[1]Satish Kumar Garapati

[2]AN. Sigappi

An Artificial Intelligence-Based Intrusion Detection System using Optimization and Deep Learning

Abstract: - The Intrusion Detection System (IDS) was developed to resolve the malicious activities on the networking. However, the existing IDS models face issues in achieving high accuracy, and it is prone to detection error. Hence, to address these challenges, an innovative Vulture-based Deep Belief System (VbDBNS) for enhancing the performance of detecting intrusion activity by monitoring their behaviour and features. This study utilized the NSL-KDD database for training and validating the system. Henceforth, pre-processing is employed to standardize the dataset using Min-Max normalization, and 1-N encoding, which transforms the categorical data variables. Moreover, feature extraction was utilized to capture the most significant 42 attributes from the pre-processed database. Further, a classification module was designed using VbDBN to categorize normal and malicious data in which the fitness of vulture was updated, which enables it to monitor and detect the intrusion continuously. The experimental results of the study illustrate that the designed methodology achieved better outcomes than the conventional techniques in terms of recall, accuracy, precision, F1-score, etc.

Keywords: Intrusion Detection, Labels, Threshold Value, Normalization, Features, Deep Belief Neural Network Person, Detection, Recurrent Network.

1. INTRODUCTION

Generally, IDS is one of the tools for protecting the system from intruders that monitors the computer network or single machine for intruders [1]. Furthermore, IDS monitors the activity and detects intrusion for breaking the security, and offers significant information to measure timely counters [2]. In the beginning stage, the IDS system resolves the anomaly data handling on computers [3]. Also identify the attempted or ongoing attacks or intrusions by performing tasks like data accumulation, data dimension minimization, behavior identification, reporting and communication [4]. In addition, the IDS framework aims to observe the user characteristics to ensure security to the data.

The prediction of malicious attacks is processed by generating an alarm [5]. However, the evolution of different malicious software over recent years has induced many issues in IDS [6]. Moreover, data security and computer security are the most significant information technology for eliminating security evasions and threats [7]. Additionally, massive use of networking and the internet activated additional security problems, confidentiality, network availability, and data integrity [8]. Those invasions are called network intrusion that performs unauthorized activity in the network [9]. It involves the process of tampering, stealing, or accessing resources and compromising the data integrity in the network [10].

Especially, some attacks occurred in the network are User-to-Root (U2R), Root-to-Local (R2L), Denial of Service (DoS), probe, etc. [11]. The most challenging task in a cloud environment is computer network security due to internet service development [12, 13]. Many applications and tools are established for increasing security such as networks, systems, and computer environments [14]. Nowadays, Artificial Intelligence (AI) is exponentially great growth and is used in many research domains including robotics, computer vision, regulation, and numerous emerging fields [15]. Additionally, AI is integrated with certain files like expert systems, neural networks, fuzzy systems, and so on [16]. The basic IDS are shown in fig.1.

[1] *[1]Research Scholar, Department of Computer Science and Engineering, Annamalai University*

Annamalainagar – 608002

gskchowdari@gmail.com

[2]Professor, Department of Computer Science and Engineering, Annamalai University

Annamalainagar – 608002

an.sigappi@gmail.com

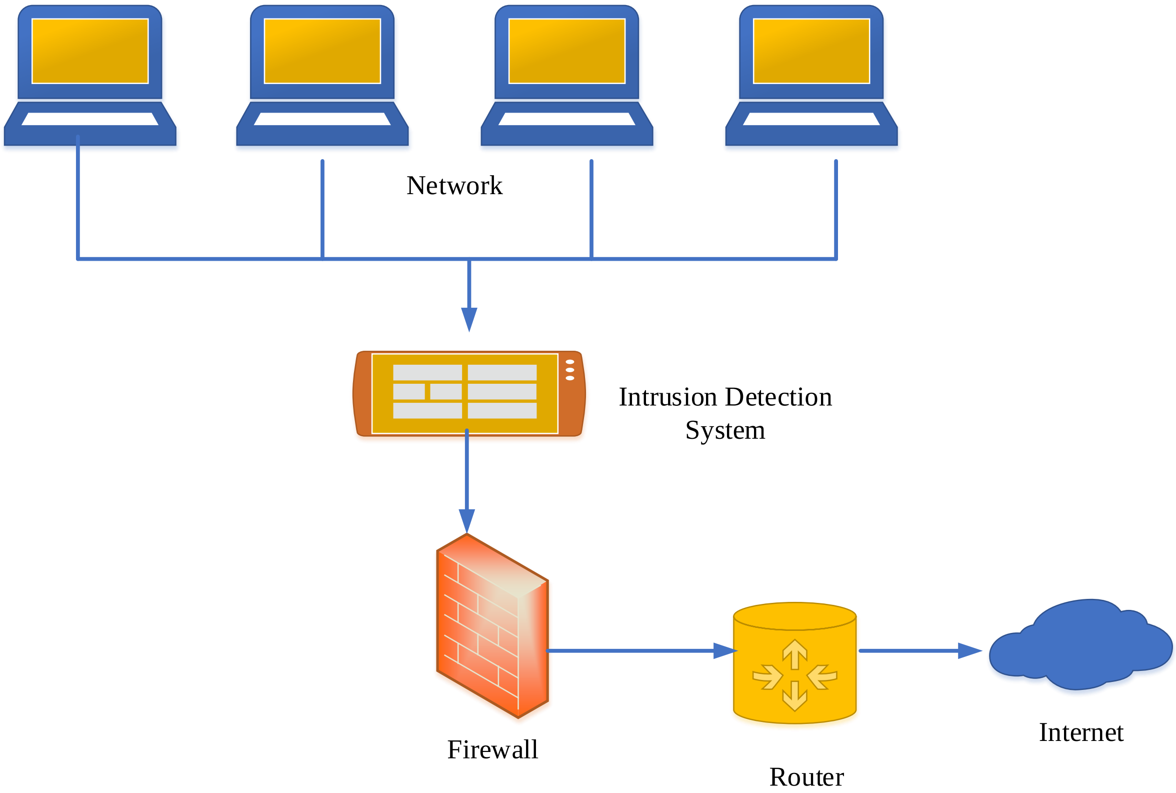

Fig.1 Basic intrusion detection system

Generally, the major problem in IDS is a lower detection rate and failure to categorize the intrusion events accurately [17]. Moreover, the generation of false alarms is another problem in IDS, which imposes huge stress on the research team for resolving intrusion activity [18]. The Machine Learning (ML) and Deep Learning (DL) methods have immense ability for boosting the existing IDS because of their efficiency of identifying distinct and variant attacks [19]. Additionally, ML-based IDS is useful to attain improved accuracy in detection with quality and sufficient data but DL-based IDS is more suitable for dealing the big data [20]. It investigates the accumulated data and absorbs the represented features for producing outcomes in a reliable manner [27]. The most common issues in IDS are less detection rate, false alarm, misclassification, overfitting, and security. Although various algorithms are designed to improve the IDS performances, a reliable solution is still not found [28]. In this research, an AI-assisted DL IDS framework is developed for enhancing the IDS by attaining better performance outcomes.

Objectives

➢ To focus on anomaly-based IDS for accurate detection.

➢ To overcome the high rate of false alarms design Anomaly-based IDS and continuous efforts to reduce the high false positive rate.

➢ To observe the classification scheme is as good with more clean data and best feature representation, and then higher accurate results.

The present research article was systematized as: part 2 discussed the relevant literatures on IDS; part 3 elaborated on the problem definition and system model. Additionally, part 4 detailed the working of the presented framework, and part 5 projected the presented model’s outcomes. Finally, part 6 discussed the article's conclusion.

2. RELATED WORKS

Some of the current works associated with IDS are analyzed below,

Neelu Khare et al [21] proposed DNN based Spider Monkey Optimization (SMO) system for minimizing the dataset and is useful for dimensional reduction. Moreover, the developed model used the NSL-KDD cup 99 datasets and attained outcomes showing 92% accuracy, 92.7% precision, and so on. Thus the designed model is validated with Principal Component Analysis (PCA) and DNN. However, it contains low-quality of data and security issues.

Generally, IDS is the system that can better the learning rate from new and previous attacks. Mohammed Hasan et al [22] proposed a fast learning framework with Particle Swarm Optimization (PSO) to reduce the false positives and negatives count and enhance the ability of learning rate. Furthermore, the KDD99 dataset was tested and validated for attaining better accuracy but generated an overfitting problem.

Vanmathi et al [23] proposed network and host-based IDS for continuous monitoring of network traffic and detecting attacks. Thus the designed IDS contain flexible hardware and software that detect network attacks by their pictures. Additionally, the KDDCUP99 dataset is used for detecting malicious attacks and the random forest system is used to predict the predominant independent variables. However, it obtains local optimum and gradient vanishing problems.

Especially, IDS is more attention to the security of dynamic information in the network. Moreover, hackers or attackers easily enter the network and hack vital information. Sekhar et al [24] developed a novel IDS-based Deep Autoencoder (DA) and Fruitfly Optimization (FO) for handling the imprecision dataset through roughest and fuzzy values. Further, the attributes are fed into the neural system for classifying attacks but data quality is low and less robust.

Deris et al [25] proposed DL-based IDS with pre-training and Autoencoder model for detecting the intrusion with Hyperparameter Optimization (HO) events. It offers an alternative solution through optimized parameters. Moreover, hyperparameter optimization enhances the detection performance and it uses the NSL-KDD database for training and validation processes. It attains the best score in multiclass classification but security and attack classification problems are generated.

The main contribution of the work is brief as follows,

- Firstly, the NSL-KDD database was accumulated from the net source and trained in the system using the Python tool.
- Accordingly, a novel VbDBNS was designed with appropriate parameters for improving IDS.
- Henceforth, preprocessing is employed to standardize the dataset that removes errors using Min-Max normalization and 1-N encoding.
- Additionally, a feature engineering module is deployed to capture the 42 attributes from the dataset.
- Further, upgrade the vulture fitness in the classification module, enabling to observe the user characteristics and intrusions precisely.
- Successively, the performance metrics of the designed model results are equated with existing appliances considering the parameters like accuracy, precision, f-measure, recall, FPR, and execution time.

3. SYSTEM MODEL AND PROBLEM STATEMENT

In the Recurrent Neural Network (RNN), the information flows from the input to hidden units [29]. It contains input, hidden, and output units. In RNN, the hidden units play a crucial role for identifying attacks. The hidden units introduce the directional loop for memorizing previous information process and applying it to output. Besides, the NSL-KDD database was employed for IDS, and preprocessing involves numericalization and normalization for converting a few non-numeric features [30]. Then forward propagation is developed to calculate output values and update the weight for differentiating normal and intrusion [31]. The problem definition and system model are shown in fig.2. Finally, the RNN-IDS framework detects the intrusion but it occurs some problems like high execution time, less detection, and a high false alarm rate [32].

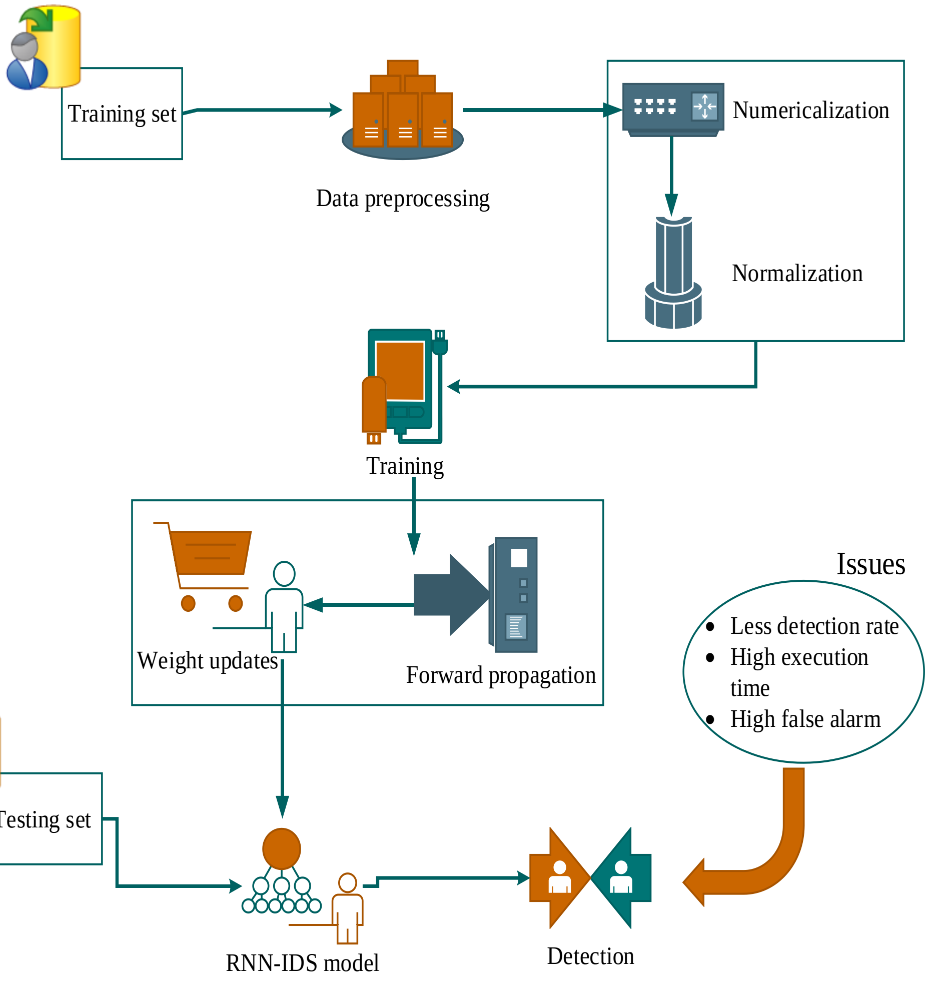

Fig.2 Problem definition and system model

However, security, overfitting, low quality of data, and multiclass issues degrade the performance of IDS. Also, attains less detection and sensitivity to detect the intrusion or attacks because of vast data and data complexity. So, AI-based ML and DL techniques are designed that attain a better detection rate but in some cases, it takes more execution time and attained less reliability and scalability due to complex data. These problems motivated this research on the IDS environment.

4. PROPOSED METHODOLOGY

The action of accessing the system in an unauthorized manner is called intrusion which badly affects the availability and confidentiality integrity of the system. The primary concern of IDS is to identify the malicious behaviour in the network. Initially, the NSL-KDD dataset was gathered from the public source, and trained using the Python tool. Accordingly, a novel VbDBNS has been designed to enhance the IDS outcomes by observing the user characteristics and attributes continuously. Henceforth, preprocessing is employed to normalize and standardize the dataset that removes errors using Min-Max normalization and 1-N encoding. Furthermore, feature extraction was utilized to capture the most significant 42 attributes from the pre-processed database. Further, a classification module was designed using VbDBN to categorize normal and malicious data in which the fitness of vulture was updated, which enables it to monitor and detect the intrusion continuously. The architecture of the proposed methodology is shown in fig.3.

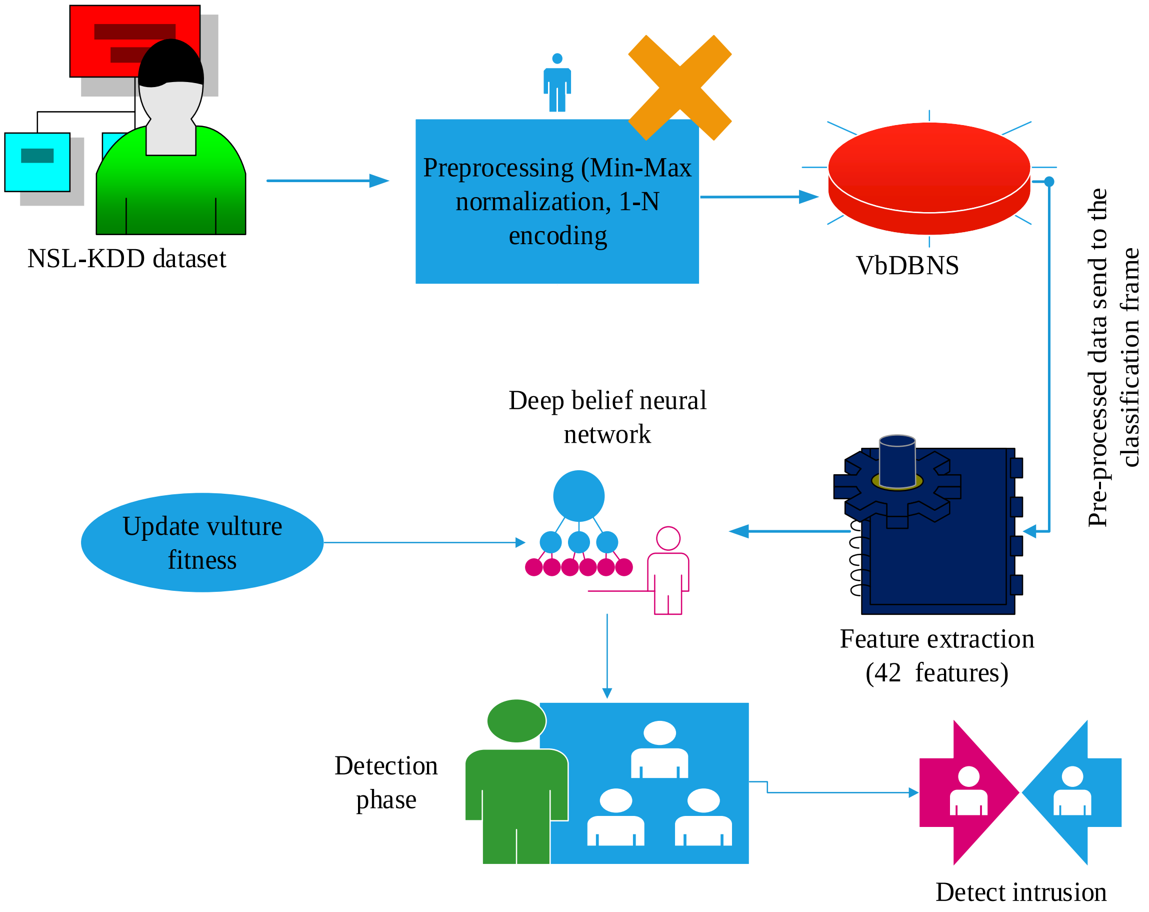
NSL-KDD dataset
Preprocessing (Min-Max normalization, 1-N encoding
VbDBNS
Pre-processed data send to the classification frame
Deep belief neural network
Update vulture fitness
Feature extraction (42 features)
Detection phase
Detect intrusion

Fig.3 Proposed methodology

Thus the designed model achieves better detection accuracy than other conventional neural techniques. Additionally, a vulture optimization [26] model was approved to tune the DL factors and attain the best intrusion classification outcomes. Additionally, the VbDBNS technique was managed as an intelligent scheme for monitoring the intrusion activity in the network.

4.1 Dataset Collection

In the designed model NSL-KDD dataset was used to illustrate the developed VbDBNS for detecting and classifying the intrusion detection data, and it is available at https://www.unb.ca/cic/datasets/nsl.htmlit. This database comprises 41 attributes and 1 class label. In the presented study, we split the database as 70% for training and 30% for model testing, enabling us to evaluate the performance of detecting intrusion in binary classification. The detailed specification of the NSL-KDD dataset is detailed in the table.1.

Table.1 NSL-KDD dataset

Type	No. of. original records	No. of. Newly generated records	Total
Normal	50569	0	50569
DoS	34443	16126	50569
U2R	37	50532	50569
R2L	8709	41860	50569
Probe	721	49848	50569

4.1 Data preprocessing

Additionally, data preprocessing is approved in the NSL-KDD dataset by relating min-max normalization and 1-N encoding. Moreover, min-max normalization is used for normalizing the dataset and 1-N encoding is used for getting an appropriate form of data for the classification process.

- **Min-max normalization**

It minimizes the diverse scales of features or dimensions and the normalization modifies the data by an exact small range by carrying out the linear transformation of the original data. Furthermore, data dimension values are normalized up to the range of [0, 1] by min-max normalization and the transformation of data is obtained by Eqn. (1)

$$m(ia) = \frac{d - \min_k}{\max_k - \min_k}\left(T\min_k - T\max_k\right) + T\min_k \tag{1}$$

Let, d is denoted as the transformed rate of the data value and k is considered as a dimension or feature. Moreover, $\min_k$ is denoted as the original minimum value of k and $\max_k$ is represented as the original maximum value of k. Additionally, $T\min_k$ is denoted as transformed minimum value with k and $T\max_k$ is considered as transformed maximum value with k.

- **1-N encoding**

1-N encoding is the essential process for converting categorical data variables into DL. Also enhances the detection and classification accuracy of the designed technique. Moreover, it is the process of converting categorical variables into numerical values. It generates the new binary features for every possible category and also assigns a value of 1 to each sample feature that corresponds to the original category. Furthermore, 1-N encoding converts the labels into numeric form and the feature set is among 0 and n-class 1, n is defined as some distinct labels.

4.2 Design of Vulture based Deep Belief Neural System (VbDBNS)

This designed procedure is one of the greedy and intelligent hunting appliances that developed layer-by-layer systems for assessing neural weights. In this phase, input variables in utilized in one layer and connect to other layers. Furthermore, the proposed model contains double layers that contain hidden units and weights. At this point, the weight layer is triggered through the input and the hidden layers that are activating in the initialization of the intrusion detection function. Additionally, the developed model process has started using Eqn. (2).

$$A_p(n) = n(0), n(0-1)......n(1) \tag{2}$$

Let, A_p is denoted as the activation parameter and n is represented as a number of the pre-processed dataset. The hidden layers of the estimated neural model are termed as h and the input layer is represented as n. The initialization of the developed method is shown in fig.4.

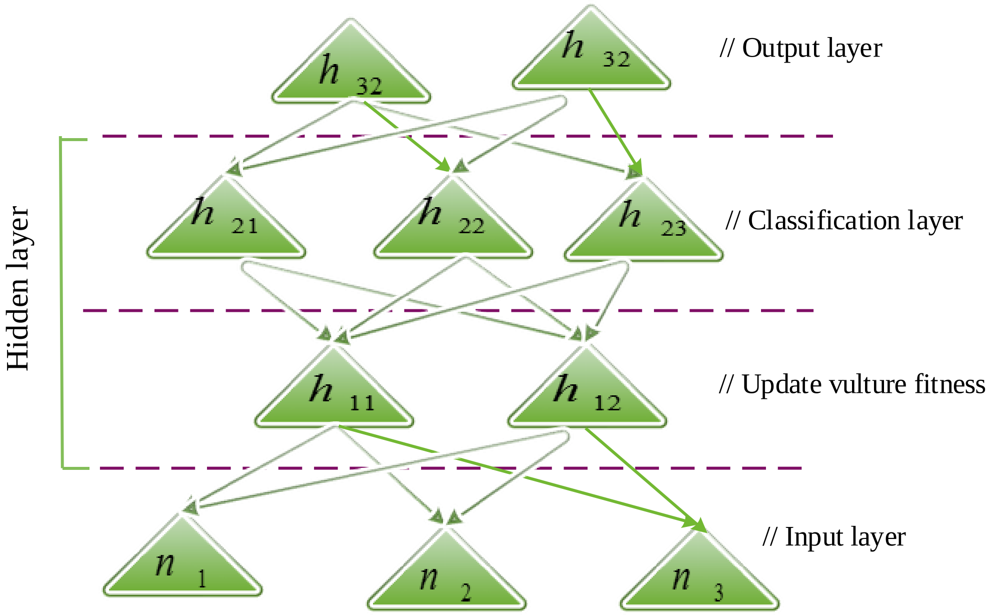

Fig.4 Primary internal process of the DBN model

Moreover, the developed model contains two varieties of nodes $p(h,n)$ such as visible and hidden hubs that are obtained using Eqn. (3)

$$p(o) = \sum_{n=1} p(h,n) = \sum_{o} \frac{1}{N} e^{-AE(h,n)} \tag{3}$$

Let, N is represented as the total neuron present in the developed model, AE is denoted as the energy activation function and the energy of every neuron is represented as e.

- **Feature extraction**

The process of feature extraction enhances the effectiveness and efficiency of detecting intrusions in the NSL-KDD dataset. Thus the feature extraction was performed with compressed matrix formatting. It extracts the relevant features from the dataset and ignores the zero values and traces non-zero values weights from the extracted features. As well, extract 42 features, where, 3 features are symbolic records and the other 39 features are numeric records. The process of feature extraction is obtained using Eqn. (4),

$$CM_f(A;B) = \sum_{e_x e_y \in (0,1)} V(A=i_x, B=i_y) \log_2 \frac{V(A=i_x, B=i_y)}{V(A=i_{xy})V(B=i_y)} \tag{4}$$

Let, V is represented as extracted features, A is called as the random variable of the real value $i_x = 1$ which contains n-gram features, the features do not contain A means $i_x = 0$. Moreover, B is represented as the random variable of the real value $i_y = 1$ and the features belong to a class, the features not in the class of B means $i_y = 0$.

- **Detection phase**

In this phase, update the fitness function of the vulture in the DBN classification layer for enhancing the detection performance by monitoring intrusion activity based on the 42 features. In this approach, the Vulture Optimization (VO) is initialized in the detection phase for intrusion prediction. The behavior of the VOA function is initiated in the classification layer that accurately predicts the intrusion using their fitness function. In VOA, the Egyptian vulture contains two main activities such as tossing pebbles and rolling with twigs which takes the better decision for hunting food. The steps of VOA are solution set initialization, tossing of pebbles, rolling with twigs, change of angle, fitness evaluation, and continue or stop decision. Moreover, tossing pebbles is helpful to break the hard eggs. It continuously tosses the pebbles and identifies the crack points or weak points for success. Additionally, rolling with twigs is the extra skill of the Egyptian vulture, which rolls the object to identify the weak points or movements. The Egyptian Vulture is thought to be rolling the solution set to modify the placements of the elements to change the meaning, which may lead to innovative approaches that could result in improved fitness values and multi-objective optimization paths. These are the two fitness functions of VOA fitness is used for continuous monitoring of user activity or intrusion activity by their two skills. The workflow of the designed model is shown in fig.4.

```
                    Algorithm: Design VBDBNS for predicting intrusion
Start
            Input: NSL-KDD dataset
            Output: predict intrusion
            {
            Initialization
            {
              Update the dataset                                      //input layer
              // NSL-KDD dataset
            }
            Pre-processing ()                                         //Hidden layer
            // normalize the data and get the appropriate form
            {
            Min-max normalization, $m(ia)$
            normalized up to the range of [0, 1]
            1-N encoding
            converting the categorical data variables
            }
               Feature extraction ()                                  //Hidden layer
               // extract the relevant features from the dataset
               {
               $CM_f(A;B) \to V(A,B)$                     $V$ - extracted features
               }
            Detection ()                                       //classification layer
            // identifies and searches the intrusion activity
            {
            The updated fitness function of VO
            // enhance the detection performance
            Threshold value
            // detect intrusion and normal user
                                      If($T(e) \le 0.5$)
                                      {
                                      Detect intrusion
                                      }
                                      End if
             Detect the intrusion
             Finest output solutions                                  // output layer
            }
End
```

Thus the developed technique searches and identifies the optimal solution and increases the appropriate number of random solutions for detecting intrusion. Moreover, intrusion detection is processed by building the relationship or features in the NSL-KDD dataset. Finally, detect the intrusion present in the collected dataset using Eqn. (5)

$$D_p(n)=V_u(t)V\frac{i_x}{i_y}\times CM_f\times m(ia)\frac{T(e)}{V} \quad (5)$$

Let, $V_u(t)$ is denoted as vulture fitness function and $T(e)$ is represented as a threshold value. Based on the threshold value of normal and intrusion, identify and detect the intrusion from the dataset. Finally, classify the intrusion present in the network based on the labels. Label 0 represents normal, label 1 represents DoS, label 2 represents probe, label 3 represents R2L, and label 4 represents U2R. Moreover, a developed framework to identify and detect the intrusion also enhances the performance of detection accuracy.

5. RESULTS AND DISCUSSIONS

The presented strategy was executed in the Python tool, and it is trained and validated using the NSL-KDD database. Accordingly, a novel VbDBN framework was proposed for enhancing the performance of IDS by accurately detecting intrusion. Additionally, pre-processing and feature extraction are employed to eliminate the flaws, and capture correlative attributes. Moreover, the fitness of the vulture is updated in the detection phase to attain the finest outcomes. It continuously monitors user behaviour and detects the intrusion present in the network. The performance evaluation displays that this methodology achieved better intrusion identification outcomes.

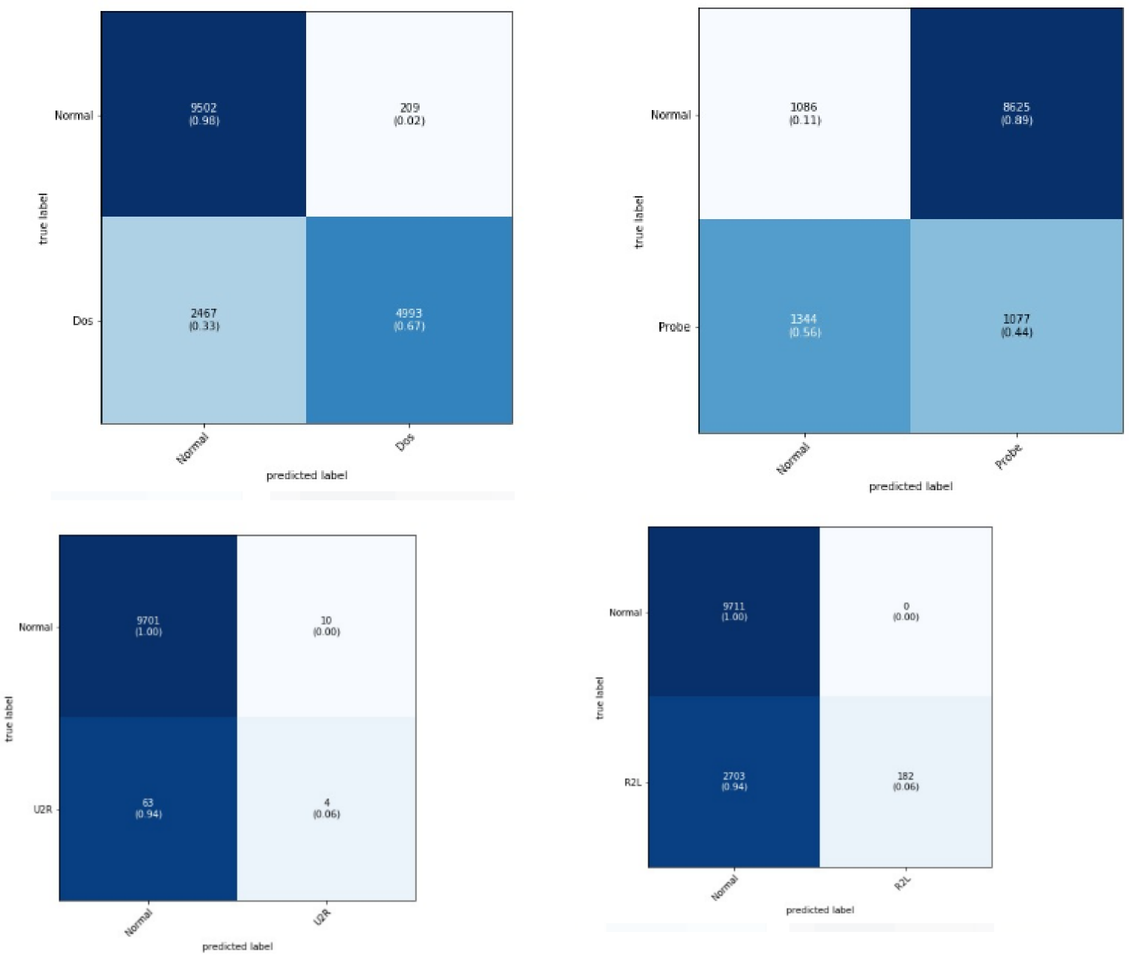

Fig.5 Confusion matrix

5.1 Performance metrics

In this section, we equate and validate the outcomes of the presented strategy with the conventional techniques such as SMO-DNN [21], IDS-DNN [23], DA-FO [24], and DL-HO [25]. The parameters utilized for evaluation include recall, accuracy, execution time, precision, False Positive Rate (FPR), and F1-score. These metrics are determined by implementing these models in the Python tool.

5.1.1 Accuracy

Accuracy metric quantifies the overall effectiveness of the presented VbDBN model in identifying intrusions in the

NSL-KDD database. It is determined by dividing the sum of true instances (positives and negatives) with the sum of all instances, and it is represented in Eqn. (6).

$$A = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \tag{6}$$

Let, T_p, T_n, F_p, and F_n denotes true positive, true negative, false positive, and false negative, respectively. Table 2 provides the evaluation of VbDBN's accuracy with current models.

Table.2 Validation of the accuracy

No. of. epochs	Accuracy (%)				
	SMO-DNN	IDS-DNN	DA-FO	DL-HO	Proposed
50	99.4	98.2	94	83.3	99.85
100	99	97.6	93.2	81	99.53
150	98.76	96	91	79.5	99.32
200	98.12	95.34	89.7	77	98.96
250	97	94	87	75.3	98.52

To validate that the designed approach gained higher accuracy, it is equated with the conventional models like SMO-DNN, IDS-DNN, DA-FO, and DL-HO. Fig 6 provides the accuracy analysis. When epoch count was 50, these models obtained accuracy rates of 99.4%, 98.2%, 94%, and 83.3%, respectively.

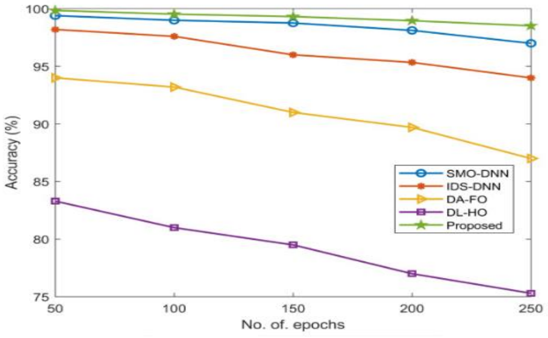

Fig.6 Comparative evaluation of accuracy

The presented methodology obtained enhanced accuracy of 99.85%. This assessment manifests that the developed algorithm outperformed the conventional models, highlighting its capacity to detect intrusion.

5.1.2 Recall

Recall indicates the system's efficiency to accurately identify and detect intrusions within a network. It quantifies the number of true positive instances accurately predicted by the VbDBN model, which is formulated in Eqn. (7).

$$S_N = \frac{T_p}{T_p + F_n} \quad (7)$$

Table.3 Validation of recall

Epoch count	Recall (%)				
	SMO-DNN	IDS-DNN	DA-FO	DL-HO	Proposed
50	99.5	98	93.6	83.3	99.94
100	98.9	96.4	91	81	99.66
150	98.1	94.56	89.6	79.4	99.15
200	96.5	92	87	76	99.01
250	95	90.03	85.4	74.21	98.64

The developed model's recall was evaluated and compared with the current models including SMO-DNN, IDS-DNN, DA-FO, and DL-HO. Fig 7 provides the recall analysis of different algorithms. The above models obtained recall rates of 99.5%, 98%, 93.6%, and 83.3%, respectively, while the designed VbDBN gained recall of 99.94%, which is high compared to those algorithms.

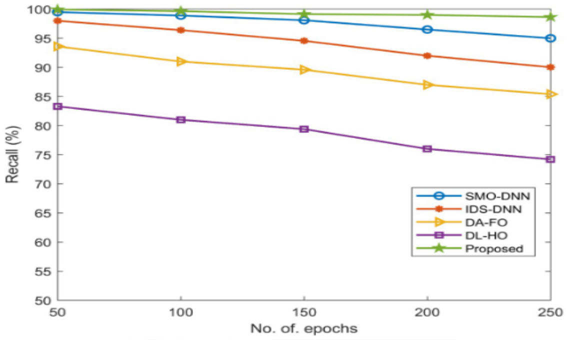

Fig.7 Comparison of recall

This intensive evaluation of recall metric for increasing epoch count (50 to 250) enables to determine model's scalability. The designed algorithm almost maintained consistent recall performance for increasing epochs, demonstrating its scalability and robustness in intrusion identification.

5.1.3 Precision

Precision defines the system's efficiency in detecting the intrusions in the NSL-KDD database. It quantifies the total correct intrusion predictions made by the VbDBN model to the total intrusions present in the database, and it is formulated in Eqn. (8).

$$P = \frac{T_p}{T_p + F_p} \tag{8}$$

Table.4 Precision assessment

Epoch count	Precision (%)				
	SMO-DNN	IDS-DNN	DA-FO	DL-HO	Proposed
50	99.5	98	92.90	86.02	99.76
100	99	96.7	90	84	99.42
150	98.4	94	88.76	82.39	99.12
200	97	92.24	86	80	98.98
250	96.3	90	84.53	78.6	98.75

To manifest that the designed strategy obtained higher precision, its precision outcome was equated with the currently existing techniques like SMO-DNN, IDS-DNN, DA-FO, and DL-HO. Fig 8 depicts the evaluation of model precision with conventional models. At 50 epochs, the above-stated models earned precision of 99.5%, 98%, 92.90%, and 86.02%, respectively.

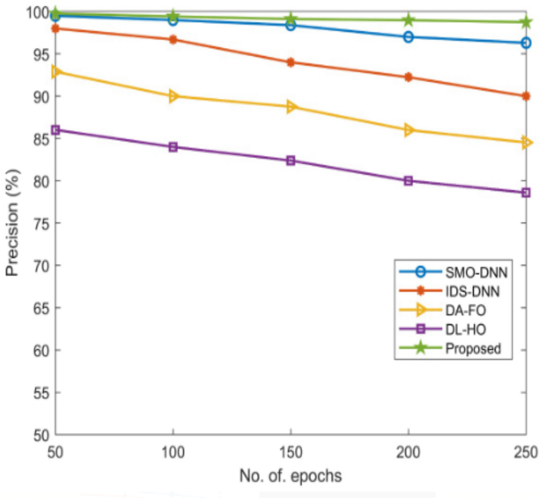

Fig.8 Precision validation

The designed methodology earned enhanced precision of 99.76%, which is greater than the current models. This validates that the VbDBN algorithm is highly efficient and reliable in detecting the attack cases.

5.1.5 F1-score

It is the degree of the test accuracy and is well-defined as the weight of harmonic mean of recall and precision. By combining the efficiency of both measurements, it enables to recognize the efficiency of detection intrusions which is measured using Eqn. (9),

$$F-measure=\frac{2*T_p}{\left(2*T_p+F_p+F_n\right)} \tag{9}$$

Table.5 F1-score evaluation

Epoch count	F1-score (%)				
	SMO-DNN	IDS-DNN	DA-FO	DL-HO	Proposed
50	99.6	98	93.24	82.04	99.88
100	99.1	97.5	91.4	80	99.64
150	98	96	89	78.6	99.38
200	96.5	94.5	87.45	77	98.95
250	95	92	86	75.23	98.77

Here, we determine and evaluate the F1-score performance of the developed algorithm with the current models including SMO-DNN, IDS-DNN, DA-FO, and DL-HO. Fig 9 presents the comparison of F1-score. These approaches earned F1-score of 99.6%, 98%, 93.24%, and 82.04%, respectively at 50 epoch count.

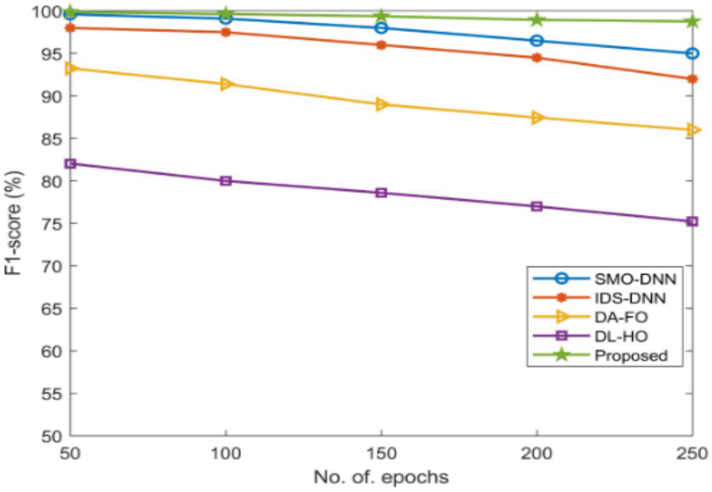

Fig.9 Comparison of F1-score

Consequently, we determined the F1-score of the proposed VbDBN, and it achieved a greater F1-score of 99.88%, which is higher compared to the current models. This depicts that the presented approach offers a balanced intrusion prediction performance considering both false-positives and negatives.

5.1.6 Execution time

Execution time defines the proportion of the time consumed to complete a task to the total time spent on data sharing. This denotes the total duration needed by the system for predicting intrusion, and it is determined using Eqn. (10). .

$$Execution\_time = \frac{Ct(t_1)}{T_r(t_1)} \times 100 \qquad (10)$$

Let, Ct is denoted as the single task completion time and T_r is represented as the total time essential to complete the task. Moreover, t_1 is represented as a task per second.

Table.6 Validation of execution time

No. of. epochs	Execution time (ms)				
	SMO-DNN	IDS-DNN	DA-FO	DL-HO	Proposed
50	13	33	19	29	7
100	16	36	22	34	10
150	21	39	26	37	12
200	24	42	29	40	14
250	26	46	32	43	15

The execution time incurred by the developed algorithm was estimated and equated with the current models such as SMO-DNN, IDS-DNN, DA-FO, and DL-HO. These techniques consumed higher execution time of 13ms, 33ms, 19ms, and 29ms, respectively for 50 epochs. Fig 10 provides the graphical visualization of execution time.

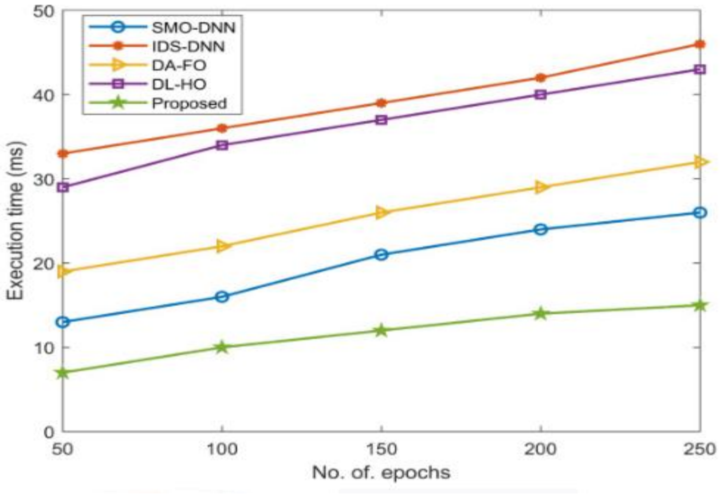

Fig.10 Comparison of execution time

On the other hand, the presented VBDBN technique consumed a minimum execution time of 7ms, which is less than the currently available approaches. This evaluation highlights that the designed algorithm obtained lower execution time, demonstrating its computational efficiency over other algorithms.

5.1.7 FPR

The FPR is termed a false alarm it is the negative statement of identifying intrusion present in the network. Moreover, IDS has the ability to identify both malicious (attack) and normal. Additionally, FPR is the measure of accuracy test and defined as the probability of incorrectly predicting the negative class, measured using Eqn. (11). Table 6 provides the statistical comparative evaluation of FPR.

$$FPR = \frac{F_n}{T_p + F_n} \tag{11}$$

Table.7 Comparison of FPR

Techniques	FPR (%)
SMO-DNN	1.51
IDS-DNN	2.03
DA-FO	1.98
DL-HO	4.48
Proposed	0.5

The implementation time of the designed VbDBN approach was determined and evaluated with currently existing techniques like SMO-DNN, IDS-DNN, DA-FO, and DL-HO. For 50 epochs, the SMO-DNN earned 1.51% FPR and the IDS-DNN obtained 2.03% FPR, DA-FO earned 1.98% FPR, and DL-HO gained 4.48% FPR. The comparative evaluation of FPR is depicted in Fig 11.

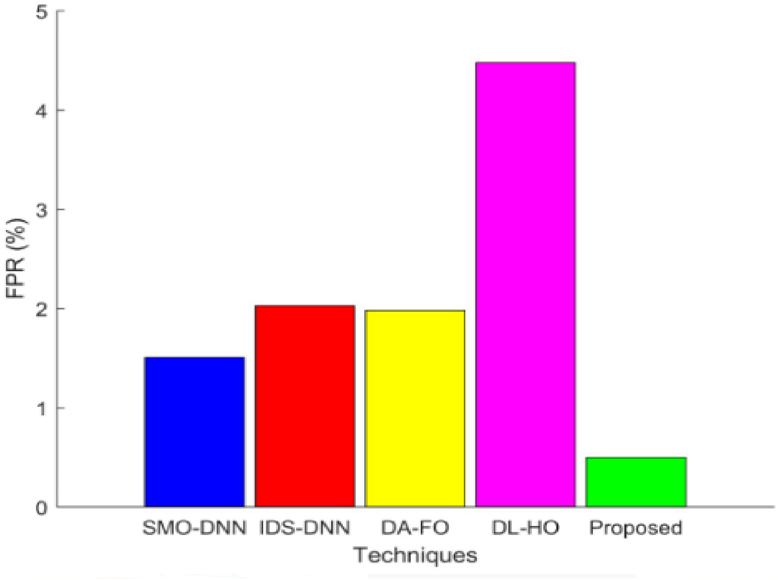

Fig.11 Comparison of FPR

This comprehensive evaluation of FPR manifests that the conventional algorithms obtained high FPR. But the designed VbDBNS algorithm earned a minimum FPR of 0.5%, which is less compared to the above-mentioned techniques.

5.2 Discussion

This study develops an innovative algorithm for identifying intrusions in the IoT system. Initially, we accumulated the NSL-KDD dataset from the public source. Then, the raw dataset was filtered to eliminate the missing values and standardizes the data representation. Further, feature engineering was done to capture the most informative 42 attributes from the database. These captured attributes are fed as input to the designed IDS model to predict the intrusions. The developed VbDBNS model has obtained greater intrusion identification performances by earning better outcomes in terms of accuracy, precision, F1-score, and recall. The experimental outcomes suggest that the designed VbDBNS algorithm obtained a minimum execution time of 7ms, greater detection accuracy of 99.85%, and improved precision of 99.76%. This validates that the proposed strategy achieved higher outcomes in predicting intrusions.

6. CONCLUSIONS

This research proposed a distinct VbDBNS for identifying intrusion in IoT systems. The developed VbDBNS aims to improve the detection performances by enhancing the results such as accuracy, precision and recall. The proposed system was validated using the NSL-KDD dataset, which is fed as input to the Python system for training. This designed VbDBNS algorithm identifies the intrusion based on the pre-defined threshold value and it uses the extracted feature sequence to precisely identify the intrusion. The simulation results manifest that this framework has obtained improved outcomes for the NSL-KDD database. From the intensive evaluation, it is observed that this algorithm earned 99.85% accuracy, 99.94% recall, and 99.76% precision in predicting intrusions. Furthermore, the comparative analysis with the conventional models depict that the accuracy performance was enhanced by 2% in the developed algorithm, illustrating its efficiency and robustness in identifying intrusions.

Compliance with Ethical Standards

Conflict of interest

The authors declare that they have no conflict of interest.

Human and Animal Rights

This article does not contain any studies with human or animal subjects performed by any of the authors.

Informed Consent

Informed consent does not apply as this was a retrospective review with no identifying patient information.

Funding: Not applicable

Conflicts of interest Statement: Not applicable

Consent to participate: Not applicable

Consent for publication: Not applicable

Availability of data and material:

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Code availability: Not applicable

Competing Interests : Not applicable

REFERENCES

[1] Li, Daming, et al. "IoT data feature extraction and intrusion detection system for smart cities based on deep migration learning," *International journal of information management*, vol. 49, pp. 533-545,2019

[2] Xiong, Wenjun, et al. "Cyber security threat modeling based on the MITRE Enterprise ATT&CK Matrix," *Software and Systems Modeling*, vol. 21.1,pp.157-177,2022

[3] Sicari, Sabrina, Alessandra Rizzardi, and Alberto Coen-Porisini. "5G In the internet of things era: An overview on security and privacy challenges," *Computer Networks*, vol.179, pp. 107345,2020

[4] Yu, Q., Ren, J., Zhang, J., Liu, S., Fu, Y., Li, Y & Zhang, W. (2020). “An immunology-inspired network security architecture,” *IEEE Wireless Communications*, vol.27, no.5, pp. 168-173.

[5] Garagad, G.Vishwanath , Nalini C. Iyer, and Heera G. Wali. "Data integrity: a security threat for internet of things and cyber-physical systems," 2020 International Conference on Computational Performance Evaluation (ComPE). *IEEE*, 2020.

[6] Kumar, Parasuraman, et al. "Analysis of intrusion detection in cyber attacks using DEEP learning neural networks," *Peer-to-Peer Networking and Applications*, vol. 14, no.4, pp. 2565-2584,2021

[7] Kimani, Kenneth, Vitalice Oduol, and Kibet Langat. "Cyber security challenges for IoT-based smart grid networks," *International Journal of Critical Infrastructure Protection*, vol. 25, pp. 36-49,2019

[8] Wang, Yuanbin, et al. "IoT-enabled cloud-based additive manufacturing platform to support rapid product development," *International Journal of Production Research*, vol. 57, no.12, pp. 3975-3991,2019

[9] Attaran, Mohsen, and Jeremy Woods. "Cloud computing technology: improving small business performance using the Internet," *Journal of Small Business & Entrepreneurship*,vol. 31, no.6, pp. 495-519,2019

[10] Eskandari, Mojtaba, et al. "Passban IDS: An intelligent anomaly-based intrusion detection system for IoT edge devices," *IEEE Internet of Things Journal,* vol. 7, no.8, pp. 6882-6897,2019

[11] Yaacoub, Jean-Paul, et al. "Security analysis of drones systems: Attacks, limitations, and recommendations," *Internet of Things,* vol. 11, pp. 100218,2020.

[12] Khan, Muhammad Ashfaq, Md Karim, and Yangwoo Kim. "A scalable and hybrid intrusion detection system based on the convolutional-LSTM network," *Symmetry*, vol.11, no.4, pp. 583,2019.

[13] Sadikin, Fal, Ton Van Deursen, and Sandeep Kumar. "A ZigBee intrusion detection system for IoT using secure and efficient data collection," *Internet of Things,* vol. 12, pp.100306,2020.

[14] Anthi, Eirini, et al. "A supervised intrusion detection system for smart home IoT devices," *IEEE Internet of Things Journal*, vol.6, no.5, pp. 9042-9053,2019.

[15] Dash, Rupa, et al. "Application of artificial intelligence in automation of supply chain management," *Journal of Strategic Innovation and Sustainability* , vol.14, no.3, pp.43-53,2019

[16] Casal-Guisande, Manuel, et al. "A decision-making methodology based on expert systems applied to machining tools condition monitoring," *Mathematics,* vol. 10.3, pp. 520,2019

[17] Alsarhan, Ayoub, et al. "Machine Learning-driven optimization for SVM-based intrusion detection system in vehicular ad hoc networks," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-10,2021

[18] Mahfouz, Ahmed, et al. "Ensemble classifiers for network intrusion detection using a novel network attack dataset," *Future Internet,* vol. 12.11, pp. 180,2020

[19] Ahmad, Zeeshan, et al. "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no.1, pp. e4150,2021

[20] Camacho, José, et al. "Multivariate Big Data Analysis for intrusion detection: 5 steps from the haystack to the needle," *Computers & Security*, vol. 87, pp. 101603,2019

[21] Khare, Neelu, et al. "SMO-DNN: spider monkey optimization and deep neural network hybrid classifier model for intrusion detection," *Electronics*, vol. 9, no.4, pp. 692,2020.

[22] Ali, Mohammed Hasan, et al. "A new intrusion detection system based on fast learning network and particle swarm optimization," IEEE Access, vol. 6, pp. 20255-20261,2018

[23] Ramaiah, Mangayarkarasi, et al. "An intrusion detection system using optimized deep neural network architecture," Transactions on Emerging Telecommunications Technologies, vol. 32, no.4, pp. e4221,2021

[24] Mohan, A., Prabha, G. and V., A. 2023. Multi Sensor System and Automatic Shutters for Bridge- An Approach. International Journal of Intelligent Systems and Applications in Engineering. 11, 4s (Feb. 2023), 278–281.

[25] Prabha , G. , Mohan, A. , Kumar, R.D. and Velrajkumar, G. 2023. Computational Analogies of Polyvinyl Alcohol Fibres Processed Intellgent Systems with Ferrocement Slabs. International Journal of Intelligent Systems and Applications in Engineering. 11, 4s (Feb. 2023), 313–321.

[26] Study On Structural Behaviour Of Ductile High-Performance Concrete Under Impact And Penetration Loads, Lavanayaprabha, S. Mohan, A. Velrajkumar, G., Mohammedharoonzubair, A. Journal of Environmental Protection and Ecology., 2022, 23(6), pp. 2380–2388.`

[27] Mohan, A., & K, S. . (2023). Computational Technologies in Geopolymer Concrete by Partial Replacement of C&D Waste. International Journal of Intelligent Systems and Applications in Engineering, 11(4s), 282–292.

[28] Mohan, A., Dinesh Kumar, R. and J., S. 2023. Simulation for Modified Bitumen Incorporated with Crumb Rubber Waste for Flexible Pavement. International Journal of Intelligent Systems and Applications in Engineering. 11, 4s (Feb. 2023), 56–60.

[29] R.Gopalakrishnan, Mohan, "Characterisation on Toughness Property of Self-Compacting Fibre Reinforced Concrete", Journal of Environmental Protection and Ecology 21, No 6, 2153–2163 (2020)

[30] Sekhar, R., et al. "A novel GPU based intrusion detection system using deep autoencoder with Fruitfly optimization," SN Applied Sciences, vol. 3, no.6, pp. 1-16,2021

[31] Kunang, Yesi Novaria, et al. "Attack classification of an intrusion detection system using deep learning and hyperparameter optimization," *Journal of Information Security and Applications,* vol. 58, pp.102804,2021

[32] Abdollahzadeh, Benyamin, Farhad Soleimanian Gharehchopogh, and Seyedali Mirjalili. "African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems," *Computers & Industrial Engineering*, vol. 158, pp. 107408,2021

[33] Papamartzivanos, Dimitrios, Félix Gómez Mármol, and Georgios Kambourakis. "Introducing deep learning self-adaptive misuse network intrusion detection systems," *IEEE Access*, vol. 7, pp. 13546-13560,2019

[34] Khan, Izhar Ahmed, et al. "Enhancing IIoT networks protection: A robust security model for attack detection in Internet Industrial Control Systems," *Ad Hoc Networks*, vol. 134, pp. 102930,2022

[35] Muhuri, Pramita Sree, et al. "Using a long short-term memory recurrent neural network (LSTM-RNN) to classify network attacks," *Information*, vol. 11, no.5, pp. 243,2019

[36] Azizjon, Meliboev, Alikhanov Jumabek, and Wooseong Kim. "1D CNN based network intrusion detection with normalization on imbalanced data," 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC). *IEEE*, 2020.

[37] Zhang, Jianwu, et al. "Model of the intrusion detection system based on the integration of spatial-temporal features." *Computers & Security*, vol. 89, pp. 101681,2021

[38] Mahdavisharif, Mahzad, Shahram Jamali, and Reza Fotohi. "Big data-aware intrusion detection system in communication networks: a deep learning approach." *Journal of Grid Computing*, vol. 19, no.4, pp. 1-28,2019