributed cloud computing: Hybrid clustering and classification methods. Measurement: Sensors, 25, p.100612.

[27] Jain, D.K., Ding, W. and Kotecha, K., 2023. Training fuzzy deep neural network with honey badger algorithm for intrusion detection in cloud environment. International Journal of Machine Learning and Cybernetics, 14(6), pp.2221-2237.

[28] https://www.kaggle.com/datasets/mrwellsdavid/unsw-nb15

[29] https://www.kaggle.com/datasets/chethuhn/network-intrusion-dataset

[30] Lu, P., Mu, Y., Gu, C., Fu, S., Cheng, Q., Zhao, K. and Shen, X., 2025. Multi-scale infrared image enhancement based on non-uniform weighted guided filtering. Optics and Lasers in Engineering, 186, p.108797.

[31] Oladejo, S.O., Ekwe, S.O. and Mirjalili, S., 2024. The Hiking Optimization Algorithm: A novel human-based metaheuristic approach. Knowledge-Based Systems, 296, p.111880.

[32] Chen, J., Yuan, Y. and Luo, X., 2024. Sdgnn: Symmetry-preserving dual-stream graph neural networks. IEEE/CAA journal of automatica sinica, 11(7), pp.1717-1719.

[33] Zhong, C., Li, G., Meng, Z., Li, H., Yildiz, A.R. and Mirjalili, S., 2025. Starfish optimization algorithm (SFOA): a bio-inspired metaheuristic algorithm for global optimization compared with 100 optimizers. Neural Computing and Applications, 37(5), pp.3641-3683.

[34] Computational and Investigational Proportional Flow Study on Cd Nozzle - JYOTHI NT, Ashwin Nair, P Ebby Darney - IJFMR Volume 5, Issue 6, November-December 2023. DOI 10.36948/ijfmr.2023.v05i06.11081

[35] N. Ramesh Reddy, P. Mohan Reddy, N. Jyothi, A. Sai Kumar, Jae Hak Jung, Sang Woo Joo, Versatile TiO2 bandgap modification with metal, non-metal, noble metal, carbon material, and semiconductor for the photoelectrochemical water splitting and photocatalytic dye degradation performance, Journal of Alloys and Compounds, Volume 935, Part 1, 2023, 167713, ISSN 0925-8388, https://doi.org/10.1016/j.jallcom.2022.167713.

[36] Suman Rana,Bhavin Soni,Dr. P. Ebby Darney,Jyothi NT, "EFFECTS OF T4 HORMONES ON HUMANBODY AND THEIR ANALYSIS", International Journal of Creative Research Thoughts (IJCRT), ISSN:2320-2882, Volume.10, Issue 10, pp.d332-d339, October 2022, Available at :http://www.ijcrt.org/papers/IJCRT2210389.pdf

[37] Systematic Review and Survey on Dominant Influence of Vedas and Ignorance Transpired in Space Science and Aviation", International Journal of Emerging Technologies and Innovative Research (www.jetir.org | UGC and issn Approved),

ISSN:2349-5162, Vol.9, Issue 7, page no. ppb490-b493, July-2022, Available at : http://www.jetir.org/papers/JETIR2207158.pdf

[38] "Systematic Survey of Wind Tunnel Test facility in India", International Journal of Emerging Technologies and Innovative Research (www.jetir.org | UGC and issn Approved), ISSN:2349-5162, Vol.9, Issue 6, page no. pph830-h840, June-2022, Available at : http://www.jetir.org/papers/JETIR2206795.pdf

[39] Ashika Parveen,Jyothi Nt,Jv Muruga lal Jeyan, "STUDY OF IMPLEMENTATION OF VALUE STREAM MAPPING AND LEAN TOOLS TO ACHIEVE LEAN", International Journal of Creative Research Thoughts (IJCRT), ISSN:2320-2882, Volume.10, Issue 10, pp.e329-e334, October 2022, Available at :http://www.ijcrt.org/papers/IJCRT2210502.pdf

[40] An Intercontinental Study of Employee and Employer Human Factor Issues Put Up in Aerospace and Aviation Industry - Jyothi NT, Hussainar A, Shilpa Rana, Muruga lal Jeyan JV - IJFMR Volume 6, Issue 1, January-February 2024. DOI 10.36948/ijfmr.2024.v06i01.12441

[41] Suman Rana,Bhavin Soni,Dr. P. Ebby Darney,Jyothi NT, "EFFECTS OF T4 HORMONES ON HUMANBODY AND THEIR ANALYSIS", International Journal of Creative Research Thoughts (IJCRT), ISSN:2320-2882, Volume.10, Issue 10, pp.d332-d339, October 2022, Available at :http://www.ijcrt.org/papers/IJCRT2210389.pdf

[42] Ashika Parveen, JV Muruga Lal Jeyan, Jyothi NT3 International Study on Application of Value Stream Mapping to Identify the Necessity of Lean System Implementation , International Journal of Scientific Research in Engineering and Management (IJSREM) Volume: 06 Issue: 09 | September - 2022 Impact Factor: 7.185 ISSN: 2582-3930

[43] JV Muruga lal Jeyan, Jyothi NT Rashi Kaushik Systematic Review and Survey on Dominant Influence of Vedas and Ignorance Transpired in Space Science and Aviation", International Journal of Emerging Technologies and Innovative Research (www.jetir.org), ISSN:2349-5162, Vol.9, Issue 7, page no.b490-b493, July-2022, Available :http://www.jetir.org/papers/JETIR2207158.pdf

[44] JV Muruga lal Jeyan, Jyothi NT , Boopesh Raja, Rajarajan G "THEORY STRATEGY OF SUBSONIC WIND TUNNEL FOR LOW VELOCITY ", International Journal of Emerging Technologies and Innovative Research (www.jetir.org), ISSN:2349-5162, Vol.9, Issue 6, page no.j572-j580, June-2022, Available :http://www.jetir.org/papers/JETIR2206973.pdf

[45] JV Muruga lal Jeyan, Jyothi NT, Reshmitha Shree, Bhawadharanee S, Rajarajan, THEORETICAL STUDY OF HYPERSONIC WIND TUNNEL TEST FACILITY IN INDIA ", International Journal of Emerging Technologies and Innovative Research (www.jetir.org), ISSN:2349-5162, Vol.9, Issue 6, page no.j512-j518, June-2022, Available :http://www.jetir.org/papers/JETIR2206967.pdf

[46] JV Muruga lal Jeyan, Jyothi NT , V S Devika Thampuratty, B Nithin, Rajarajan, CONCEPT DESIGN AND DEVELOPMENT OF SUPERSONIC WIND TUNNEL ", International Journal of Emerging Technologies and Innovative Research (www.jetir.org | UGC and issn Approved), ISSN:2349-5162, Vol.9, Issue 6, page no. ppj209-j217, June-2022, Available at : http://www.jetir.org/papers/JETIR2206925.pdf

[47] Muthu Venkatesh, Rajarajan G Jyothi NT JV Muruga Lal Jeyan "Systematic Survey of Wind Tunnel Test facility in India", International Journal of Emerging Technologies and Innovative Research (www.jetir.org), ISSN:2349-5162, Vol.9, Issue 6, page no.h830-h840, June-2022, Available :http://www.jetir.org/papers/JETIR2206795.pdf

[48] Ashika Parveen, JV Muruga Lal Jeyan, Jyothi NT "Investigation Of Lean Developments And The Study Of Lean Techniques Through Event Studies" Internation Journal for Science and Advance Research In Technology, 8(4)

[49] P Gopala Krishnan, JV Muruga Lal Jeyan, Jyothi NT "Novel Evaluation Of Aircraft Data Structure Optimization Techniques And Opportunities" International Journal for Science and Advance Research In Technology, 8(4)

[50] Suryansh Upadhyay, JV Muruga lal Jeyan, Jyothi NT Preliminary Study on Brain Computer Interface © August 2021| IJIRT | Volume 8 Issue 3 | ISSN: 2349-6002 IJIRT 152537 INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN TECHNOLOGY 720

[51] Sruthi.s.kumar, Jyothi Nt , Jv Muruga lal jeyan . Computational Turbine Blade Analysis with Thermal Barrier Coating International Journal of Engineering Research and Applications www.ijera.com ISSN: 2248-9622, Vol. 12, Issue 4, (Series-I) April 2022, pp. 01-08, DOI: 10.9790/9622-1204010108

[52] Jyothi, N. T., Ganesan, H., & Jeyan, J. V. (2024, April). Methodical assessment and truth flow analysis of wind tunnels. In AIP Conference Proceedings (Vol. 3037, No. 1). AIP Publishing.

[53] lal Jeyan, J. M., Jyothi, N. T., Raja, B., & Rajarajan, G. THEORY STRATEGY OF SUBSONIC WIND TUNNEL FOR LOW VELOCITY. International Journal of Emerging Technologies and Innovative Research (www. jetir. org), ISSN, 2349-5162.

[54] Venkatesh, M. Rajarajan G Jyothi NT JV Muruga Lal Jeyan" Systematic Survey of Wind Tunnel Test facility in India. International Journal of Emerging Technologies and Innovative Research (www. jetir. org), ISSN, 2349-5162.

[55] lal Jeyan, J. M., Jyothi, N. T., Thampuratty, V. D., Nithin, B., & Rajarajan, C. D. DEVELOPMENT OF SUPERSONIC WIND TUNNEL. International Journal of Emerging Technologies and Innovative Research (www. jetir. org| UGC and issn Approved), ISSN, 2349-5162.

[56] Parveen, Ashika, JV Muruga Lal Jeyan, and N. T. Jyothi. "Investigation Of Lean Developments And The Study Of Lean Techniques Through Event Studies." Internation Journal for Science and Advance Research In Technology 8.4.

[57] Krishnan, P. Gopala, JV Muruga Lal Jeyan, and N. T. Jyothi. "Novel Evaluation Of Aircraft Data Structure Optimization Techniques And Opportunities." International Journal for Science and Advance Research In Technology 8.4.

[58] Parveen, Ashika, JV Muruga Lal Jeyan, and N. T. Jyothi. "International Study on Application of Value Stream Mapping to Identify the Necessity of Lean System Implementation." International Journal of Scientific Research in Engineering and Management (IJSREM) Volume 6.

[59] lal Jeyan, JV Muruga, et al. "Rajarajan, THEORETICAL STUDY OF HYPERSONIC WIND TUNNEL TEST FACILITY IN INDIA." International Journal of Emerging Technologies and Innovative Research (www. jetir. org), ISSN: 2349-5162.

[60] lal Jeyan, JV Muruga, et al. "THEORY STRATEGY OF SUBSONIC WIND TUNNEL FOR LOW VELOCITY." International Journal of Emerging Technologies and Innovative Research (www. jetir. org), ISSN: 2349-5162.

[61] Venkatesh, Muthu. "Rajarajan G Jyothi NT JV Muruga Lal Jeyan" Systematic Survey of Wind Tunnel Test facility in India." International Journal of Emerging Technologies and Innovative Research (www. jetir. org), ISSN: 2349-5162.

[62] lal Jeyan, JV Muruga, et al. "DEVELOPMENT OF SUPERSONIC WIND TUNNEL." International Journal of Emerging Technologies and Innovative Research (www. jetir. org| UGC and issn Approved), ISSN: 2349-5162.

[63] lal Jeyan, JV Muruga. "Jyothi NT Rashi Kaushik Systematic Review and Survey on Dominant Influence of Vedas and Ignorance Transpired in Space Science and Aviation." International Journal of Emerging Technologies and Innovative Research (www. jetir. org), ISSN: 2349-5162.

[64] lal Jeyan, JV Muruga. "Jyothi NT." VS Devika Thampuratty, B Nithin, Rajarajan, CONCEPT DESIGN AND DEVELOPMENT OF SUPERSONIC WIND TUNNEL", International Journal of Emerging Technologies and Innovative Research (www. jetir. org| UGC and issn Approved), ISSN: 2349-5162.

[65] A.John B, J. V. M. L. Jeyan, J. NT, A. Kumar, Assessment of the Properties of Modified Pearl Millet Starch. Starch. 2023, 75, 2200160.

[66] Shaik Ameen, Velagala Susmitha, Poranki Vyshnavi, Talapaneni Venkata Sai Teja, Dr. Nm Jyothi, and Dr. M. Madhusudhana Subramanyam. “Detection of Plant Diseases Using Advanced Deep Learning Methods”. Educational Administration: Theory and Practice, vol. 30, no. 5, May 2024, pp. 8836-43, doi:10.53555/kuey.v30i5.4466.

[67] N. T. Jyothi, H. Ganesan, J. V. Muruga Lal Jeyan; Methodical assessment and truth flow analysis of wind tunnels. AIP Conf. Proc. 2 April 2024; 3037 (1): 020016. https://doi.org/10.1063/5.0196120

[68] B. c. Mathew, K. S. Priyanka and J. V. M. Lal Jeyan, "Computational study on chamber morphing wing concept for efficient lift at various angle of attack," 2020 International Conference on Interdisciplinary Cyber Physical Systems (ICPS), Chennai, India, 2020, pp. 68-71, doi: 10.1109/ICPS51508.2020.00020.

[69] MurugalalJeyan, J. V., and Dr M. Senthil Kumar. "Performance evaluation of yaw meter with the aid of computational fluid dynamic." *International Review of Mechanical Engineering (IREME). ISSN* 8734 (1970).

[70] LAL JEYAN, JV MURUGA, and M. SENTHIL KUMAR. "PERFORMANCE EVALUATION FOR MULTI-HOLE PROBE WITH THE AID OF ARTIFICIAL NEURAL NETWORK." *Journal of Theoretical & Applied Information Technology* 65.3 (2014).

[71] Ann Steffy Thomas, Anu John B, T G Ansalam Raj, JV Muruga Lal Jeyan. (2025). The Effects on Sleep Quality, Consistency, and Duration to Improve Human Reproductive and Health Outcomes. Frontiers in Pharmaceutical, Medical and Health Sciences (FPMHS), 6(3), 9-21.

Citation: Vineshraj S, P Ebby Darney. (2025). An Intelligent Intrusion Detection System for Cloud Computing Environments Using Symmetry-Preserving Dual-Stream Graph Neural Networks. International Journal of Advanced Research in Engineering and Technology (IJARET), 16(4), 10-50.

Abstract Link: https://iaeme.com/Home/article_id/IJARET_16_04_002

Article Link:
https://iaeme.com/MasterAdmin/Journal_uploads/IJARET/VOLUME_16_ISSUE_4/IJARET_16_04_002.pdf

✉ **editor@iaeme.com**