Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

An Approach for Could IDS Based on Deep Learning technique to Novel Optimized ANCNNED for Secure Intrusion Detection Method

Dr.Faisal Nabi

faisal.nabi@yahoo.com

De Monfort University

Carsten Maple

University of Warwick

H M Attaullah

Maju

Research Article

Keywords: Intrusion Detection Systems (IDS), DC-Res2Net, Genghis Khan Shark Optimizer (GKSO), ANCNNED, Enhanced Zebra Optimization (EZO)

Posted Date: March 25th, 2025

DOI: https://doi.org/10.21203/rs.3.rs-6289788/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare potential competing interests as follows: NON

An Approach for Could IDS Based on Deep Learning technique to Novel Optimized ANCNNED for Secure Intrusion Detection Method

Dr.Faisal Nabi ,Dr.Carsten Maple ,Hafiz Muhammad Attaullah

Demonford University ,Department of Computer Science ,Great Britain faisal.nabi@yahoo.com
Warwick University ,Department of Computer Science ,Great Britain
Muhammad Ali Jinnah University, Department of Computer Science ,Great Britain

Abstract:

The spread adoption of cloud computing resulted from recent improvements but security concerns persist because of how cloud environments operate across multiple nodes. The analysis of heterogeneous network data through IDS in cloud networks generates high false positive rates together with low accuracy and overfitting problems. This paper introduces an optimized security framework for cloud-based intrusion detection which uses an Ensemble Optimized deep learning method.The IDS framework operates through a comprehensive process starting from data collection and proceeding through preprocessing and feature extraction and selection and ending with detection and classification followed by performance evaluation. Analysis based on DC-Res2Net extracts features before GKSO optimizes the selection process. The Novel ANCNNED approach uses an ensemble of materialized by AlexNet CNN and EfficientDet to conduct classificatory operations. The detection system employs the Enhanced Zebra Optimization algorithm for optimization purposes.

Keywords: Intrusion Detection Systems (IDS), DC-Res2Net, Genghis Khan Shark Optimizer (GKSO), ANCNNED, Enhanced Zebra Optimization (EZO),

1. Introduction

Computer network development has gained vast improvements during the last several decades. The amount of devices in the network grows rapidly as the network expands, presenting it additional security concerns to network devices [1]. These security concerns have the potential to affect asset confidentiality, integrity, and availability. Therefore, it is critical to defend networks and information systems against potential attacks on the network. IDS has evolved as one of the best alternatives for offering security solutions against a variety of network intrusion threats [2, 3]. Feature-based detection involves investigating and examining existing attacks before encoding the derived attack characteristics into rules or signatures. When network traffic meets the established rules, an alarm will be triggered [4]. Although it may efficiently prevent known dangers, defense effectiveness is significantly diminished when confronted with unknown threats. Furthermore, anomaly-based detection can reduce the impact of unidentified hazards by studying and modelling regular data. It sounds an alarm when it detects anomalous network traffic or deviates from the baseline [5-7].

Several existing models are examined for intrusion detection systems. Although existing models perform better [8-10], they are unable to identify infiltration effectively due to various constraints. Existing models have shortcomings such as taking longer to identify certain types of threats, being inapplicable to large-scale AMI networks, and relying on low-dimensional security data [11, 12]. These difficulties are inspired by the suggested model's ability to suppress limits. Thus, an Efficient Feature Selection assisted NIDS based on Machine Learning is employed to enhance network security and prevent attacks [13-15]. Existing approaches, such as traditional SmoT (Synthetic Minority Oversampling Technique), are incapable of solving two minority closest neighbour samples separated by a long distance and removing outliers [16]. The SVD (Singular Value Decomposition) model extracts key features from input data. Several forms of assaults cannot be easily classified [17] by a single

classifier approach, such as LR, RF, or SVM with huge datasets [18-20]. These constraints are addressed by introducing a novel ensemble of optimal deep learning-based techniques.

1.1 Motivation

The motivation for this research derives from the growing reliance on cloud-based environments and the critical need to defend them against developing cyber threats. As cloud computing expands, it has substantial security concerns due to its distributed nature and enormous volumes of network data, making it an appealing target for intrusion and assaults. Traditional intrusion detection systems (IDS) struggle with difficulties such as overfitting, low accuracy in categorizing, and high false positives, particularly when dealing with diversified and large-scale data in cloud environments. These limitations highlight the critical need for more effective and precise IDS solutions. This study aims to address these challenges by creating an advanced, optimized intrusion detection model that uses deep learning, feature selection techniques, and optimization algorithms to improve detection accuracy, reduce false positives, and improve overall cloud infrastructure security. This paper proposes a novel optimized ensemble model to provide a strong security mechanism capable of efficiently identifying and mitigating modern cyber threats in cloud computing.

The major key contributions of this research are as follows,

- Utilization of an Improved Self-Paced Ensemble Generative Adversarial Networks with Auxiliary Classifier (SPE-ACGAN) to address the data imbalance issue effectively.
- An improved Densely Connected Res2Net (DC-Res2Net) approach is utilized to extract the essential features from the preprocessed data.
- To effectively select important features, the Genghis Khan Shark Optimizer (GKSO) algorithm is employed, reducing dimensionality and computational complexity.
- The proposed Novel Efficient Optimized ANCNNED model is used to classify the attacks which is the ensemble of AlexNet CNN and EfficientDet approach that offers high classification accuracy and effectively detects intrusions and anomalies.
- For improved training optimization, the Enhanced Zebra Optimization (EZO) algorithm is utilized for efficient convergence.
- Evaluation of the proposed approach using the IoTID23, N-BaIoT, ToN-IoT, and CICIoT-2023 datasets, the experimental outcome demonstrating over above 99% in all metrics, establishing the model's robustness in detecting and classifying network intrusions within cloud infrastructures.

Arrangement of research: The research article is organized into the following sections: Part 2 discusses existing papers; Part 3 gives a summary of the proposed methodology; Part 4 provides the findings and analysis; and Part 5 concludes the investigation.

2 Prior Research Review

IDSs have been employed in various studies because they are crucial for protecting computer networks from virtual threats. This paper describes the ML and DL approaches used in previous research for IDS structures.

Aljehane et al. [21] presented a Golden Jackal Optimization Algorithm for Network Security (GJOADL-IDSNS) technology. The GJOADL-IDSNS system's primary goal was to accomplish network security by effectively recognizing and classifying attacks. The GJOA-based selection of characteristics (GJOA-FS) method can be used in the GJOADL-IDSNS strategy provided here to pick an optimal subset of features. This methodology then employs the A-BiLSTM approach.

Table 1: Summary of Literature Review – Methods, Advantages, Limitations, Datasets, and Accuracy

Reference	Method Used	Advantages	Limitations	Dataset Used	Accuracy
Aljehane et al. [21]	GJOADL-IDSNS	Effective recognition and classification of attacks	May require extensive computational resources	NSL-KDD, UNSW, CICiddos	96.54%
Hanafi et al. [22]	IBGJO with	Enhanced GJO	LSTM training can be time-	ToN-IoT,	97.08%

	LSTM	efficiency through opposition-based learning	consuming	BoT-IoT, KDD	
Hazman et al. [23]	IDS-SIoEL	Combines multiple feature selection techniques for robustness	Potential overfitting with ensemble methods	NSL-KDD, UNSW, CiCiddos 2019	98.21%
Sabitha et al. [24]	HNN-FPSO	High detection rate for IoT attacks	May struggle with complex patterns	Bot-IoT, IoT ToN-IoT	99.25%
Kasongo [25]	LSTM, GRU, Simple RNN	XGBoost reduces feature space, improving accuracy	Performance may decrease with increasing feature dimensions	KDD, UNSW, CICiddos	98.74%

The IBGJO method with a Long Short-Term Memory (LSTM) network was introduced by Hanafi et al. [22] to create an IDS framework for IoT networks. In the beginning, OBL improves the GJO. A binary mode of the modified GJO method was employed to choose characteristics from IDS data in order to find the best subset. IBGJO employs the OBL method to increase GJO efficiency and avoid the algorithm from being trapped in local optima by managing the initial population. The IBGJO-LSTM strategy uses LSTM for sample classification. IDS-SIoEL was an IDS framework for Internet of Things-based intelligent settings with ensemble learning, was presented by Hazman et al. [23]. In general, the framework suggested an ideal anomaly detection model that combines various feature selection methods with AdaBoost.

Sabitha et al. [24], developed the AIS model that incorporates the advantages of the Hopfield Neural Network (HNN) for categorization. The weight of the HNN classifier was fixed because its backpropagation property does not allow for weight changes. By employing F-PSO to choose the fixed weight optimally, this research contributes to improving the HNN classifier's accuracy. Next, the classifier model distinguishes between IoT assaults that have a normal signal and a high detection rate.

Kasongo [25] provided an IDS architecture with Machine Learning (ML) approaches. This system makes use of three different kinds of recurrent neural networks, or RNNs: simple RNN, gated RRU, and long-short-term memory (LSTM). Furthermore, when the feature dimension increases, current IDSs have trouble identifying novel threats due to low test accuracy scores.

2.1 Research Gap

The growing use of cloud-based settings has resulted in major technological improvements, but it has also created critical security challenges. The distributed structure of cloud solutions makes them especially vulnerable to many types of assaults, posing significant dangers to user confidentiality and data integrity. As network traffic grows, conventional IDS frequently suffer from high FPR, poor classification accuracy, and overfitting when dealing with diverse and huge datasets. These constraints reduce the efficiency of current IDS solutions in protecting cloud infrastructures. To solve these present limitations, this study provides an optimized intrusion detection model based on an ensemble deep learning methodology, specifically the optimized ANCNNED model, paired with advanced feature selection methods. This study intends to provide a resilient security solution for cloud computing environments by focusing on improving detection accuracy and reducing computational complexity, thereby boosting overall security measures against developing cyber threats.

3 Overview of proposed methodology

Network security is a major concern when sharing data over the Internet, especially with the surge in cyber-attacks in the IoT ecosystem, which are frequently the result of lax security measures. Existing models have shortcomings, such as synthetic sample imbalance, low detection accuracy, vulnerability to zero-day assaults, and a lack of complete taxonomy. To overcome these challenges, this study presents a novel network IDS model based on deep learning techniques.

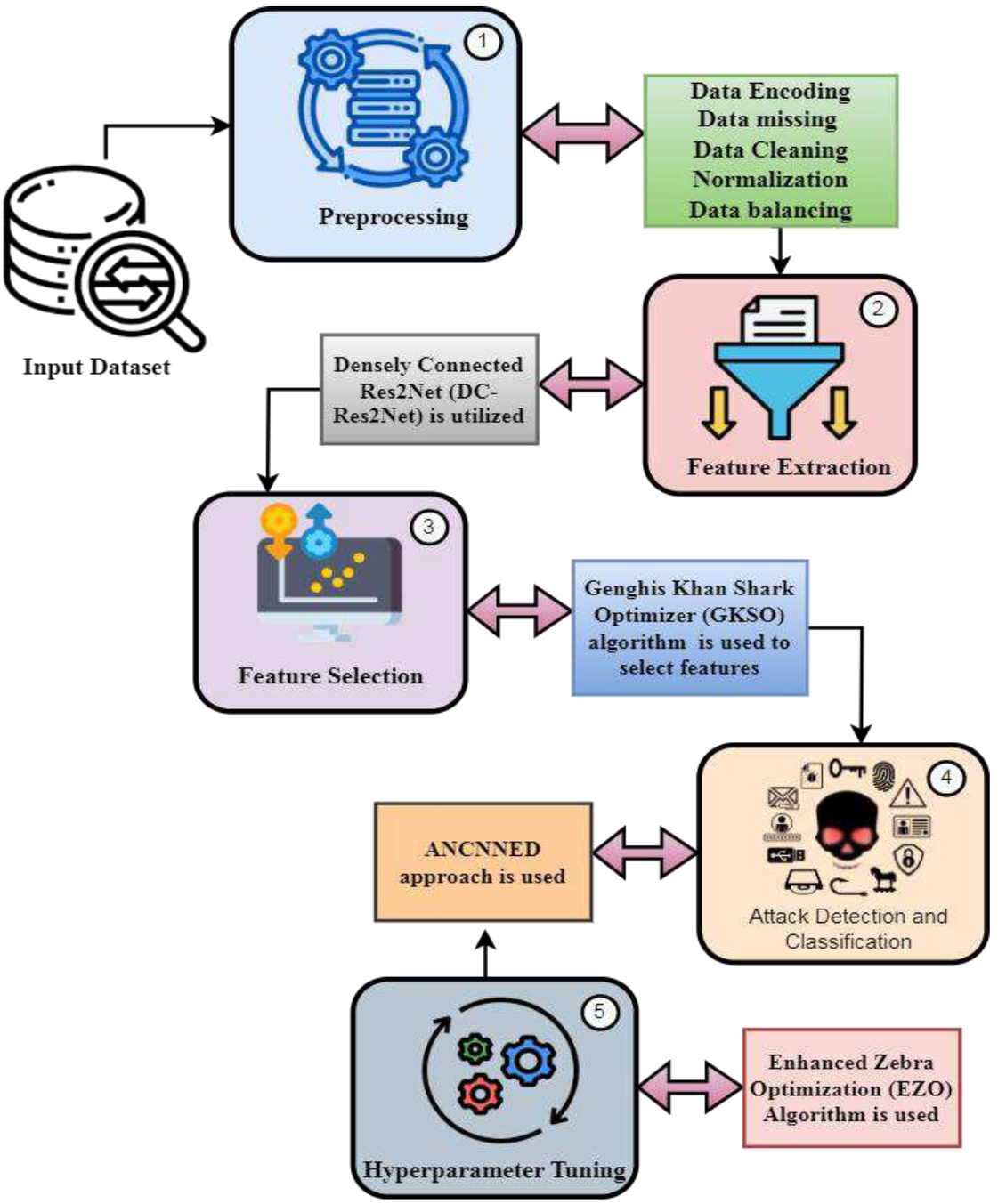


Fig. 1. Overview of the proposed framework

The proposed methodology begins with gathering the data, then uses Densely Connected Res2Net (DC-Res2Net) to extract features and the Genghis Khan Shark Optimizer (GKSO) Algorithm to select features. The categorization is performed using an ensemble model called ANCNNED, which combines the strengths of AlexNet Convolutional Neural Network (ACNN) and EfficientDet to provide automated attack detection. The Enhanced Zebra Optimization (EZO) algorithm is used to improve hyperparameters, resulting in increased accuracy and effectiveness in intrusion detection in cloud and IoT contexts.

3.1 Preprocessing

Data preprocessing is critical in intrusion detection systems (IDS) because it prepares raw network traffic data by removing noise, addressing missing values, and normalizing features. Effective

preprocessing ensures that the IDS can accurately identify and classify possible threats, hence improving overall network security.

3.1.1 Data cleaning

Data cleaning is used to identify and rectify (or delete) incorrect or missing value data in a dataset. Data cleaning in the HDA-IDS consists of removing repeated and incomplete data. Additionally, the various settings of the "label" feature are substituted by integers.

3.1.2 Data normalization

The primary purpose of data normalization is to scale the supplied data into a usable format. The first step in the data preparation stage is to clean up the information by filling in any values that are missing and removing redundant data. Selecting the most significant and relevant elements from the information flow improves performance and reduces errors in traffic categorization. The processing stage involves standardizing and translating the data, which turns categorical values into arithmetic values as needed by the learning model. Z-score normalization is employed to accelerate training and reduce variation in large datasets.

$$z = \frac{x-\mu}{\sigma} \tag{1}$$

In Eq. (1), z indicates z-score normalization, x represents the value, σ represents the standard deviation (SD), and μ represents the mean of instances.

3.1.3 Improved Self-Paced Ensemble Generative Adversarial Networks with Auxiliary Classifier (SPE-ACGAN)

The problem of data imbalance in intrusion detection systems (IDS) is addressed by the Improved SPE-ACGAN. Due to a lack of training samples, traditional IDS frequently have trouble identifying minority class assaults, which results in poor classification performance. Self-paced learning is used by SPE-ACGAN to enable the model to incorporate simpler samples progressively before taking on more complicated ones. This method produces high-quality synthetic samples that improve the representation of minority classes while stabilizing training. Furthermore, many GANs are combined in the ensemble framework to increase resilience and enhance learning overall.

Additionally, the SPE-ACGAN architecture's incorporation of an auxiliary classifier allows for simultaneous data production and classification, guaranteeing that the synthetic samples produced are accurate and pertinent to the attack classes. This feature improves the model's detection performance across a range of attack types while also enriching the dataset. Evaluations show that SPE-ACGAN is a potent technique for boosting the efficacy of IDS in recognizing a variety of dynamic cyber threats since it greatly increases detection rates for minority classes.

3.2 Feature Extraction

The goal of employing Densely Connected Res2Net (DC-Res2Net) for feature extraction in IoT-based Intrusion Detection Systems (IDS) is to effectively capture complex network traffic patterns while retaining high accuracy and efficiency. DC-Res2Net's dense connections allow for improved multi-scale feature representation, which aids in the detection of minor irregularities and potential security concerns in IoT environments. Its ability to decrease redundancy and react to various data increases the IDS's robustness, allowing it to scale for the fast-expanding landscape of connected devices and providing a more secure IoT infrastructure.

3.2.1 Densely Connected Res2Net (DC-Res2Net)

Accurate intrusion detection in IDS relies heavily on efficient feature extraction from network traffic data. Traditionally, many approaches use widely known feature extractors like VGG16 or ResNet50 to capture deep features directly from the data. However, VGG16 suffers from degradation issues as the network deepens, which ResNet partially overcomes through its residual connections. Res2Net, an enhancement over ResNet, was introduced to further optimize feature extraction. It clusters convolution kernels into smaller groups while maintaining the overall size and number of kernels. This approach builds hierarchical residual-like connections between these kernel branches, effectively increasing the receptive field of each layer without altering the network's complexity. Building on Res2Net and inspired by DenseNet, we propose **DC-Res2Net** for more efficient feature extraction in

IDS. In this model, each group of convolution filters processes the output from the previous batch, resulting in a broader range of receptive field sizes. This enables the model to capture more detailed patterns in network traffic, improving detection accuracy. To optimize computation time and reduce network parameters, group convolution is used in place of traditional convolution, making the model faster and more efficient for intrusion detection.

By utilizing the DC-Res2Net module for multi-scale feature extraction from network traffic data, the IDS's ability to capture intricate patterns across multiple scales has been significantly enhanced. To further interpret and validate this multi-scale extraction capability, Class Activation Mapping (CAM), specifically Grad-CAM, is applied. Grad-CAM helps visualize the discriminative regions used by the network for classifying different types of traffic, providing insights into how DC-Res2Net effectively identifies key features for accurate intrusion detection.

Compared to ResNet29, where activation maps cover only portions of key regions, the CAM results from DC-Res2Net29 show more focused and concentrated activation areas. This demonstrates DC-Res2Net's enhanced ability to accurately capture semantic information in network traffic data, effectively distinguishing between normal and malicious patterns. Its precise localization of CAM regions enables more reliable and accurate detection of anomalies in IDS.

Assuming the input feature $x \in Rw \times h \times b \times C$ where ww and hh represent the spatial dimensions of the feature map, bb its spectral dimension, and CC the number of feature channels, we introduce a new dimension called "scale" to enable hierarchical operations. A 3×3 filter with "D" total channels is divided into "D" groups, where each group contains D channels, i.e., $D = D \times D$. Additionally, the input feature map is sliced into multiple groups Di (with $D \in \{1,2,...,s\}$ for multi-scale analysis. This structure improves the IDS's ability to capture fine-grained, multi-scale patterns in network traffic, enhancing its capacity to detect anomalies across various scales and channels.

$$y_i = \begin{cases} x_i, & i = 1 \\ G_i(x_i + y_{i-1} + \cdots + y_1) & 1 < i \leq s \end{cases} \quad (2)$$

Formula (1) defines $G_i(*)$ as a 3×3 clustered convolution. While increasing the number of channels s expands the receptive fields, it also adds more parameters, which can hinder training and prediction. In this work, the channel number s is fixed at four, as Res2Net with $s = 4$ has been shown to deliver optimal results without compromising efficiency. This choice ensures effective multi-scale feature extraction for IDS.

3.3 | Feature Selection

The purpose of feature selection utilizing the Genghis Khan Shark Optimizer (GKSO) in IDS is to improve model performance by identifying the most relevant characteristics from big and complicated datasets. GKSO efficiently lowers dimensionality, which helps to offset difficulties with computational complexity and overfitting. By focusing on essential features, GKSO assures that the IDS detects abnormalities and security threats with high accuracy and low false positives. Furthermore, this feature selection technique streamlines the training phase, resulting in faster convergence and enhanced overall efficiency of the IDS, allowing it more capable of handling the dynamic nature of network traffic.

3.3.1 | Genghis Khan Shark Optimizer (GKSO)

An algorithm inspired by nature, the GKSO is derived from the migratory habits of freshwater GKS fish, which are found in rivers such as the Mekong and Chao Phraya. Compared to their more extended male counterparts, female GKS fish have stronger, flatter bodies. These fish have sleek bodies for quick swimming, which helps in hunting, and they spawn every year. To mislead predators, they also change the colour of their skin. To tackle optimization issues, the GKSO algorithm models the foraging, out-to-hunt, and escaping behaviours of GKS fish.

Initialization: The GKSO algorithm is swarm-based and begins with individuals of possible solutions that are all assessed according to an objective function. N solutions make up the population, represented by the initial X, and every response X_i, has D dimensions that match the variables in the issue. The GKSO equations are used to improve the initial population, which is created randomly.

$$X_i = LB + rand(0, 1) \times (UB - LB) \quad (3)$$

While X_i represents the present agent's positioning and LB and UB represent the optimization problem's boundary bounds.

Hunting Phase (Exploration): GKS fish scan the ocean depths in their natural habitat to make sure no greater predators are around. They can safely widen their search range and find the best prey thanks to this behavior. This is emulated in the GKSO algorithm by searching both the upper and lower bounds of the search space to identify a new random point that would be the best hunting target. By expanding the exploration area, this technique makes sure the optimum solution is discovered. Equation 4) defines a mathematical framework for this behaviour.

$$XP_i = X_i(t) + \frac{LB + r_1 * (UB - LB)}{t} \tag{4}$$

In the GKSO method, agent ii's newly upgraded spot during the first phase is indicated as XP_i, and its current position is represented as $X_i(t)$. Within the interval [0, 1], a random value r1r1 is generated, and tt denotes the optimization process's iteration at that point. If the updated position of every agent enhances the fitness function, the newly estimated position is approved for the update; if not, the agent keeps its old position. Equation (5) provides a mathematical formulation of this updating rule.

$$X_i(t+1) = \begin{cases} XP_1, & f(XP_1) < f(X_i) \\ X_i(t), & \text{otherwise} \end{cases} \tag{5}$$

Moving Phase (Exploitation): During the exploitation stage, the GKSO algorithm makes use of the GKS fish's highly developed sense of smell to pinpoint the ideal target. This stage makes sure that agents are always sent to the best hunting position. In this stage, there are two updates to the agent spot: one for the top agent in individuals and the other agents. Equation (6) is used to upgrade the spot of the first agent.

$$XP_2 = s * (X_{best}(t) - X_i(t)) \tag{6}$$

The optimal location at the present iteration is denoted by $X_{best}(t)$ and the olfactory concentration s determines the amount of attraction regarding the optimal spot.

$$s = ml' \tag{7}$$

Stronger agents have more influence in the migrating phase, as II represents an agent's fitness and m balances exploration and exploitation. The amount of prey odor perceived is determined by the random number r(0 to 1):

r=0: The agent resumes its exploration after detecting no smell.

r=1: The agent detects every odor and locates the local maximum.

While higher s values aid in agent direction, they may also limit exploration. Eq. (8) is used to upgrade the placements of the remaining agents to get closer to the ideal solution.

$$\hat{X}_i(t) = s * (X_{best}(t) - X_i(t)) \tag{8}$$

$$XP_2 = \frac{X_2(t) + X_{i-1}(t)}{2} \tag{9}$$

If a newly estimated spot for every agent is based on either Equation. (8) or Eq. (9) enhances the objective function, it is approved in this phase of GKSO. The agent keeps their prior position if not. In Eq. (10), this updating rule is provided.

$$X_i(t+1) = \begin{cases} XP_2, & f(XP_2) < f(X_i) \\ X_i(t), & \text{otherwise} \end{cases} \tag{10}$$

Parabolic Foraging: When GKS detects prey, it quickly moves forward to attack by using the prey as a point of reference. The attack takes a parabolic route, using its mouth cavity and speed to deliver a deadly strike. This behavior is described in Eq. (11), which formulates the GKSO algorithm, to upgrade the spot of search agents.

$$XP_3 = X_{best}(t) + r_2 \times (X_{best}(t) - X_i(t)) + \Delta \times P^2 \times (X_{best}(t) - X_i(t)) \tag{11}$$

The GKSO algorithm allows for both 1 and -1 for Δ, and random values for r_2 between 0 and 1. Acting as the convergence factor, the nonlinear parameter pp regulates the movement step size of the GKS. While local exploitation rises with a lower pp value, worldwide exploration is prioritized with a higher pp value.

Self-Protection Mechanism: GKS fish frequently come into conflict with rivals or predators that endanger their prey or safety when foraging. Like cuttlefish, GKS fish alter their color to avoid these dangers. Their skin and tails change color when they are scared, which keeps predators away and makes it easier for them to flee quickly. In the subsequent stage of the GKSO algorithm, this colour-changing behaviour is incorporated, and it is quantitatively represented by a mathematical framework.

$$\begin{cases} X_i(t) + k_1(a_1X_{best}(t) - a_2X_k(t)) + k_{2\rho}(a_3(X2(t) - X1(t))) & \text{if } a_1 < 0.5 \\ X_{best}(t) + k_1(a_1X_{best}(t) - a_2X_k(t) + k_{2\rho}(X2(t) - X1(t))) & \text{otherwise} \end{cases} \quad (12)$$

3.3.1.1 | Enhanced solution quality (ESQ) strategy

The ESQ technique is applied in RUN to ensure that every solution advances to a higher level and to improve the standard of solutions before the commencement of the subsequent iteration. The algorithm performs more effectively due to the ESQ strategy, which promotes departure from local optimum solutions. The following is a full explanation of this strategy's mathematical formulation:

$$\begin{gathered} \text{if rand} < 0.5 \\ \text{if } w < 1 \text{ is } w < 1 \\ X_{new2} = X_{new1} + r * w * |(X_{new1} - X_{avg}) + m| \\ \text{else} \\ X_{new2} = (X_{new1} - X_{avg}) + r * w * |(X_{new1} - X_{avg}) + m| \\ \text{end if} \\ \text{end if} \end{gathered} \quad (13)$$

Wherein X_{avg} represents the average spot of three population-derived, generated at a random solution.

$$X_{avg} = \frac{X_{r1}+X_{r2}+X_{r3}}{} \quad (14)$$

$$w = \exp\left(-c\left(\frac{i}{maxi}\right)^3\right) \times \text{rand}(0,2) \quad (15)$$

$$X_{new1} = (1 - \beta) \times X_{best} + \beta \times X_{avg} \quad (16)$$

While c is a randomly selected amount set to 5 $\times$ rand, w is a decreasing variable that is random as the number of iterations increases, and β is a uniformly distributed random variable ranging from 0 to 1. A number denoted by the variable r can be either 1, $-$ 1, or 0. $X_{best}t$ is the name of the best possible option. It's possible that the X_{new3} solution won't perform better than the present one. Consequently, X_{new3} is a new solution that was developed to produce a second opportunity to produce a good solution, as described in:

$$\begin{gathered} \text{if rand} < w \\ X_{new3} = (X_{new2} - \text{rand} \times X_{new2}) + SF \times (\text{rand} \times X_{RK} + (v \times X_b - X_{new2})) \quad (17) \\ \text{end if} \end{gathered}$$

The ESQ is utilized at the final stage of every iteration based on the random value rand, as indicated by Eq. (17), while w is an integer at random that is set to 2 $\times$ rand. The ESQ approach makes the population's members fitter than they were, which improves the I-GKSO exploration phase and leads the modified individuals to choose a new search location. Furthermore, by integrating ESQ, the exploration and exploitation phases will be balanced, which lowers the possibility of settling into local optima while looking for global solutions.

3.3.1.2 Time complexity analysis

One of the most important metrics for assessing optimization methods is complexity. A performance comparison of the I-GKSO in terms of its big-oh is given by this work. The number of individuals, the number of iterations as the termination circumstance, the size of the functions' dimensions, and the pheromone of each individual are the key factors we take into account. Consider that the individual of

individual agents has N members, the optimum number of iterations is T, and the average size of the optimization issues is D. So, the following formula can be used to determine the complexity of I-GKSO:

- $O(N \times D)$ is needed for individual initialization that occurs before the main loop of the proposed IGKSO.
- The primary GKSO update behaviours and proposed enhancements are included in the I-GKSO procedure that follows.
- $O(N \times D \times T)$is needed due to the intricacy of the unique poor persons' classification technique.
- The primary behaviours of I-GKSO need $O(4 \times N \times D \times T)$, since the primary phases of GKSO consist of four main loops for mobility, hunting, defense mechanism, and foraging behaviours, every of which requires$O(N \times D \times T)$.
- $O(N \times D \times T)$) is needed to apply the ESQ technique to each individual in the population after each iteration.

Thus, the following formula can be used to determine the total complexity of I-GKSO, taking into account both the primary behaviours and the proposed enhancements:

$$O(I - GKSO) = O(N \times D) + O(N \times D \times T) + O(4 \times N \times D \times T) + O(N \times D \times T) \quad (18)$$

$$= O(N \times D \times (1 + 6 \times T)$$

3.4 Classification

To maximize the advantages of both architectures, the Novel ANCNNED method was designed for classification in intrusion detection systems (IDS). While EfficientDet is renowned for its effective architecture that strikes a balance between accuracy and computational efficiency, AlexNet, with its deep convolutional layers, is superior at extracting features and intricate patterns. By merging both models, the ANCNNED method takes use of EfficientDet speed and scalability, as well as AlexNet's strong representation capabilities, allowing the IDS to efficiently detect and categorize a range of network intrusions. By improving detection accuracy and lowering false positives, this ensemble approach offers a more dependable protection mechanism against cyber threats in dynamic network environments.

3.4.1 AlexNet based CNN model

Effective decision-making plays a major role in IDS classification. In this work, we present an architecture that uses a TFR-based AlexNet CNN model. AlexNet's deep structure and convolutional layers, among other properties, make it an excellent choice for extracting hierarchical characteristics. Here are some further details describing the causes: (1) Compared to previous CNN models, AlexNet has a deeper architecture that makes it easier to collect more complex aspects of the data. Local response normalization is a technique that AlexNet uses to improve the model's adaptability by standardizing answers across different channels. This normalizing process helps to capture relevant patterns and strengthens the model's resistance to fluctuations. (3) AlexNet's convolutional filters are built to recognize spatial patterns of various sizes. This feature is useful for recording local and global variations in data, which is an important feature for accurate classification.

An advanced CNN is called AlexNet. It has eight layers total three fully linked layers and five convolutional layers. Convolutional layers comprise the first five layers, which are sometimes succeeded by max-pooling layers. The final three tiers were completely interconnected. Two clones of the network, each running on a single GPU, are created to enable parallel processing.

The convolutional layer's filters vary in size and number. To investigate deep characteristics, the filter is transported as data or a tensor. After receiving the obtained deep features, the next input proceeds to extract deep features from that location. The model is terminated by the Softmax layer, which assigns the class using a probabilistic methodology. The schematic framework of the AlexNet model is shown in Fig. 2.

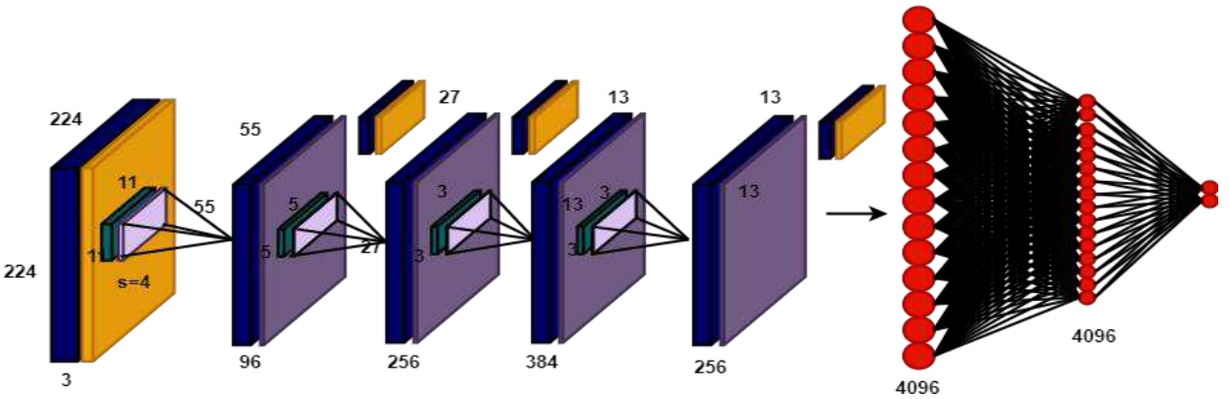


Fig. 2. Framework of AlexNetCNN

3.4.2 EfficientDet

In this study, we employ EfficientDet, a sophisticated algorithm, to enhance intrusion detection in network traffic analysis. EfficientDet utilizes a compound scaling technique that optimizes both accuracy and processing complexity by gradually increasing the backbone network's resolution, depth, and breadth. This approach enables the model to effectively capture complex patterns in network data. It incorporates BiFPN (Bi-directional Feature Pyramid Network) to efficiently fuse features across different scales, which is crucial for distinguishing between normal and malicious activities that may vary significantly.

To address the class imbalance, we implement focus loss, a mechanism that emphasizes challenging cases, thereby improving detection performance. EfficientDet was trained using our selected datasets, achieving high recall and accuracy scores for intrusion detection tasks. Its stability and capacity for real-time applications make it particularly suitable for network surveillance systems. The compound scaling technique allows EfficientDet to learn complex patterns without significantly increasing computational costs. Additionally, the effective feature fusion algorithms enhance the model's ability to integrate multi-scale data, capturing both detailed and broad characteristics of network traffic. Overall, EfficientDet proves to be an effective solution for intrusion detection, delivering high accuracy while maintaining low CPU overhead due to its innovative architecture and training methodologies.

3.5 Hyperparameter Tuning

The Enhanced Zebra Optimization (EZO) Algorithm is a critical step in hyperparameter tweaking the Novel ANCNNED method, which maximizes IDS effectiveness. Zebra social behaviour and movement patterns serve as an inspiration for EZO, which enables it to efficiently search the hyperparameter space for ideal configurations. Through methodical manipulation of parameters including learning rate, batch size, and number of layers, EZO improves the model's capacity to generalize from training data, resulting in decreased overfitting and increased accuracy. Through this optimization process, the ANCNNED model is ensured to function at peak efficiency, optimizing its ability to precisely identify and categorize network intrusions while reducing false positives. As a result, the IDS is stronger and better equipped to adjust to the changing threats and complexities found in network environments.

3.5.1 Zebra Optimization Algorithm (ZOA)

The social interactions of zebras in the wild serve as the model for the ZO), a novel meta-heuristic optimization algorithm. The system models the zebra's feeding strategy and defensive reaction to predators, enabling it to tackle intricate optimization problems in mathematics with remarkable search and convergence speeds. Three steps comprise the ZOA solution process:

Initialization of Individuals: The ZOA generates a random individual of zebras as the space of search for the issue and uses iterative analysis to find the ideal zebra individual placements to find an optimal approach to the targeted issue. Here is the randomly produced initialized zebra positional matrix:

$$X_Z = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,d} \end{bmatrix} \quad (19)$$

In this case, d and N represent the optimized dimension of the issue and the total amount of population in the zebra, respectively; is the location of the ith zebra in the jth dimension. The following is the zebra fitness estimation equation:

$$F_Z = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_N \end{bmatrix} \quad (20)$$

For each zebra, F_i represents its value of fitness, while F_Z is the zebras' fitness matrix. When addressing the DES configuration issue, the zebra population's function of fitness is designated as the issue's objective function.

Foraging behaviour: The zebra population upgrades its location during the subsequent phase of the optimization process by foraging under the direction of the best zebra individuals. The definition for the spot upgrade is as follows:

$$x^{n1}_{i,j} = x_{i,j} + a_1 \bullet \left(Z^P_j - a_2 \bullet x_{i,j}\right) \quad (21)$$

$$\begin{cases} a_1 = \text{rand} \\ a_2 = \text{round}(\text{rand} + 1) \end{cases} \quad (22)$$

$$X_i = \begin{cases} X^{n1}_i & F_{ii} \gtrless F^{n1}_i \end{cases} \quad (23)$$

Whereas rand represents a number chosen at random in the interval [0, 1], X^{n1}_i is the zebra's new spot following the position upgrade and F^{n1}_i denotes the value of fitness.

Strategy for Defending: During the third phase of optimization, the zebra population modifies its location based on the defensive tactic. The two modes of the defense strategy are M_1 and M_2, and the zebra operates in one of the two modes based on the randomized integer a_3; the revised equation is displayed below:

$$v^{n2}_{i,j} = \begin{cases} M_1: x_{i,j} + R \bullet (2 \bullet \text{rand} - 1) \bullet (1 - t/T_{max}) \bullet x_{i,j} & (a_3 \le 0.5) \\ M_2: x_{i,j} + \text{rand} \bullet \left(Z^A_j - a_2 \bullet x_{i,j}\right) & (a_3 > 0.5) \end{cases} \quad (24)$$

$$X_i = \begin{cases} X^{n2}_i & F_{ii} < F^{n2}_i \end{cases} \quad (25)$$

In this case, R is an intentionally defined constant; $X^{n2}_{i,j}$ is the i^{th} zebra's new spot following the foraging stage position update; F^{n2}_i is the associated fitness value; Z^A_j is the quantity of a randomly chosen zebra.

3.5.1.1 | Enhanced Zebra Optimization Algorithm

3.5.1.1.1 | Tent chaotic mapping

During the population's initialization step, the ZOA algorithm creates an entirely random zebra population. Therefore, tent chaotic mapping is introduced in this study to initialize the zebra population and provide a more consistently distributed population in the search space. The revised initialization equation can be written as follows:

$$x_i = r_i(u_b - l_b) + l_b \quad (26)$$

$$r_i = \begin{cases} r_{i-1}/\beta & r_{i-1} \in [0,\beta] \\ (1 - r_{i-1})/(1 - \beta) & r_{i-1} \in [\beta,1] \end{cases} \quad (27)$$

While r_i represents the produced chaotic sequence, x_i represents the individual zebra following tent chaos startup, u_b and l_b represent the lower and upper bounds.

3.5.1.1.2 | Sine cosine strategy

In the foraging phase, the sine cosine approach was implemented to adjust the position update Equation (28) of individual zebras by the sine cosine paradigm, strengthening the ZOA's global search ability. Following the development of the SCA, the following is the revised formula for foraging behaviour:

$$x_{i,j}^{n1} = \begin{cases} x_{i,j} + b_1 \bullet \sin b_2 \bullet \left| b_3 \bullet Z_j^p - a_2 \bullet x_{i,j} \right| & \text{rand} < 0.5 \\ x_{i,j} + b_1 \bullet \cos b_2 \bullet \left| b_3 \bullet Z_j^p - a_2 \bullet x_{i,j} \right| & \text{rand} \geq 0.5 \end{cases} \quad (28)$$

$$b_1 = \left[1 - (t/T_{max})^k\right]^{1/k} \quad (29)$$

3.5.1.1.3 | Dynamic adaptive weight factor

The defense strategy of ZOA benefits from the addition of dynamic adaptive weight factors, which help to dynamically balance the algorithm's capacity for both local exploration and global search. Below are the dynamic adaptive weight factor (ω_z) and updated position update formula for the zebra imperial strategy:

$$X_{i,j}^{n2} = \begin{cases} M_1 : \omega_z \bullet x_{i,j} + 0.01 \bullet (2 \bullet \text{rand} - 1) \bullet \left(Z_j^A - a_2 \bullet x_{i,j}\right) \bullet x_{i,j} & (a_3 \leq 0.5) \\ M_2 : \omega_z \bullet x_{i,j} + 0.01 \bullet (2 \bullet \text{rand} - 1) \bullet \left(Z_j^A - t/T_{max} \bullet x_{i,j}\right) \bullet x_{i,j} & \end{cases} \quad (30)$$

$$\omega_z = -\left[1 + \cos\left(\frac{3}{2}\pi + \frac{1}{2}\pi \bullet \frac{t}{T_{max}}\right)\right] \quad (31)$$

Tent chaos mapping is used to initialize IZOA, and iterative optimization is achieved by combining the defensive equation with a dynamic weighting factor and the foraging equation with a sine cosine strategy.

4 | Result and discussions

This section provides a thorough study of the performance of the suggested methodology. It assesses the Novel ANCNNED approach's efficacy for IDS classification, emphasizing important metrics like F1-score, accuracy, precision, and recall. The benefits of the suggested system are illustrated through comparisons with current models, and discussions are held regarding the effects of hyperparameter adjustment with the Enhanced Zebra Optimization Algorithm (EZO). This section also looks at how the model handles data imbalance and increases detection rates for different kinds of attacks, providing an in-depth overview of its advantages and disadvantages. Hyperparameters and their values are represented in table 2.

Table 2 parameter and their values

Parameter	Values
Learning rate	0.001
Batch Size	64
Epoch	200
Optimizer	EZOA
Dropout Rate	0.2 – 0.5
L1 Regularization	0.01

4.1 | Experimental Setup

With 32 GB of RAM and an Intel Corporation i5 2.60GHz processor, it operates on the most recent version of Windows. The assessments performed on the Anaconda 3 platform make use of Python, KERAS, and TensorFlow. The setup for the experiment is represented in Table 3.

Table 3 Environmental setup

Project	Setup
Software	Python version 3.11

Framework	Keras
Memory	32 GB
CPU	i3-7100U
Operating System	Windows 10

4.2 | Dataset Description

To simulate and evaluate the performance of the proposed approach we utilized four standard datasets.

CICIoT2023: 105 IoT devices made up the IoT topology that was created. The testbed is made up of cameras, sensors, smart home appliances, and microcontrollers that are networked and set up to facilitate the execution of different types of assaults.

ToN-IoT Dataset: In 2019, the ACCS published a varied dataset. The traffic network component collected via an IoT ecosystem has been utilized. The majority of the traffic is made up of assault instances, with a ratio of 21,542,641 attack flows to 796,380 benign flows.

N-BaIoT dataset: The N-BaIoT dataset, which contains traffic information for nine IoT devices, is the second dataset that is used. These data comprise a range of cyber-attacks as well as benign traffic. There are ten distinct types of hostile cyberattacks, two of which are classified as botnets.

IoTID23 Dataset: This dataset is a collection of data from IoT ecosystems for intelligent homes. The sensors and AI speakers in this dataset indicated the IoT victim equipment, whereas the other equipment represented the attacker's equipment. Nmap tools were used to mimic attacks such as man-in-the-middle, scanning, and denial-of-service.

4.3 | Evaluation Criteria

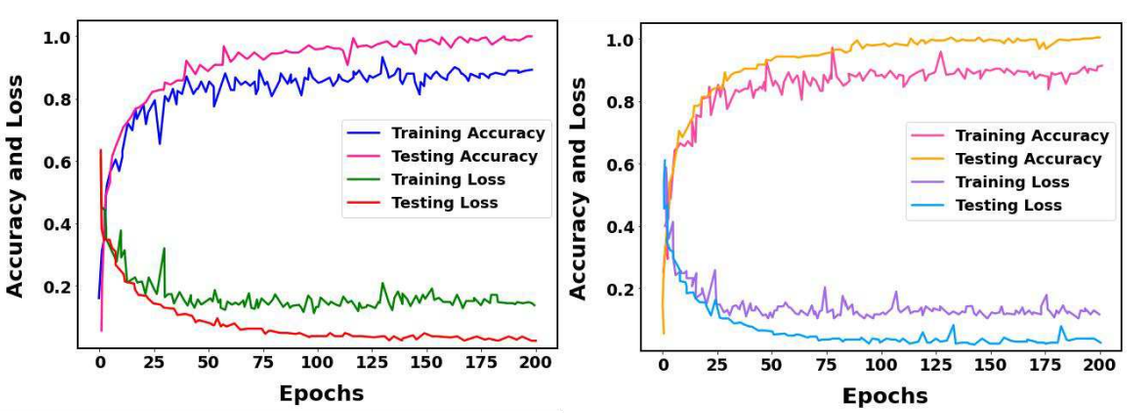
To thoroughly evaluate the proposed method, we employed five distinct assessment metrics: Accuracy is used to evaluate the general classification's efficacy. The calculation of accuracy involves dividing the total number of cases by the number of correctly classified instances. The network traffic examples that are accurately or wrongly identified as normal or abnormal are displayed in the confusion matrix. Equations (32), (33), (34) and (35) display the evaluation matrices.

$$\text{Recall} = \frac{\text{Fp}}{\text{Tn+Fp}} \tag{32}$$

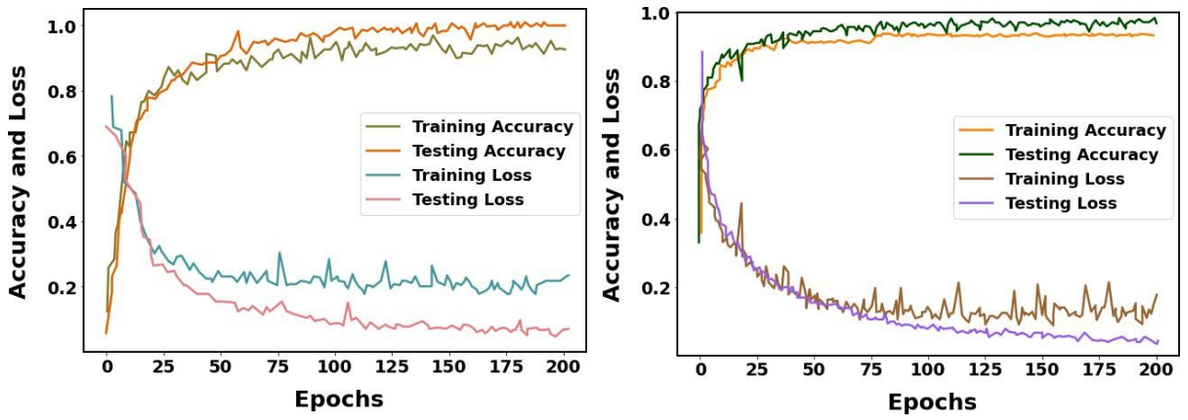
$$\text{Precision} = \frac{\text{Tp}}{\text{Fp+Tp}} \tag{33}$$

$$\text{F1 - Score} = 2 * \frac{\text{precison*recall}}{\text{Precision+Recall}} \tag{34}$$

$$\text{Accuracy} = \frac{\text{Tn+Tp}}{\text{Tn+Fp+Tp+Fn}} \tag{35}$$



(a) (b)



(c) (d)

Fig. 3 Evaluation of performance analysis (a) training and testing loss and accuracy for dataset 1 (b) for dataset 2 (c) for dataset 3 (d) for dataset 4

The differentiation of training and testing performance of proposed dataset is shown in Fig. 3. It is trained over 200 epochs.

4.3.1 | Performance analysis on preprocessing

In this study, we utilized multiple datasets, including ToN-IoT, N-BaIoT, CICIoT2023, and IoTID20. The original ToN-IoT dataset contained an imbalanced distribution of classes, with a total of 1108609 normal and 270251 attack packets. To address this imbalance, we employed down-sampling techniques, resulting in a balanced dataset of 270251 normal and 270251 attack packets.

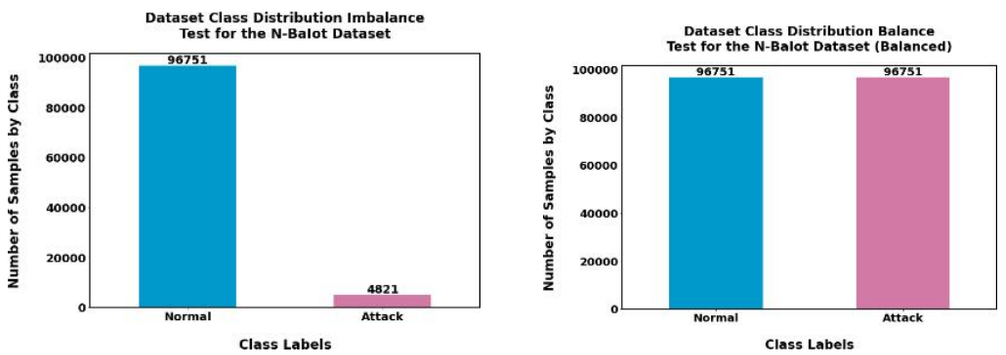


Fig. 4. Class Distribution imbalance and balance set for the N-BaIoT dataset

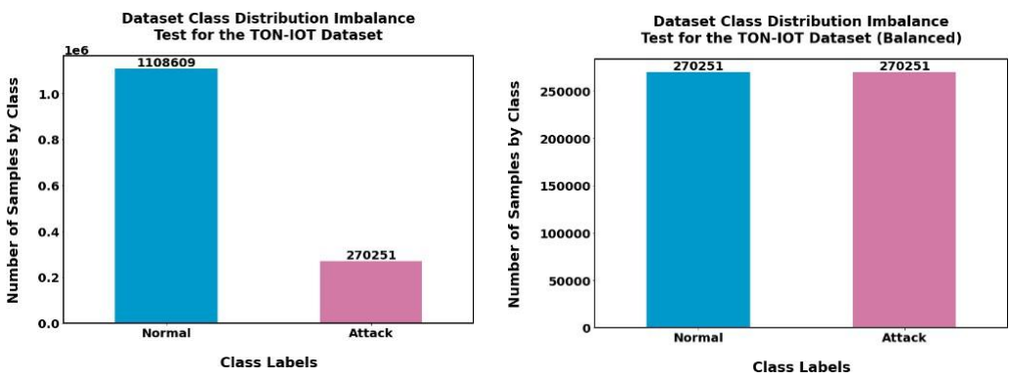


Fig. 5. Class Distribution imbalance and balance set for the ToN-IoT dataset

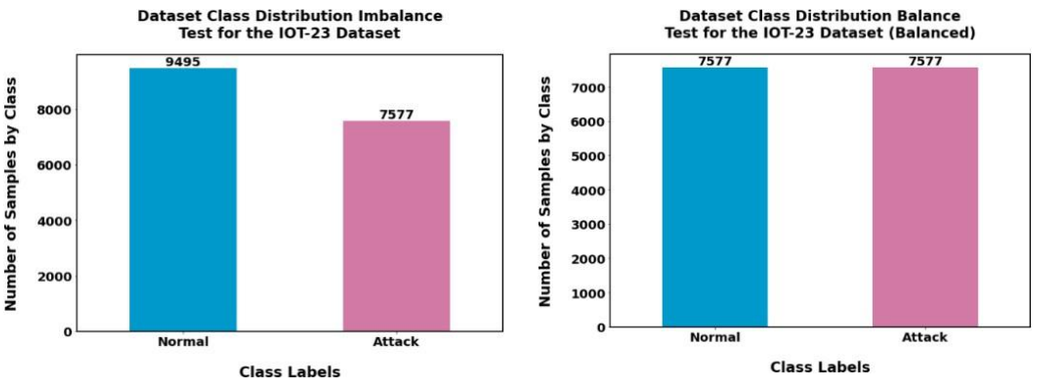


Fig. 6. Class Distribution imbalance and balance set for IoT-23 dataset

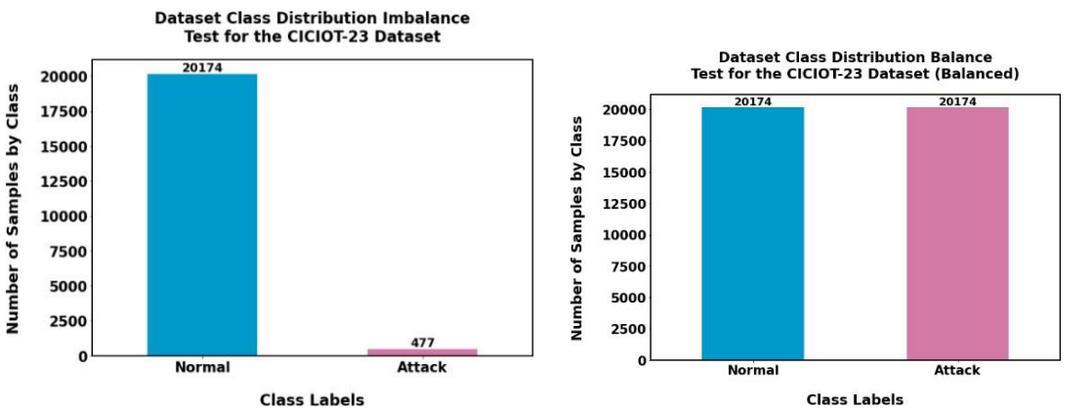
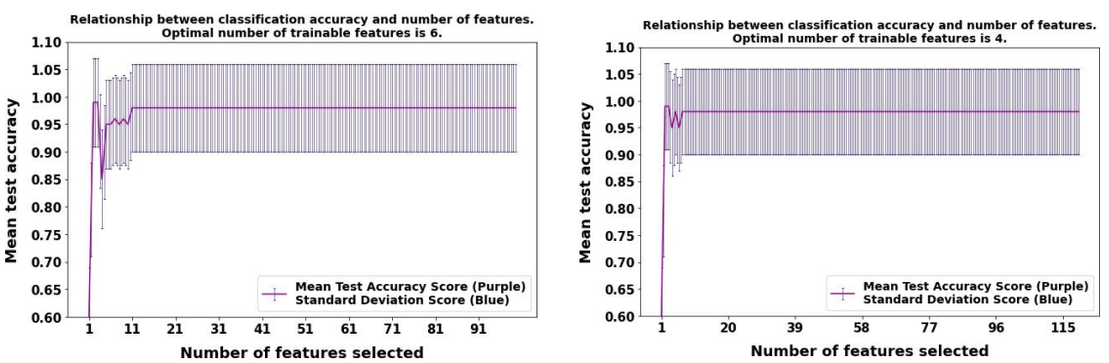


Fig. 7. Class Distribution imbalance and balance set for CICIOT-23 dataset

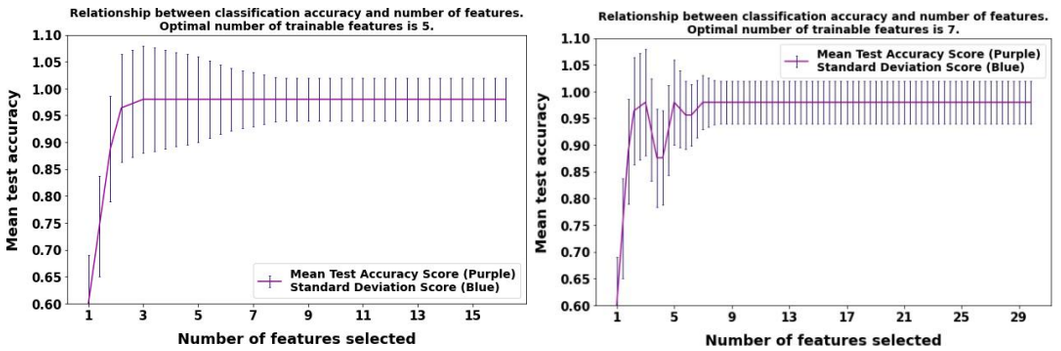
Similarly, the N-BaIoT dataset was characterized by an uneven distribution, featuring 96751 normal packets and 4821 attack packets. After down-sampling, this dataset was balanced to include 96751 normal and 96751 attack packets. The CICIoT2023 and IoTID20 datasets were also preprocessed to ensure an equitable representation of normal and attack packets. The distribution of normal and attack packets across these datasets, both before and after balancing, is illustrated in Fig.s 4, 5, 6 and 7.

4.3.2 | Relevant Feature Subsets in IDS

In this study, we selected twelve subsets of efficient and relevant features from the ToN-IoT, N-BaIoT, CICIoT2023, and IoTID20 datasets using a single feature selection algorithm. Each subset consists of six optimal features, as shown in Fig. 8, derived from the original feature sets of 93,750, 62,154, 40, and 15 features, respectively. The choice of six features was based on the observation that the SGDC algorithm achieved the highest detection accuracy with this subset size. The feature selection process assessed the relationship coefficients among the independent (input) features to identify the most relevant ones. Relationship between categorization model accuracy and the optimum amount of trainable features



(a) ToN-IoT dataset (b) CICIOT-23



(c) N-BaloT dataset (d) IoT23 dataset

Fig. 8. Relationship between categorization model accuracy and the optimum amount of trainable features

4.3.3 | Feature Importance Score

The importance of each feature in the dataset for the intrusion detection task is gauged by the feature importance score. We determine the essential characteristics that set malicious from regular network activity distinct by analyzing individual contributions to model predictions.

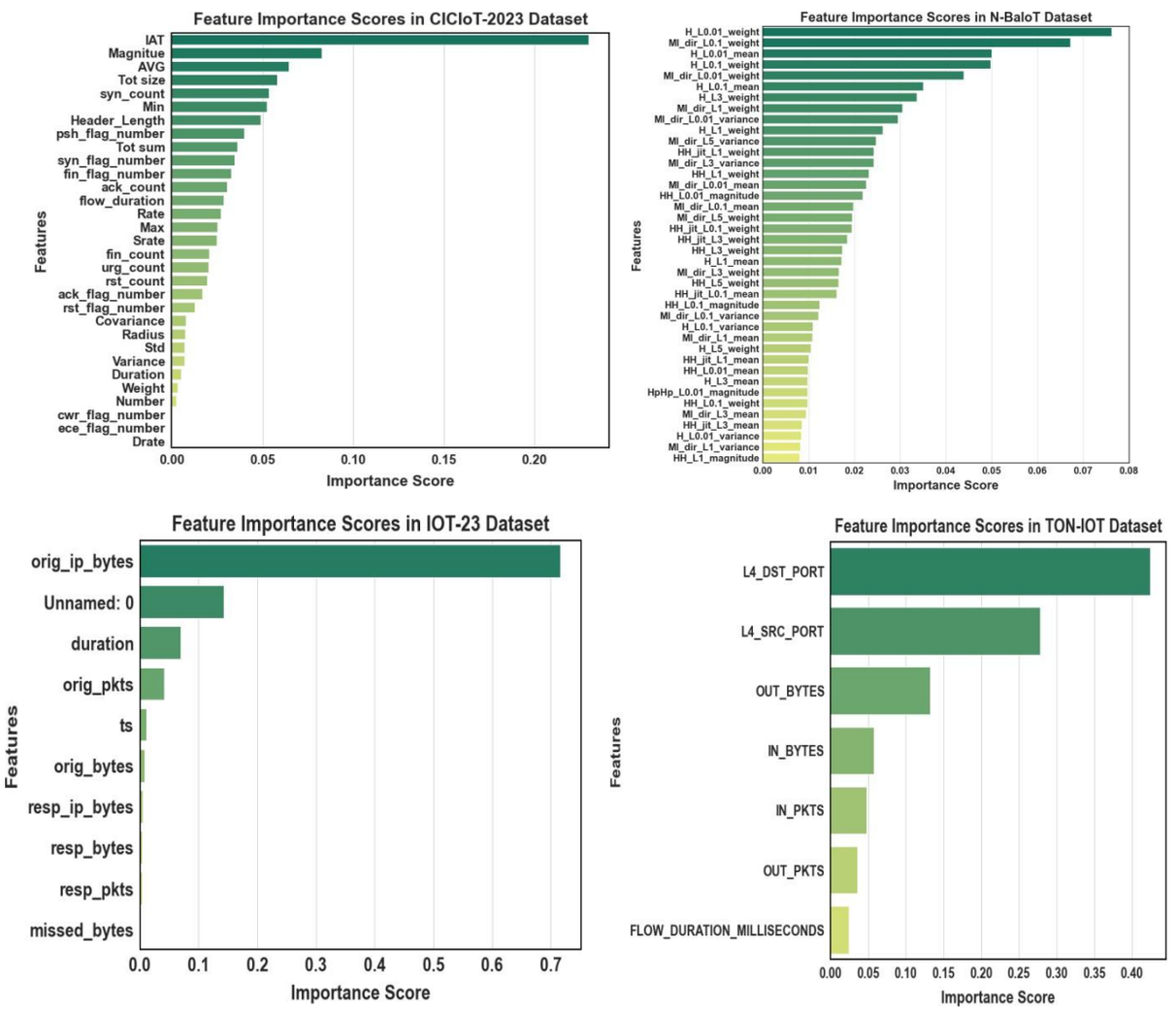


Fig. 9. Feature importance score for proposed datasets

The most important elements for enhancing model performance are highlighted by this score, which is produced by our feature selection method. Recognizing the significance of a feature allows for focused improvements and effective use of resources, which may improve detection accuracy by concentrating on the most important aspects of network traffic. Feature importance score for proposed datasets is illustrated in Fig. 9.

Performance analysis on the CICIoT2023 dataset

This section utilizes the CICIoT2023 dataset to provide a thorough assessment of our proposed IDS methodology. We examine important performance indicators to evaluate how well the model detects different types of attacks in IoT network traffic.

Table 4. Multi-class categorization result for the CICIoT2023 dataset

Approaches	Recall	Precision	Accuracy	F1-Score
LSTM	98.63%	98.71%	98.66%	98.67%
MLP	99.91%	99.91%	99.91%	99.91%
LR	99.43%	99.44%	99.43%	99.43%
CNN	99.90%	99.93%	99.90%	99.91%
DT	86.34%	82.39%	86.34%	82.69%
GB	96.88%	99.88%	99.88%	99.88%
KNN	99.86%	99.86%	99.86%	99.86%
RF	96.58%	96.98%	96.58%	96.51%
NB	99.09%	99.13%	99.09%	99.10%
DL-BiLSTM	-	-	93.13%	-
Proposed	99.92%	99.95%	99.95%	99.94%

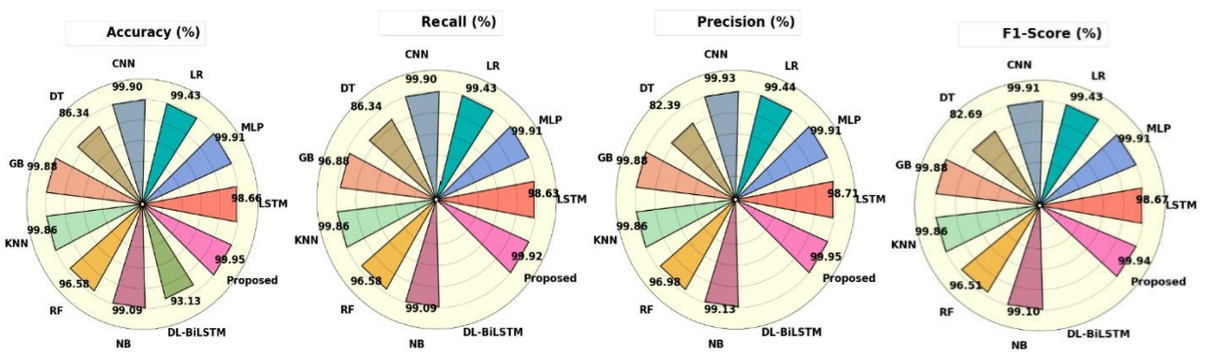


Fig. 10. Performance differentiation over proposed with existing approaches using the CICIoT2023 dataset

Multi-class categorization results for the CICIoT2023 dataset show the performance of various approaches in terms of Recall, Precision, Accuracy, and F1-Score. The results indicate that the proposed method achieved the highest performance, with a Recall of 99.92%, Precision of 99.95%, Accuracy of 99.95%, and an F1-Score of 99.94%. This is closely followed by MLP and KNN, which both demonstrated high scores across all metrics, with MLP achieving 99.91% in all categories. Other notable methods include CNN, with a Recall of 99.90%, and LR, with a Recall of 99.43%. Conversely, the Decision Tree (DT) showed the lowest performance, with Recall at 86.34% and F1-Score at 82.69%. The DL-BiLSTM model presented an accuracy of 93.13%. Overall, the results highlight the efficacy of the proposed approach and other machine learning models, demonstrating their capability in multi-class categorization tasks within IoT environments. Table 4 and Fig. 10 show the performance differentiation of the CICIoT2023 dataset.

Table 5. Binary and Multi categorization of the CICIoT2023 dataset

Types	Label	Acc (%)	Prec (%)	Rec (%)	FS (%)
Binary	Normal	99.23	99.41	99.32	99.30
	Attack	99.21	99.40	99.24	99.32
Multi-Class	Backdoor,	99.36	99.26	99.45	99.35
	DoS,	99.42	99.31	99.51	99.41
	DDos,	99.18	99.61	99.16	99.38
	Injection,	99.43	99.22	99.31	99.26
	MITM,	99.32	99.43	99.24	99.33
	Normal,	99.31	99.26	99.19	99.22

	Password,	99.19	99.11	99.29	99.20
	Ransomware,	99.20	99.35	99.38	99.36

The effectiveness of our intrusion detection model for both binary and multi-class categorization on the CICIoT2023 dataset is compiled in Table 5. In terms of binary classification, the model obtained high results for accuracy, precision, recall, and F1-score: 99.23%, 99.41%, 99.32%, and 99.30% for normal traffic, and 99.21%, 99.40%, 99.24%, and 99.32% for attack traffic. The model's efficacy in identifying both benign and malevolent actions in the Internet of Things (IoT) environment was demonstrated by the accuracy with which different attack types performed in multi-class classification, ranging from 99.18% to 99.43%.

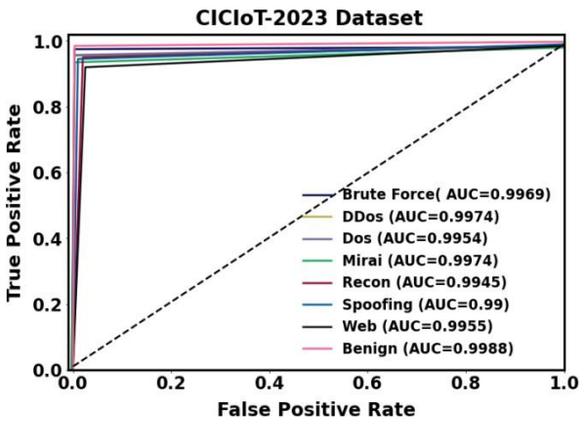


Fig. 11. Roc-Curve for CICIoT2023 dataset

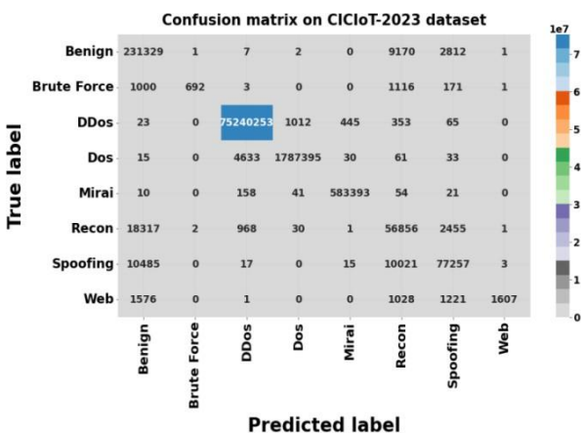


Fig. 12. Confusion matrix

Figures 11 and 12 show the roc curve and confusion matrix for the CICIoT2023 dataset.

Performance analysis on the ToN-IoT dataset

A thorough analysis of our IDS model with the ToN-IoT dataset is given in this subsection. Table 6 shows the performance differentiation of the proposed with existing approaches.

Table 6 Performance comparison of proposed with existing models using the ToN-IoT dataset

Approaches	Precision	Accuracy	F1-Score	Detection Rate
RF	86.86%	93.94%	91.81%	97.32%
SAE-IDS	67.57%	83.15%	80.48%	99.5%
SVM	65.12%	82.21%	78.81%	74.74%
NB	77.68%	90.62%	72.43%	77.7%
DT	74.42%	95.34%	76.33%	80%
E-ADS	90.54%	96.35%	95.03%	99.98%
SATIDS	97.3%	96.56%	97.35%	97.4%
Proposed	99.56%	99.43%	99.37%	99.99%

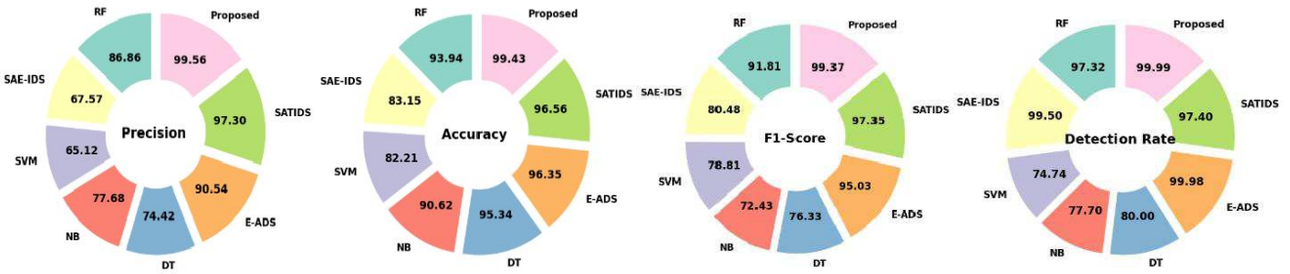


Fig. 13. performance differentiation using the ToN-IoT dataset

The Proposed method outperformed all others, achieving the highest Precision (99.56%), Accuracy (99.43%), F1-Score (99.37%), and Detection Rate (99.99%), showcasing its superior capability in detecting and classifying threats. SATIDS also performed well, with a Precision of 97.3% and F1-Score of 97.35%, while E-ADS achieved strong results across the board, with a Detection Rate of 99.98% and F1-Score of 95.03%. In contrast, SVM and SAE-IDS showed lower performance, particularly in Precision and F1-Score, indicating room for improvement in handling multi-class categorization. RF and NB presented moderate results, with RF showing a strong Detection Rate (97.32%) but relatively lower Precision (86.86%). Overall, the proposed approach demonstrates its robustness and effectiveness compared to other models on the ToN-IoT dataset. Fig. 13 illustrates the performance differentiation using the ToN-IoT dataset.

Table 7 Binary and Multi categorization of the ToN-IoT dataset

Types	Label	Acc (%)	Prec (%)	Rec (%)	FS (%)
Binary	Normal	99.65	99.45	99.33	99.39
	Attack	99.54	99.38	99.19	99.28
Multi-Class	Backdoor,	99.25	99.27	99.25	99.26
	DoS,	99.39	99.36	99.31	99.34
	DDos,	99.30	99.43	99.31	99.37
	Injection,	99.61	99.61	99.28	99.44
	MITM,	99.44	99.45	99.19	99.32
	Normal,	99.32	99.34	99.23	99.29
	Password,	99.27	99.41	99.15	99.28
	Ransomware,	99.57	99.39	99.16	99.27
	Scanning,	99.19	99.33	99.28	99.30
	XSS	99.22	99.28	99.48	99.38

Our IDS model's performance metrics for binary and multi-class classifications are shown in table 7. Strong precision, recall, and F1-scores were demonstrated by the model, which reached 99.65% accuracy for normal traffic and 99.54% accuracy for attack traffic in the binary classification. Accuracy for multi-class categorization varied from 99.19% to 99.61% for different attack types like DDoS, Backdoor, and more. These outcomes demonstrate how well the model detects and categorizes both benign and malevolent network activity.

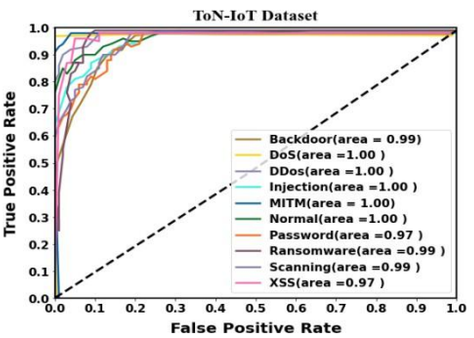


Fig. 14 Roc-Curve for ToN-IoT dataset

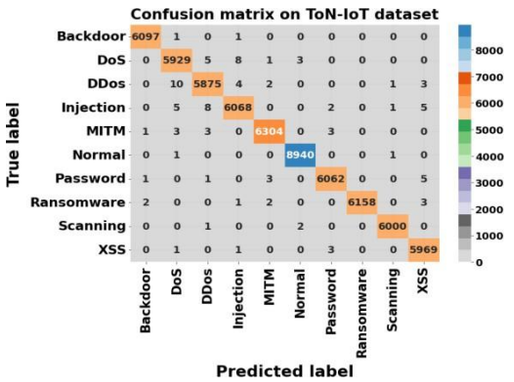


Fig. 15 Confusion matrix for ToN-IoT dataset

Figures 14 and 15 show the Roc-Curve and confusion matrix for the ToN-IoT dataset.

Performance analysis on the IoT23 dataset

A thorough analysis of our IDS model with the IoT-23 dataset is given in this subsection. Table 8 shows the performance analysis of various approaches using the IoT-23 dataset.

Table 8 Performance differentiation of Proposed with existing models using IoT23 dataset

Method	Acc	Prec	Rec	F-S
Random Forest	99.5%	99.5%	99.5%	-
Adaboost	87%	86%	87%	-
Shallow Model	95.22%	-	97.50%	-
Catboost	99.92%	99.88%	99.92%	99.90%
ANN	66%	71%	66%	68.05%
Proposed	99.96%	99.93%	99.91%	99.92%

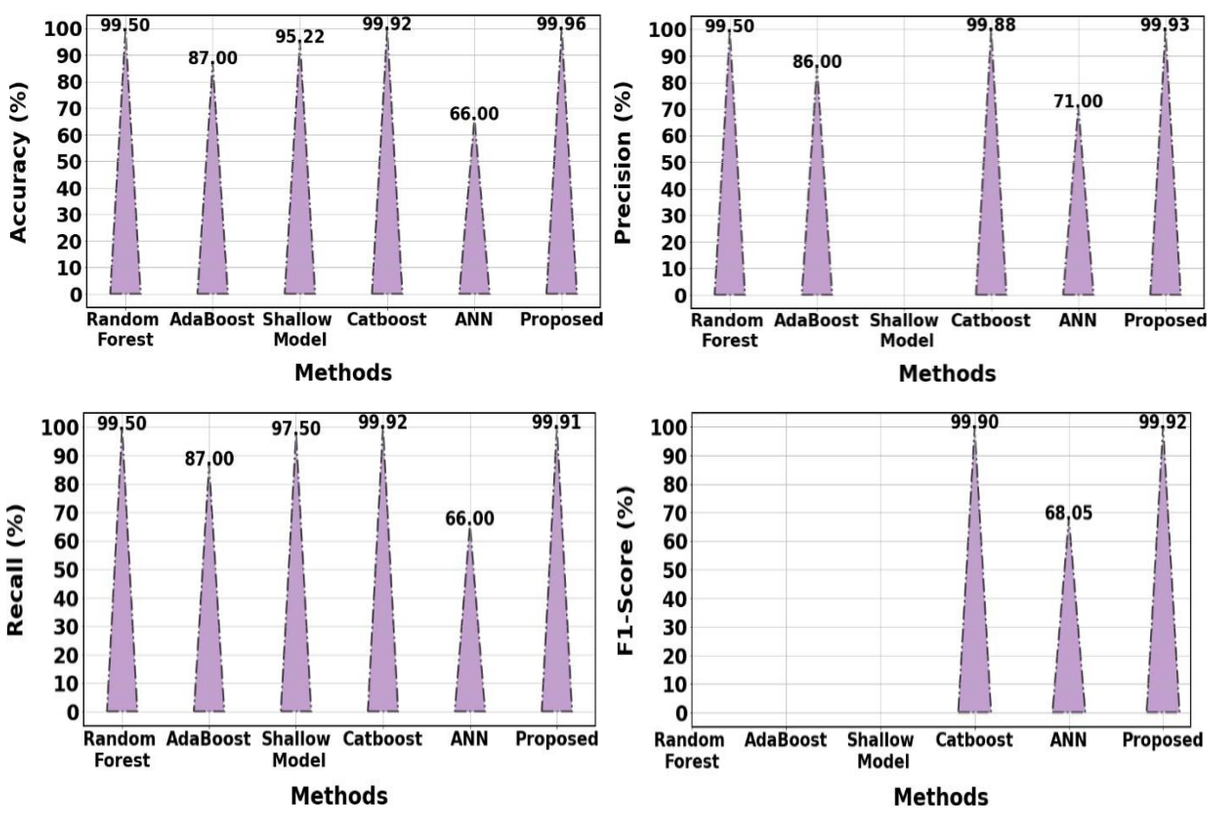


Fig. 16. Performance differentiation of proposed with existing models using IoT-23 dataset

Differentiation results for the IoT23 dataset show that the proposed method outperformed others with 99.96% Accuracy, 99.93% Precision, 99.91% Recall, and a 99.92% F1-score. CatBoost also performed well, achieving 99.92% Accuracy and 99.90% F1-Score. Random Forest followed with 99.5% in both Accuracy, Precision, and Recall. Adaboost and the Shallow Model showed moderate results, with Adaboost achieving 87% Accuracy and the Shallow Model reporting 95.22% Accuracy and 97.50% Recall. ANN had the lowest performance with 66% Accuracy and a 68.05% F1-Score. Fig. 16 shows the performance differentiation using the IoT-23 dataset.

Table 9 Binary and Multi-class categorization of IoT23 dataset

Types	Label	Acc (%)	Prec (%)	Rec (%)	FS (%)
Binary	Normal	99.38	99.29	99.32	99.30
	Attack	99.32	99.32	99.41	99.36
Multi-Class	Benign	99.46	99.18	99.29	99.23
	DDoS	99.52	99.36	99.36	99.36
	Mirai	99.31	99.20	99.41	99.30
	Okiru	99.42	99.43	99.35	99.39
	Torii,	99.35	99.34	99.28	99.31
	Part of Horizontal-PortScan	99.27	99.52	99.27	99.39
	Heartbeat	99.54	99.43	99.65	99.54
	FileDownload	99.39	99.39	99.59	99.49
	Command and control	99.46	99.42	99.42	99.42

The performance indicators for binary and multi-class classification of the IoT23 dataset are compiled in Table 9. With F1-scores of 99.30% and 99.36%, respectively, the model obtained 99.38% accuracy for regular traffic and 99.32% accuracy for attacks in binary classification. The Heartbeat category received the greatest F1-score of 99.54% in multi-class classification, indicating the model's excellent performance against a variety of attack types. The algorithm can accurately identify both benign and harmful behaviours in Internet of Things environments, according to the overall results.

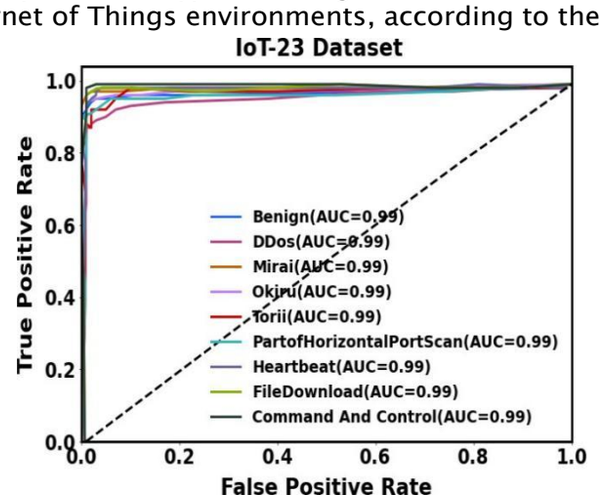


Fig. 17. Roc-Curve for IoT-23 dataset

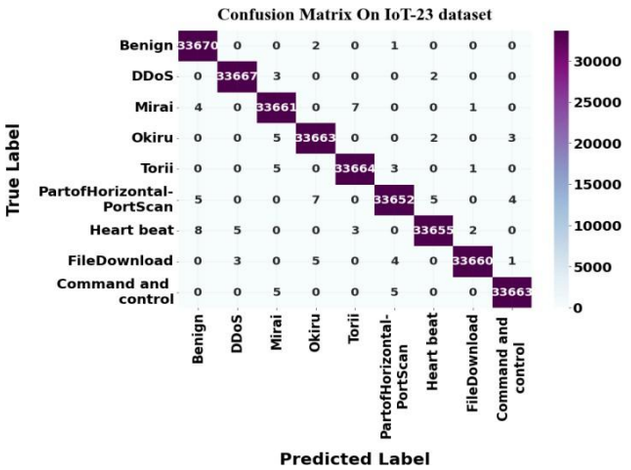


Fig. 18. Confusion matrix for IoT-23 dataset

Figures 17 and 18 show the Roc-Curve and confusion matrix for the IoT-23 dataset.

Performance analysis on the N-BaIoT dataset

Using the N-BaIoT dataset, we thoroughly evaluate our intrusion detection model in this section. Table 10 shows the performance differentiation of approaches using the N-BaIoT dataset.

Table 10. Multiclass differentiation of various approaches using the N-BaIoT dataset

Methods	Accuracy	Precision	Recall	F-Score
Ensemble Naïve Bayes	99%	-	-	-
Bat algorithm	90%	85%	90%	86.6%
KELM	99.4%	98.8%	99.25%	99.05%
Proposed	99.64%	99.52%	99.48%	99.50%

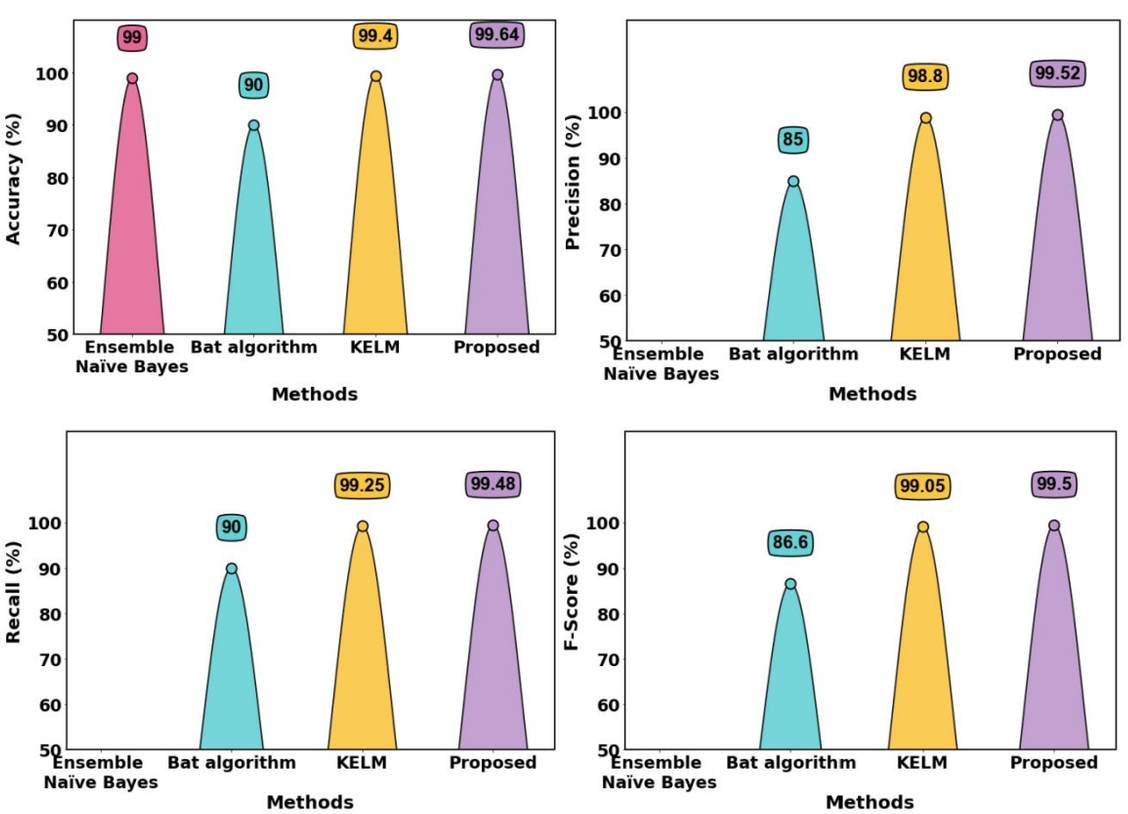


Fig. 19. Performance differentiation using N-BaIoT dataset

Using the N-BaIoT dataset indicates that the proposed method achieved high performance, with 99.64% Accuracy, 99.52% Precision, 99.48% Recall, and a 99.50% F-Score. KELM also performed well, with 99.4% Accuracy and an F-Score of 99.05%. The Bat algorithm showed moderate results, achieving 90% Accuracy, 85% Precision, and 90% recall, with an F-Score of 86.6%. The Ensemble Naïve Bayes method reported 99% Accuracy, although Precision, Recall, and F-Score metrics were not available. Overall, the proposed approach demonstrated superior effectiveness in classifying N-BaIoT data compared to other methods which is shown in Fig. 19.

Table 11. Binary and multi-class categorization

Types	Label	Acc (%)	Prec (%)	Rec (%)	FS (%)
Binary	Normal	99.29	99.36	99.25	99.30
	Attack	99.32	99.31	99.34	99.32
Multi-Class	benign,	99.41	99.42	99.45	99.43
	gafgyt.combo,	99.38	99.18	99.57	99.37
	gafgyt.junk,	99.42	99.20	99.21	99.21
	gafgyt.scan,	99.49	99.53	99.38	99.45
	gafgyt.tcp,	99.54	99.62	99.52	99.57
	gafgyt.ud,	99.46	99.43	99.76	99.59
	mirai.ack,	99.47	99.42	99.41	99.42
	mirai.scan,	99.59	99.54	99.39	99.46
	mirai.syn,	99.39	99.18	99.52	99.35
	mirai.udp,	99.54	99.32	99.51	99.41
	mirai.udpplain	99.35	99.35	99.32	99.34

The binary and Multi-class classification of the N-BaIoT dataset is shown in table 11.

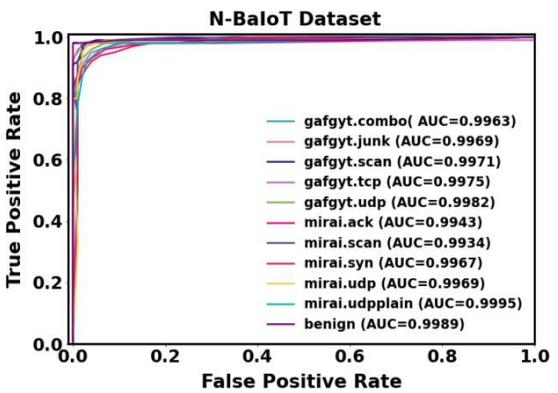


Fig. 20. Roc-Curve for N-BaIoT dataset

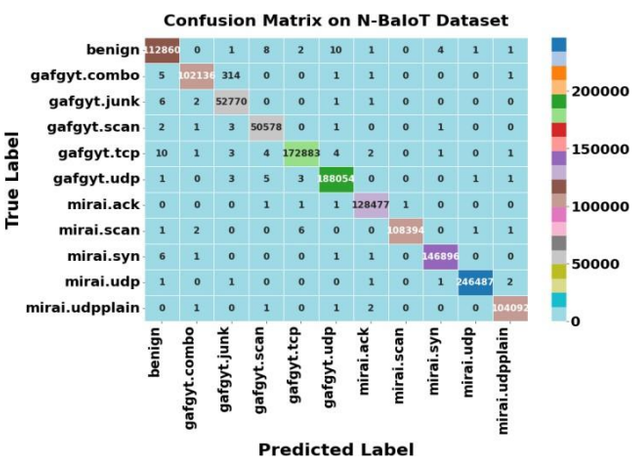


Fig. 21. Confusion matrix for N-BaIoT dataset

Figures 20 and 21 show the Roc-Curve and confusion matrix for the N-BaIoT dataset.

Overall Performance evaluation of the proposed approach

The thorough performance analysis of our proposed intrusion detection method using many datasets is provided in this part. The outcomes show how well our approach detects different types of attacks, which indicates that it can be used to improve security in IoT.

Table 12. Differentiation over Related Works

Reference	Method Used	Advantages	Disadvantages	Dataset Used	Accuracy
Aljehane et al. [21]	GJOADL-IDSNS	Effective recognition and classification of attacks	May require extensive computational resources	NSL-KDD, UNSW, CICiddos	96.54%
Hanafi et al. [22]	IBGJO with LSTM	Enhanced GJO efficiency through opposition-based learning	LSTM training can be time-consuming	ToN-IoT, BoT-IoT, KDD	97.08%
Hazman et al. [23]	IDS-SIoEL	Combines multiple feature selection techniques for robustness	Potential overfitting with ensemble methods	NSL-KDD, UNSW, CiCiddos 2019	98.21%
Sabitha et al. [24]	HNN-FPSO	High detection rate for IoT attacks	May struggle with complex patterns	Bot-IoT, ToN-IoT	99.25
Kasongo [25]	LSTM, GRU, Simple RNN	XGBoost reduces feature space, improving accuracy	Performance may decrease with increasing feature dimensions	KDD, UNSW, Ciciddos	98.74%
Proposed approach					99.71%

The differentiation between proposed with existing approaches is shown in table 13. Contrasting with existing methods proposed approach yields superior performances.

Table 13. Training and testing time comparison

Models	Training Time (sec)	Testing Time (sec)
GA+SVM	24.37	19.47
Modified KNN	10.33	6.23
CNN+LSTM	24.17	16.66
Proposed	5.01	1.96

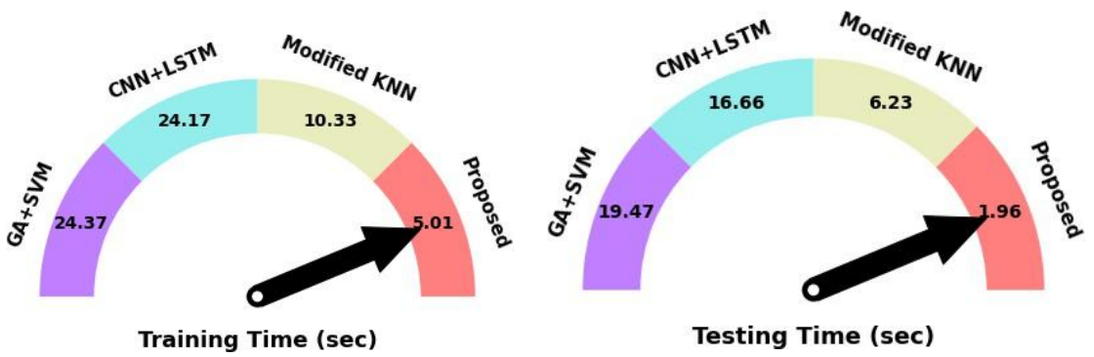


Fig. 22. training and testing time comparison

Training and testing time comparison for various models highlights the efficiency of the proposed method, which demonstrated the shortest training time of 5.01 seconds and testing time of 1.96 seconds. Modified KNN followed, with a training time of 10.33 seconds and a testing time of 6.23 seconds. In contrast, CNN+LSTM required 24.17 seconds for training and 16.66 seconds for testing, while GA+SVM took the longest, with 24.37 seconds for training and 19.47 seconds for testing. Overall, the proposed method proved to be the fastest in both the training and testing phases as shown in table 12 and Fig. 22.

5 | Conclusion and Future Scope

An intelligent network intrusion detection system is essential for protecting against various types of malicious attacks. The representation of network behaviour encompasses a wide range of features, including attack references, host details, and malicious scripts. However, typical network traffic can present an imbalanced variety of hazardous intrusions, making it challenging to develop a transfer

learning-based method for detecting specific types of network attacks due to complex attributes and data imbalance issues.

To address these challenges, we propose an Ensemble Optimized deep learning approach that significantly enhances the performance of Intrusion Detection Systems (IDS) in cloud computing environments. Our methodology begins with feature extraction using the Densely Connected Res2Net (DC-Res2Net), which captures multi-scale features, ensuring improved detection of both common and rare attacks while reducing redundancy for faster processing. The Genghis Khan Shark Optimizer (GKSO) is employed for effective feature selection, minimizing computational complexity while retaining high-quality data for accurate detection. The classification process is strengthened through the Ensemble of AlexNet CNN and EfficientDet (ANCNNED), which combines the strengths of both models to provide comprehensive coverage of security threats. Additionally, the Enhanced Zebra Optimization (EZO) Algorithm ensures optimal convergence during the training process, resulting in superior model performance. Experimental results demonstrate that our proposed model achieves over 99% accuracy, precision, and recall, effectively addressing the challenges of imbalanced data and complex attributes in network traffic. This robust solution significantly enhances the security of cloud infrastructures, providing a reliable defense mechanism against evolving cyber threats.

Our proposed IDS-INT method can be used on a variety of mobile devices, including Android and Windows-based network edges. In the future, we will try to develop a client-server communication model using an advanced deep learning approach, i.e., federated learning, to detect various types of attacks on network edges.

Declaration

Conflict of interests: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We declare that this manuscript is original, has not been published before and is not currently being considered for publication elsewhere.

Funding: No funding was received to assist with the preparation of this manuscript.

References

1. L. Narengbam, S. Dey, Harris hawk optimization trained artificial neural network for anomaly based intrusion detection system. Concurrency and Computation: Practice and Experience, 35(23) (2023) e7771. https://doi.org/10.1002/cpe.7771
2. B. Sharma, L. Sharma, C. Lal, S. Roy, Anomaly based network intrusion detection for IoT attacks using deep learning technique. Computers and Electrical Engineering, 107 (2023) 108626. https://doi.org/10.1016/j.compeleceng.2023.108626
3. M. H. L. Louk, B. A. Tama, Dual-IDS: A bagging-based gradient boosting decision tree model for network anomaly intrusion detection system. Expert Systems with Applications, 213 (2023) 119030. https://doi.org/10.1016/j.eswa.2022.119030
4. S. Zavrak, M. Iskefiyeli, Flow-based intrusion detection on software-defined networks: a multivariate time series anomaly detection approach. Neural Computing and Applications, 35(16) (2023) 12175-12193. https://doi.org/10.1007/s00521-023-08711-w
5. N. Hamdi, Federated learning-based intrusion detection system for Internet of Things. International Journal of Information Security, 22(6) (2023) 1937-1948. https://doi.org/10.1007/s10207-023-00727-6
6. M. Soltani, B. Ousat, M. J. Siavoshani, A. H. Jahangir, An adaptable deep learning-based intrusion detection system to zero-day attacks. Journal of Information Security and Applications, 76 (2023) 103516. https://doi.org/10.1016/j.jisa.2023.103516
7. Z. Li, P. Wang, Z. Wang, D. C. Zhan, Flowgananomaly: Flow-based anomaly network intrusion detection with adversarial learning. Chinese Journal of Electronics, 33(1) (2024) 58-71. **DOI:** 10.23919/cje.2022.00.173
8. M. J. Idrissi, H. Alami, A. El Mahdaouy, A. El Mekki, S. Oualil, Z. Yartaoui, I. Berrada, Fed-anids: Federated learning for anomaly-based network intrusion detection systems. Expert Systems with Applications, 234 (2023) 121000. https://doi.org/10.1016/j.eswa.2023.121000
9. R. Alghamdi, M. Bellaiche, A cascaded federated deep learning based framework for detecting wormhole attacks in IoT networks. Computers & Security, 125 (2023) 103014. https://doi.org/10.1016/j.cose.2022.103014
10. S. Bacha, A. Aljuhani, K. B. Abdellafou, O. Taouali, N. Liouane, M. Alazab, Anomaly-based intrusion detection system in IoT using kernel extreme learning machine. Journal of Ambient Intelligence and Humanized Computing, 15(1) (2024) 231-242. https://doi.org/10.1007/s12652-022-03887-w

11. H. Xu, Z. Sun, Y. Cao, H. Bilal, A data-driven approach for intrusion and anomaly detection using automated machine learning for the Internet of Things. Soft Computing, 27(19) (2023) 14469-14481. https://doi.org/10.1007/s00500-023-09037-4
12. M. Vishwakarma, N. Kesswani, A new two-phase intrusion detection system with Naïve Bayes machine learning for data classification and elliptic envelop method for anomaly detection. Decision Analytics Journal, 7 (2023) 100233. https://doi.org/10.1016/j.dajour.2023.100233
13. D. M. Ngo, C. Pham-Quoc, T. N. Thinh, (2020). Heterogeneous hardware-based network intrusion detection system with multiple approaches for SDN. Mobile Networks and Applications, 25(3), 1178-1192. https://doi.org/10.1007/s11036-019-01437-x
14. N. M. Yungaicela-Naula, C. Vargas-Rosales, J. A. Perez-Diaz, SDN-based architecture for transport and application layer DDoS attack detection by using machine and deep learning. IEEE Access, 9 (2021) 108495-108512. **DOI:** 10.1109/ACCESS.2021.3101650
15. G. O. Anyanwu, C. I. Nwakanma, J. M. Lee, D. S. Kim, Optimization of RBF-SVM kernel using grid search algorithm for DDoS attack detection in SDN-based VANET. IEEE Internet of Things Journal, (2022). **DOI:** 10.1109/JIOT.2022.3199712
16. A. Sebbar, K. Zkik, Y. Baddi, M. Boulmalf, M. D. Ech-Cherif El Kettani, Secure data sharing framework based on supervised machine learning detection system for future SDN-based networks. Machine Intelligence and Big Data Analytics for Cybersecurity Applications, (2021) 355-371. https://doi.org/10.1007/978-3-030-57024-8_16
17. V. Ravi, R. Chaganti, M. Alazab, Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system. Computers and Electrical Engineering, 102 (2022) 108156. https://doi.org/10.1016/j.compeleceng.2022.108156
18. I. Aliyu, M. C. Feliciano, S. Van Engelenburg, D. O. Kim and C. G. Lim, A blockchain-based federated forest for SDN-enabled in-vehicle network intrusion detection system. IEEE Access, 9 (2021) 102593-102608. **DOI:** 10.1109/ACCESS.2021.3094365
19. H. Aldabbas, R. Amin, A novel mechanism to handle address spoofing attacks in SDN based IoT. Cluster Computing, 24(4) (2021) 3011-3026. https://doi.org/10.1007/s10586-021-03309-0
20. R. M. A. Ujjan, Z. Pervez, K. Dahal, A. K. Bashir, R. Mumtaz, J. González, Towards sFlow and adaptive polling sampling for deep learning based DDoS detection in SDN. Future Generation Computer Systems, 111 (2020) 763-779. https://doi.org/10.1016/j.future.2019.10.015
21. N. O. Aljehane, H. A. Mengash, M. M. Eltahir, F. A. Alotaibi, S. S. Aljameel, A. Yafoz, M. Assiri, Golden jackal optimization algorithm with deep learning assisted intrusion detection system for network security. Alexandria Engineering Journal, 86 (2024) 415-424. https://doi.org/10.1016/j.aej.2023.11.078
22. A. V. Hanafi, A. Ghaffari, H. Rezaei, A. Valipour, B. Arasteh, Intrusion detection in Internet of things using improved binary golden jackal optimization algorithm and LSTM. Cluster Computing, 27(3) (2024) 2673-2690. https://doi.org/10.1007/s10586-023-04102-x
23. C. Hazman, A. Guezzaz, S. Benkirane, M. Azrour, IIDS-SIoEL: intrusion detection framework for IoT-based smart environments security using ensemble learning. Cluster Computing, 26(6) (2023) 4069-4083. https://doi.org/10.1007/s10586-022-03810-0
24. R. Sabitha, S. Gopikrishnan, B. J. Bejoy, V. Anusuya, V. Saravanan, Network based detection of IoT attack using AIS-IDS model. Wireless Personal Communications, 128(3) (2023) 1543-1566. https://doi.org/10.1007/s11277-022-10009-4
25. S. M. Kasongo, A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework. Computer Communications, 199 (2023) 113-125. https://drg/10.1016/j.comcom.2022.12.010