
EMPOWERING SECURITY IN IOT: SPRING SEARCH ALGORITHM WITH OPTIMAL MACHINE LEARNING ENABLED INTRUSION DETECTION

Dr. Manisha Kumari

Department of Science,
Technology and Technical Education,
Patna, Bihar.

Crossref DOI - <https://doi.org/10.63665/rh.v7i2.73>

Abstract :

Devices, sensors, and physical items are all part of what is known as the Internet of Things (IoT) embedded with technology that allows them to gather, exchange, and act upon information without human intervention. Security in IoT is essential to secure sensitive information, safeguard interconnected networks and devices from cyberattacks, and ensure the trustworthiness and reliability of IoT environment, thereby alleviating risks of privacy breaches, potential physical harm, and financial losses. IoT security includes authentication, encryption, access control, and intrusion detection measures to alleviate security attacks and protect against potential vulnerabilities. One efficient method of addressing security concerns in an Internet of Things (IoT) setting is to implement a system for intrusion detection (IDS), which employs ML and DL approaches to fortify protections. Therefore, this study introduces a new Spring Search Algorithm based on Feature Selection with Optimal Machine Learning Empowered Intrusion Detection (SSAFS-OMLID) in IoT framework. The projected SSAFS-OMLID model intends to detect the occurrence and recognition of intrusions in the IoT framework. Data pre-processing is the first step in preparing the data for analysis. After the data has been pre-processed, the SSAFS model is employed to pick features. Then, hop field neural network (HNN) classifier is exploited for the intrusion detection in the network. Finally, Sooty Tern Optimization Algorithm (STOA) is utilized to effectually adjust the parameters included in the HNN performance. In order to prove that the SSAFS-OMLID method is the best, researchers conducted extensive experiments, and the results showed that the method is very efficient according to several criteria.

Keywords : Intrusion detection system, Machine learning, Optimization algorithms, Security, Feature Selection, Data classification

Introduction :

Connected devices that enhance people's lives, professions, and cultures are known as the Internet of Things (IoT) [1]. IoT structures are open around the globe, significantly consisting of compelled resources and enhanced through lossy connection [2]. Correspondingly, critical modifications of available security ideas for remote systems and data might be realized to offer compelling IoT security methods. While using recent security



devices, for example, network protection, authentication, application control, encryption, and access control consumes more time and considers inadequate for a greater system amongst most of the linked appliances, with each portion of the system having its own susceptibility [3]. IoT devices, being resource-constrained and reliant on lossy communication links, are inherently vulnerable to cyber threats. These limitations make it easier for attackers to exploit weaknesses, leading to privacy breaches, operational disruptions, and financial losses. Among the different cyber threats, Distributed Denial of Service (DDoS) attacks and data breaches are particularly alarming due to their widespread impact and ability to compromise sensitive information. This heightened vulnerability necessitates focused research efforts to develop robust security measures for IoT ecosystems [4].

In IoT networks, intrusion detection systems are essential for spotting and stopping hostile activity [5]. They detect malicious activity by analysing network traffic for unusual patterns. IDS are classified into three types: anomaly-based, which detects deviations from typical network behaviour; signature-based, which detects known attack patterns; and specification-based, which rely on established system behaviour. Among these, anomaly-based IDS are favored for their capacity to detect previously unknown threats, a critical requirement in dynamic IoT environments. Despite their advantages, anomaly-based IDS face significant challenges, including high rates of false alarms and computational complexity. The reliance on machine learning models to analyze network behavior adds to the complexity, as these models must balance detection accuracy with efficient resource utilization. Addressing these challenges requires innovative approaches that optimize IDS performance while ensuring scalability and reliability for resource-constrained IoT systems [6].

Privacy and Security factors of the IoT plays a primary role that makes it effective in becoming one of the universally accepted technology in the future. But, open character and self-configuring of IoT cause prone to several outsider and insider attackers. Attackers can endanger the users' privacy and security with a view to obtain access to users' personal data, make financial losses, and eavesdrop. IoT gadgets become an easy task for attackers that abuses their vulnerabilities for the purpose of executing distributed of services (DoS) attacks [7]. Therefore, safe guarding of such gadgets is considered a significant factor for researchers in recent times. For solving this problem, a greater number of researchers are undergoing intrusion detection (ID) is globally [8]. IDS are classified into 3 classes depending on the identification technique that is specification, signature, and anomaly, Anomaly-related IDS consistently verifies network traffic for any diversion over normal network profile [9]. Fig. 1 shows the IDS architecture.



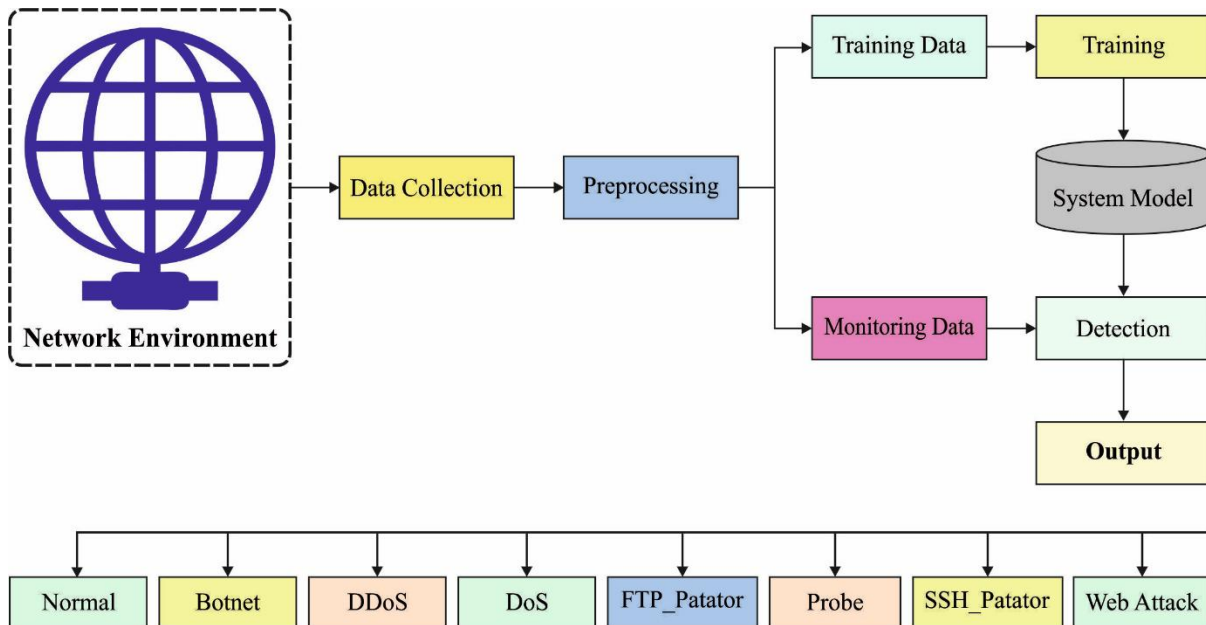


Fig 1: Architecture of IDS

Suppose a deviation surpasses the threshold, an alarm can be raised so that we can able to recognize the attack. The normal network profile is studied by (ML) algorithms [10]. An anomaly-related IDS is recommended over signature and specification related IDS due to its capability for identifying new attacks, only disadvantage is the cost of high false alarm rate. The proficiency of anomaly related IDS is based on the goodness of detection engine (classifier or model), and this goodness has a quality of network traffic pattern (data set samples) is utilized for training of engine [11]. ID in IoT network was featured as a binary classifier issue whereas a trained classifier targets in classifying network traffic to attack or normal class with minimal false alarms (FAR) and maximal accuracy [12]. The great efficiency of the classification in relation to accuracy and FAR is mainly based on the choice of training data and classifier algorithm. Security specialist always chooses best performing classifier for detecting the intrusion [13].

This study develops a new SSAFS-OMLID in IoT framework. Data preparation is the first step in the suggested SSAFS-OMLID technique, which normalises the input into an appropriate format. Then, an optimum HNN model is used as the classifier, and the SSAFS method is used for feature selection. At last, Sooty Tern Optimization Algorithm (STOA) is applied to effectually adjust the parameters included in the HNN algorithm. The superiority of the SSAFS-OMLID technique was confirmed through a thorough experimental examination on benchmark datasets.

Literature Review :

In [14], a hybrid weighted deep belief network (HW-DBN) methodology was developed to offer a trustworthy and effective intrusion detection system (IDS) (DeepIoT.IDS) method for detecting both existing and future threats. The proposed approach uses a weighted deep neural network (WDNN) classification methodology coupled with an upgraded Gaussian-Bernoulli restricted Boltzmann machine (Deep GB-RBM) for feature



learning. The authors presented a new intrusion detection system (IDS) in [15] that makes use of a cutting-edge DL technique. A smart intrusion detection system (IDS) capable of detecting intrusions on an Internet of Things (IoT) network was introduced, inspired by the benefits of Long Short Term Memory (LSTM), by the whale integrated Long Short Term Memory (WILS) network.

A new FS and extraction method for the IDS scheme was developed by Fatani et al. [16] by utilising the advantages associated with swarm intelligence (SI) algorithm. A conventional neural network (CNN) based on a feature extraction model is then created. Next, we presented an alternate FS model using the newly proposed SI method, Aquila optimizer (AQU). Gad et al. [17] tested different ML methodologies in the binary and multiple class classification issues. The study then uses the Chi-square (χ^2) approach to feature selection and SMOTE for class imbalance correction. The results show that compared to other ML approaches, the XGBoost method performs better. In [18], the sequential approach is the main focus, and the model's characteristic presents a fresh approach. Using system procedures at the application layer and tcpdump packets at the network layer, the technique may collect features. Since GRU and Text-CNN models can interpret the sequential dataset as a language model, they are recommended.

To find security risks in IoT frameworks, Otoum et al. [19] suggest unique DL-based intrusion detection system (DL-IDS). Despite the fact that many intrusion detection systems (IDSs) are investigated in their research, many of them have poor feature learning and dataset management, which severely impairs their ability to detect attacks. Overcoming these limitations, the suggested method improves detection efficacy by combining a stacked deep polynomial network (SDPN) with the spider monkey optimisation (SMO) algorithm. Raghuvanshi et al. [20] presented architecture for classifying and detecting intrusion into IoT networks utilized in agriculture. Privacy and Security are the primary consideration in agriculture based IoT networks as well as in each application of the IoT. Feature extraction can be done by principal component analysis. Next, ml approaches like RF, LR, and SVM, are utilized for classifying pre-processed datasets.

For the purpose of detecting network intrusions, Moizuddin and Victor [21] suggested a bioinspired hybrid deep learning (HDL) model. The two stages of this IDS model's construction comprise feature selection using a bio-inspired algorithm and attack categorisation based on deep learning. The ElasticNet Contractive Autoencoder (CAE) was used to classify attacks, and the Generalised Mean Grey Wolf (MGW) technique remained used to pick features. The NSL-KDD and BoT-IoT datasets were used in this work, which focused on binary and multi-class classification models. An increase to 0.999 in accuracy was discernible for both classification groups. But computing complexity continued to be a problem.

To solve the issue of class imbalance, Rani and Manisha [22] created an effective network intrusion detection system idea utilising DNN multimedia tools and applications. Challenges for current machine learning (ML)-based classifiers in Network Intrusion Detection Systems (NIDS) include the dynamic nature of the environment, the diversity of



network data, and the difficult task of detecting assaults that haven't been observed before. As a data pretreatment method, min-max normalisation was used to overcome these difficulties. The Deep Neural Network (DNN), which used a modified cross-entropy function to address the problem of class imbalance, was then fed the normalised data. The UNSW-NB15 and NSL-KDD datasets were used to assess the suggested model and show how effective it is. Longer training periods and possible overfitting were among the model's shortcomings, nevertheless.

In order to identify and categorise different kinds of cyberattacks, Ravi et al. [23] suggested a network intrusion detection system (IDS) that makes use of deep feature fusion and an ensemble meta-classifier. The system was created for an environment that combines software-defined networks and cyber-physical systems (CPS+SDN). For feature extraction, recurrent deep learning models like LSTM, RNN, and GRU were used. Using the Kernel Principal Component Analysis (KPCA) approach, the best features were chosen. Concatenation was then employed to merge these traits. A two-stage meta-classifier was used to detect and classify network attacks. SVM and Random Forest classifiers were utilised for prediction in the first stage, while Logistic Regression (LR) was utilised for attack classification in the second stage. Training and evaluation were conducted using the SDN-IoT dataset. The deep learning model's sensitivity to unbalanced data in the network dataset was one of its drawbacks.

Andresini et al. [24] employed deep learning with an auto-encoder to identify network intrusions by extracting flow-based attributes from incoming traffic. To identify network breaches, we used a triplet network and auto-encoders. We employed embedding-based learning criteria to anticipate network attacks. Adding auto-encoders to the detection model resolved the issue of convergence during triplet learning.

In order to classify incoming traffic, Balasundaram [25] presented an optimised extreme learning-based intrusion detection technique that first extracts both time-independent and time-dependent information. By fixing inefficiencies in managing predispositions, the optimisation method improves accuracy. Optimising the bias and weights of the ELM improved its global minima, resulting in reliable detection. Using a basic design and minimum computing effort resulted in faster attack detection.

Folino et al. [26] presented the elastic stack (ELK) architecture for real-time processing and storage of log data from many users and applications. An ensemble of models may classify user behaviour and identify issues in real-time by utilising system benefits. In order to categorise people, a distributed evolutionary algorithm is employed to examine digital footprints gathered from diverse data sources. Using two real-world datasets, the efficacy of this method in detecting unusual user behaviour, handling missing data, and lowering false alarms has been evaluated.

Alrayes et al. [27] employ deep learning to develop an Enhanced Artificial Gorilla Troops Optimiser (EAGTO) for detecting cybersecurity vulnerabilities popular IoT cloud networks. The EAGTODL-CTD technique includes threat detection for IoT cloud environments. The EAGTODL-CTD model converts binary data into colour images in order



to classify photos and detect malware. The model appropriately preprocesses the incoming data to guarantee compatibility. In order to find the labels for the classes required for vulnerability detection and classification, a CGRU model is employed. Our study stands out because we used the EAGTO technique to improve the settings of the CGRU. A dataset with two classes malignant and benign is used to evaluate the efficacy of the EAGTODL-CTD model. The EAGTODL-CTD model obtains the best accuracy of 99.47%, according to experimental data.

The Proposed system :

To identify and prevent intrusions into the IoT infrastructure, this study presents a novel SSAFS-OMLID method. The first step in preparing data is to normalise it into a suitable format. The best features from the pre-processed data are then found and chosen using the SSAFS model. For data classification, the HNN technique was executed for determining appropriate class labels to it. Finally, the STOA is exploited for effective parameter adjustment of the HNN algorithm. The overall process of SSAFS-OMLID approach is demonstrated in Fig. 2.

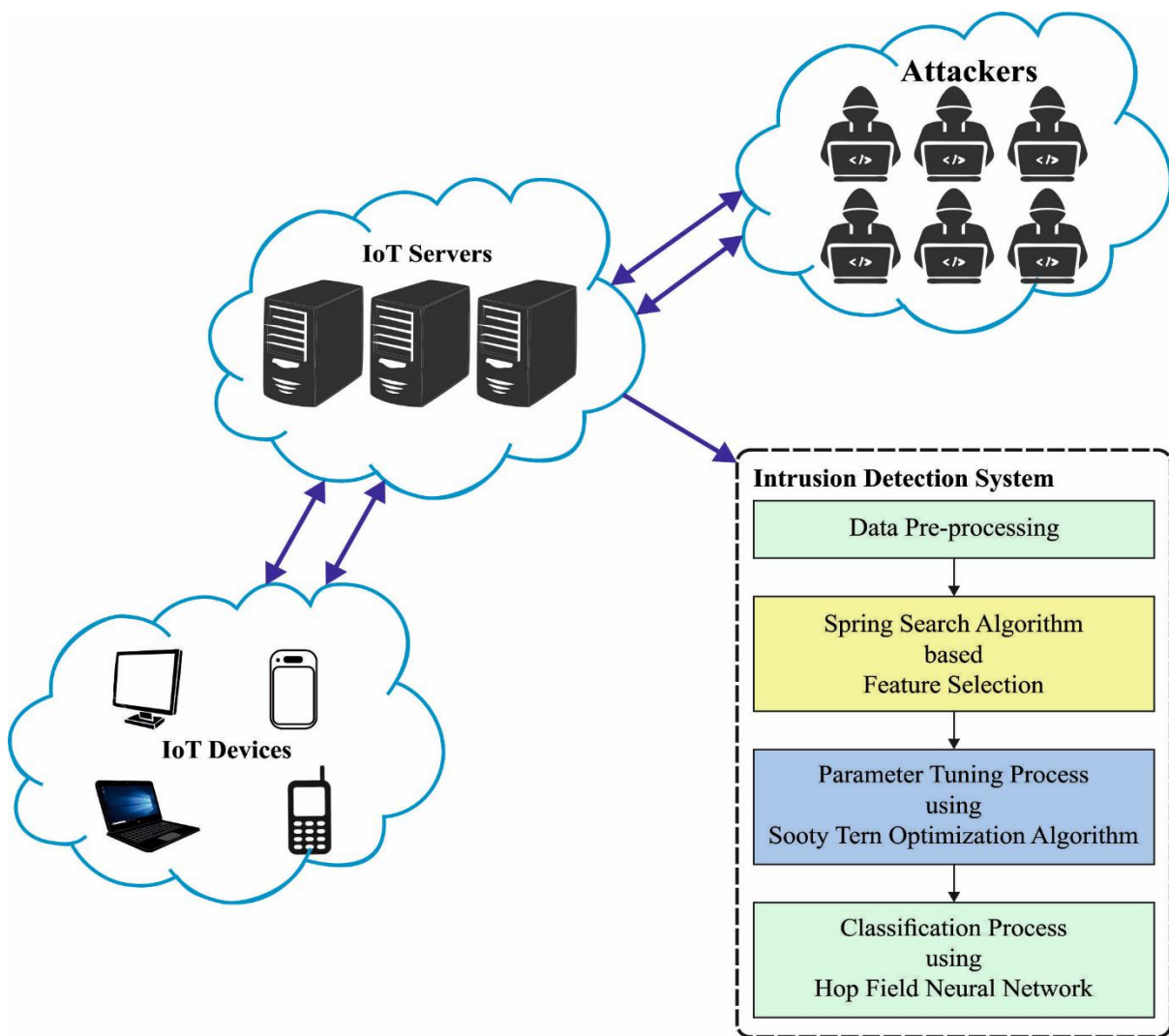


Fig 2: Block diagram of SSAFS-OMLID method

1. Process involved in SSAFS Method :

Initially, the SSAFS model chooses an optimal feature subset. SSA approach can be described as follows: (1) passage of time up to stop time. Initially, system space is determined. The spring stiffness coefficient can be defined based on the fitness of objects connected with one another, and (2) form an artificial system with the initial positioning of object, discrete time problem environments, adjusting the parameters, and determining laws [28]. The location of all the objects is a point in space that resolve the issue. In the next formula, the location of d dimension of object i can be given as x_i^d

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n) \quad (1)$$

Initially, the first location of object can be randomly defined in problem space. This object proceeds to system equilibrium (solution) based on the force applied by the spring among themselves. To evaluate the spring stiffness coefficient, the below equation is applied.

$$K_{i,j} = K_{max} |F_n^i - F_n^j| \max(F_n^i, F_n^j) \quad (2)$$

Where, $K_{i,j}$ denotes spring stiffness coefficient amongst objects i and j , F_n represents the objective function, K_{max} indicates the maximal spring stiffness coefficient defined according to the kind of problematic, and F_n^i and F_n^j are the normalized objective function of i^{th} and j^{th} objects. For normalizing the objective function, the below equations are applied:

$$F_n^i = \frac{f_{obi}^i}{\min(f_{obj})} \quad (3)$$

$$F_n^i = \min(F_n^i) \times \frac{1}{F_n^i} \quad (4)$$

Here, f_{obi} and f_{obj}^i are objective functions for i^{th} object. The strong point of all the objects is has more enhanced location. Thus, on every axis, two resulting forces are used for the object: the resulting force of right and left sides utilized for the calculation:

$$F_{total_R}^{j,d} = \sum_{i=1}^{n_R^d} K_{i,j} x_{i,j}^d \quad (5)$$

$$F_{total_L}^{j,d} = \sum_{l=1}^{n_L^d} K_{l,j} x_{l,j}^d \quad (6)$$

Where, $F_{total_R}^{j,d}$ indicates the resulting force used for j^{th} objects from the right side and $F_{total_L}^{j,d}$ denotes resulting force used for the object from left side at d dimension. $K_{i,j}$ and $K_{l,j}$ indicates spring stiffness connected coefficient among j object and strong point. n_R^d and n_L^d denotes strong points on right and left sides of d^{th} dimensional.



$$dX_R^{j,d} = \frac{F_{totalR}^{j,d}}{K_{equalR}^i} \quad (7)$$

$$dX_L^{j,d} = \frac{F_{totalL}^{j,d}}{K_{equalL}^i} \quad (8)$$

Now the replacement value of j^{th} objects to right and left side at d dimension is represented as $dX_R^{j,d}$ and $dX_L^{j,d}$ correspondingly.

$$dX^{j,d} = dX_R^{j,d} + dX_L^{j,d} \quad (9)$$

In the equation, $dX^{j,d}$ denotes the last displacement value of j^{th} objects at d dimension that value is positive and negative based on the below equation:

$$X^{j,d} = X_0^{j,d} + r_1 \times dX^{j,d} \quad (10)$$

Where, $X^{j,d}$ indicates equilibrium point and position of d dimension and j^{th} objects. Furthermore, $X_0^{j,d}$ indicates the first position of object i equilibrium at d dimension. The ending criteria are defined afterward some time. The steps of spring force model are given below:

1. Initializing and determining the system environment.
2. The first placement of object
3. Normalization and Evaluation of object fitness
4. Update the variable K
5. Form spring force laws for every object
6. Calculate displacement of object
7. Update displacement of object
8. If the end criteria are satisfied, repeat steps 3 to 7
9. Stop

The SSAFS algorithm's aim is to define the optimal feature subset for the given dataset that has high classifier accuracy and less number of features. Both indicators have different effects on classifier performance. Here, we integrated them with a single weighted indicator and employed a similar FF in the following:

$$fitness = \omega_1 \times acc(classifier) + \omega_2 \times \left(1 - \frac{s}{p}\right), \quad (11)$$

Now p represents the overall quantity of features, and s indicates the amount of features selected. The values of ω_1 and ω_2 correspondingly indicates 1 and 0.001. The $acc(classifier)$ characterizes classifier accuracy obtained commencing the VAE classification that can be shown in the following.

$$acc(classifier) = \frac{n_c}{n_c + n_i} \times 100\%. \quad (12)$$



Whereas, n_i and n_c correspondingly characterizes the amount of inaccurately and correctly classified samples.

2. Classification using HNN Model :

Next to optimal choice of features, the data classification process is implemented by using the HNN algorithm. The HNN has powerful feature of content addressable memory and inter-connected neurons which is of great significance to resolve combinatorial and optimized tasks. The HNN method structure contains organization neurons called as Ising variable [29].

$$S_i = \{1, \text{if } \sum_j W_{ij} S_j > \psi, -1, \text{Otherwise} \} \quad (13)$$

Whereas W_{ij} signifies the synaptic weighted vector from j^{th} to i^{th} neurons. S_i is determined as the state of i^{th} neuron in HNN, and ψ signifies the present value. The HNN method is same intricate details as the Ising method of magnetism.

$$S_i \rightarrow sgn [h_i(t)] \quad (14)$$

Where h_i is the local field vector.

$$h_i = \sum_k^N \sum_j^N W_{ijk} S_j S_k + \sum_j^N W_{ij} S_j + W_i. \quad (15)$$

The local field evaluates the last neuron state and create 3-SAT-induced logic. The generalization fitness function, $E_{FRANKSAT}$, control the groups of neurons from HNN. $E_{FRANKSAT}$ was projected in the following:

$$E_{FRANKAT} = \sum_{i=1}^{NN} \prod_{j=1}^V T_{ijk} \quad (16)$$

Whereas V and NN are the amount of variables and amount of neurons created in $F_{RANKSAT}$. The inconsistency of $F_{RANKSAT}$ is determined as:

$$T_{ij} = \left\{ \begin{array}{l} \frac{1}{2}(1 - S_\rho), \text{if } \neg \rho \\ \frac{1}{2}(1 + S_\rho), \text{otherwise} \end{array} \right. \quad (17)$$

The value $F_{RANKSAT}$ has proportionate to inconsistencies assessment of the logical clauses. The recommendations for updating the neuronal state are:

$$\begin{aligned} S_i(t+1) &= \{1, h_i = \sum_k^N \sum_j^N W_{ijk} S_j S_k + \sum_j^N W_{ij} S_j + W_i \geq 0 - 1, h_i \\ &= \sum_k^N \sum_j^N W_{ijk} S_j S_k + \sum_j^N W_{ij} S_j + W_i < 0 \} \quad (18) \end{aligned}$$



The Hopfield Neural Network's (HNN) Lyapunov energy function is represented by equation (19).

$$H_{FRINNAT} = -\frac{1}{3} \sum_{i=1, i \neq j, j \neq k, k=1}^N \sum_{i \neq j, j \neq i, k=1}^N \sum_{i \neq j, k \neq i}^N W_{ijk} S_i S_j S_k - \frac{1}{2} \sum_{i=1, i \neq j}^N \sum_{j=1, i \neq j}^N W_{ij} S_i S_j - \sum_{i=1}^N W_i S_i \quad (19)$$

For classification, the solution attains local minimal energy. The HNN create an optimal assignment if the induced neuron state attains global minima.

3. STO based Parameter Optimization :

The STO is exploited for the effective parameter adjustment of the HNN model during the parameter optimization process. STO simulates the attacking and migration behaviors of sooty terns. The arithmetical model of attacking and migration behaviors is described as follows [30]. In the migration process, a sooty tern must fulfil the three criteria that are given below: Now, S_A is utilized for calculating the searching agent location to prevent collision avoidance among its neighboring searching agents (that is sooty terns).

$$\vec{C}_{st} = S_A \times \vec{P}_{st}(z) \quad (20)$$

In the equation, \vec{C}_{st} indicates the location of searching agent could not collide with other searching agents, \vec{P}_{st} signifies the predefine location of searching agent, z characterizes the existing iteration, and the movement of searching agent is S_A .

$$S_A = C_f - \left(z \times \left(\frac{C_f}{Max_{iterations}} \right) \right) \quad (21)$$

Where $z = 0, 1, 2, \dots, Max_{iterations}$, C_f is a controlling parameter for adjusting the S_A that is reduced linearly from C_f to 0. The search agent converges to the direction of optimal neighbour after collision avoidance.

$$\vec{M}_{st} = C_B \times \left(\vec{P}_{st}(z) - \vec{P}_{st}(z) \right) \quad (22)$$

Here \vec{M}_{st} and \vec{P}_{st} are the location and the optimum fittest search agents, C_B represent an arbitrary parameter that is accountable for effective exploration. C_B can be estimated by:

$$C_B = 0.5 \times R_{and} \quad (23)$$

Whereas R_{and} specifies an arbitrary value in [0,1]. At last, the sooty tern or searching agent could upgrade their place.

$$\vec{D}_{st} = \vec{C}_{st} + \vec{M}_{st} \quad (24)$$



Whereas \vec{D}_{st} highlights the difference between a searching agent and the ideal search agents. As they migrate, sooty terns alter their speed and attack angle. The wings increase their altitude. The following air spiral behaviour is generated during the prey attack:

$$x' = R_{adius} \times \sin(i) \quad (25)$$

$$y' = R_{adius} \times \cos(i) \quad (26)$$

$$z' = R_{adius} \times i \quad (27)$$

$$r = u \times e^{kv} \quad (28)$$

Whereas R_{adius} indicates radius of every rotation of the spiral, e characterizes the base of natural logarithm. i constant variable within $[0 \leq k \leq 2\pi]$. constants u and v estimate the spiral shape, and The constant values u and v is 1 are considered.

$$\vec{P}_{st}(z) = (\vec{D}_{st} \times (x' + y' + z')) \times \vec{P}_{bst}(z) \quad (29)$$

In Eq. (29), $\vec{P}_{st}(z)$ update the position of other searching agents and save the optimum solution.

A fitness function (FF) is addressed by the STOA technique in order to attain high classifier accuracy. It designates a positive integer to indicate a potential solution's higher performance. It was expected that FF would minimize its classification error rate in this case.

$$fitness(x_i) = ClassifierErrorRate(x_i) = \frac{No. of misclassified samples}{Total No. of samples} * 100 \quad (30)$$

Performance Validation :

Two benchmark datasets are used to assess the SSAFS-OMLID technique's performance analysis. In addition, a comprehensive study of the classification results is performed using 70% and 30% of training (TR) and testing (TS) dataset.

1. Result Analysis on CICIDS-2017 dataset :

The CICIDS-2017 dataset includes 80 features with eight class labels. A set of 39 features were chosen in this study. The dataset includes a total of 15500 instances.

The SSAFS-OMLID approach generated confusion matrices on 70% of the TRA dataset and 30% of the TES dataset, as shown in Figure 3. The consequences specified that the SSAFS-OMLID technique has appropriately identified the class labels on both datasets.



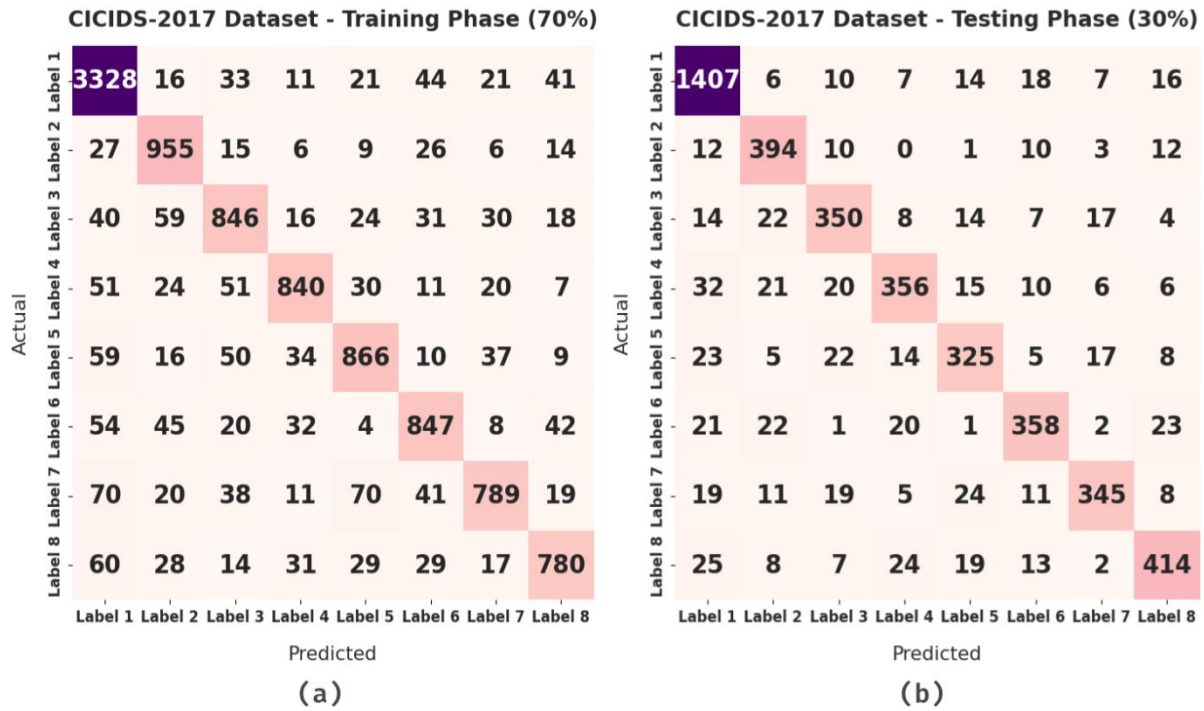


Fig 3: Confusion matrix of SSAFS-OMLID method on CICIDS-2017 dataset

The SSAFS-OMLID method's classifier results on 70% of the TRA dataset and 30% of the TES dataset are reported in Table 1 and Fig. 4, respectively. The simulation consequence implied that the SSAFS-OMLID method has accomplished better results on both datasets. For instance, with 70% of TRA dataset, the SSAFS-OMLID technique has provided $accu_y$ of 96.32%, $prec_n$ of 83.74%, DR of 82.48%, $spec_y$ of 97.81%, F_{score} of 83.02%, and $G_{measure}$ of 83.06%. Besides, with 30% of TES data, the SSAFS-OMLID technique has provided $accu_y$ of 96.23%, $prec_n$ of 83.16%, DR of 82.12%, $spec_y$ of 97.77%, F_{score} of 82.56%, and $G_{measure}$ of 82.60%.

Table 1: Evaluation of the suggested SSAFS-OMLID approach using several metrics on the CICIDS-2017 dataset

Class Label	$Accu_y$	$Prec_n$	Detection Rate	$Spec_y$	F_{score}	$G_{measure}$
Training Phase (70%)						
Label 1	94.95	90.21	94.68	95.08	92.39	92.42
Label 2	97.13	82.12	90.26	97.88	86.00	86.09
Label 3	95.95	79.29	79.51	97.74	79.40	79.40
Label 4	96.91	85.63	81.24	98.56	83.37	83.40
Label 5	96.29	82.24	80.11	98.09	81.16	81.17
Label 6	96.34	81.52	80.51	98.04	81.01	81.02
Label 7	96.24	85.02	74.57	98.58	79.46	79.63
Label 8	96.70	83.87	78.95	98.48	81.33	81.37
Average	96.32	83.74	82.48	97.81	83.02	83.06
Testing Phase (30%)						
Label 1	95.18	90.60	94.75	95.39	92.63	92.65



Label 2	96.92	80.57	89.14	97.74	84.64	84.75
Label 3	96.24	79.73	80.28	97.89	80.00	80.00
Label 4	95.96	82.03	76.39	98.14	79.11	79.16
Label 5	96.09	78.69	77.57	97.92	78.12	78.13
Label 6	96.47	82.87	79.91	98.24	81.36	81.38
Label 7	96.75	86.47	78.05	98.72	82.05	82.15
Label 8	96.24	84.32	80.86	98.14	82.55	82.57
Average	96.23	83.16	82.12	97.77	82.56	82.60

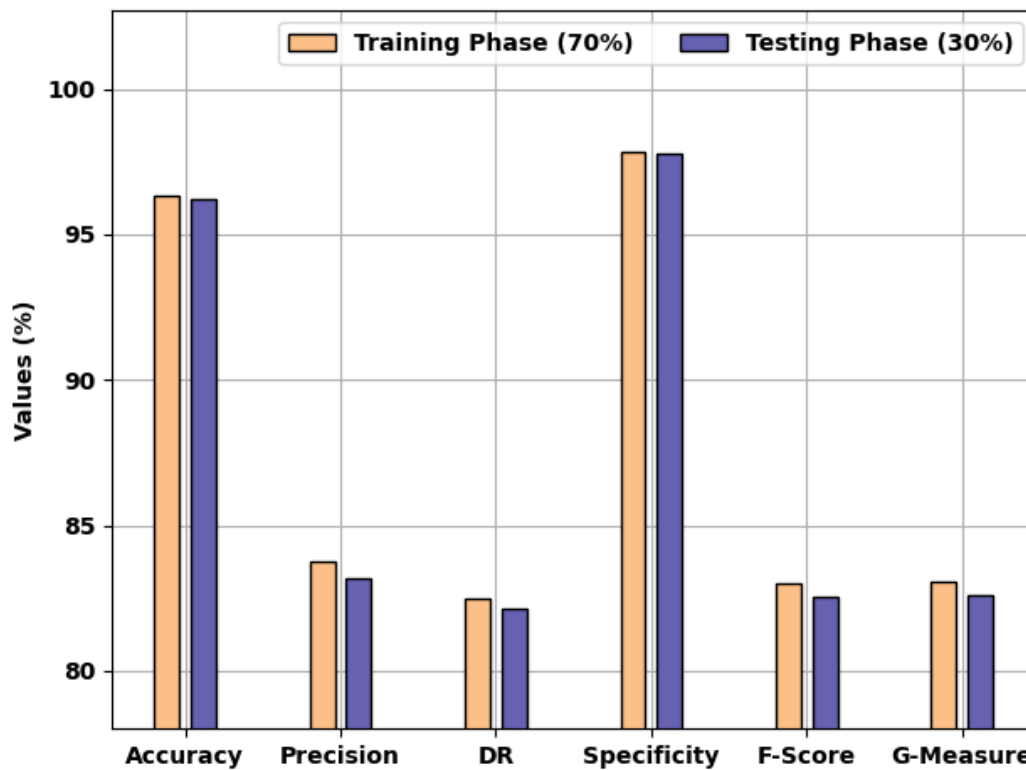


Fig 4: Result analysis of SSAFS-OMLID method on CICIDS-2017 dataset

Figure 5 shows the SSAFS-OMLID method's training accuracy (TRAC) and validation accuracy (VLAC) happening the CICIDS-2017 dataset. The SSAFS-OMLID method demonstrated the best TRAC and VLAC scores. Particularly, the VLAC is seemed to be higher than TRAC.

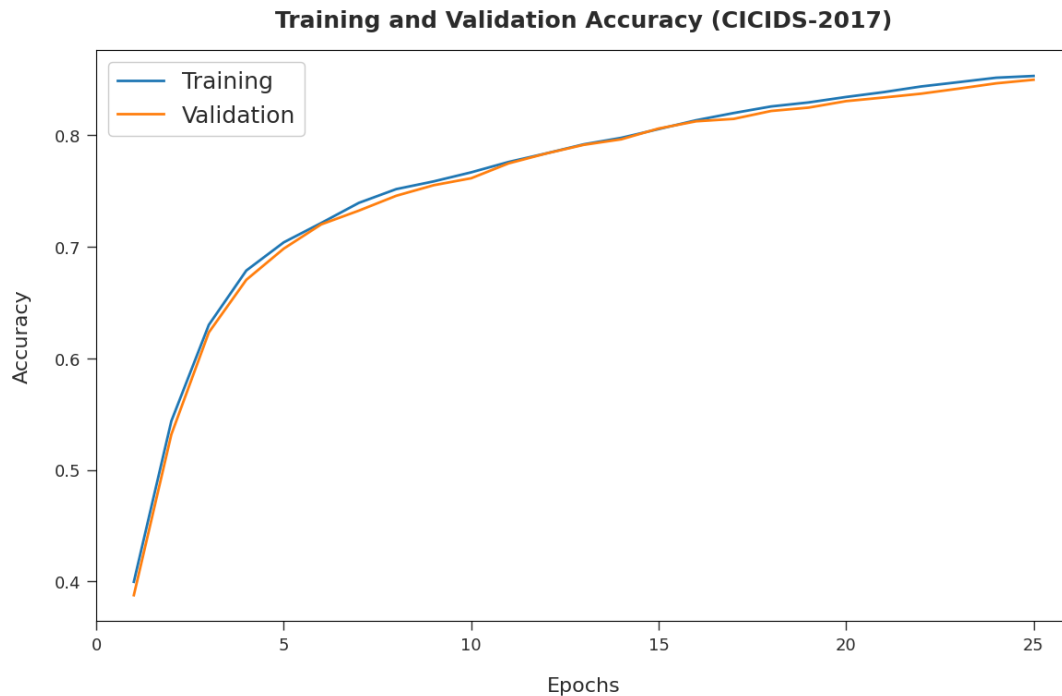


Fig 5: TRAC and VLAC curve of SSAFS-OMLID method on CICIDS-2017 dataset

Figure 6 displays the SSAFS-OMLID technique's prediction of both training loss (TLOS) and validation loss (VLOS) for the CICIDS-2017 dataset. The SSAFS-OMLID method was found to have the best TLOS and VLOS results. The VLOS is located below the TLOS in particular.

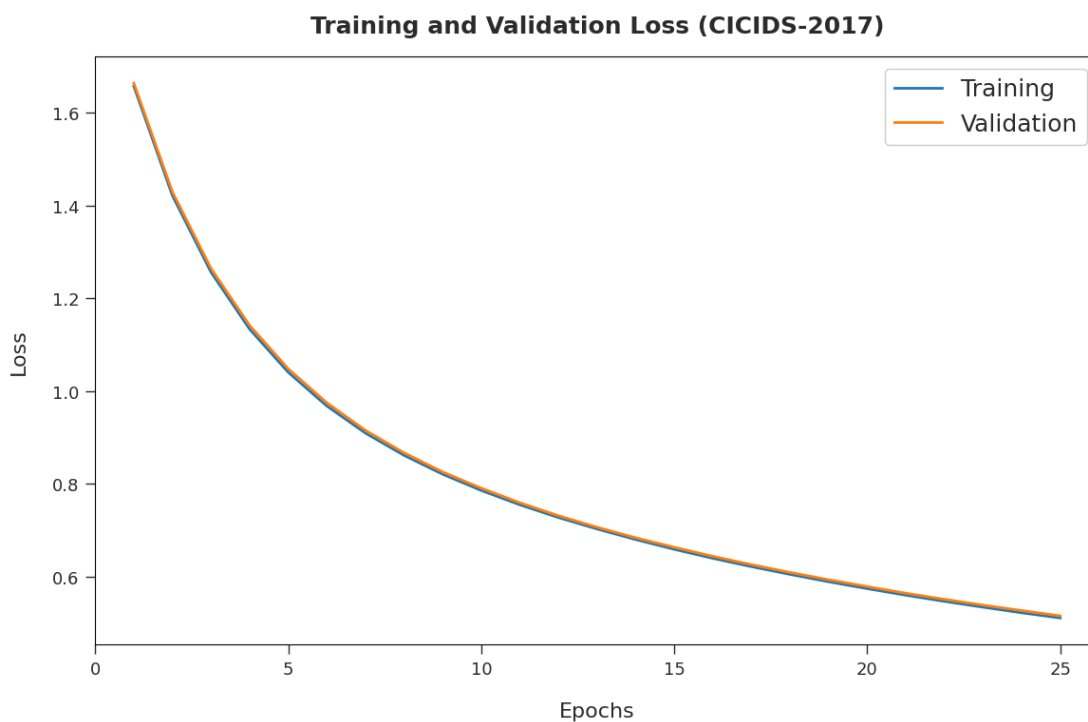


Fig 6: TLOS and VLOS curve of SSAFS-OMLID algorithm on CICIDS-2017 dataset



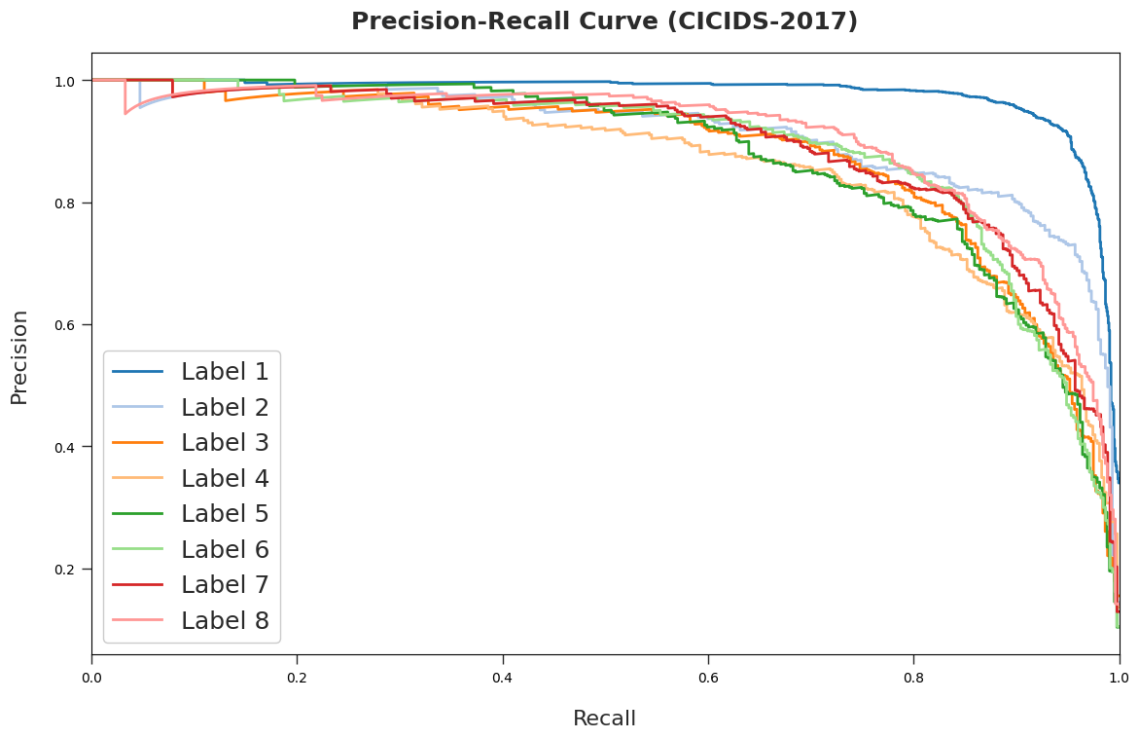


Fig 7: PR analysis of SSAFS-OMLID method on CICIDS-2017 dataset

The SSAFS-OMLID model's detailed PR curve on the CICIDS-2017 dataset is shown in Figure 7. The SSAFS-OMLID approach produces high PR values in every class, as the graphic illustrates.

2. Result Analysis on UNSW-NB15 dataset :

There are ten class labels and forty-two characteristics in the UNSW-NB15 dataset that is provided. A total of 27 features have been chosen for this investigation.

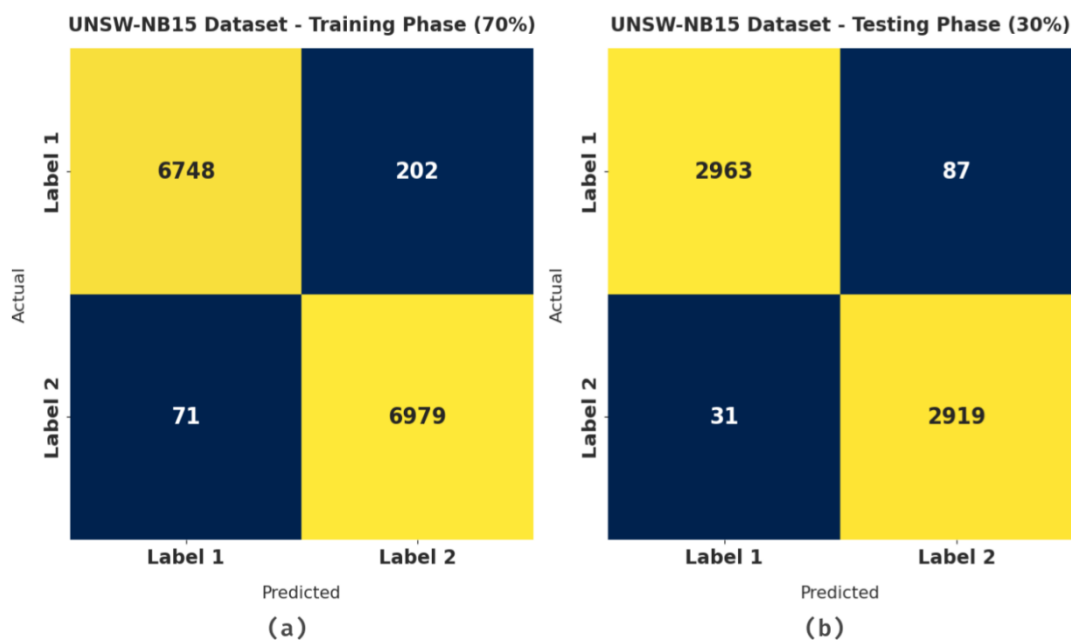


Fig 8: Confusion matrix of SSAFS-OMLID method on UNSW-NB15 dataset

The confusion matrices generated by the SSAFS-OMLID method on 70% and 30% of the TRA and TES datasets, respectively, are shown in Fig. 8. The outcomes specified that the SSAFS-OMLID methodology has properly identified the classes on both sub datasets.

Table 2 and Fig. 9 demonstrate the wide-ranging category results of the SSAFS-OMLID method on 70% and 30% of TRA and TES dataset TES data. The consequence inferred that the SSAFS-OMLID technique has accomplished better results on both dataset. For instance, with 70% of TRA data, the SSAFS-OMLID algorithm has provided $accu_y$ of 98.05%, $prec_n$ of 98.07%, DR of 98.04%, $spec_y$ of 98.04%, F_{score} of 98.05%, and $G_{measure}$ of 98.05%. Eventually, with 30% of TES data, the SSAFS-OMLID method has provided $accu_y$ of 98.03%, $prec_n$ of 98.04%, DR of 98.05%, $spec_y$ of 98.05%, F_{score} of 98.03%, and $G_{measure}$ of 98.04%.

Table 2: Result analysis of SSAFS-OMLID method with numerous measures on UNSW-NB15 dataset

Class Label	$Accu_y$	$Prec_n$	Detection Rate	$Spec_y$	F_{score}	$G_{measure}$
Training Phase (70%)						
Label 1	98.05	98.96	97.09	98.99	98.02	98.02
Label 2	98.05	97.19	98.99	97.09	98.08	98.09
Average	98.05	98.07	98.04	98.04	98.05	98.05
Testing Phase (30%)						
Label 1	98.03	98.96	97.15	98.95	98.05	98.05
Label 2	98.03	97.11	98.95	97.15	98.02	98.02
Average	98.03	98.04	98.05	98.05	98.03	98.04



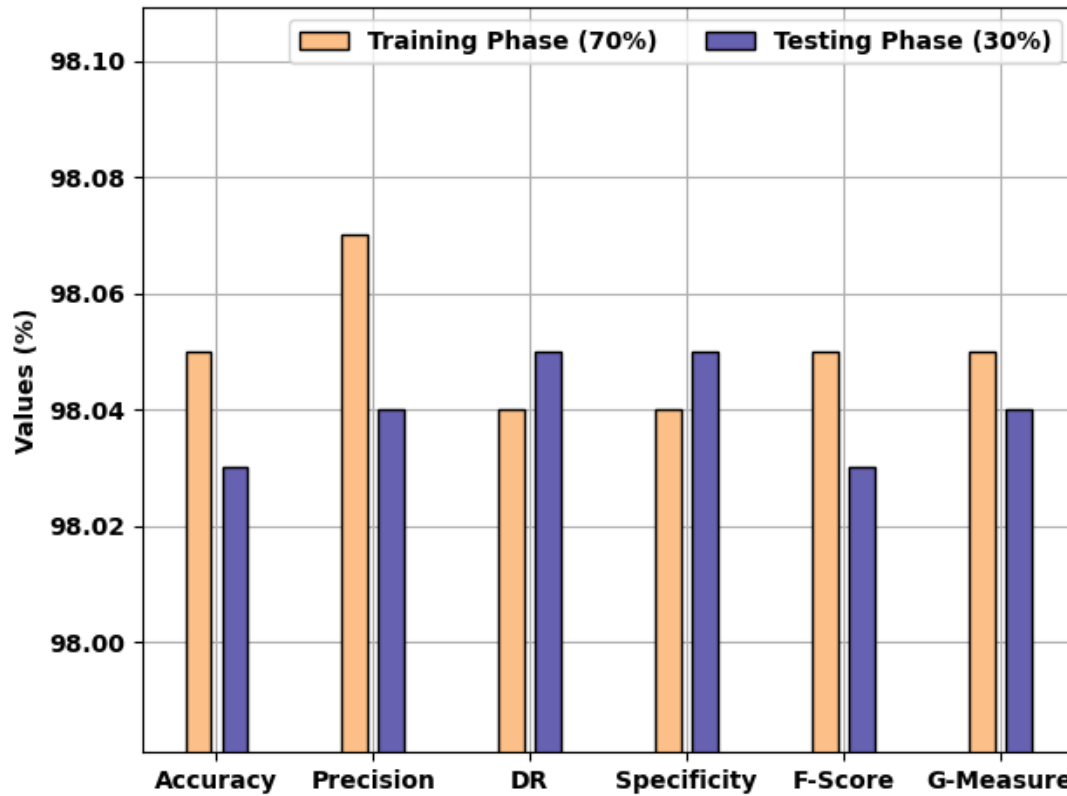


Fig 9: Result analysis of SSAFS-OMLID algorithm on UNSW-NB15 dataset

Figure 10 displays the TRAC and VLAC values that were acquired using the SSAFS-OMLID method on UNSW-NB15 dataset. The results show that VLAC was better than TRAC, and the usage of the SSAFS-OMLID method yielded the best results for both techniques.

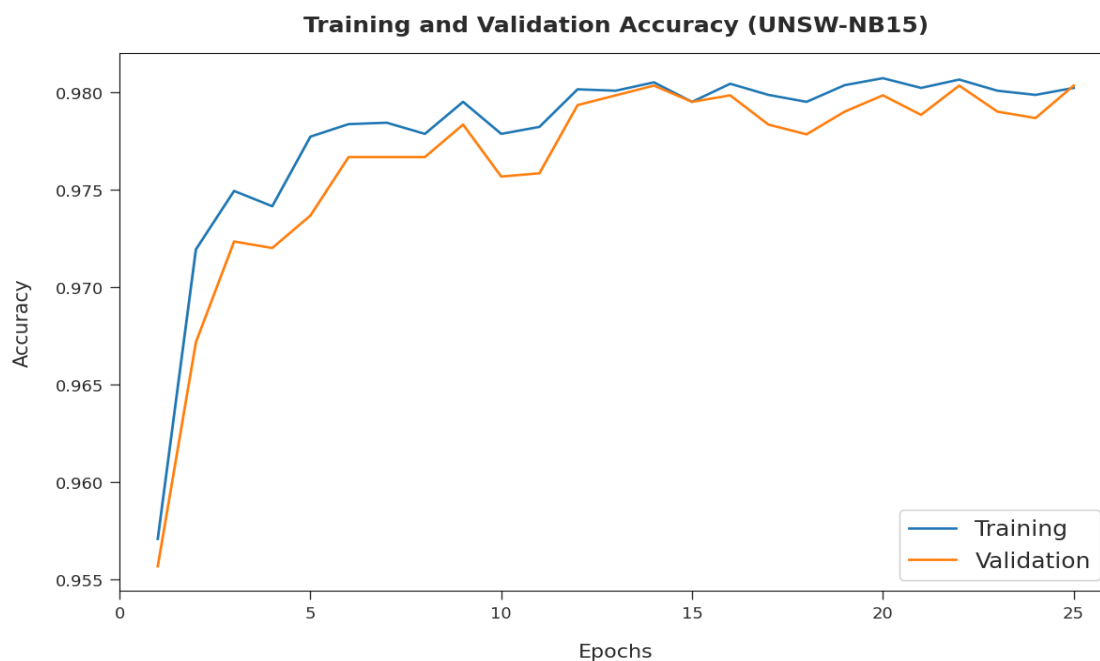


Fig 10: TA and VA analysis of SSAFS-OMLID technique on UNSW-NB15 dataset

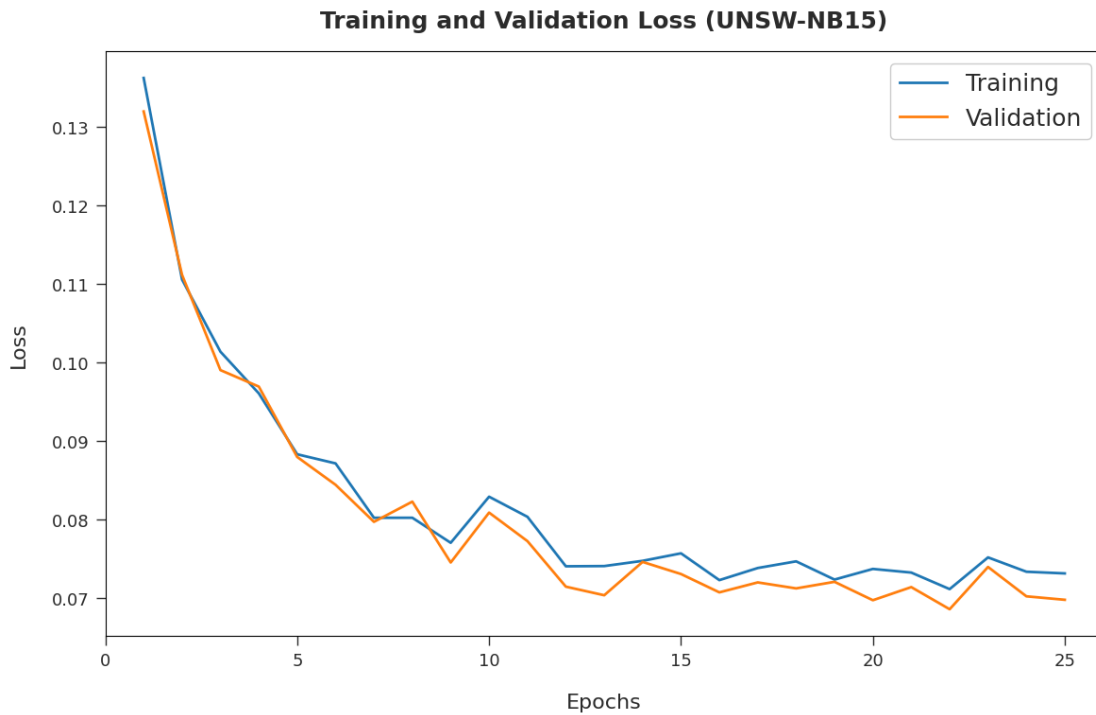


Fig 11: TLOS and VLOS investigation of SSAFS-OMLID algorithm on UNSW-NB15 dataset

Figure 11 shows that the SSAFS-OMLID system was able to achieve TLOS and VLOS on the UNSW-NB15 dataset. The results showed that the SSAFS-OMLID method achieved the lowest TLOS and VLOS values. In example, VL is less than TLOS.

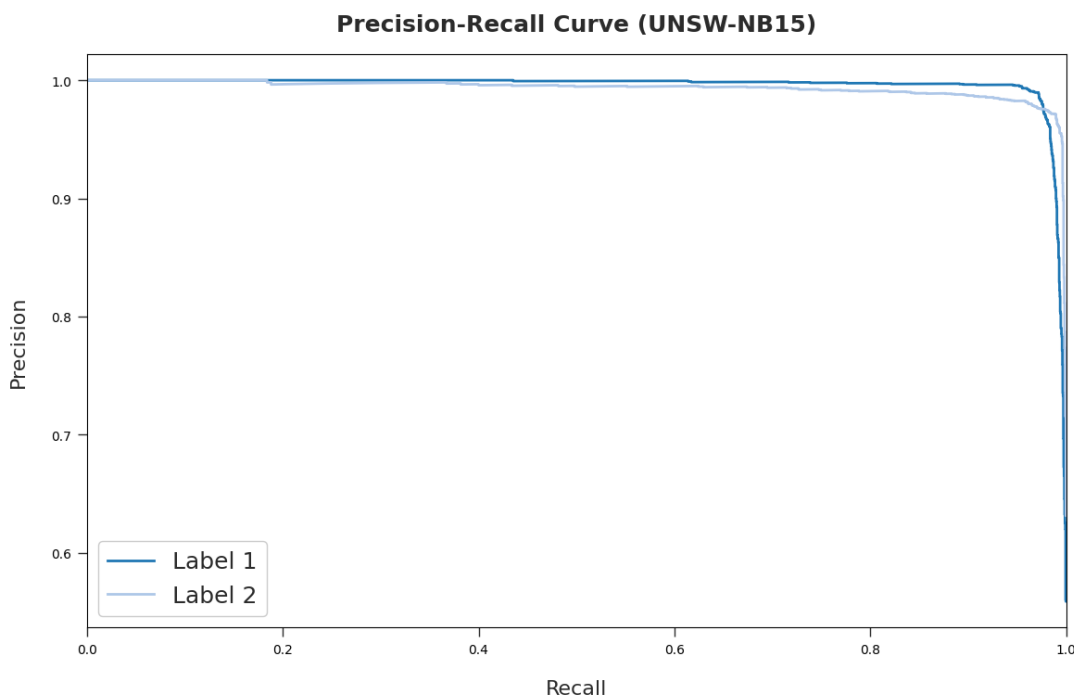


Fig 12: PR analysis of SSAFS-OMLID technique on UNSW-NB15 dataset

A detailed PR curve of the SSAFS-OMLID approach on UNSW-NB15 dataset is represented in Fig. 12, the SSAFS-OMLID technique has attained maximal PR values under all classes.

Discussion :

In order to prove that the SSAFS-OMLID methods provide better results, a comparison analysis is presented in Table 3. Fig. 13 illustrates detection rate (DR) and acc_y inspection of the SSAFS-OMLID method with existing techniques. With respect to DR, the outcomes inferred that the SSAFS-OMLID methodologies has obtained maximum DR of 98.05% whereas the GCNSE, CNN-Bagging, DT-Bagging, NB-Bagging, CNN-Adaboost, DT-Adaboost, NB-Adaboost, RBF, and SVM techniques have obtained minimal DR of 82.80%, 76.83%, 76.15%, 73.85%, 72.64%, 74.37%, 74.82%, 75.22%, and 77.77% correspondingly. Besides, interms of acc_y , the results specified that the SSAFS-OMLID method has obtained maximum acc_y of 98.03% whereas the GCNSE, CNN-Bagging, DT-Bagging, NB-Bagging, CNN-Adaboost, DT-Adaboost, NB-Adaboost, RBF, and SVM methodologies have reached minimal acc_y of 80.76%, 76.18%, 76.13%, 71.36%, 75.28%, 72.18%, 74.85%, 74.36%, and 77.36% correspondingly.

Table 3: Evaluation of the SSAFS-OMLID method in comparison to other approaches

Methods	Detection Rate	$Accu_y$	$Prec_n$	F_{score}
SSAFS-OMLID	98.05	98.03	98.04	98.03
GCNSE Model	82.80	80.76	81.57	81.74
Conv. Neural Network-Bagging	76.83	76.18	82.59	79.84
Decision Tree-Bagging	76.15	76.13	83.53	78.17
Naïve Bayes-Bagging	73.85	71.36	70.95	72.38
Conv. Neural Network-Adaboost	72.64	75.28	70.29	69.25
Decision Tree-Adaboost	74.37	72.18	75.69	74.98
Naïve Bayes-Adaboost	74.82	74.85	80.12	77.77
Random Forest Algorithm	75.22	74.36	81.68	78.10
Support Vector Machine	77.77	77.36	79.85	78.57



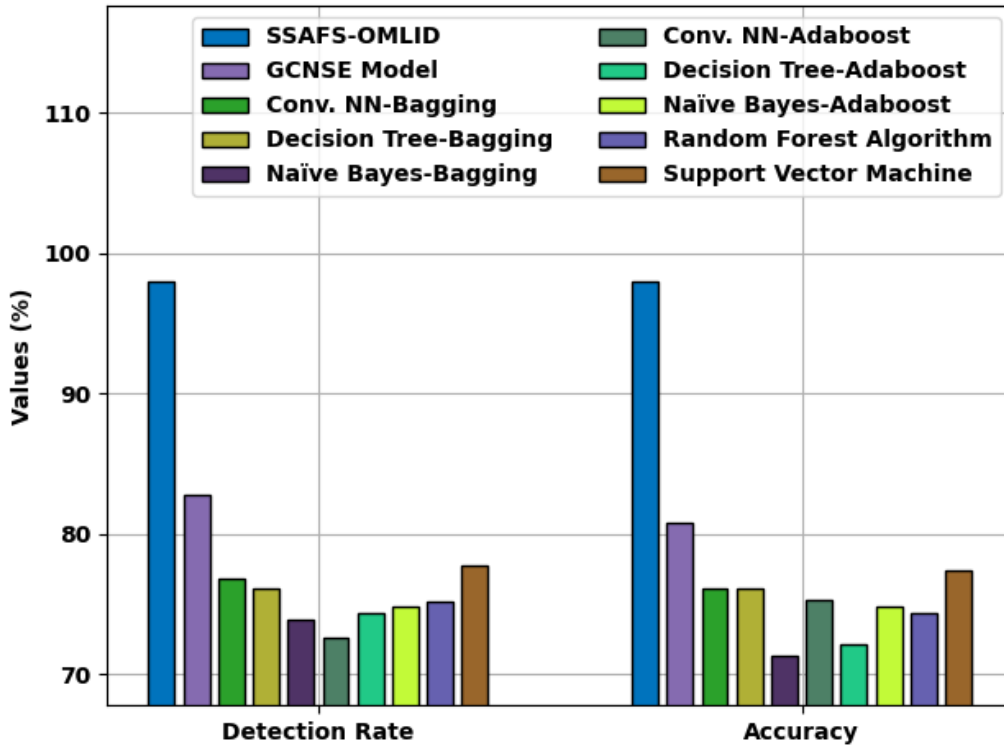


Fig 13: DR and acc_y analysis of SSAFS-OMLID method with existing approaches

Fig. 14 showcases $prec_n$ and F_{score} analysis of the SSAFS-OMLID method with other techniques. With respect to DR, the results indicated that the SSAFS-OMLID method has attained high $prec_n$ of 98.04% whereas the GCNSE, CNN-Bagging, DT-Bagging, NB-Bagging, CNN-Adaboost, DT-Adaboost, NB-Adaboost, RBF, and SVM approaches have obtained minimal $prec_n$ of 81.57%, 82.59%, 83.53%, 70.95%, 70.29%, 75.69%, 80.12%, 81.68%, and 79.85% respectively. Besides, interms of F_{score} , the results specified that the SSAFS-OMLID algorithm has gained maximal F_{score} of 98.03% whereas the GCNSE, CNN-Bagging, DT-Bagging, NB-Bagging, CNN-Adaboost, DT-Adaboost, NB-Adaboost, RBF, and SVM models have obtained minimal F_{score} of 81.74%, 79.84%, 78.17%, 72.38%, 69.25%, 74.98%, 77.77%, 78.10%, and 78.57% correspondingly.

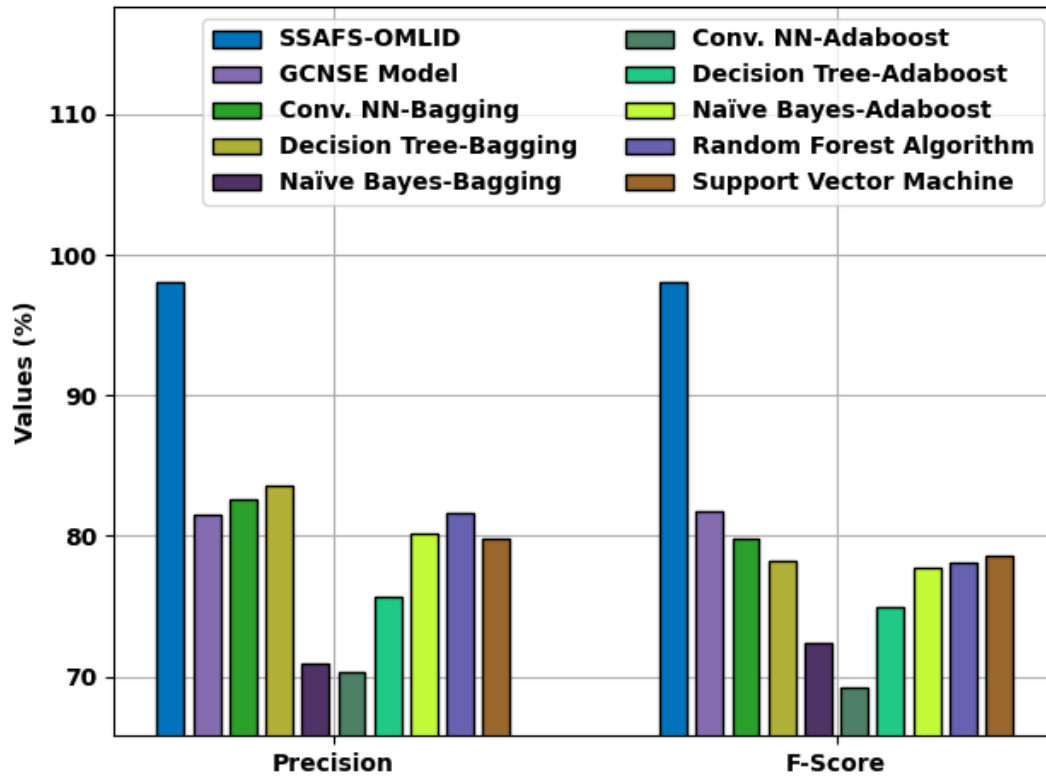


Fig 14: $Prec_n$ and F_{score} analysis of SSAFS-OMLID method with existing approaches

Finally, a detailed training time (TRAT) and testing time (TEST) examination of the SSAFS-OMLID method with existing models are made in Table 4 and Fig. 15. The consequences specified that the SSAFS-OMLID method has incurred minimal TRAT and TSET over the other approaches. For instance, with respect to TRAT, the SSAFS-OMLID method has offered lower TRAT of 125s whereas the SVM model has attained higher TRAT of 297s.

Table 4: TRAT and TEST time analysis of SSAFS-OMLID algorithm with existing techniques

Methods	TRAT (sec)	TEST (sec)
SSAFS-OMLID	125	113
GCNSE Model	183	174
Conv. Neural Network-Bagging	239	202
Decision Tree-Bagging	265	203
Naïve Bayes-Bagging	167	164
Conv. Neural Network-Adaboost	273	213
Decision Tree-Adaboost	148	142
Naïve Bayes-Adaboost	272	247
Random Forest Algorithm	208	183
Support Vector Machine	297	198

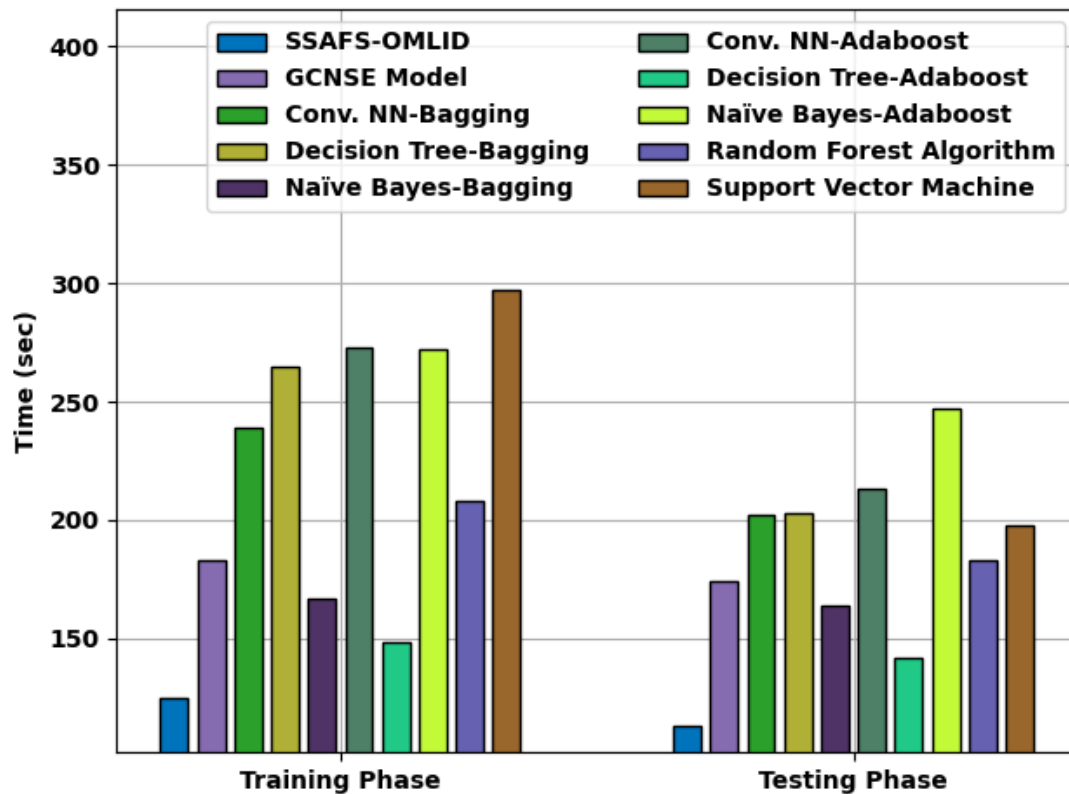


Fig 15: TRAT and TEST analysis of SSAFS-OMLID method with existing algorithms

Similarly, with respect to TEST, the SSAFS-OMLID model has presented inferior TEST of 113s whereas the Naïve Bayes-Adaboost model has reached greater TEST of 247s. The above mentioned result analysis specified that the SSAFS-OMLID technique has obtained effectual outcomes over the other techniques.

Conclusion :

In order to identify and prevent intrusions into the IoT framework, this study created a new SSAFS-OMLID method. The first step is to do data preprocessing to standardise the data into an appropriate format. From the preprocessed data, the best features are then found and chosen using the SSAFS approach. Data categorisation using the HNN approach yields accurate class labels. Using the STOA, the parameters of the HNN algorithm are finally fine-tuned. A thorough experimental analysis was carried out on benchmark datasets to validate the improved SSAFS-OMLID technique performance. Numerous comparison results showed that the SSAFS-OMLID strategy outperformed competing approaches in a number of evaluation metrics. In the future, the presented method was deployed in the big data environment with heterogeneous data sources.

References :

- Asharf, J., Moustafa, N., Khurshid, H., Debie, E., Haider, W. and Wahab, A., 2020. A review of intrusion detection systems using machine and deep learning in internet of things: challenges, solutions and future directions. *Electronics*, 9(7), p.1177.



- da Costa, K.A., Papa, J.P., Lisboa, C.O., Munoz, R. and de Albuquerque, V.H.C., 2019. Internet of Things: A survey on machine learning-based intrusion detection approaches. *Computer Networks*, 151, pp.147-157.
- Alsoufi, M.A., Razak, S., Siraj, M.M., Nafea, I., Ghaleb, F.A., Saeed, F. and Nasser, M., 2021. Anomaly-based intrusion detection systems in iot using deep learning: A systematic literature review. *Applied Sciences*, 11(18), p.8383.
- Maheshwar Reddy, V., Ravi Prakash Reddy, I., Adi Narayana Reddy, K., 2020. An Efficient Intrusion Detection System with Convolutional Neural Network. In: *Chillarige R., Distefano S., Rawat S. (eds) Advances in Computational Intelligence and Informatics. ICACII 2019. Lecture Notes in Networks and Systems*, 119. Springer, Singapore. https://doi.org/10.1007/978-981-15-3338-9_22
- Mekala, S., Jatothu, R., Kodati, S., Pradeep Reddy, K., Sreekanth, N., 2023. Network Intrusion Detection Using Machine Learning for Virtualized Data. In: Mandal, J.K., Hinchey, M., Rao, K.S. (eds) *Innovations in Signal Processing and Embedded Systems. Algorithms for Intelligent Systems. Springer*, Singapore. https://doi.org/10.1007/978-981-19-1669-4_21
- Shankar, D., Victo Sudha George, G., & Kanya, N. 2023. OptiBiNet_GRU: Robust Network Intrusion Detection System Using Optimum Bi-Directional Gated Recurrent Unit. *International Journal of Intelligent Engineering and Systems*, 16(3), pp. 75-91
- Verma, A. and Ranga, V., 2020. Machine learning based intrusion detection systems for IoT applications. *Wireless Personal Communications*, 111(4), pp.2287-2310.
- Susilo, B. and Sari, R.F., 2020. Intrusion detection in IoT networks using deep learning algorithm. *Information*, 11(5), p.279.
- Thamilarasu, G. and Chawla, S., 2019. Towards deep-learning-driven intrusion detection for the internet of things. *Sensors*, 19(9), p.1977.
- Bala, B., & Behal, S. (2024). AI techniques for IoT-based DDoS attack detection: Taxonomies, comprehensive review and research challenges. *Computer science review*, 52, 100631.
- Bakhsh, S. A., Khan, M. A., Ahmed, F., Alshehri, M. S., Ali, H., & Ahmad, J. (2023). Enhancing IoT network security through deep learning-powered Intrusion Detection System. *Internet of Things*, 24, 100936.
- Dina, A. S., Siddique, A. B., & Manivannan, D. (2023). A deep learning approach for intrusion detection in Internet of Things using focal loss function. *Internet of Things*, 22, 100699.
- Wang, S., Xu, W., & Liu, Y. (2023). Res-TranBiLSTM: An intelligent approach for intrusion detection in the Internet of Things. *Computer Networks*, 235, 109982.
- Maseer, Z.K., Yusof, R., Mostafa, S.A., Bahaman, N., Musa, O. and Al-rimy, B.A.S., 2021. DeepIoT. IDS: hybrid deep learning for enhancing IoT network intrusion detection. *CMC-Computers, Materials & Continua*, 69(3), pp.3945-3966.
- Jothi, B. and Pushpalatha, M., 2021. WILS-TRS—A novel optimized deep learning based intrusion detection framework for IoT networks. *Personal and Ubiquitous Computing*, pp.1-17.
- Fatani, A., Dahou, A., Al-Qaness, M.A., Lu, S. and Elaziz, M.A., 2021. Advanced



feature extraction and selection approach using deep learning and Aquila Optimizer for IoT intrusion detection system. *Sensors*, 22(1), p.140.

- Gad, A.R., Nashat, A.A. and Barkat, T.M., 2021. Intrusion Detection System Using Machine Learning for Vehicular Ad Hoc Networks Based on ToN-IoT Dataset. *IEEE Access*, 9, pp.142206-142217.
- Zhong, M., Zhou, Y. and Chen, G., 2021. Sequential model based intrusion detection system for IoT servers using deep learning methods. *Sensors*, 21(4), p.1113.
- Otoum, Y., Liu, D. and Nayak, A., 2019. DL-IDS: a deep learning-based intrusion detection framework for securing IoT. *Transactions on Emerging Telecommunications Technologies*, p.e3803.
- Raghuvanshi, A., Singh, U.K., Sajja, G.S., Pallathadka, H., Asenso, E., Kamal, M., Singh, A. and Phasinam, K., 2022. Intrusion detection using machine learning for risk mitigation in IoT-enabled smart irrigation in smart farming. *Journal of Food Quality*, 2022.
- Moizuddin, M.D., and Jose, M.V., 2022. A bioinspired hybrid deep learning model for network intrusion detection. *Knowledge-Based Systems*, 238, p. 107894.
- Rani, M., 2022. Effective network intrusion detection by addressing class imbalance with deep neural networks multimedia tools and applications. *Multimedia Tools and Applications*, 81(6), pp. 8499-8518.
- Ravi, V., Chaganti, R., and Alazab, M., 2022. Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system. *Computers and Electrical Engineering*, 102, p. 108156.
- Andresini, G., Appice, A., and Malerba, D., 2021. Autoencoder-based deep metric learning for network intrusion detection. *Information Sciences*, 569, pp. 27-706.
- Balasundaram, J., 2021. Retracted: A novel optimized Bat Extreme Learning intrusion detection system for smart Internet of Things networks. *International Journal of Communication Systems*, 34(7), p. e4729.
- Folino, G., Godano, C.O., Pisani, F.S., (2023). An ensemble-based framework for user behaviour anomaly detection and classification for cybersecurity. *The Journal of Supercomputing*, 79, pp. 11660-11683. <https://doi.org/10.1007/s11227-023-05049-x>
- Alrayes, F.S., Alotaibi, N., Alzahrani, J.S., Alazwari, S., Alhogail, A., Al-Sharafi, A.M., Othman, M., and Hamza, M.A., (2023). Enhanced gorilla troops optimizer with deep learning enabled cybersecurity threat detection. *Computer Systems Science and Engineering*, 45(3): 3037-3052. <https://doi.org/10.32604/csse.2023.033970>
- Dehghani, M., Montazeri, Z., Dehghani, A. and Seifi, A., 2017, December. Spring search algorithm: A new meta-heuristic optimization algorithm inspired by Hooke's law. In *2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI)* (pp. 0210-0214). IEEE.
- Muhammad Sidik, S. S., Zamri, N. E., Mohd Kasihmuddin, M. S., Wahab, H. A., Guo, Y., & Mansor, M. A. (2022). Non-systematic weighted satisfiability in discrete hopfield neural network using binary artificial bee colony optimization. *Mathematics*, 10(7), 1129.

