Journal of Information Systems and Telecommunication
Vol.13, No.1, January-March 2025, 50-62
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

A Novel Hybrid Convolutional-Attention Recurrent Network (HCARN) for Enhanced Cybersecurity Threat Detection

Archana R. Laddhad [1*], Gurveen Vaseer [1]

[1].Faculty of Computer Science, Oriental University, Indore – Madhya Pradesh, India - 453555

Received: 09 Nov 2024/ Revised: 04 Mar 2025/ Accepted: 19 Apr 2025

Abstract
Cybersecurity solutions are critical for the protection of networks against constantly evolving threats. Traditional intrusion detection systems (IDS) struggle to adapt to the rapidly varying attack patterns, encouraging the exploration of advanced techniques such as deep learning. This study introduces a novel framework utilizing a Hybrid Convolutional-Attention Recurrent Network (HCARN) for identifying cybersecurity threat. Utilizing the CSE-CIC-IDS2018 dataset, the data preparation process includes data cleanup, feature extraction, and Information Gain-based feature choice. The HCARN architecture, integrates convolutional layers, attention mechanisms, and recurrent layers, is employed for categorization. Convolutional layers effectively capture spatial features in the dataset, attention mechanisms highlight critical features, and recurrent layers model temporal dependencies. This allows HCARN to process and analyze complex patterns in network traffic, leading to more accurate threat diagnosis. The proposed model proves significant efficacy in distinguishing between major, moderate, and minor threats, attaining high accuracy and robustness in threat recognition. The incorporation of attention mechanisms allows the model to emphasize on critical features, while the recurrent layers pay attention to temporal dependencies in the dataset. The HCARN architecture determines classification accuracy, achieving 94.7% in K-fold validation, 95.4% in model training, and 92.3% in model testing while classifying major, moderate, minor threats satisfactorily, confirming its effectiveness in cybersecurity threat detection. This novel attempt underscores the potential of hybrid deep learning models in enhancing cybersecurity defenses against sophisticated attacks, paving the way for adaptive security systems.

Keywords: Intrusion Detection Systems; CSE-CIC-IDS2018; Deep Learning; Hybrid Convolutional-Attention Recurrent Network; Cybersecurity

1- Introduction

In today's cybersecurity landscape, Intrusion Detection Systems (IDS) hold significant importance by serving as a vital security measure against the continuously growing array of digital threats. An Intrusion Detection System (IDS) is an active security measure devised to detect and counteract unauthorized or malicious activities occurring within a network or system [1]. The principal objective of this system is to actively observe network traffic and analyze system behavior, with the purpose of promptly identifying any peculiar patterns or discrepancies that could potentially serve as indications of a security breach. In contemporary cybersecurity landscape, Intrusion Detection Systems (IDS) play a pivotal role in preserving the authenticity, secrecy, and accessibility of digital assets, thus imbuing them with utmost significance as a formidable deterrent against the ever-changing realm of security threats [2].

Conventional Intrusion Detection Systems (IDS) rely on pre-established rules and signatures in order to detect and classify recognized attack patterns. Nevertheless, it is frequently challenging for them to effectively adjust to the rapidly evolving risks and complex methods employed by potential assailants [3]. The aforementioned constraint has prompted researchers to investigate sophisticated methodologies, such as the amalgamation of artificial intelligence (AI), machine learning (ML), and deep learning (DL) approaches, with the intent of augmenting the precision and responsiveness of Intrusion Detection Systems (IDS). These approaches facilitate Intrusion Detection Systems (IDS) to acquire knowledge from data, identify novel attack patterns, and generate prompt decisions, rendering them indispensable instruments for

✉ **Archana R. Laddhad**
archanaladdhad@gmail.com

enterprises intending to enhance their cybersecurity safeguards. By utilizing these methodologies, Intrusion Detection Systems (IDS) can transcend rule-based methodologies, which encounter difficulties in accommodating emerging threats, and instead become adept in identifying innovative attack patterns. The utilization of artificial intelligence (AI), machine learning (ML), and deep learning (DL) methodologies in intrusion detection systems (IDS) improves their capacity to identify both familiar and novel threats, decrease instances of erroneous positive detections, and effectively react to security incidents in a timely manner. The CSE-CIC-IDS2018 dataset offers researchers a significant opportunity to implement these methodologies in practical situations, enabling the training and evaluation of intrusion detection systems based on artificial intelligence, machine learning, and deep learning [4]. It also permits the assessment of their efficacy in tackling contemporary cybersecurity challenges, which are characterized by their dynamic and intricate nature.

In their development of two deep neural network models for intrusion detection in cloud environments, Alzughaibi & El Khediri [5] achieve high accuracy rates on with 98.97% for binary classification and 98.41% for multi-class classification. Göcs & Johanyák [6] concentrate on feature selection for intrusion detection systems. They use six feature selection techniques and classification algorithms to find pertinent elements essential for differentiating between benign and malicious network traffic. In their comparison of bio-inspired optimization algorithms for cybersecurity attack detection, Najafi Mohsenabad & Tut [7] found that Ant Colony Optimization, Flower Pollination Algorithm, and Artificial Bee Colony feature-selection strategies produced detection accuracies of over 98.6%.

On the CSE-CIC-IDS2018 dataset, Göcs & Johanyák [8] describe a novel ensemble feature-ranking strategy that improves on existing feature-ranking score combinations and achieves superior classification metrics, particularly for specific attack types. XGBoost, DT, and RF models are highlighted for their superior performance in terms of ROC values and CPU runtime by Songma, Sathuphan, and Pamutha [9] as they optimize intrusion detection systems using data preprocessing, dimensionality reduction with PCA and RF, and various machine learning techniques on the CSE-CIC-IDS-2018 dataset. Using 19 features chosen using the decision tree technique, Khan & Haroon [10] offer an Artificial Neural Network (ANN)-based intrusion detection system that achieves great performance on the CSE-CIC-IDS2018 dataset. Farhan & Jasim [11] use deep learning, namely LSTM, for intrusion detection. They achieve an impressive detection accuracy of up to 99%, demonstrating the usefulness of deep learning techniques for cybersecurity applications. The summary of literature review is organized in Table 1. Masoudi & Ghaffari [26] conducted a comprehensive investigation on Software Defined Networks (SDN), focusing on performance issues and solutions in SDN-based data centers. They grouped solutions into different clusters and identified key challenges and future research directions in the field. Further, Shirmarz & Ghaffari [27] focused on enhancing the performance of software defined network through an autonomic system based on deep neural networks. Their architecture demonstrates improved performance metrics such as blocking probability, delay, and packet loss rate. Shirmarz & Ghaffari [28] continued the research and presented improving DDoS attack detection in SDN using Self-Organizing Maps and Learning Vector Quantization. The approach significantly improves the detection rate, making SDN more resilient against cyber threats.

Table 1: Summary of literature review

Objectives	***Attacks***	***Algorithms***	***Authors***
IDS	DoS, U2L R2L, Probe	K-means + KNN	Tsai et al. [12]
Encrypted traffic classification	Malicious instances		Bar et al. [13]
IDS	DoS, U2L R2L, Probe		Lin et al. [14]
Malware detection	High-risk malwares	SVM + KNN	Comar et al. [15]
IDS	DoS, U2L R2L, Probe	SVM + kNN + PSO	Aburomman et al. [16]
Android malware detection	-	SVM + DT	Li et al. [17]
IDS	All attacks	K-Support Vector	Bamakan et al. [18]
IDS	Anomalous connections	PCA Filtering + Probabilistic SOM	Hoz et al. [19]
IDS	DoS, U2R, R2L, probe	K-Means + NB + BNN	Dubey et al. [20]
IDS		RF + AODE	Jabbar et al. [21]
IDS	Botnet	DT + NB + ANN	Moustafa et al. [22]
NADS	DoS, U2R, R2L, probe	NB + KNN	Pajouh et al. [23]
DoS attack detection	DoS	Multivariate Correlation + Triangle Area	Tan et al. [24]
Network forensics	DDoS, DARPA	FL + ES	Liao et al. [25]
IDS	All attacks	Hybrid Convolutional-Attention Recurrent Network	Present model

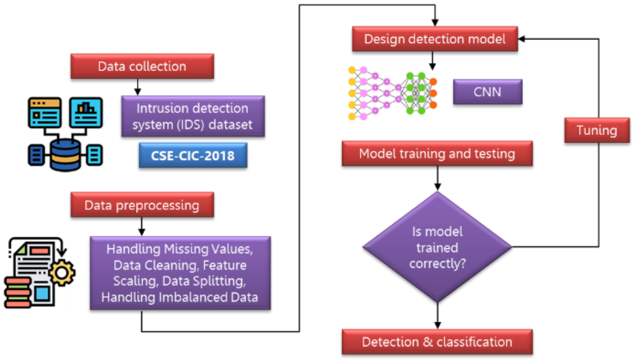

Data collection
Intrusion detection system (IDS) dataset
CSE-CIC-2018
Data preprocessing
Handling Missing Values, Data Cleaning, Feature Scaling, Data Splitting, Handling Imbalanced Data
Design detection model
CNN
Tuning
Model training and testing
Is model trained correctly?
Detection & classification

Fig. 1 Methodology

In the context of a swiftly progressing digital environment, the significance of highly resilient intrusion detection systems holds substantial weight. This paper makes a contribution to the continuous endeavors in enhancing network security and protecting the digital realm against an increasingly wide range of threats by utilizing the CSE-CIC-IDS2018 dataset and the capabilities of deep learning. The findings presented in this manuscript not only provide valuable insights into the effectiveness of deep learning-based intrusion detection, but also elucidate the trajectory for developing intelligent, versatile, and proactive cybersecurity measures. The flow of the research is depicted in Figure 1. The core objective of this investigation is to develop and assess the HCARN for cybersecurity threat detection. It aims to:

- improve the effectiveness of intrusion detection systems (IDS) by employing advanced deep learning method
- integrate recurrent, attention, and convolutional mechanisms to efficiently capture network traffic with temporal, critical, and spatial patterns
- evaluate the HCARN’s performance on the CSE-CIC-IDS2018 dataset to categorize cybersecurity threats into major, moderate, and minor sets reliably.

2- CSE-CIC-IDS2018 Dataset

The dataset consists of a substantial amount of annotated network traffic data, which is indispensable in the process of training, testing, and validating the efficacy of intrusion detection systems. The CSE-CIC-IDS2018 dataset [29] exemplifies a meticulous emphasis on realism, effectively replicating the complex and ever-evolving characteristics of contemporary network environments. The current dataset encapsulates a wide array of network activities encompassing both legitimate and malicious traffic. Such inclusion permits researchers to evaluate the efficacy of Intrusion Detection Systems (IDS) in distinguishing between the two types of traffic within an environment that closely mimics real-world conditions. The CSE-CIC-IDS2018 dataset presents a comprehensive range of characteristics, each augmenting the comprehension of network traffic patterns. The aforementioned characteristics encompass details regarding individual packets, aggregated data on network flows, and diverse metadata pertaining to the network traffic. The extensive amount of information available to enhance IDS that possess the capability to precisely identify security risks.

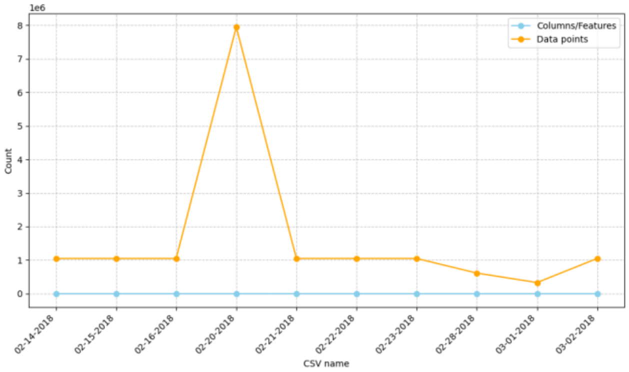


Fig. 2 Comparison of features and data points

- Number of observations: The dataset is acknowledged to encompass a considerable quantity of observations, frequently reaching in the millions. The precise quantity of observations may fluctuate contingent upon the particular iteration or subset of the dataset under consideration. Knowing the number of features and observations is essential for training the HCARN model as it helps in deciding the input shape and the convolution of the dataset and thus figure 2 represents comparison of features and data points.
- Number of features and attributes: The dataset typically encompasses a multitude of features and attributes which serve to depict network traffic data. The range of features within the dataset's version can vary from tens to hundreds. Features often include data related to the contents of packets, patterns of network traffic, and a range of additional attributes that characterize the network. The output of the application is in CSV file format with six columns labeled for each flow, namely FlowID, SourceIP, DestinationIP, SourcePort, DestinationPort, and Protocol with more than 80 network traffic features. Number of features and data points in each csv file as shown in Table 2 guides in setting applicable hyperparameters and design of architecture, confirming efficient training and optimization. Here, data points refer to individual instances or observations in a dataset and each row or entry in the dataset represents a single data point. Each row signifies a specific date in February and March 2018, with the subsequent number of features in

various columns and the total number of data points logged for that date.

Table 2: Number of features and data points in each csv file

CSV Name	*Features*	*Data Points*
02-14-2018	79	1,048,574
02-15-2018	79	1,048,574
02-16-2018	79	1,048,574
02-20-2018	83	7,948,746
02-21-2018	79	1,048,574
02-22-2018	79	1,048,574
02-23-2018	79	1,048,574
02-28-2018	79	613,103
03-01-2018	79	331,124
03-02-2018	79	1,048,574

- Number of attacks: The dataset known as CSE-CIC-IDS2018 encompasses a diverse array of cyberattacks and network anomalies. The dataset typically encompasses various categories.
- Denial of Service Attacks (DoS): The primary objective of these attacks is to inundate a targeted system or network, resulting in its unavailability to genuine users. Instances of these types of attacks incorporate SYN flood attacks and UDP flood attacks.
- Port Scanning: Port scanning attacks encompass the practice of systematically investigating a target system to ascertain the availability of open ports with the aim of identifying potential security vulnerabilities or discerning the services currently active on the system.
- Malware: The dataset potentially encompasses traffic affiliated with the propagation or transmission of malicious software, including botnets, worms, and viruses.
- Intrusions: Intrusions refer to a multitude of unauthorized activities occurring within a network, such as unauthorized access, privilege escalation, or other forms of network exploitation.
- Botnet Activity: The dataset may contain instances of Botnet activity, whereby the activities related to a network of compromised devices controlled by a malevolent individual are detected.

Understanding category wise traffic distribution by respective shares (%) is vital as it presents insights into the prevalence and importance of different attacks (See Figure 3). This helps in prioritizing the emphasis of feature engineering, training, and evaluation schemes, certifying model's robustness to effectively classify and detect the most prevalent real-world threats. Furthermore, it helps in resource allocation and decision-making for mitigating specific types of attacks based on relative occurrence frequencies. These attacks are explained herewith.

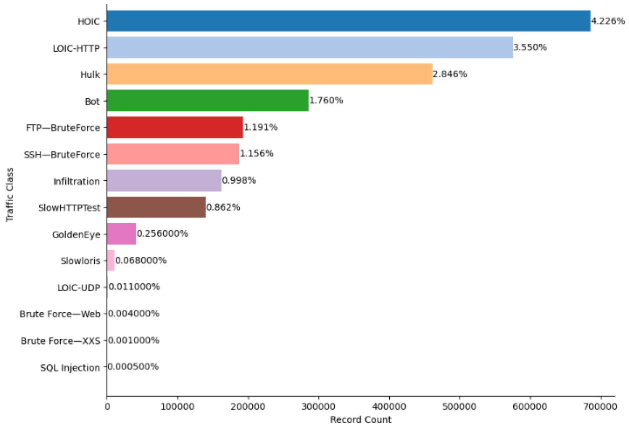


Fig. 3 Types of attacks and counts

- HOIC: It has 686,012 samples, corresponds to 4.226% of the total traffic flow. HOIC (High Orbit Ion Cannon) is a common tool that launches DDoS attacks.
- LOIC-HTTP: This category contains 576,191 samples, making up 3.550% of the total records. LOIC (Low Orbit Ion Cannon) is another DDoS tool, and the "HTTP" suffix suggests it targets web servers.
- Hulk: With 461,912 samples, Hulk forms 2.846% of the total records. Hulk is a Python script employed to execute DDoS attacks by flooding web servers with HTTP demands.
- Bot: This category incorporates 286,191 samples, representing 1.76% of the total records. Bots are automated program that executes several tasks, including detection of malicious behavior such as data theft, spamming, or DDoS.
- FTP—BruteForce: There are 193,360 samples here, with 1.191% of the total records. FTP (File Transfer Protocol) Brute Force attack attempts to gain unauthorized access to FTP servers by methodically trying unique combinations of username/password.
- SSH—BruteForce: This category has of 187,589 samples, that takes 1.156% of the total records. SSH (Secure Shell) Brute Force attacks attempts guessing of SSH login credentials to secure unauthorized access.
- Infiltration: With 161,934 samples, Infiltration denotes 0.998% of the total records. Infiltration stand for to the unauthorized invasion of a system or network with the intentation of retrieving confidential data or causing destruction.
- SlowHTTPTest: This category involves 139,890 samples, with 0.862% of the total records. SlowHTTPTest is a tool employed for testing HTTP DoS exposures by launching slow HTTP POST or GET requests.

- GoldenEye: GoldenEye includes 41,508 samples, indicating 0.256% of the total records. GoldenEye is a DDoS tool that targets web servers by flooding them with TCP SYN packets.
- Slowloris: With 10,990 samples, Slowloris takes 0.068% of the total records. Slowloris is a DoS attack tool that targets web servers by multiple connections kept open for longest possible time, exhausting server resources.
- LOIC-UDP: This category gets 1,730 samples, with 0.011% of the total records. LOIC-UDP is a variation of the Low Orbit Ion Cannon tool that implements UDP-based DDoS attacks.
- Brute Force—Web: There are 611 samples here, demonstrating 0.004% of the total records. Brute Force—Web indicates brute force attacks pursuing web apps.
- Brute Force—XXS: This type consist of 230 samples, with 0.001% of the total records. Brute Force—XXS is a brute force attack focusing on cross-site scripting vulnerabilities in web apps.
- SQL Injection: With 87 samples, SQL Injection signifies 0.0005% of the total records. SQL Injection is very usual attack responsible for exploitation of vulnerabilities in web apps for executing malicious SQL queries.

In order to handle the imbalance a class-weighted categorical cross-entropy loss function was itself employed. It ensured that:

- Under-represented attack categories supported considerably to model learning, targeting no bias towards frequent attack.
- Over-represented attacks had lesser loss weights, guaranteeing the network did not overly favor them.

3- Deep learning Pipeline

3-1- Data pre-processing

Data cleaning involves identifying and rectifying errors, inconsistencies, and inaccuracies in datasets to ensure their quality and reliability. This process greatly impacts the validity and credibility of research findings and statistical analyses. Consequently, it is essential to carefully and systematically perform data cleaning to enhance data integrity and minimize the potential for biased or misleading conclusions. Furthermore, adhering to best practices and employing appropriate software tools can streamline and facilitate the data cleaning process, leading to more robust and accurate research outcomes. Commence with the unprocessed data as the primary dataset. To initiate the preprocessing workflow, it is necessary to commence with the data cleaning process. The initial cleaning process facilitates dimensionality reduction, ultimately providing various benefits. In the current analytical study, the inclusion of the time parameter is deemed unnecessary, and any columns consisting solely of zero values are excluded due to their lack of influence on the final result. Following completion of the cleaning procedure, a total of 11 columns were excluded from the initial set of 80 columns, leaving 69 columns remaining. A few fields were eliminated from the dataset before features were chosen. Metrics like 'Bwd_Avg_Bulk_Rate', 'Fwd_Avg_Bytes_Bulk', 'Bwd_Avg_Packets_Bulk', and 'Fwd_Avg_Bulk_Rate' are associated with bulk transfer rates and packet sizes. Furthermore, flags like 'Bwd_PSH_Flags' and 'Bwd_URG_Flags' that indicated Push (PSH) and Urgent (URG) actions in forward and backward packets were removed. For simplifying the dataset and concentrating on features more pertinent to the current task, some fields were probably removed. This could increase the efficacy and efficiency of later feature selection algorithms and machine learning models. Eliminating these fields makes the information easier to handle and could improve the intrusion detection system's accuracy and interpretability.

3-2- Feature Extraction

The process of feature extraction plays a crucial role in the initial stages of our data preprocessing. During this phase, we meticulously curate and convert pertinent attributes from the extensive pool of 80 features present in the CSE-CIC-IDS2018 dataset. This procedure plays a vital role in optimizing the dataset and identifying the most informative attributes for our analysis of intrusion detection. In this article, we present a succinct compilation of characteristics, each accompanied by a succinct explanation. The parameter "Flow Duration" (fl_dur) quantifies the length of time that a network flow persists, thereby offering valuable observations regarding the lifespan of network-based operations. The metric "Total Packets in the Forward Direction" (tot_fw_pk) denotes the aggregate count of transmitted packets in the forward direction, serving as a pivotal indicator for analysis of network traffic. The variable "Total Packets in the Backward Direction" (tot_bw_pk) corresponds to the number of packets that flow in the opposite direction. It bears resemblance to the previously discussed variable "Total Packets in the Forward Direction" (tot_fw_pk). The Average Time Between Flows (fl_iat_avg) parameter serves to measure the mean duration between consecutive network flows, contributing to the examination of flow timing. The fw_psh_flag metric quantifies the frequency at which the Push (PSH) flag in forward direction packets is enabled, thereby bearing significance in comprehending the dynamics of data transmission. The parameter "pkt_len_min" corresponds to the minimum length observed in a data flow, which serves as a significant metric in assessing the magnitude of data being transferred. The Download and Upload Ratio (down_up_ratio) is a measurement that quantifies the proportion of download activities to upload activities, providing insights into network usage patterns. The variable "atv_max" denotes the maximum duration of flow activity prior to transitioning into an idle state. This subset of carefully chosen features is a selection from the dataset's available 80 attributes, determined based on their relevance to the task of intrusion detection. The aforementioned feature extraction process plays an integral role in enhancing the performance of the model and enabling it to accurately differentiate between benign and malicious network traffic. The identified features are anticipated to make a substantial contribution to our analytical and categorization endeavors, ultimately bolstering the overall level of network security.

3-3- Feature Selection

Feature selection using Information Gain (IG) is a widely utilized method within the decision tree framework to discern and preserve the most informative features pertinent to classification or regression tasks. The concept of Information Gain pertains to quantifying the decrease in uncertainty, as represented by entropy, attained through the division of a dataset by a specific attribute. Features that result in a substantial decrease in entropy are regarded as possessing a higher degree of information. The subsequent step-by-step guide delineates the procedure in a systematic manner. To ascertain the Entropy of the Target Variable, rigorous calculations are required. To commence the process, it is pertinent to compute the entropy of the target variable, which refers to the variable under consideration that is sought to be predicted. Entropy is a quantitative metric used to quantify the degree of impurity or randomness present within the target variable. The entropy of the target variable should be calculated for each feature. The entropy of the target variable should be computed for each feature by partitioning the dataset according to that specific feature. This metric essentially quantifies the effectiveness of a feature in partitioning the data into distinct classes. The purpose of this exercise is to determine the value of information gain. Information Gain (IG) is determined by subtracting the entropy of the initial target variable from the weighted average of the entropies of the target variable for each partition based on the feature at hand. The information gain (IG) is calculated as the difference between the entropy before the splitting operation and the weighted average of the entropies after the splitting operation. Using a decision tree to choose features led to the selection of several feature sets for intrusion detection. Key characteristics found are ‘forward active data packets’, ‘forward segment size minimum’, ‘backward packets per second’, ‘forward inter-arrival time minimum’, and ‘destination port’. A slightly different set of features, on the other hand, were given priority by calculating Gini index. These features included ‘destination port’, ‘forward packet length minimum’, ‘flow packets per second’, ‘backward packets per second’, ‘forward inter-arrival time minimum’, ‘count of the ACK flag’, ‘count of the explicit congestion notification (ECE) flag’, ‘forward segment size minimum’, and ‘forward active data packets’. By efficiently reducing the feature space to the most pertinent characteristics for intrusion detection, these techniques may improve the precision and effectiveness of later machine learning models.

3-4- Feature Classification using HCARN

The Hybrid Convolutional-Attention Recurrent Network (HCARN) is an innovative architecture designed to improve threats detection. By incorporating convolutional layers, attention mechanisms, and recurrent neural networks, HCARN take advantage of the strengths of each component in delivering superior performance while diagnosing the attacks. This section explains the architecture, components, and the rationale behind the design choices of HCARN.

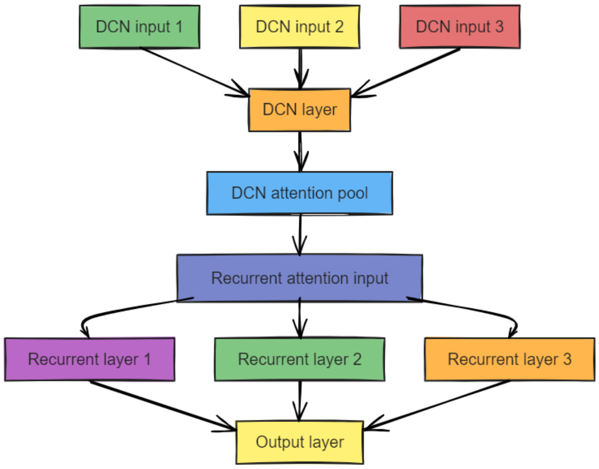


Fig. 4 HCARN architecture

Architecture Overview

HCARN is designed to address the constraints of traditional CNNs in cybersecurity threat detection by utilizing advanced features like attention mechanisms and recurrent layers as shown in Figure 4. The architecture is composed of the key components as stated herein.

- Input Layer: The input layer receives dataset with 79 features per sample, reshaped to meet the requirements of the convolutional layers.
- Convolutional Layers: These layers are accountable for initial feature extraction. Convolutional layers with ReLU activation capture spatial patterns. The residual connections are then used to mitigate the vanishing gradient challenge and enhance learning.

- Attention Mechanism: The attention mechanism focuses on the most significant parts of the input sequence, improving the categorization accuracy by emphasizing on critical features.
- Recurrent Layers: The Bidirectional Long Short-Term Memory (BiLSTM) layers capture temporal dependencies, making the architecture adept at recognizing sequences of events in network.
- Fully Connected Layers: Dense layers incorporated with ReLU activation function then process the extracted features. Dropout layers take care of overfitting.
- Output Layer: The final layer has softmax activation function to categorize the input into one of the three threat sets: major, moderate, and minor threats.
- Weighted Loss Function: The dataset, particularly on 20th February 2018, had a disproportionately highest samples, which could have directed the network to favor dominant category. Thus, to avoid this bias, a weighted categorical cross-entropy loss function is employed during training of HCARN which assigned higher weights to under-represented observations and lesser weights to over-represented ones. This class-weighted loss function is an external training element applied to adjust the loss contribution of each category, guaranteeing that under-represented categories had a robust impact while model being training.

Detailed Component Description

- Convolutional Layers: The convolutional layers in HCARN are devised to extract local features from the input dataset. The network begins with a Conv1D layer with 64 filters and a kernel size of 3, batch normalization and max pooling one after the other. A residual connection is included to retain the original input, which stabilizes the learning process. Subsequent convolutional layers increase the number of filters, improves complex patterns recognition.
- Attention Mechanism: The presence of a multi-head attention mechanism allows HCARN to dynamically weigh the position of several features in the input sequence. This mechanism focuses on the most critical parts of the data, which is principally effective in recognizing subtle yet substantial anomalies in network traffic.
- Recurrent Layers: HCARN further incorporates BiLSTM layers to capture temporal dependencies. The bidirectional nature ensures that the network can learn from both past and future instances within the series, presenting a comprehensive interpretation of the temporal dynamics in network traffic. The detailed architecture of HCARN is stated in Table 3.

Table 3: HCARN architecture designed

Layer (Type)	*Output*	*Parameters*	*Description*
Input	(None, 79, 1)	0	Input data with 79 features per sample.
Reshape	(None, 79, 1)	0	Reshape input to fit Conv1D layers.
Conv1D	(None, 79, 64)	256	64 filters, kernel size 3, ReLU activation.
Batch Norm	(None, 79, 64)	256	Batch normalization for stability.
MaxPooling1D	(None, 39, 64)	0	Max pooling with pool size 2.
Residual Add	(None, 39, 64)	0	Residual connection to stabilize learning.
Conv1D	(None, 39, 128)	24,704	128 filters, kernel size 3, ReLU activation.
Batch Norm	(None, 39, 128)	512	Batch normalization for stability.
MaxPooling1D	(None, 19, 128)	0	Max pooling with pool size 2.
Attention	(None, 19, 128)	0	Multi-head attention mechanism.
Residual Add	(None, 19, 128)	0	Residual connection for stability.
Conv1D	(None, 19, 256)	98,560	256 filters, kernel size 3, ReLU activation.
Batch Norm	(None, 19, 256)	1,024	Batch normalization for stability.
MaxPooling1D	(None, 9, 256)	0	Max pooling with pool size 2.
BiLSTM	(None, 9, 256)	394,240	Bidirectional LSTM with 128 units.
Residual Add	(None, 9, 256)	0	Residual connection for stability.
Flatten	(None, 2,304)	0	Flattening the data for dense layers.
Dense	(None, 256)	589,440	Fully connected 256 units, ReLU activation.
Dropout	(None, 256)	0	Dropout with rate 0.5 to prevent overfitting.
Dense	(None, 128)	32,896	Fully connected 128 units, ReLU activation.
Dropout	(None, 128)	0	Dropout with rate 0.5 to prevent overfitting.
Output	(None, 3)	387	Fully connected layer with 3 units (for major, moderate, minor threats), Softmax activation.

- Fully Connected Layers: After feature extraction and sequence modeling, the dataset is flattened and sent through fully connected dense layers. These layers perform classification with high-level reasoning. Dropout layers with a 0.5 dropout rate randomly deactivates neurons during training which are purposefully employed to reduce overfitting.

- Output Layer: The final layer of HCARN is a dense layer with softmax activation function, which outputs probabilities for each of the three threat categories – major, moderate, and minor. This probabilistic output helps in confident classification of threats.

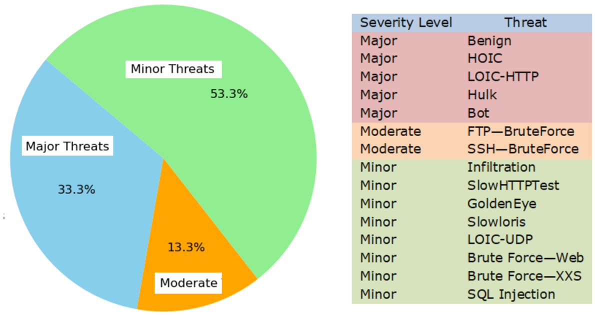
The pseudocode outlining the HCARN model's pipeline into phases such as preprocessing, training, and classification is presented in Annexure I.

Advantages of HCARN

- Hybrid Architecture: The hybrid architecture of HCARN, which integrates convolutional layers, attention mechanisms, and recurrent layers, utilizes the strengths of each component. Convolutional layers effectively capture spatial features in the dataset, attention mechanisms highlight critical features, and recurrent layers model temporal dependencies. This allows HCARN to process and analyze complex patterns in network traffic, leading to more accurate threat diagnosis.
- Scalability and Adaptability: HCARN is inherently scalable, thus, can be applied to large and complex datasets with no worries about significant performance degradation. Its adaptability to different types of threats and capability to maintain high performance across various metrics make it appropriate for a wider range of cybersecurity applications.
- Enhanced Feature Representation: The employment of attention mechanisms enhances feature representation by focusing on the most relevant part. This is particularly beneficial in intrusion detection, where critical features might be sparse and dispersed throughout. By emphasizing these important features, HCARN can perform even better.

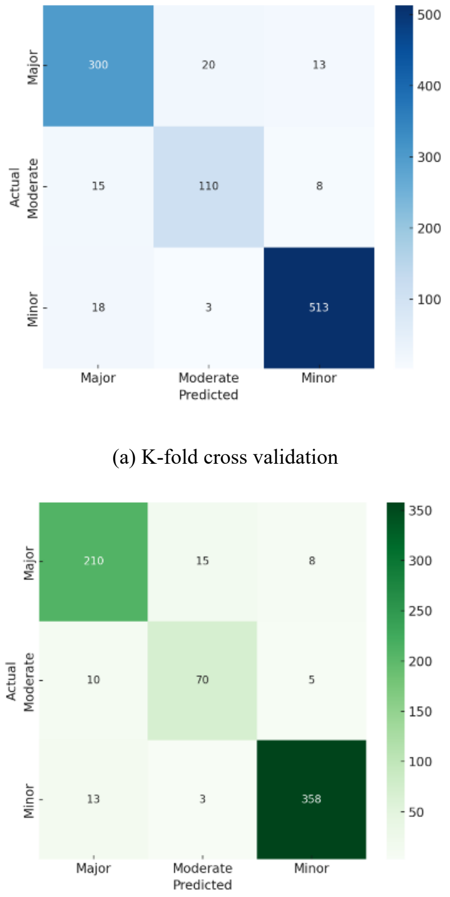
4- Results & Discussion

In this section, the performance of the Hybrid Convolutional-Attention Recurrent Network model on the CSE-CIC-IDS2018 dataset is presented and discussed. Figure 5 illustrate the distribution of threats in the CSE-CIC-IDS2018 dataset, which groups threats into levels i.e., major (33.3%), moderate (13.3%), and minor (53.3%). Major level includes high impact attacks making the model to precisely distinguish between normal and malicious traffic to avoid significant disruptions. Moderate level has FTP and SSH brute force attacks, demand a balanced detection method to prevent false positives and negatives. Minor level has various denial of service (DoS) attacks and infiltration methods, though less severe individually, dominate the dataset and require the model to maintain high precision and recall in effectively manage the frequent occurrences. This distribution impacts the performance, demonstrating the HCARN model's robustness and success in handling a wider range of cybersecurity threats.



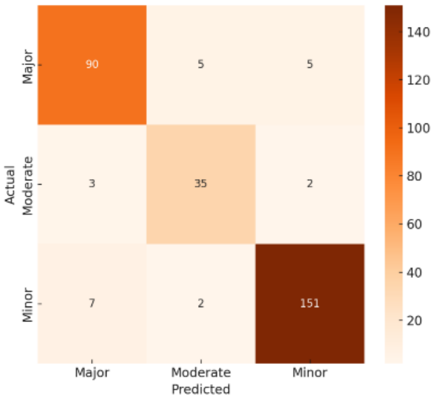
Severity Level	Threat
Major	Benign
Major	HOIC
Major	LOIC-HTTP
Major	Hulk
Major	Bot
Moderate	FTP—BruteForce
Moderate	SSH—BruteForce
Minor	Infiltration
Minor	SlowHTTPTest
Minor	GoldenEye
Minor	Slowloris
Minor	LOIC-UDP
Minor	Brute Force—Web
Minor	Brute Force—XXS
Minor	SQL Injection

Fig. 5 Distribution of traffic classes by threat level



(a) K-fold cross validation

(b) Training with 70% samples



(c) Testing with 30% samples

Fig. 6 Confusion matrix depicting classification of different threats

4-1- Observations & Discussion

The performance parameter includes precision, recall, F1-score, specificity, Matthews Correlation Coefficient (MCC), accuracy, ROC-AUC, average precision (AP) and confusion matrices for each threat category. The results are derived from splits for training on 70% dataset, testing 30% and k-fold cross-validation. The confusion matrices in Figure 6 provide a detailed view of the classification results across major, moderate, and minor threats. The relatively low misclassification in each phase underlines the model's accuracy and reliability. They also highlight areas where the model could potentially amend, such as cutting the number of moderate threats misclassified in major or minor levels.

Table 4: Detailed performance of classification

Split	*Class*	*F1-Score*	*Specificity*	*MCC*	*Accuracy*	*ROC-AUC*	*AP*
K-Fold	Major	0.90	0.95	0.81	94.7	0.93	0.92
	Moderate	0.85	0.97	0.79		0.93	0.92
	Minor	0.96	0.96	0.92		0.96	0.97
Train	Major	0.92	0.95	0.81	95.4	0.93	0.93
	Moderate	0.82	0.97	0.75		0.92	0.91
	Minor	0.96	0.97	0.93		0.97	0.97
Test	Major	0.88	0.93	0.74	92.3	0.91	0.91
	Moderate	0.85	0.97	0.78		0.91	0.91
	Minor	0.94	0.94	0.88		0.95	0.95

Table 5 summarizes the key performance metrics including precision, recall, F1-score, specificity, MCC, accuracy, ROC-AUC, and AP. The HCARN model presents high precision and recall across major, moderate, and minor levels, indicating its ability to correctly identify and classify threats. High precision confirms that most identified threats are at actual level, lowering the incidence of false alarms which overwhelm security analysts. High recall confirms that the model catches most actual threats, reducing the risks of missed attacks which could lead to possible breaches. The F1-score balances precision and recall, is especially high for all threat classes, indicating that the HCARN model maintains an excellent trade-off between these two critical parameters. This balance is crucial for practical cybersecurity domain where both false positives and false negatives can have serious concerns.

High specificity values indicate that the model is adept at appropriately identifying non-threats, reducing the probability of false positives. The Matthews Correlation Coefficient (MCC), a comprehensive measure of the quality of binary categorization, further supports the model's effectiveness. High MCC scores across all classes authorize that the model performs well across different types of threats, offering a balanced measure that reflects on all four confusion matrix possibilities (true positives, false positives, true negatives, and false negatives) as stated in Table 5. The ROC curves for each class shown in Figure 7 illustrate the trade-off between the true positive rate and false positive rate. The area under the curve (AUC) value indicates strong discriminatory power for all classes. The Precision-Recall curves shown in Figure 8 highlight the balance amongst precision and recall for different thresholds.

Table 5: Detailed parameters from confusion matrices

Threats	*False Positive (%)*	*False Negative (%)*	*True Positive (%)*	*True Negative (%)*
		K fold cross validation		
Major	300	33	33	634
Moderate	110	23	15	852
Minor	513	21	21	445
		Training with 70% samples		
Major	210	23	15	452
Moderate	70	18	13	599
Minor	358	11	16	315
		Testing with 30% samples		
Major	90	14	10	186
Moderate	35	7	5	253
Minor	151	9	9	131

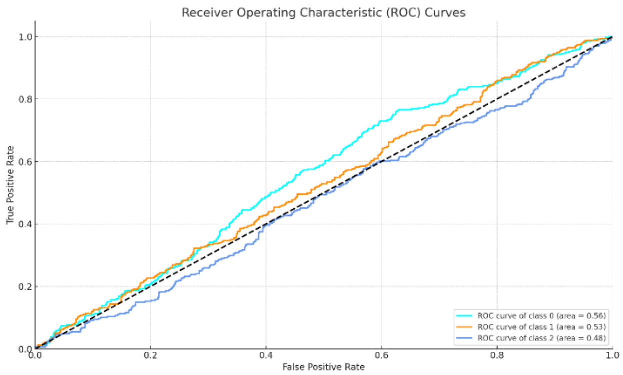


Fig. 7 Receiver Operating Characteristic (ROC) Curves

4-2- Comparative Performance

When compared to conventional IDS models, HCARN extends several advantages as shown in Table 1. Conventional models often rely on hand-crafted features and shallow learning approaches, which may not successfully obtain the complex and evolving nature of modern cyber threats. In contrast, HCARN's deep learning method allows it to automatically learn and extract features from raw data, leading to superior execution. The results from the k-fold cross-validation, training, and testing phases indicate that HCARN consistently outperforms in terms of precision, recall, F1-score, specificity, and overall accuracy. This consistent execution across different data splits and threat levels underscores HCARN's reliability and robustness.

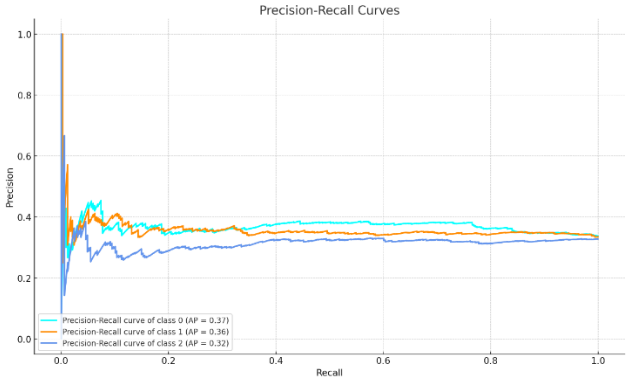


Fig. 8 Precision-Recall Curves

The table 6 summarizes the models used, their reported accuracies, and how they compare to the proposed HCARN model. Unlike traditional models that classify attacks based on specific types, the HCARN model categorizes threats into Major, Moderate, and Minor levels, improving generalization and scalability. It leverages a hybrid architecture combining CNN, Attention, and BiLSTM layers, which allows it to efficiently capture spatial, temporal, and contextual dependencies in network traffic. Additionally, it reduces computational complexity compared to high-dimensional multi-class models, making it more suitable for real-time intrusion detection.

Table 6: Comparison of various ML models applied on CSE-CIC-IDS2018 dataset

Study	*Model(s) Used*	*Accuracy*
R. I. Farhan et al. [30]	DNN	90.25%
A. Elhanashi et al. [31]	Random Forest, GaussianNB, and multilayer perceptron	85.70%
J. Kim et al. [32]	CNN and RNN	93.00%
M. Mayuranathan et al. [33]	LSTM-SGDM	66.38%
Proposed HCARN	**Hybrid Convolutional-Attention Recurrent Network**	**95.40%**

4-3- Practical Implications

The high performance of HCARN has significant practical implications for cybersecurity operations. By accurately detecting and diagnosing threats, HCARN can decrease the workload on security analysts, assisting them to emphasis on the most critical concerns. Its high precision and low false positive rate reduce the occurrence of false alarms, leading to efficient threat management. Moreover, HCARN's ability to adapt to diverse threats ensures that it remains effective in dynamic and evolving threat environments. This adaptability is crucial for modern cybersecurity conditions, where new and sophisticated attack vectors are constantly arising.

5- Conclusions

This study introduced the hybrid convolutional-attention recurrent network, a novel architecture that leverages a combination of convolutional layers, attention mechanisms, and recurrent layers to effectively addresses the limitations of traditional CNN-based models. The proposed HCARN model demonstrated significant efficacy in distinguishing between major, moderate, and minor threats, achieving high accuracy and robustness in threat diagnosis. The attention mechanism enabled the model to prioritize relevant features, enhancing its ability to identify subtle yet substantial anomalies. Meanwhile, the recurrent layer ensured the model comprehends the temporal dynamics of network events, providing a widespread threat diagnosis framework. Extensive assessment through k-fold cross-validation, training, and testing phases showed the model's consistent performance and low false positive rates. The combination of residual connections and dropout layers further strengthened the model by mitigating overfitting and steadying the training process. Overall, HCARN represents a considerable advancement in cybersecurity threat diagnosis. The novel combination of convolutional, attention, and recurrent layers within a single framework underscored the capability of hybrid deep learning algorithms in designing adaptive security systems. This investigation not only demonstrated the efficacy of HCARN in enhancing cybersecurity defenses but also paves the way for future research and development in this critical area. While the current findings are promising, there are several opportunities for future work to further enhance performance and applicability. Optimization of the HCARN architecture by experimenting with different configurations of convolutional, attention, and recurrent layers can be undertaken. Implementing data augmentation techniques to synthetically expand the dataset can aid the model generalize better to blind data. Developing real-time execution framework for HCARN could enable its operation in live cybersecurity environments. This involves optimizing the model for low-latency predictions and incorporating it with present cybersecurity infrastructure. Future research can be directed towards detecting multi-stage attacks for understanding how minor attacks escalate into critical ones for strategies aimed at early mitigation. In addition, the adaptive learning will allow the network to update dynamically on its own and increase its capacity to identify zero-day threats exclusive of full retraining. Furthermore, federated learning will be investigated to assist collaborative training while guaranteeing data privacy in distributed security circumstances. To boost real-time efficiency, efforts to be made for optimizing latency and computational cost in high-speed networks.

Annexure I

Pseudocode: Hybrid Convolutional-Attention Recurrent Network (HCARN)

```
Start
Input: Network traffic dataset $\mathcal{D}$ (CSE-CIC-IDS2018)
Output: Predicted threat category $\mathcal{C} \in$ {Major, Moderate, Minor}
Step 1: Data Preprocessing
  $\mathcal{D} \leftarrow$ Load dataset
  $\forall x \in \mathcal{D}$: If $x$ contains NaN, remove or impute missing values
  $\forall x \in \mathcal{D}$: Normalize features $\rightarrow x' = (x - \min(x)) / (\max(x) - \min(x))$
  $\mathcal{K} \leftarrow$ Select top k features using Information Gain
  $\{X_{tr}, y_{tr}\}, \{X_{val}, y_{val}\}, \{X_{test}, y_{test}\} \leftarrow$ Split dataset (70%-Training, 30%-Testing)
Step 2: Define HCARN Model $\mathcal{M}$
  $\mathcal{M} \leftarrow$ Initialize input layer $I \in \mathbb{R}^{79}$
  # Convolutional Feature Extraction
  $C_1 \leftarrow$ Conv1D($I$, $F_1$=64, $K_1$=3, activation=ReLU)
  $C_1 \leftarrow$ BatchNorm($C_1$), MaxPool($C_1$, $P_1$=2)
  $R_1 \leftarrow$ Add($I$, $C_1$)  # Residual Connection
  $C_2 \leftarrow$ Conv1D($R_1$, $F_2$=128, $K_2$=3, activation=ReLU)
  $C_2 \leftarrow$ BatchNorm($C_2$), MaxPool($C_2$, $P_2$=2)
  $R_2 \leftarrow$ Add($R_1$, $C_2$)  # Residual Connection
  $C_3 \leftarrow$ Conv1D($R_2$, $F_3$=256, $K_3$=3, activation=ReLU)
  $C_3 \leftarrow$ BatchNorm($C_3$), MaxPool($C_3$, $P_3$=2)
  # Attention Mechanism
  $A \leftarrow$ MultiHeadAttention($C_3$, $h$=4, $k_e y$=64)
  $R_3 \leftarrow$ Add($C_3$, $A$)  # Residual Connection
  # Temporal Dependency Learning
  $H \leftarrow$ BiLSTM($R_3$, $u$=128, bidirectional=True)
  # Fully Connected Layers
  $H' \leftarrow$ Flatten($H$)
  $D_1 \leftarrow$ Dense($H'$, $u_1$=256, activation=ReLU)
  $D_1 \leftarrow$ Dropout($D_1$, $p$=0.5)
  $D_2 \leftarrow$ Dense($D_1$, $u_2$=128, activation=ReLU)
  $D_2 \leftarrow$ Dropout($D_2$, $p$=0.5)
  # Output Layer
  $\mathcal{C} \leftarrow$ Softmax($D_2$, $u$=3)
Step 3: Model Compilation and Training
  $L \leftarrow$ Weighted Categorical Cross-Entropy Loss
  $O \leftarrow$ Adam(learning rate=0.001)
  $\forall e \in [1, N]$:  # Training for N epochs
    $\forall B \in X_{tr}$:  # Mini-batch training
      $B' \leftarrow$ Forward($B$, $\mathcal{M}$)
      $l \leftarrow L(B', y_{tr})$
      Backpropagate($l$, $O$)
      Update($\mathcal{M}$, $O$)
    If Validation Loss Converges:
      Break training
Step 4: Model Evaluation
  $\hat{y}_{test} \leftarrow$ Predict($X_{test}$, $\mathcal{M}$)
  Compute:
    $\mathcal{A}cc$ = Accuracy($\hat{y}_{test}$, $y_{test}$)
    $\mathcal{P}$ = Precision($\hat{y}_{test}$, $y_{test}$)
    $R$ = Recall($\hat{y}_{test}$, $y_{test}$)
    $F_1$ = F1-score($\hat{y}_{test}$, $y_{test}$)
    $ROC$ = ROC-AUC($\hat{y}_{test}$, $y_{test}$)
  Generate Confusion Matrix
Step 5: Deployment for Real-Time Threat Detection
  $\forall x \in$ Incoming_Network_Traffic:
    $x' \leftarrow$ Normalize($x$)
    $\mathcal{C}' \leftarrow$ Predict($x'$, $\mathcal{M}$)
    Output Threat Class: $\mathcal{C}' \in$ {Major, Moderate, Minor}
End
```

Abbreviations and symbols

$\mathcal{D}$ = Input dataset
X_{tr}, X_{test}, X_{val} = Training, Testing, Validation Sets
I = Input Layer (79 features)
C_1, C_2, C_3 = Convolutional Layers
R_1, R_2, R_3 = Residual Connections
A = Multi-Head Attention Layer
H = BiLSTM Layer

D_1, D_2 = Fully Connected Layers
$\mathcal{C}$ = Softmax Output (Threat Classes)
L = Loss Function
O = Optimizer (Adam)
l = Computed Loss
$\mathcal{Acc}$, $\mathcal{P}$, R, F_1, ROC = Performance Metrics

References

[1] M. Markevych and M. Dawson, “A review of enhancing intrusion detection systems for cybersecurity using artificial intelligence (AI),” in *International Conference Knowledge-Based Organization*, vol. 29, no. 3, pp. 30–37, July 2023.

[2] A. Dunmore, J. Jang-Jaccard, F. Sabrina, and J. Kwak, “A comprehensive survey of generative adversarial networks (GANs) in cybersecurity intrusion detection,” *IEEE Access*, 2023.

[3] J. M. Storm, J. Hagen, and Ø. A. A. Toftegaard, “A survey of using process data and features of industrial control systems in intrusion detection,” in *2021 IEEE International Conference on Big Data (Big Data)*, Dec. 2021, pp. 2170–2177.

[4] B. J. Asaju, “Advancements in Intrusion Detection Systems for V2X: Leveraging AI and ML for Real-Time Cyber Threat Mitigation,” *Journal of Computational Intelligence and Robotics*, vol. 4, no. 1, pp. 33–50, 2024.

[5] S. Alzughaibi and S. El Khediri, “A cloud intrusion detection systems based on DNN using backpropagation and PSO on the CSE-CIC-IDS2018 dataset,” *Applied Sciences*, vol. 13, no. 4, p. 2276, 2023.

[6] L. Göcs and Z. C. Johanyák, “Identifying relevant features of CSE-CIC-IDS2018 dataset for the development of an intrusion detection system,” *Intelligent Data Analysis*, preprint, 2023.

[7] H. Najafi Mohsenabad and M. A. Tut, “Optimizing cybersecurity attack detection in computer networks: A comparative analysis of bio-inspired optimization algorithms using the CSE-CIC-IDS2018 dataset,” *Applied Sciences*, vol. 14, no. 3, p. 1044, 2024.

[8] L. Göcs and Z. C. Johanyák, “Feature selection with weighted ensemble ranking for improved classification performance on the CSE-CIC-IDS2018 dataset,” *Computers*, vol. 12, no. 8, p. 147, 2023.

[9] S. Songma, T. Sathuphan, and T. Pamutha, “Optimizing intrusion detection systems in three phases on the CSE-CIC-IDS2018 dataset,” *Computers*, vol. 12, no. 12, p. 245, 2023.

[10] M. Khan and M. Haroon, “Artificial neural network-based intrusion detection in cloud computing using CSE-CIC-IDS2018 datasets,” in *2023 3rd Asian Conference on Innovation in Technology (ASIANCON)*, Aug. 2023, pp. 1–4.

[11] B. I. Farhan and A. D. Jasim, “Performance analysis of intrusion detection for deep learning model based on CSE-CIC-IDS2018 dataset,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 26, no. 2, pp. 1165–1172, 2022.

[12] C. F. Tsai and C. Y. Lin, “A triangle area based nearest neighbors approach to intrusion detection,” *Pattern Recognition*, vol. 43, no. 1, pp. 222–229, 2010.

[13] R. Bar-Yanai, M. Langberg, D. Peleg, and L. Roditty, “Realtime classification for encrypted traffic,” in *Proceedings of the International Symposium on Experimental Algorithms*, Springer, Berlin, Heidelberg, May 2010, pp. 373–385.

[14] W. C. Lin, S. W. Ke, and C. F. Tsai, “CANN: An intrusion detection system based on combining cluster centers and nearest neighbors,” *Knowledge-Based Systems*, vol. 78, pp. 13–21, 2015.

[15] P. M. Comar, L. Liu, S. Saha, P. N. Tan, and A. Nucci, “Combining supervised and unsupervised learning for zero-day malware detection,” in *Proceedings of the 2013 IEEE INFOCOM*, Apr. 2013, pp. 2022–2030.

[16] A. A. Aburomman and M. B. I. Reaz, “A novel SVM-kNN-PSO ensemble method for intrusion detection system,” *Applied Soft Computing*, vol. 38, pp. 360–372, 2016.

[17] J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-an, and H. Ye, “Significant permission identification for machine-learning-based android malware detection,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3216–3225, 2018.

[18] S. M. H. Bamakan, H. Wang, and Y. Shi, “Ramp loss K-support vector classification-regression; a robust and sparse multi-class approach to the intrusion detection problem,” *Knowledge-Based Systems*, vol. 126, pp. 113–126, 2017.

[19] E. De la Hoz, A. Ortiz, J. Ortega, and B. Prieto, “PCA filtering and probabilistic SOM for network intrusion detection,” *Neurocomputing*, vol. 164, pp. 71–81, 2015.

[20] S. Dubey and J. Dubey, “KBB: A hybrid method for intrusion detection,” in *Proceedings of the 2015 International Conference on Computer, Communication and Control (IC4)*, Sept. 2015, pp. 1–6.

[21] M. Jabbar, R. Aluvalu, et al., “RFAODE: A novel ensemble intrusion detection system,” *Procedia Computer Science*, vol. 115, pp. 226–234, 2017.

[22] N. Moustafa, B. Turnbull, and K. K. R. Choo, “An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things,” *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4815–4830, 2018.

[23] H. H. Pajouh, R. Javidan, R. Khayami, D. Ali, and K. K. R. Choo, “A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks,” *IEEE Transactions on Emerging Topics in Computing*, vol. 7, no. 2, pp. 314–323, 2016.

[24] Z. Tan, A. Jamdagni, X. He, P. Nanda, and R. P. Liu, “A system for denial-of-service attack detection based on multivariate correlation analysis,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 447–456, 2013.

[25] N. Liao, S. Tian, and T. Wang, “Network forensics based on fuzzy logic and expert system,” *Computer Communications*, vol. 32, no. 17, pp. 1881–1892, 2009.

[26] R. Masoudi and A. Ghaffari, "Software Defined Networks: A Survey," *Journal of Information Systems and Telecommunication*, vol. 67, no. 5, pp. 1–25, 2016.

[27] A. Shirmarz and A. Ghaffari, "Autonomic Software Defined Network (SDN) Architecture With Performance Improvement," *Journal of Information Systems and Telecommunication*, vol. 8, no. 2, pp. 120-128, April-June 2020.

[28] A. Shirmarz and A. Ghaffari, "A Novel SDN-Based Architecture for Distributed Denial-of-Service (DDoS) Detection," *Journal of Information Systems and*

Telecommunication, vol. 10, no. 2, pp. 120-131, April-June 2022.

[29] Canadian Institute for Cybersecurity. (2018). CSE-CIC-IDS2018: A Large-Scale Dataset for Intrusion Detection Systems. Retrieved from https://registry.opendata.aws/cse-cic-ids2018/

[30] Farhan, R. I., Maolood, A. T., & Hassan, N. (2020). Performance analysis of flow-based attacks detection on CSE-CIC-IDS2018 dataset using deep learning. *Indones. J. Electr. Eng. Comput. Sci*, *20*(3), 1413-1418.

[31] Elhanashi, A., Gasmi, K., Begni, A., Dini, P., Zheng, Q., & Saponara, S. (2022, September). Machine learning techniques for anomaly-based detection system on CSE-CIC-IDS2018 dataset. In *International Conference on Applications in Electronics Pervading Industry, Environment and Society* (pp. 131-140). Cham: Springer Nature Switzerland.

[32] Kim, J., Kim, J., Kim, H., Shim, M., & Choi, E. (2020). CNN-based network intrusion detection against denial-of-service attacks. *Electronics*, *9*(6), 916.

[33] Mayuranathan, M., Saravanan, S. K., Muthusenthil, B., & Samydurai, A. (2022). An efficient optimal security system for intrusion detection in cloud computing environment using hybrid deep learning technique. *Advances in Engineering Software*, *173*, 103236.