IJIE

INTERNATIONAL JOURNAL OF INTEGRATED ENGINEERING
ISSN: 2229-838X e-ISSN: 2600-7916

Vol. 18 No. 4 (2026) 111-129
https://publisher.uthm.edu.my/ojs/index.php/ijie

# Self-Adaptive Intrusion Detection Framework for IoT Networks Using Addax Optimization and Echo State Networks

**Mohd Abdul Rahim Khan[1]***

[1] *Department of Electrical Engineering and Computer Science, College of Engineering, A'sharqiyah University, Ibra, 400, SULTANATE OF OMAN*

*Corresponding Author: mohd.khan@asu.edu.om
DOI: https://doi.org/10.30880/ijie.2026.18.04.008

**Article Info**

Received: 27 December 2025
Accepted: 27 April 2026
Available online: 22 June 2026

**Keywords**

Internet of Things, sophisticated attacks, deep denoising autoencoders, tune hyperparameters, intrusion detection

**Abstract**

The rapid expansion of the Internet of Things (IoT) has rendered its networks vulnerable to sophisticated attacks, necessitating the employment of improved intrusion detection techniques. This paper presents an innovative approach to IoT intrusion detection that combines advanced preprocessing approaches, hyperparameter optimization, and effective classification algorithms. The preprocessing pipeline employs Deep Denoising Autoencoders (DDA) to remove noise from IoT sources, Kalman Filters (KF) to smooth temporal data, and Dynamic Range Compression (DRC) to standardize sensor readings. The framework dynamically adjusts the hyperparameters and integrates the Self-Adaptive Addax Optimization Algorithm (AOA) to select the optimum model architecture for various attack situations, resulting in optimal model performance. This involves integrating ESNs, GRUs, and GANs into a single network architecture through the use of a attack categorization model. Its primary impact is to strengthen the system as a whole while maintaining precise detection even at intricate levels and with a high degree of pattern flexibility. The system achieves outstanding performance with 98.45% accuracy, 97.92% precision, 98.65% recall, and an F1 score of 98.27% when evaluated on the NSL-KDD dataset. Scalable, adaptable, and effective intrusion detection systems are provided by the integrated solution, which has resolved the security challenges posed by dynamic IoT networks.

## 1. Introduction

The IoT has revolutionized several industries by enabling real-time data interchange, connecting devices, and boosting operational efficiency [1]. However, because of this increased connection, IoT networks are now more susceptible to sophisticated cyberattacks, such as malware penetration, denial-of-service (DoS) assaults, and data breaches [2]. The dynamic and diverse nature of IoT networks frequently makes it difficult for traditional intrusion detection systems (IDS) to keep up, necessitating the urgent need for cutting-edge solutions that can recognize and neutralize changing threats [3]. Innovative approaches to intrusion detection that incorporate sophisticated algorithms, flexible structures, and high-performance models are necessary to provide safe communication and uninterrupted service as IoT networks manage enormous volumes of data in real-time [4]. By utilizing cutting-edge machine learning techniques and optimization methodologies to counter a variety of

intricate attack vectors, this study tackles the requirement for a reliable, scalable IDS framework designed for IoT contexts.

IoT networks' attack surfaces increase as a possible target for hackers due to their growing scale, complexity, and interconnection [5]. When faced with the challenges of an IoT environment such as the fluctuation of data patterns the devices limited resources, and the evolution of attack strategies traditional intrusion detection systems [6]. Current IDS approaches frequently lack the flexibility to adequately handle the wide variety of IoT devices and the constantly changing attack patterns [7-8]. Additionally, noisy high-dimensional data from several IoT sources is difficult for traditional models to understand, leading to poor performance. In this study creates a sophisticated self-adaptive IDS system that incorporates state-of-the-art methods like Addax Optimization and hybrid neural networks to overcome these constraints and improve detection efficiency and accuracy over dynamic IoT networks.

Developing an intrusion detection system for Internet of Things networks is fraught with difficulties [9-10]. IoT settings provide large, varied, and frequently noisy data streams that need to be pre-processed using advanced methods to identify trends. The construction of a universal detection model is made more difficult by the wide variations in processing power, communication protocols, and security setups among IoT devices [11-12]. IDS frameworks must constantly evolve due to the dynamic nature of cyberattacks to properly identify new threats [13]. Additionally, as IoT devices frequently have limited resources, striking a balance between detection accuracy and computing efficiency is crucial. To handle the quick expansion of IoT ecosystems without sacrificing security or functionality, intrusion detection solutions must be scalable [14].

The Self-Adaptive Intrusion Detection Framework for IoT Networks presented in this paper uses hybrid classification models, optimization, and sophisticated preprocessing to overcome these difficulties. DRC, KF, and DDA are all used in the preprocessing pipeline to deal with noisy and diverse data. Model hyperparameters are dynamically adjusted by the AAOA, guaranteeing peak performance in a range of attack situations. To improve the system's accuracy, resilience, and flexibility, a hybrid model that combines ESN, GRU, and GAN is used for classification. Utilizing the NSL-KDD dataset for evaluation, the framework outperforms current techniques in detecting a wide range of attack types, providing a scalable and effective means of protecting IoT networks. The Contribution of study is given below:

- To develop a preprocessing pipeline that uniformizes sensor readings in IoT networks, smoothes temporal data, and effectively reduces noise using dynamic range compression, Kalman filters, and deep denoising autoencoders.
- The best intrusion detection system should be optimized and its hyperparameters dynamically adjusted to provide flexibility against various IoT attack types.
- It is possible to detect dynamic attack patterns and complicated attack signatures by using hybrid classification with ESN, GRU and GAN.
- Verify the suggested intrusion detection framework's capacity to identify a variety of attack types and outperform the current techniques in terms of performance metrics using the NSL-KDD dataset.
- Creating a resource-efficient intrusion detection system that strikes a compromise between high-performance real-time detection on dynamic networks and the processing constraints of IoT devices.

The paper is organized into the following sections: Section 2 examines pertinent studies and reviews of literature; Section 3 presents the suggested framework; Section 4 includes a thorough examination of the findings and debates; and Section 5 presents the study's final evaluation.

## 2. Literature Review

In Keshk et al. [15] investigated to comprehend how well attack surface and vector identification work. The new uses of machine/deep learning models in cyber defense, particularly anomaly-based intrusion detection systems, need analyzing the models' architecture and elucidating their predictions to investigate the potential entry path. A novel explainable intrusion detection method for IoT networks is presented in this study. To detect cyberattacks and explain the model's behaviour, they have created an IDS that uses an LSTM model.

Silivery et al., [16] introduced innovative DL techniques. Developing a trustworthy intrusion detection system to assist in spotting malicious attempts is the goal here. Three methods make up the created DL-based solution framework. Seven optimizer functions—adamax, SGD, adagrad, adam, RMSprop, nadam, and adadelta—are included in the first method, LSTM-RNN. The NSL-KDD dataset and multi-attack classification are used to assess the model. In terms of accuracy, detection rate, and low false alarm rate, the model has done better than the Adamax optimizer.

Sajid et al., [17] analyse the advancement of communication technology, network systems must transport heterogeneous and varied data in scattered settings while avoiding security flaws and dangers. This work addresses these issues by integrating ML and DL methods to develop a hybrid model for ID. For feature extraction, the suggested model uses CNN and XGBoost, which are then combined with LSTM for classification. This study

made use of four benchmark datasets. For each distinct model in this study, CNN and XGBoost feature selection techniques are used to decrease the feature space of each dataset.

Elsaid et al., [18] assessed Higher processing complexity is the result of classic IDSs incapacity to manage duplicate characteristics. To overcome these problems, this work suggests two enhanced IDSs that combine DL models with GWO. The first system combines GRU-GWO, whereas the second system employs LSTM-GWO. These methods aim to enhance feature selection by reducing dimensionality and raising detection accuracy. The NSL-KDD and UNSW-NB15 datasets, which are representative of contemporary network environments, were used to test the proposed solutions.

Huma et al., [19] analyzed to the use of conventional Internet of Things principles in industrial settings. Smart cities, smart homes, linked automobiles, supply chain management, and smart grids are a few of the IIoT's uses. For IIoT networks, strong security procedures may be developed and put into place using deep learning and big data analytics. In this paper, a unique HDRaNN for IIoT cyberattack detection is described. A deep random neural network and multilayer perceptron with dropout regularization are combined in the HDRaNN. DS2OS and UNSW-NB15, two datasets pertaining to IIoT security, are used to assess the suggested methodology.

Ahmad et al. [20] proposed that the vast volume of data generated by IIoT devices necessitates the use of clever data processing methods to create cybersecurity defenses. This is where DL might be a good option. In this study, a rapid and dependable attack detection strategy for IIoT contexts based on DRaNN is proposed. An improved version of the classic ANN, the RaNN is more widely dispersed and has superior generalization abilities. Using hybrid PSO and SQP, the suggested RaNN is optimally trained to achieve greater attack detection accuracy.

Shahin et al. [21] researched IoT devices have been widely employed and technologically advanced to monitor, collect share, analyze, and transmit data in manufacturing settings. To protect industrial Internet of things equipment from cyberattacks, effective intrusion detection solutions that use deep learning algorithms have since been developed. In this study, three benchmark industrial IoT datasets (BoT-IoT, UNSW-NB15, and TON-IoT) that employ different deep learning approaches to address cyber-security challenges are used to test two different classification and detection utilizing the LSTM architecture.

Abassi et al., [22] investigated DSs, designed mainly to support IT systems, are trained on specific cyberattacks and mostly depend on pre-existing models. To detect cyberattacks from the new representations, the attack detection model introduced in this study employs DT and DNN classifiers. The performance of the proposed model is evaluated on two real ICS datasets using ten-fold cross validation.

Nizamudeen [23] introduced a sophisticated IDF to identify threats based on applications and networks. Three phases make up the suggested framework: feature selection, categorization, and data pre-processing. To guarantee a fair-scale data transformation procedure, the gathered datasets are pre-processed using the I-GN approach. The second objective of the feature selection phase is the OBL-RIO. Rats' progressive nature selects the essential characteristics. The stability of the OBL-RIO characteristics is guaranteed by the fittest value. Lastly, the suggested binary class classifier is a 2D-ACNN.

Latif et al., [24] proposed an enhanced intrusion detection system designed especially for their diverse IIoT networks and based on DTL. The tri-layer architectural approach of our framework synergistically integrates CNNs, GA and bootstrap aggregation ensemble approaches. Three crucial steps make up the technique's implementation: We first turn the cutting-edge cybersecurity dataset, Edge IIoTset, into picture data to make CNN-based analysis easier. Second GA is used to adjust each basic learning model's hyperparameters, improving the model's performance and adaptability.

The IoT network is becoming more active, extensive, and diverse because of its quick growth across several domains, which increases the possibility of cyberattacks. Their incapacity to manage high-dimensional, loud, and unbalanced data from IoT devices is one of the many problems with current IDSs. Conventional intrusion detection systems rely on static models and pre-existing attack signatures. APTs and zero-day attacks are examples of more complex and dynamic threats that traditional models cannot address. Despite the existence of some ML and DL-based methods, they typically fail to scale, adapt, or endure in practical IoT scenarios. To achieve high detection accuracy while preserving processing efficiency, the methods do not dynamically modify their architectures and hyperparameters. Closing this gap requires a new intrusion detection system that is dependable, adaptable, and tailored to the unique challenges posed by IoT networks.

## 3. Research Methodology

This research uses a multi-stage approach to provide an improved intrusion detection framework for IoT networks. This includes a preprocessing technique that employs DDA to reduce noise, KF to smooth out temporal data, and DRC to normalize sensor values. The framework uses the Self-adaptive Self-Adaptive AOA to dynamically alter hyperparameters to determine the optimal model architecture for intrusion detection across a range of attack situations. To classify attacks, a hybrid model that incorporates ESN, GRU, and GAN is used. The system is developed using the NSL-KDD dataset and outperforms existing approaches, especially in identifying a broad variety of attack types.

**Table 1** *Research gap from the existing work*

| Author & Year | Aim | Technique Used | Findings |
|---|---|---|---|
| Keshk et al., [15] | To propose an explainable IDS that identifies cyberattacks and explains the decisions of the model. | LSTM model. | Developed an IDS using LSTM, enhancing interpretability and detection of cyberattacks. |
| Silivery et al., [16] | To create a reliable intrusion detection mechanism using deep learning methodologies for identifying malicious attacks. | LSTM-RNN with seven optimizer functions (adamax, SGD, adagrad, adam, RMSprop, nadam, adadelta); evaluated on NSL-KDD dataset. | Adamax optimizer outperformed others in terms of accuracy, detection rate, and false alarm rate. |
| Sajid et al., [17] | To tackle security limitations in network systems with a hybrid model combining ML and DL techniques. | Hybrid model: XGBoost and CNN for feature extraction, combined with LSTM for classification; evaluated on four benchmark datasets. | Improved feature selection and dimensionality reduction, enhancing detection accuracy across various datasets. |
| Elsaid et al., [18] | To address computational complexity and redundant features in traditional IDSs. | Optimized IDSs using GRU-GWO and LSTM-GWO evaluated on NSL-KDD and UNSW-NB15 datasets. | Reduced dimensionality and improved detection accuracy, addressing challenges in handling redundant features. |
| Huma et al. [19] | To apply IIoT concepts in industrial applications and develop robust security mechanisms using deep learning and big data analytics. | HDRaNN combining Deep Random Neural Network and Multilayer Perceptron with dropout regularization; evaluated on DS2OS and UNSW-NB15 datasets. | Enhanced security for IIoT networks with improved cyberattacks detection. |
| Ahmad et al., [20] | To process massive IIoT data intelligently and develop a fast, reliable attack detection scheme. | DRaNN optimized with PSO and SQP. | Improved detection accuracy by combining DRaNN with hybrid optimization techniques. |
| Shahin et al., [21] | To develop effective IDSs for Industrial IoT devices using deep learning algorithms. | LSTM-based IDS; evaluated on BoT-IoT, UNSW-NB15, and TON-IoT datasets. | Reliable detection and classification of cyberattacks on Industrial IoT devices. |
| Abassi et al., [22] | To enhance traditional IDSs that rely on predefined models and detect novel cyberattacks. | Attack detection model combining DNN and DT; evaluated using 10-fold cross-validation on two ICS datasets. | Detected novel cyberattacks with improved accuracy using a hybrid model. |
| Nizamudeen, [23] | To develop an intelligent IDF for network and application-based attacks. | Framework with three phases: I-GN for preprocessing, OBL-RIO for feature selection, and 2D-ACNN for classification. | Achieved stability and efficiency in feature selection and binary classification of network attacks. |
| Latif et al., [24] | To design an optimized IDS for heterogeneous IIoT networks using DTL. | Tri-layer approach combining CNNs, GA and bootstrap aggregation; Edge_IIoTset dataset converted to image data for CNN-based analytics. | Enhanced model adaptability and performance, achieving high detection rates for IIoT networks. |

### 3.1.1 Data Set Description

One of the popular benchmark datasets for network security is NSL-KDD, which was created specifically to assess IDS. Along with other elements that depict different aspects of network packets and connections, it replicates actual network traffic. Among the essential characteristics that make the data set extremely feasible for both validations and training intrusion detection models are packet length, type, protocol, service, and others. Redundancy and duplicate records were among the flaws of the KDD dataset 99, the data set's predecessor, that the present data set fixes. It includes labelled data that is divided into regular and several forms of attacks, including DoS, Probe, U2R, and R2L. Researchers may create models that can recognize and categorize both frequent and complex incursions by including a variety of assault behaviors. The dataset link: https://www.kaggle.com/datasets/hassan06/nslkdd/data

### 3.1.2 Preprocessing

To make sure the data is in the proper format for deep learning models, we preprocess it using several techniques. KF smooth temporal data, DDA reduce noise from IoT sources, and DRC standardizes sensor inputs. By improving the data's suitability for deep learning models, these methods increase efficiency and performance. By improving the data's suitability for deep learning models, these techniques boost effectiveness and performance.

## 3.2 Deep Denoising Autoencoders

The fundamental principle of the DAE is identifying the main source of the input, whether it's a corrupt or bad one. Thus, it could have an excellent representation from raw or attacked data, be able to discover better discriminative representations of latent-space and avoid learning an identity mapping between input and reconstruction in an AE. There are two common types of noise applied in the corrupting process, and they include additive Gaussian noise (GDAE) and zero-masking noise (ZDAE). The encoder, code/latent-space, and decoder form the structure of the DAE, same with an AE. From a given input x, a corrupted/attacked input data, $\tilde{y}$ instead of original input data, x, is encoded into a hidden or latent-space encoding, h.

$$h = f(W_1 y + c) \tag{1}$$

Where f(.) is an example of a nonlinear activation function. .$W1 \in IR^{m \times n}$is the weight matrix and $b \in IR^m$ is optimized for the m node latent space encoding. This hidden space is then transformed into the recovered vector by using the decoder through nonlinear activation as shown in Eqn. (2).

$$\rho = \hat{y} = g(W_1 y + d) \tag{2}$$

where $g(.)$ is an activation function that is nonlinear like the sigmoid function. The linked weights were used to improve learning efficiency. as $W_1 = WT_2$ . The reconstruction error can be computed for a given input training set, $\{x_i\}_{i=1}^m$, as Eqn (3)

$$\rho \sum_{i=1}^m \|y_i - \hat{y}_i\|^2 \tag{3}$$

The DAE's training goal is to identify the ideal parameters $\psi = \{W_1, c, d\}$ that minimize the reconstruction error, as given Eqn (4)

$$\min_{\psi} \sum_{i=1}^m \|y_i - \hat{y}_i\|^2 \tag{4}$$

The equation explicitly defines the reconstruction error as the difference between the real meter readings and the rebuild output, not the attacked measurements. DAE is taught to provide output that is more akin to the original, in other words input $x$ even while utilizing the attacked input, $\tilde{y}$ .

## 3.3 Kalman Filters

For achieving the optimum result, the Kalman filter was designed to include the effects of the noisy measurements in a recursive least squares adjustment process. In combination, the two models form a linear dynamical system where the state vector from the previous epoch i-1 along with the process model information is integrated with the (noisy) observations to yield the state vector $X_i$of the epoch i. The observations utilized by the Kalman filter are included in the observation model from which the Kalman filter itself is computed. The observations li are related to the unknown parameters $X_i$that describe the gravity field at epoch i through the matrix Ai in the following system of linear equations, Eqn. (5).

$$I_i = A_i X_i + V_i \text{ with } v_i \sim N(0, R_i) \quad (5)$$

The covariance matrix of the corrections that can be used is given by $R_i$, and it is assumed that the expected value of the corrections $v_i$ will be 0. The matrix $A_i$ consists of $\Phi k\ (x, x_k)$, the observation covariance matrix$R_i$ consists of the Gaussian filtered covariance matrix of the GRACE grid values, and the computed time series are the observations li used in this case. This Kalman filter uses geophysical models that are stochastic in nature and try to represent the temporal behavior of the Earth's gravity field. It assumes that the behavior of the gravity field will be predictable within a particular integration period rather than random at each epoch $x_{i-1}$ to the succeeding epoch $x_i$ . Therefore, a discrete first order Markov process is used based on Eqn. (6).

$$x_i = Bx_{i-1} + w \text{ with } w \sim N(0, Q) \quad (6)$$

The errors of the model regarding the predicted 0 and covariance matrices Q are defined in terms of the constant-state transition matrix B of the state vector that is modeled and the respective noise process w. The entire correlation structure at the time step i - 1 and i is used to derive the state transition matrix B and the covariance matrix Q of the noise process.

$$\left\{\begin{pmatrix} x_i \\ x_{i-1} \end{pmatrix}\right\} = \begin{pmatrix} \Sigma & \Sigma_\Delta \\ \Sigma_\Delta^T & \Sigma \end{pmatrix} \quad (7)$$

Where, Σ is the auto covariance matrix calculation given Eqn (8)

$$\Sigma \coloneqq C\{x_i, x_i\} \quad (8)$$

It shows the spatial variation in the estimated gravity field, while the cross-covariance matrix $\Sigma_\Delta$ shows the temporal variation at each successive step as follows:

$$\Sigma_\Delta \coloneqq C\{x_i, x_{i-1}\} \quad (9)$$

B, the state transition matrix, is defined by Eqn. (10).

$$B = \Sigma_\Delta \Sigma^{-1} \quad (10)$$

This is based on the premise that the covariance matrix associated with the error vector of estimator B in Eq. 11 attains its minimum value. Secondly, Q represents the covariance matrix of the process noise w.

$$Q = \Sigma - \Sigma_\Delta \Sigma^{-1} \Sigma_\Delta^T \quad (11)$$

Because the exact values of the states of the gravitational field of the Earth cannot be determined, Σ and $\Sigma_\Delta$can only be estimated from empirical geophysical data. This can be done by utilizing the residual ESA ESM time series $\tilde{V}_I$as presented in Equation 1. The time series will again be filtered using a Gaussian filter of 300 km and will be represented in terms of the radial basis function scaling coefficients ai (where each vector ai consists of the K scaling coefficients ak). The time series will therefore consist of I state vectors ai representing the scaling coefficients:

$$x_i \approx m_i = a_i^T \quad (12)$$

From these state vectors at time points i, the empirical auto covariance matrix

$$\rho\Sigma \approx \bar{\Sigma} = \frac{1}{I} \sum_{i=1}^{1} m_i m_i^T \quad (13)$$

and empirical cross covariance matrix

$$\Sigma_\Delta \approx \bar{\Sigma}_\Delta = \frac{1}{I-1} \sum_{i=2}^{1} m_i m_{i-1}^T \quad (14)$$

## 3.4 Dynamic Range Compression

DRC is a widely used audio dynamics control technique. It is widely utilized in live performances, radio transmission, sound recording, and music production. A basic compressor's job is to reduce the loud portions of the sound and increase the quiet ones, resulting in a reduced dynamic range. For any transmission channel with a bandwidth restriction, this can help lower the coding rate or improve the perceived loudness. A compressor can be made to amplify quiet sounds that fall below a given threshold or to compress loud sounds that exceed a specific threshold. The two types are referred to as upward and downward compressors, respectively.

Int. Journal of Integrated Engineering Vol. 18 No. 4 (2026) p. 111-129

$$G = \begin{cases} G1 & if \quad X < threshold \\ G2 * (X - L) & if \quad X \geq threshold \end{cases} \tag{15}$$

$$ratio = \frac{G2}{G1}, normally\ G1 = 1, \ \ so\ ratio = G2 \tag{16}$$

Where G1 and G2 are signal gain, L is threshold in dB scale, and X is signal magnitude. Attack time and release time are also two key variables of the compressor. These two variables determine the speed at which the compressor reacts to changes in sound levels or recovers back. If the compressor reacts directly to the sound as soon as it crosses the threshold, then this may cause some unwanted effects. Attack time can be referred to as the time taken by the compressor to cross the threshold. The compression's approximate recovery time is known as the release time. Eq.2,3, where τ is the attack or release time, illustrates how these time constants are often transmitted through a smoothing factor in implementation. Another often used parameter that can be added to the compressor's output to modify the total energy level is make-up gain. Understanding the fundamental workings of the DRC is not the same as being able to use this audio effect

$$\beta = 1 - \exp[-2.2/(f_s * \tau)] \tag{17}$$

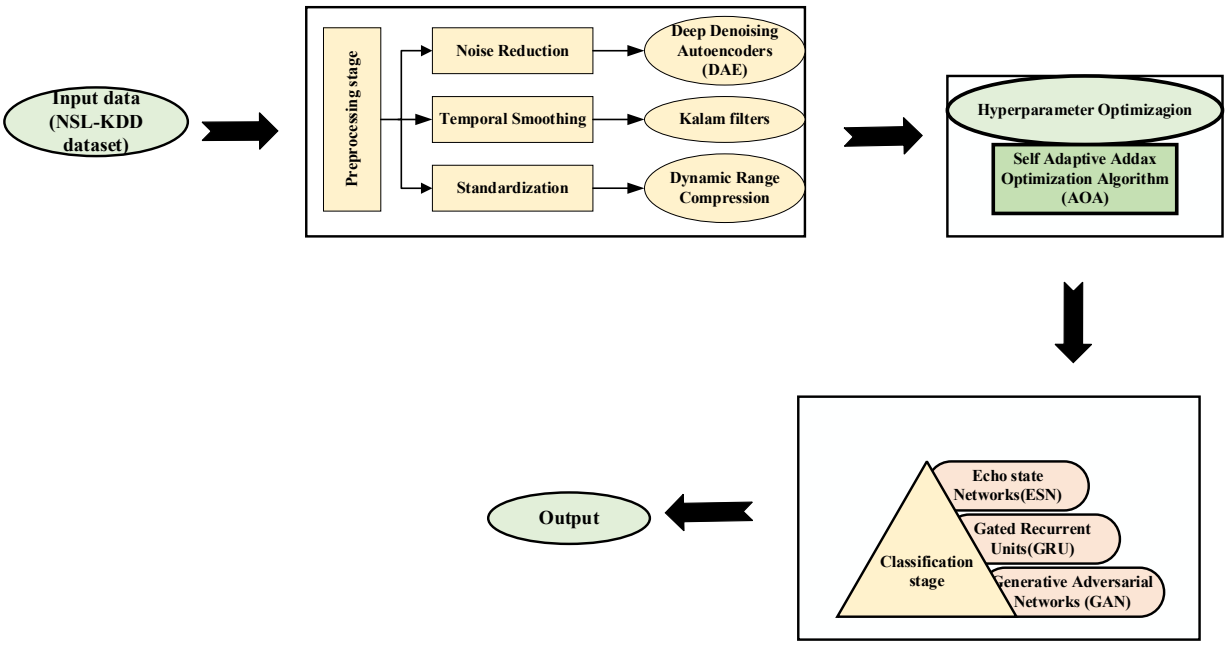


**Fig. 1** *Conceptual framework*

## 3.5 Self-Adaptive Addax Optimization Algorithm (SA-AOA)

The framework uses the Self adaptive Self-Adaptive Addax Optimization Algorithm (AOA) to dynamically tune hyperparameters, guaranteeing that the most effective model architecture for intrusion detection is chosen over a wide range of attack situations. It is used to enable systems to adapt autonomously to changing network conditions and resource requirements. It reduces the need for manual intervention, enhances efficiency, and improves performance by continuously learning from the environment. This approach is particularly useful in dynamic, large-scale systems like 6G, ensuring real-time optimization.

### 3.5.1 Initialization

The suggested AOA method, the population-based metaheuristic algorithm, portrays members of the population as addaxes. Through an iterative procedure, AOA efficiently handles optimization problems. Each addax sets design variable values according to its position in the problem-solving space. The position of each addax, represented as a vector equal to the number of choice variables, represents a potential solution. The AOA population, consisting of addaxes and their associated vectors, can be modeled as a matrix using Eq. (18). The initial placement of addaxes is randomly initialized according to Eq. (19).

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N\times m} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,d} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ x_{i,1} & \cdots & x_{i,d} & \cdots & x_{i,m} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,d} & \cdots & x_{N,m} \end{bmatrix}_{N\times m} \tag{18}$$

$$x_{i,d} = lb_d + r \cdot (ub_d - lb_d) \tag{19}$$

.The population matrix of AOA is denoted by $X$, where the ith addax (possible solution) is represented by $X_i$and its dth dimension (decision variable) in the search space is indicated by $x_{i,d}$, N is the population's addaxes, m is the number of decision variables, and r is a random value between 0 and 1. In Eq. (19), lb_d and ub_d stand for the lower and upper bounds of the dth choice variable, respectively.

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N\times 1} \tag{20}$$

$F$ represents the vector of assessed objective functions, with $F_i$ denoting the evaluation of the objective function for the $ith$ addax. These computed values serve as performance indicators for assessing each addax's potential solution. The best calculated objective value corresponds to the top addax while the worst aligns with the poorest-performing addax. AOA iteratively updates Addax positions, ensuring the best member remains the prime candidate solution in each iteration. Mathematical Modelling: The suggested AOA approach is a population-based metaheuristic algorithm that effectively detects rainfall and identifies foraging areas with higher flora, drawing inspiration from the addax's natural behavior. To create suitable resting spots and sandstorm shelters, addaxes are also skilled at digging. These behaviors are fundamental to the design of AOA, in which each addax updates its position in the problem-solving space through recurrent rounds of exploration (based on foraging) and exploitation (based on digging). These phases are mathematically modeled and thoroughly described.

Foraging process (Exploration phase): Based on addaxes extended foraging behavior in dry environments, the first phase of AOA models how colony members' locations are adjusted. Addaxes use their abilities to follow rainfall and choose regions with an abundance of vegetation to eat on grasses, shrubs, legumes, herbs, and bushes. Each Addax searches independently despite swarming tendency, which results in notable location shifts. These modifications improve the algorithm's capacity for worldwide investigation. In AOA, addaxes seek areas with better objective function values, using these as guiding locations, as defined by Eq. (21).

$$CA_i = \{X_k : F_k < k \neq i\} \text{ where } i = 1,2,\dots,N \text{ and } k \in \{1,2,\dots,N\} \tag{21}$$

$CA_i$ represents the possible locations for foraging by the $ith$ addax, and $X_k$ is a member of the population that has an objective function value better than that of the $ith$ addax, with$F_k$being the objective function value of $X_k$. The foraging operation of the AOA algorithm involves an addax choosing any one of these locations for foraging randomly. Using Eq. (18), the new location of each addax is determined based on the foraging operation in Eq. (20). The improved autonomous exploration and exploitation by presenting a self-adaptive method that boosts search efficiency. Through regular modification of search parameters and random exploration of other regions, it ensures that the algorithm adapts to changing conditions, effectively avoiding local optima and converging to global solutions.

$$Cx_{i,j}^{p1} = x_{i,j} + F \cdot r_{i,j} \cdot \left(SA_{i,j} - I_{i,j} \cdot x_{i,j}\right) \tag{22}$$

$$\text{where } \boldsymbol{F} = \frac{\mathbf{1}}{\mathbf{1}+\boldsymbol{exp}\left(\propto * \frac{\mathbf{1}-\frac{\boldsymbol{NP}}{\mathbf{2}}}{\boldsymbol{NP}}\right)}$$

$$X_i = \begin{cases} X_i^{p1}, & F_i^{p1} \leq F_i \\ X_i, & else, \end{cases} \tag{23}$$

$SA_i$ is the foraging place selected by the ith addax, and $SA_{i,j}$ is the jth dimension of that foraging place. $X_i^{p1}$ is the new location of the ith addax after foraging, where $x_{i,j}^{p1}$ is the jth dimension of it. $F_i^{p1}$ indicates the function value of the ith addax. Further, $r_{i,j}$ is a set of random numbers in [0, 1], while $II_{i,j}$ refers to randomly chosen 1 or 2.

### 3.5.2 Digging Skill (Exploitation Phase)

During the second phase of AOA, the population members move their positions through the emulation of the sand digging process undertaken by the addaxes. Addaxes make depressions in shaded regions of the desert as resting places and shelters from sandstorms. The minor movement that comes about because of this is simulated in AOA to improve the algorithm's exploitation performance during local search. In AOA, guided by the simulated movements of addaxes due to the digging process, the new position of everyone in the population is calculated based on Eqn. (24).

$$x_{i,j}^{p2} = x_{i,j} + F \cdot (1 - 2r_{i,j}) \cdot \frac{ub_j - lb_j}{t} \tag{24}$$

$$X_i = \begin{cases} X_{i,j}^{p2}, & F_i^{p2} \leq F_i \\ X_i, & else \end{cases} \tag{25}$$

$X_i^{p2}$ denotes the new calculated position of the $ith$ addax during the digging stage, wherein $X_i^{p2}$ denotes the jth dimension of that addax. On the other hand, $F_i^{p2}$ is the objective function value for the ith addax, $r_{i,j}$ are random values in the range of [0, 1] and t is the iteration number in Eqn. (26).

$$Q(s_t, a_t) = r_t + \gamma \max_a Q'(s_{t+1}, a) * X_i \tag{26}$$

AOA is resilient in changing contexts because of its self-adaptive nature, which enables the algorithm to constantly enhance its search method without requiring outside assistance. Better performance is guaranteed by this flexibility, particularly in challenging optimization tasks like hybrid beamforming in communication systems.

## 3.6 Classification

The most advantageous aspects of GAN, GRU, and ESN are combined in an attack categorization. To effectively capture the historical patterns and temporal relationships in the input data, ESN has been employed as a dynamic reservoir. The accuracy of the predictions would then increase because of GRUs refining these characteristics through their gating mechanisms in modelling sequential dependencies. High-quality synthetic datasets are created using GANs to improve the model's capacity to generalize over a variety of unknown assault patterns. High accuracy and adaptability in the detection system are made possible by the model's structure, which enables it to robustly detect dynamic assault behaviors and complex and evolving attack signatures. This makes the model suitable for advanced threat detection systems operating in dynamic environments.

### 3.6.1 Echo State Networks (ESN)

The dynamical reservoir, a recurrent topology of nonlinear PEs, and the readout, a memoryless linear network, are two distinct components of the RNN architecture that are separated by the ESN approach. The dynamical reservoir's echo states the input pattern's history. After reading the reservoir and obtaining the outputs from the internal PEs, a readout network creates the network output. ESNs are interesting because they only train the memoryless linear readout, as opposed to recurrent systems that have fixed connections. The neural network is a recurrent discrete-time neural network that has M input neurons, N internal processing elements (PEs), and L output neurons. The input neuron's value at time n is u(n), the internal neurons' value is x(n), and the output neurons' value is y(n). The activation of the internal Pes (echo states) is updated according to:

$$x(n+1) = f(W^{in}u(n+1) + Wx(n)) \tag{27}$$

If x(n) represents echo state vector, W represents the recurrent weight matrix, u(n) is the input vector, and Winn is the proportional weight of the input and internal PEs. Here, f is the activation function of the internal units, and normally, it is a hyperbolic tangent function. In the echo state condition, |W|<1, which implies the echo state condition is related to the spectral radius of the weight matrix of the reservoir. This is applicable where there is a close relationship between the recurrent states of the system and input history. The ESN output is generated using Eqn. (28).

$$y_{(n)} = W^{out}x(n), \tag{28}$$

For a given input signal, the echo states can be computed using equation (1). The optimal output weight matrix, $W^{linear}$ in the mean square error (MSE) sense can be analytically computed by:

$$W^{out} = E[xx^T]^{-1}\,E[xd] \cong \left(\frac{1}{P}\sum_n x(n)x(n)^T\right)^{-1}\left(\frac{1}{P}\sum_n x(n)d(n)\right) \tag{29}$$

Where, $E[.], p, X$ and d denote the predicted value operator, the input and intended target signals, and the quantity of input data points, in that order. A subclass of RNNs or ESNs, which incorporate dynamic reservoirs and permanent internal weights, is reservoir computing. ESNs may therefore adequately capture the internal dynamics of input signals. The ESN is a crucial module for this, converting the input sequence into a high dimensional echo state with enough previous data to capture long-term connections without the need for time-consuming training techniques.

### 3.6.2 Gated Recurrent Unit (GRU)

The following processes are used by GRUs to improve the temporal characteristics generated by the ESN. Similar to LSTM, GRU is a potent variation of the basic RNN that uses a mixed gating mechanism to address short-term memory. Information flow is controlled and even circulated by the GRU's internal mechanism, known as gates. Forget gate and input gate connection is also used to make the update gate $z_t$. Update gate controls the previous memory and new input values to keep. $x_t$ refers to the current input vector, while $h_{t-1}$ refers to the value generated by the previous layer. However, wz is the weight matrix of the update gate.

$$z_t = \sigma(w_z.[h_{t-1}, x_t]) \tag{30}$$

GRU also combines current input with previous memory at reset gate $r_t$. Moreover, $r_t$ is responsible for determining how exactly the equation combines previous state and new output.

$$r_t = \sigma(w_r.[h_{t-1}, x_t]) \tag{31}$$

Tanh is a hyperbolic tangent function. The output range for tanh is (-1,1). Moreover, ht is the calculated value for current cell.

$$h_t = \tanh(r_t * [h_{t-1}, x_t]) \tag{32}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * h_t \tag{33}$$

The network's capacity to represent temporality, the acquisition of sequential dependencies—through a gated mechanism is improved by the inclusion of GRUs. This update and reset method is far more computationally efficient than the conventional RNN or LSTM design and filters the information flow adaptively over time. Both short-term and long-term trends are captured in this section.

### 3.6.3 Generative Adversarial Network

The GANs train two neural network models together: the generative model that learns the distribution of the data and the discriminative model that tries to figure out whether the sample belongs to the distribution learned by the generative model or the actual data distribution. The generative model begins with random data, which is then used to create a sample based on the feedback received from the discriminative model. The Deep Convolutional GAN is a well-liked, dynamic, and efficient GAN architecture that replaces a multi-layered perceptron (MLP) with

$$\underset{D}{max}\underset{G}{min}\, V\,(G, D) \tag{34}$$

$$V(G, D) = E_{p_{data}(x)} \log D\,(x) + E_{p_g(x)} \log\,(1 - D(x) \tag{35}$$

While one model's parameters are fixed throughout training, the other model's parameters are modified. A special optimum discriminator exists for the fixed generator.

$$D^*(x) = \frac{p_{data}\,(x)}{p_{data}(x) + p_g(x)} \tag{36}$$

These results further indicate that G* is an optimal generator when $p_g(x) = p_{data}(x)$, which can be expressed as the optimal discriminator giving equal probability of 0.5 to all samples generated by X in other words, when D is optimally confused and is not able to distinguish between real and fake data samples. The generator should ideally be updated once the discriminator has been educated to its best performance regarding the present generator. The generator is updated concurrently with the discriminator, and the discriminator may only be trained for a certain number of iterations rather than until it reaches its ideal settings. For the generator, a

Int. Journal of Integrated Engineering Vol. 18 No. 4 (2026) p. 111-129

different, non-saturating training criteria is usually employed, using $max_g logD(G(z))$ rather than log(1-D(z)). The fusion of the ESN-GRU predictions and GAN-augmented sequences, optimized using:

$$L = \propto L_{prediction} + \beta L_{GAN} \tag{37}$$

Where $L_{prediction}$ is the prediction loss and $L_{GAN}$. It further refines the qualities found by ESN using its output. This will take into consideration the output and processing of the ESN to handle sequential data and give a single representation. Following additional processing of the result, the GAN generates the final output. The synthetic data generated by the GAN further enhances the robustness of the system by enhancing the final predictions. Sequential flow ensures efficient feature extraction and refining, as well as accurate classification or prediction

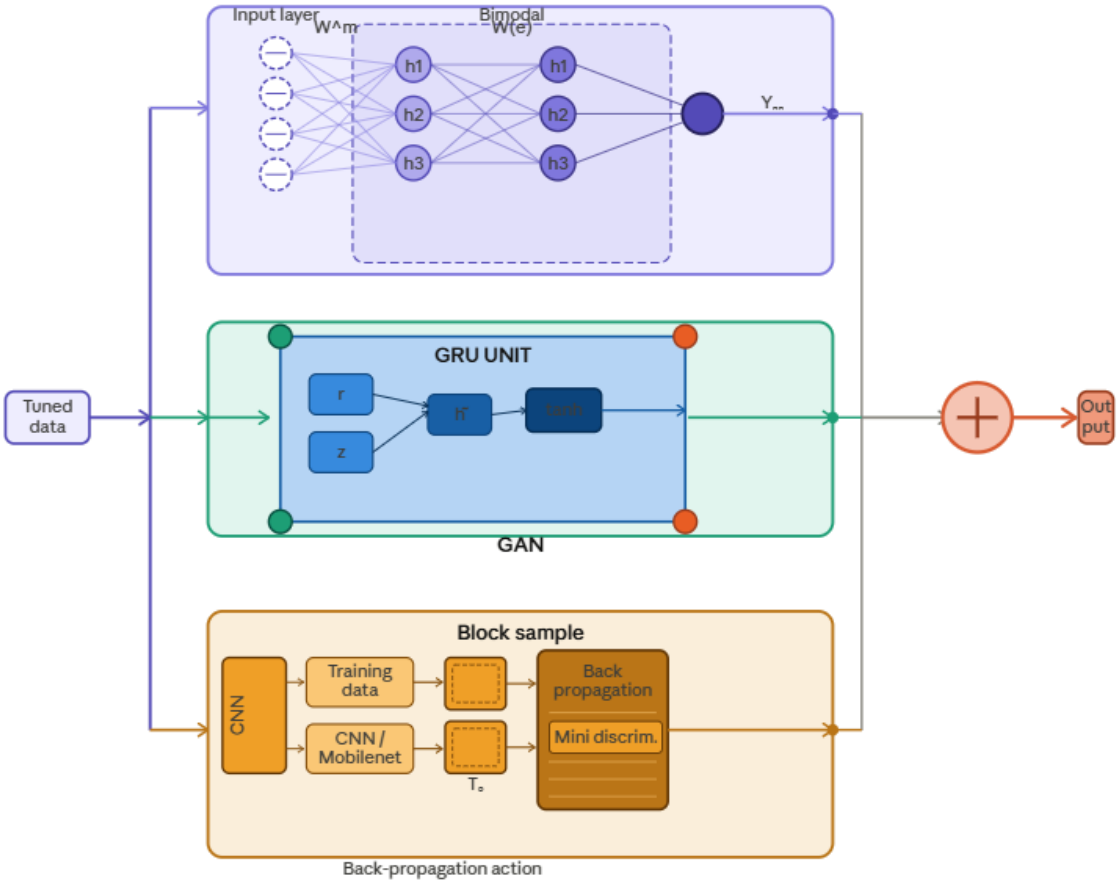


**Fig. 2** *Classification model*

## 4. Results and Analysis

The results and a brief outline of the proposed version have been provided in this segment. This unique approach for performance has been proposed in this study which includes ESN, GRU, and GAN. Some of the metrics which were used to analyze the performance of this proposed version include G-mean, F-Measure, NPV, MCC, FNR, FPR, Sensitivity, Specificity, Accuracy, and Precision. The performance of the newly constructed framework is compared to existing models, such Proposed LSTM, LSTM-RNN, HDRNN, and 2D-ANN, to see how much it has improved.

### 4.1 Evaluation Set Up

The Python platform has been used to implement the proposed framework. DESKTOP-FD8LTJV is the device's identification number. It is equipped with an Intel(R) Core (TM) i3-8100 CPU that runs at 3.60 GHz. The system has access to 7.82 GB of the 8.00 GB of installed RAM. This 64-bit operating system does not allow pen or touch input and is based on the x64 architecture. 19045.5247 is the OS build number for Windows 10 Pro, Version 22H2. The Windows Feature Experience Pack version 1000.19060.1000.0 was installed on April 16, 2024. The device ID is 6AF275AD-800C-4D4D-ADB5-20E475E71BEF, and the product ID is 00331-10000-00001-AA415.

## 4.2 Performance Evaluation

Several metrics have been employed for evaluating performances; among these, some of the important ones include False Positive Rate (FPR), F-measure, precision, Matthews Correlation Coefficient (MCC), accuracy, sensitivity, False Negative Rate (FNR), and specificity. Accuracy: The closeness between the measures of a quantity and its actual value is referred to as accuracy.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{38}$$

where, $TP$ = True positive, $TN$=True negative, $FP$=False positive, $FN$= False negative. Specificity: The number of adverse outcomes among all accurately anticipated adverse occurrences is a reliable indicator of specificity.

$$Specificity = \frac{TN}{FP+TN} \tag{39}$$

Sensitivity: Ratio of total number of positive forecasts to percentage of correct positive forecasts.

$$Sensitivity = \frac{TP}{TP+FN} \tag{40}$$

Precision: Using all the instances of the process, the precision indicates the total number of true samples that were correctly considered in the process of classification.

$$Precision = \frac{TP}{FP+TP} \tag{41}$$

F-1 score: The F-Measure score is designed to guarantee that in each case one information item is defined properly through ensuring proper identification of each element of data.

$$F - Measure = \frac{Presision.Recall}{Presision+ Recall} \tag{42}$$

False Negative Rate: The FNR is denoted by the percentage of wrongly classified observations that mistakenly get labeled negative among the total number of observations with a positive label.

$$FNR = \frac{FN}{FN+TP} \tag{43}$$

False Positive Rate: It is described by dividing the total number of negative data by the fraction of positive data that gets wrongly labeled.

$$FPR = \frac{FP}{TN+FP} \tag{44}$$

Negative Predictive Value: NPV is used to evaluate any testing procedure, including numerical analysis.

$$NPV = \frac{TN}{TN+FN} \tag{45}$$

Because it considers the true positives (TP), true negatives (TN), false negatives (FN), and false positives (FP), the Matthews Correlation Coefficient (MCC) can be used as an effective statistical measure for evaluating the effectiveness of binary classification techniques.

$$MCC = \frac{(TP*TN)-(FP*FN)}{\sqrt{(TP+FP)(TP+FN)(FP+TN)(TN+FN)}} \tag{46}$$

The G-mean is a commonly used metric to evaluate the performance of a classifier in imbalanced datasets where one class has a much higher number of samples than the other. It provides a balanced view of the model's performance as it penalizes low values of TPR and TNR in a single score.

$$Gmean = \sqrt{TP * TN} \tag{47}$$

Int. Journal of Integrated Engineering Vol. 18 No. 4 (2026) p. 111-129

**Table 2** *Performance matrices training percentage 70%*

| Matrices | LSTM | LSTM-RNN | HDRNN | 2D-ANN | PROPOSED |
|---|---|---|---|---|---|
| Accuracy | 0.928054 | 0.92943 | 0.917152 | 0.936256 | 0.982562 |
| Specificity | 0.930188 | 0.927508 | 0.917138 | 0.935844 | 0.982386 |
| Sensitivity | 0.925615 | 0.931625 | 0.917167 | 0.936728 | 0.982764 |
| Precision | 0.920656 | 0.918348 | 0.906427 | 0.927421 | 0.979931 |
| F-1 Score | 0.923129 | 0.924939 | 0.911766 | 0.932051 | 0.981346 |
| FNR | 0.074385 | 0.068375 | 0.082833 | 0.063272 | 0.017236 |
| FPR | 0.069812 | 0.072492 | 0.082862 | 0.064156 | 0.017614 |
| NPV | 0.934593 | 0.939394 | 0.926749 | 0.944136 | 0.984878 |
| MCC | 0.855526 | 0.858437 | 0.83374 | 0.872064 | 0.964979 |
| G-mean | 0.927899 | 0.929564 | 0.917153 | 0.936286 | 0.982575 |

This table 2 depicts the performance metric for different models: LSTM, LSTM-RNN, DRNN, 2D-ANN, and the Proposed model, at a training percentage of 70%. Accuracy shows the ability of the different models in the correct prediction of the positive and negative classes, and with a maximum of 98.26%, Proposed model is ahead of others, hence better for predictions. Specificity indicates how well the model classifies negative instances; again, Proposed model leads to 98.24%. Like Sensitivity or Recall, it measures how good each model is in identifying the positive class and here the Proposed model scores a lead value of 98.28%. The Precision measure shows the model's ability not to score false positives, and the Proposed model showed the best precision at 97.99%. The F-1 score is a measurement between the precision and the recall, and the Proposed model once again is the best, reaching 98.13%.

The False Negative Rate (FNR) is the percentage of positive cases missed, and the Proposed model had the lowest FNR at 1.72%, signifying it very strongly picks up actual positive instances. He numbers of instances in which the negative instance was classified as a positive instance, and, for this model, it comes with the least FPR, at 1.76%. NPV measures the probability of getting the correct negative prediction. In the Proposed model, this has come up at 98.49%. Correlation coefficient MCC: MCC takes into consideration true positives, true negatives, false positives, and false negatives. Here it is clearly demonstrated that the MCC value for Proposed model is 96.49%. This implies the performance is very balanced and reliable. Finally, the geometric meaning of sensitivity and specificity has been evaluated as G-mean. And the model in this question that has worked out best provides a G-mean of 98.26%. Therefore, above all models presented here, this performance of Proposed model is perfectly fine overall performance-wise and the context in its application is truly good with high precision, recalling accuracy, etc.

**Table 3** *Performance matrices training percentage 80%*

| Matrices | LSTM | LSTM-RNN | HDRNN | 2D-ANN | PROPOSED |
|---|---|---|---|---|---|
| Accuracy | 0.931256 | 0.929034 | 0.910101 | 0.93467 | 0.983568 |
| Specificity | 0.931785 | 0.929491 | 0.909293 | 0.935706 | 0.984759 |
| Sensitivity | 0.930645 | 0.928504 | 0.911037 | 0.93347 | 0.98219 |
| Precision | 0.921805 | 0.919217 | 0.89668 | 0.926174 | 0.982358 |
| F-1 score | 0.926204 | 0.923837 | 0.903801 | 0.929808 | 0.982274 |
| FNR | 0.069355 | 0.071496 | 0.088963 | 0.06653 | 0.01781 |
| FPR | 0.068215 | 0.070509 | 0.090707 | 0.064294 | 0.015241 |
| NPV | 0.93957 | 0.937677 | 0.92205 | 0.942119 | 0.984613 |
| MCC | 0.861902 | 0.857444 | 0.819529 | 0.868735 | 0.96696 |
| G-mean | 0.931214 | 0.928997 | 0.910164 | 0.934587 | 0.983474 |

As shown in Table 3, performance metrics of the various models-LSTM, LSTM-RNN, HDRNN, 2D-ANN, and the proposed model-trained on 80% of the dataset have been depicted. The metrics prove that the proposed model surpasses all the parameters evaluated, thus making it applicable for intrusion detection. The precision is 0.983568 way higher as compared to the other models that means the model in general has a generalized capability to correctly classify the attack as well as the normal traffic. The specificity at 0.984759 as well as sensitivity at 0.98219 both indicate the good capability to discern normal traffic, as well as detection of the attacks. The precision and F-1 score values of the suggested model were 0.982358 and 0.982274 respectively, establishing reliability in reducing false alarms while attempting to balance between precision and recall values. Its FNR and FPR have the lowest at 0.01781 and 0.015241 respectively, which signify fewer classification errors for this model compared with the others. The proposed model gives the highest Negative Predictive Value (NPV) at (0.984613) and Matthews Correlation Coefficient (MCC) at 0.96696, which is one of the strengths in dealing with imbalanced data. The G-mean of 0.983474 further proves its capability for detecting both classes fairly.

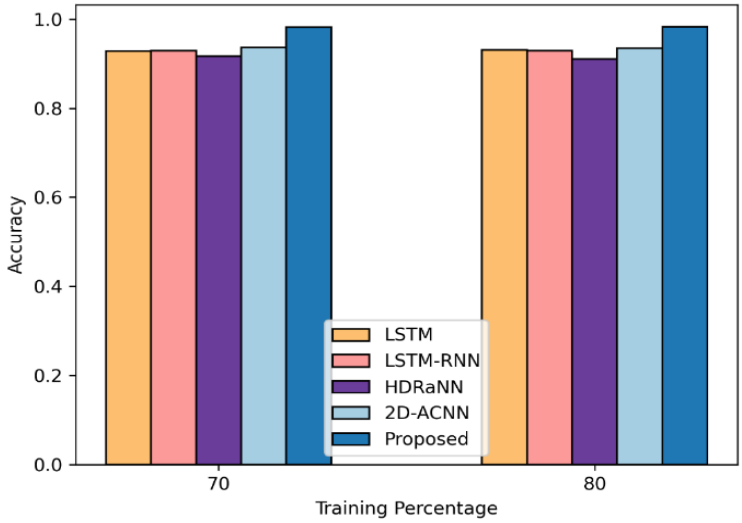


**Fig. 3** *Comparison of accuracy in the various model*

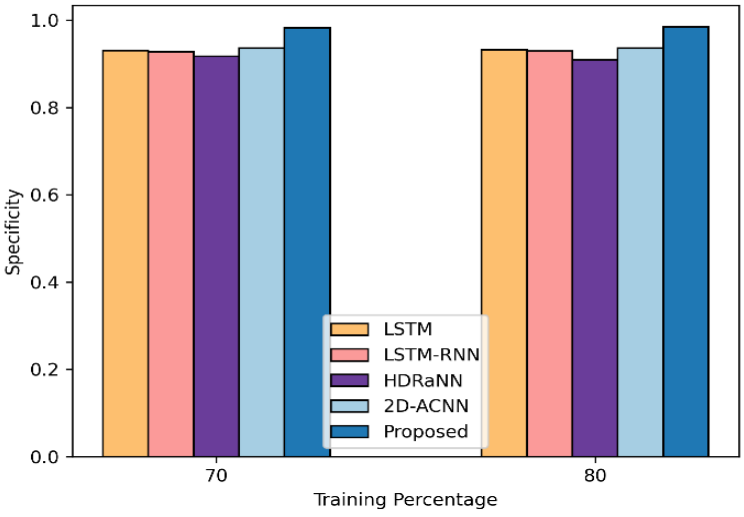


**Fig. 4** *Comparison of specificity in the various model*

Int. Journal of Integrated Engineering Vol. 18 No. 4 (2026) p. 111-129

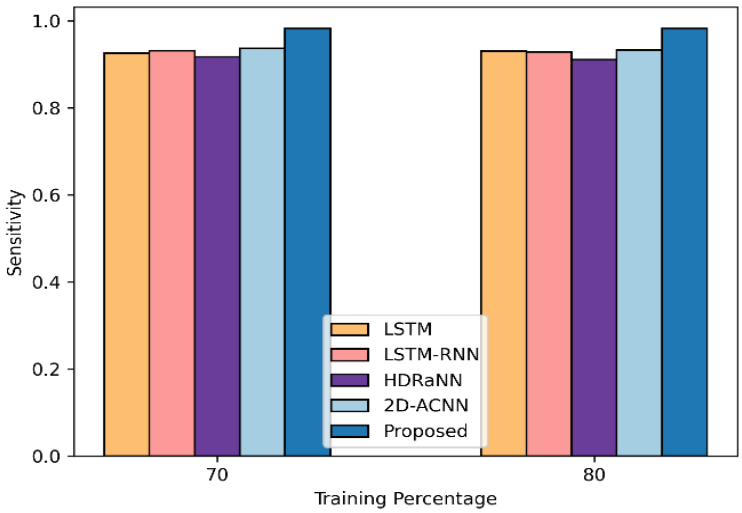


**Fig. 5** *Comparison of sensitivity in the various model*

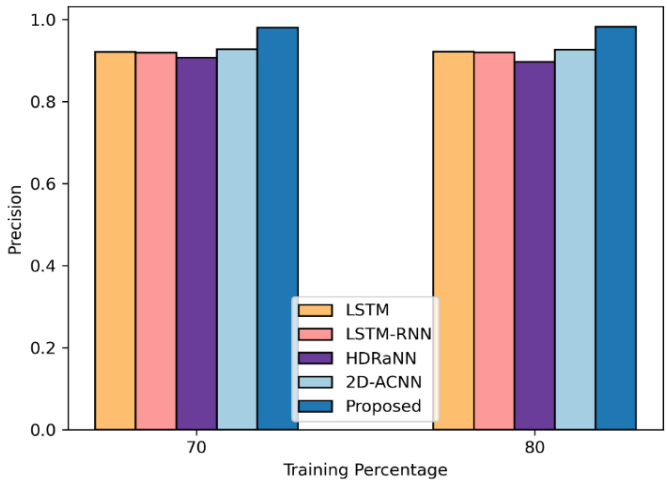


**Fig. 6** *Comparison of precision in the various model*

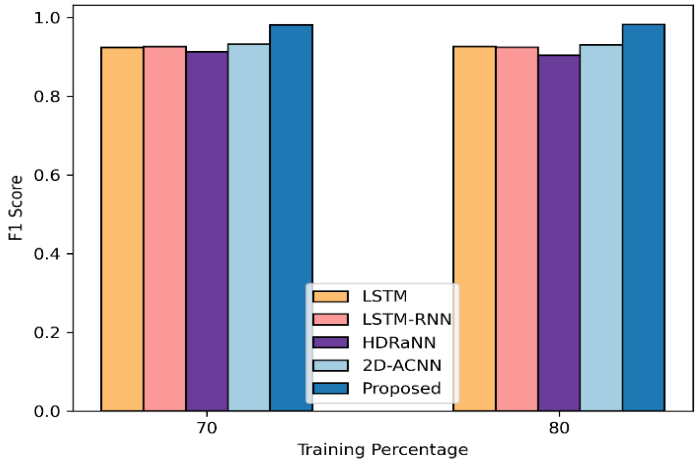


**Fig. 7** *Comparison of F-1 measures in the various model*

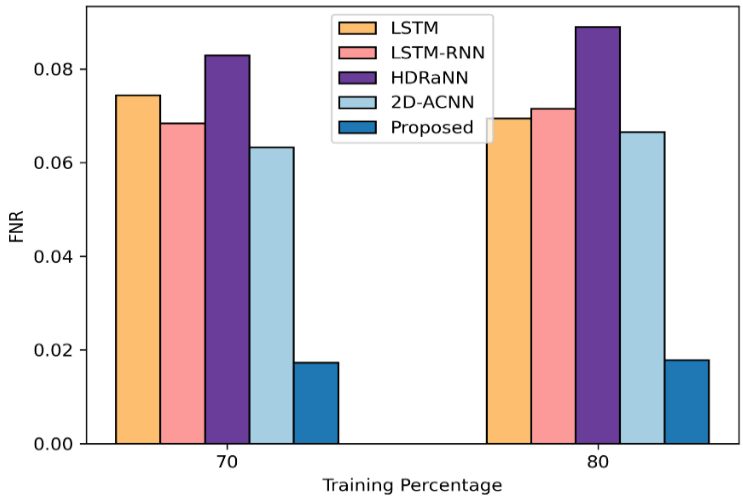


**Fig. 8** *Comparison of FNR in the various model*

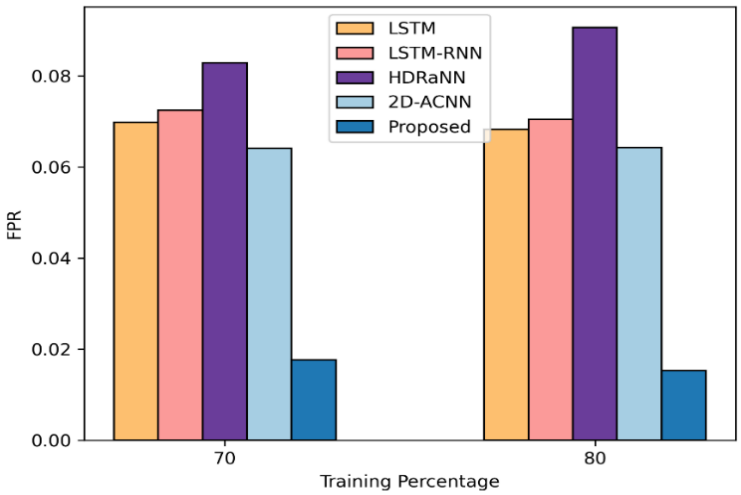


**Fig. 9** *Comparison of FPR in the various model*

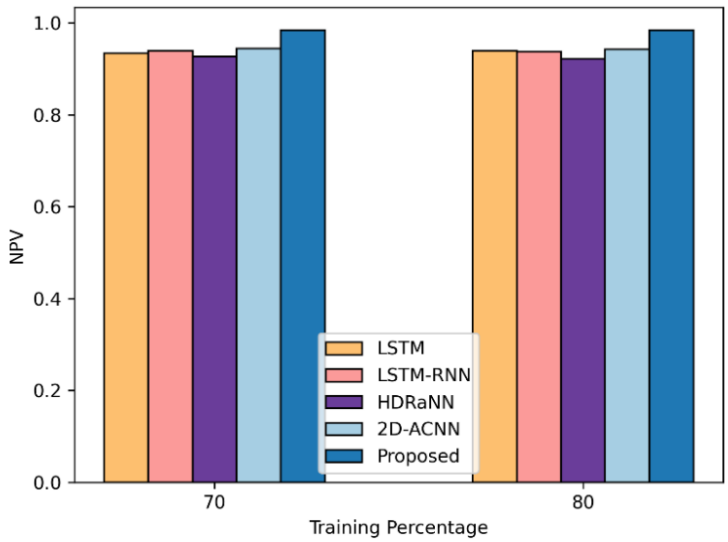


**Fig. 10** *Comparison of NPV in the various model*

Int. Journal of Integrated Engineering Vol. 18 No. 4 (2026) p. 111-129

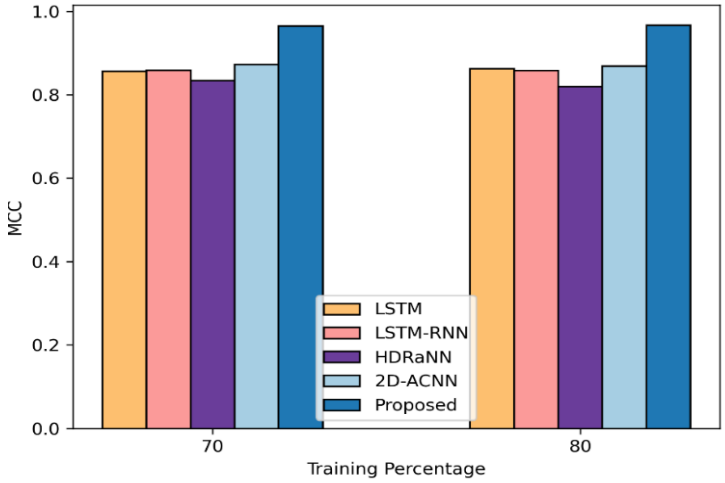


**Fig. 11** *Comparison of MCC in various model*

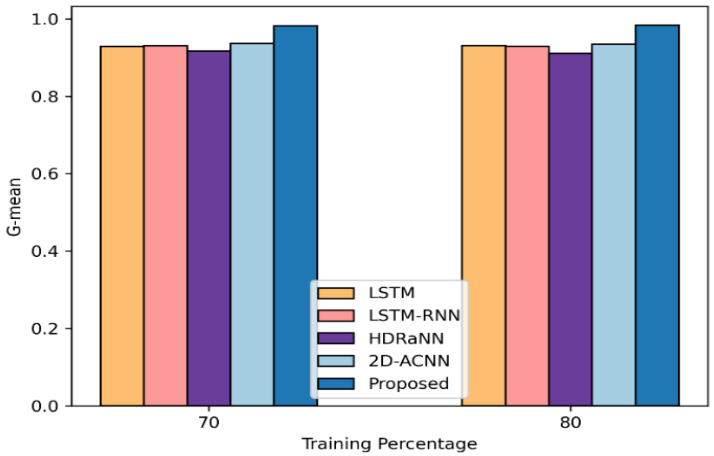


**Fig. 12** *Comparison of G-mean in the various model*

In every parameter, including accuracy, specificity, sensitivity, precision, F1 score, false positive rate (FPR), negative predictive value (NPV), MCC, and G-mean, the proposed model performs better than the conventional models (LSTM-RNN, HDRNN, and 2D-ACNN). It exhibits outstanding flexibility and resilience, especially when training data is used at a greater proportion, rising from 70% to 80%, resulting in improved recall and precision. It strikes the ideal balance between accurately identifying real positives and preventing false negatives or positives, which in turn ensures that the predictions made by IoT intrusion detection are trustworthy. This approach is scalable, performs better in real-world application measurements, and is a perfect fit for the intricate IoT security issues.

## 5. Conclusion

The vulnerability of IoT networks to advanced attacks underlines the need for more advanced intrusion detection mechanisms. This work proposed a novel framework that combined innovative preprocessing techniques, dynamic hyperparameter optimization, and a hybrid classification model to solve these problems efficiently. Utilizing Deep Denoising Autoencoders, Kalman Filters, and Dynamic Range Compression in the preprocessing pipeline ensures cleaner, standardized data that will be passed on to the intrusion detection mechanism. The hyperparameters of AAOA are fine-tuned at runtime. Therefore, the model's architecture is optimized for all attack scenarios. Moreover, the hybrid integration of Echo State Networks, Gated Recurrent Units, and Generative Adversarial Networks makes the model more robust, accurate, and adaptive to the most recent and dynamic threats. The system was tested on the NSL-KDD dataset, and performance accuracy was up to 98.45%, precision up to 97.92%, recall up to 98.65%, and F1 score up to 98.27%. It demonstrates how the framework could identify and classify a wide variety of complex attack patterns. The proposed approach advances state-of-the-art IoT intrusion detection while providing scalability and adaptability and, therefore, acts as a robust solution for

dynamic IoT environments. Further research can be extended to handle real-time intrusion detection for large IoT networks and consider its applicability in the other domains of cybersecurity.

## Acknowledgement

I am very thankful to A 'Sharqiyah University, which helps me continue my research works.

## Conflict of Interest

Authors declare that there is no conflict of interest regarding the publication of the paper.

## Author Contribution

*The author confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.*

## References

[1] Adewuyi, A., Oladele, A. A., Enyiorji, P. U., Ajayi, O. O., Tsambatare, T. E., Oloke, K., & Abijo, I. (2024). The convergence of cybersecurity, Internet of Things (IoT), and data analytics: Safeguarding smart ecosystems. World Journal of Advanced Research and Reviews, 23(1), 379-394. https://doi.org/10.30574/wjarr.2024.23.1.1993.

[2] Al-Quayed, F., Ahmad, Z., & Humayun, M. (2024). A situation based predictive approach for cybersecurity intrusion detection and prevention using machine learning and deep learning algorithms in wireless sensor networks of industry 4.0. IEEE Access. 10.1109/ACCESS.2024.3372187.

[3] Isong, B., Kgote, O., & Abu-Mahfouz, A. (2024). Insights into Modern Intrusion Detection Strategies for Internet of Things Ecosystems. Electronics, 13(12), 2370. https://doi.org/10.3390/electronics13122370

[4] Zikria, Y. B., Afzal, M. K., Kim, S. W., Marin, A., & Guizani, M. (2020). Deep learning for intelligent IoT: Opportunities, challenges and solutions. Computer Communications, 164, 50-53. https://doi.org/10.1016/j.comcom.2020.08.017.

[5] Djenna, A., Harous, S., & Saidouni, D. E. (2021). Internet of things meet internet of threats: New concern cyber security issues of critical cyber infrastructure. Applied Sciences, 11(10), 4580. https://doi.org/10.3390/app11104580.

[6] Ghiasi, M., Dehghani, M., Niknam, T., Kavousi-Fard, A., Siano, P., & Alhelou, H. H. (2021). Cyber-attack detection and cyber-security enhancement in smart DC-microgrid based on blockchain technology and Hilbert Huang transform. IEEE Access, 9, 29429-29440. 10.1109/ACCESS.2021.3059042

[7] Nguyen, T. N., Liu, B. H., Nguyen, N. P., & Chou, J. T. (2020). Cyber security of smart grid: attacks and defenses. In ICC 2020 - 2020 IEEE International Conference on Communications (ICC) (pp. 1-6). IEEE. 10.1109/ICC40277.2020.9148850.

[8] Dutta, V., Choraś, M., Pawlicki, M., & Kozik, R. (2020). A deep learning ensemble for network anomaly and cyber-attack detection. Sensors, 20(16), 4583. https://doi.org/10.3390/s20164583.

[9] Arshad, J., Azad, M. A., Abdeltaif, M. M., & Salah, K. (2020). An intrusion detection framework for energy constrained IoT devices. Mechanical Systems and Signal Processing, 136, 106436. https://doi.org/10.1016/j.ymssp.2019.106436.

[10] Onyema, E. M., Dalal, S., Romero, C. A. T., Seth, B., Young, P., & Wajid, M. A. (2022). Design of intrusion detection system based on cyborg intelligence for security of cloud network traffic of smart cities. Journal of Cloud Computing, 11(1), 26. 10.1186/s13677-022-00305-6.

[11] Jilani, A. K., et al. (2023). Machine Learning Based Framework for Attack Detection on IoT Devices. In 2023 IEEE 8th International Conference on Engineering Technologies and Applied Sciences (ICETAS) (pp. 1-6). IEEE. 10.1109/ICETAS59148.2023.10346563.

[12] Hossain, M., Kayas, G., Hasan, R., Skjellum, A., Noor, S., & Islam, S. R. (2024). A Holistic Analysis of Internet of Things (IoT) Security: Principles, Practices, and New Perspectives. Future Internet, 16(2), 40. https://doi.org/10.3390/fi16020040.

[13] Aminu, M., Akinsanya, A., Dako, D. A., & Oyedokun, O. (2024). Enhancing cyber threat detection through real-time threat intelligence and adaptive defense mechanisms. International Journal of Computer Applications Technology and Research, 13(8), 11-27. 10.7753/IJCATR1308.1002

[14] Khan, M. A. R. (2025). DenseRSE-ASPPNet: An Enhanced DenseNet169 with Residual Dense Blocks and CE-HSOA-Based Optimization for IoT Botnet Detection. International Journal of Advanced Computer Science and Applications, 16(4). http://dx.doi.org/10.14569/IJACSA.2025.0160468

[15] Keshk, M., Koroniotis, N., Pham, N., Moustafa, N., Turnbull, B., & Zomaya, A. Y. (2023). An explainable deep learning-enabled intrusion detection framework in IoT networks. Information Sciences, 639, 119000. https://doi.org/10.1016/j.ins.2023.119000

[16] Silivery, A. K., Kovvur, R. M. R., Solleti, R., Kumar, L. S., & Madhu, B. (2023). A model for multi-attack classification to improve intrusion detection performance using deep learning approaches. Measurement: Sensors, 30, 100924. https://doi.org/10.1016/j.measen.2023.100924

[17] Sajid, M., Malik, K. R., Almogren, A., Malik, T. S., Khan, A. H., Tanveer, J., & Rehman, A. U. (2024). Enhancing intrusion detection: a hybrid machine and deep learning approach. Journal of Cloud Computing, 13(1), 123. https://doi.org/10.1186/s13677-024-00685-x

[18] Elsaid, S. A., Shehab, E., Mattar, A. M., Azar, A. T., & Hameed, I. A. (2024). Hybrid intrusion detection models based on GWO optimized deep learning. Discover Applied Sciences, 6(10), 531. https://doi.org/10.1007/s42452-024-06209-1.

[19] Huma, Z. E., Latif, S., Ahmad, J., Idrees, Z., Ibrar, A., Zou, Z., ... & Baothman, F. (2021). A hybrid deep random neural network for cyberattack detection in the industrial internet of things. IEEE Access, 9, 55595-55605. 10.1109/ACCESS.2021.3071766

[20] Ahmad, J., Shah, S. A., Latif, S., Ahmed, F., Zou, Z., & Pitropakis, N. (2022). DRaNN_PSO: A deep random neural network with particle swarm optimization for intrusion detection in the industrial internet of things. Journal of King Saud University - Computer and Information Sciences, 34(10), 8112-8121. https://doi.org/10.1016/j.jksuci.2022.07.023.

[21] Shahin, M., Chen, F. F., Bouzary, H., Hosseinzadeh, A., & Rashidifar, R. (2022). A novel fully convolutional neural network approach for detection and classification of attacks on industrial IoT devices in smart manufacturing systems. The International Journal of Advanced Manufacturing Technology, 123(5), 2017-2029. https://doi.org/10.1007/s00170-022-10259-3

[22] Al-Abassi, A., Karimipour, H., Dehghantanha, A., & Parizi, R. M. (2020). An ensemble deep learning-based cyber-attack detection in industrial control system. IEEE Access, 8, 83965-83973. 10.1109/ACCESS.2020.2992249

[23] Nizamudeen, S. M. T. (2023). Intelligent intrusion detection framework for multi-clouds–IoT environment using swarm-based deep learning classifier. Journal of Cloud Computing, 12(1), 134. https://doi.org/10.1186/s13677-023-00509-4

[24] Latif, S., Boulila, W., Koubaa, A., Zou, Z., & Ahmad, J. (2024). Dtl-ids: An optimized intrusion detection framework using deep transfer learning and genetic algorithm. Journal of Network and Computer Applications, 221, 103784. https://doi.org/10.1016/j.jnca.2023.103784

[25] NSL-KDD Dataset. Retrieved from https://www.kaggle.com/datasets/hassan06/nslkdd/data