Journal of Intelligent Systems and Internet of Things
Vol. 16, No. 01, PP. 199-210, 2025
DOI: https://doi.org/10.54216/JISIoT.160117

Cat-Feed-Nets: A Novel Cat Evoked Deep Feedforward Networks for Detection of Dos Attacks in IoT-Cloud Environment

P. Jagdish Kumar [1,*], S. Neduncheliyan[2]

[1]Research Scholar, Department of Computer Science and Engineering, Bharath Institute of Higher Education and Research, Chennai 600073 Tamil Nadu, India

[2]Dean of School Computing, Bharath Institute of Higher Education and Research

Chennai 600073 Tamil Nadu, India

Emails: pjkcse123@gmail.com; dean.cse@bharathuniv.ac.in

Abstract

Internet of things (IoT) is an intelligent combination of embedded systems, cloud computing and wireless communications. However, the data privacy and leakage problems are considered as the major deadlocks for deploying the IoT devices in the real time fields. Nevertheless, the complication of Distributed Denial of Service (DDoS) hazard on the IoT devices recent surge has seen an uptick, making it prone to numerous threat complications. For this reason, prompt detection of these attacks plays a pivotal role to safeguard the user's data. The AI methodology of Machine and Deep Learning Models engaged for the designing the intelligent systems to provide the secured environment to safeguard the network against the various attacks. However, the computational overhead of deep learning model handicaps to deploy it in the IoT-Cloud environment. To tackle this issue, the present article suggests the novel hybrid learning based detection system called CAT-FEED-NETS that incorporates the Deep feed forward neural networks (DFFNN) where the hyper parameters are tuned by the Cat Swarm Intelligence Algorithms. Comprehensive trials and analysis are performed using NSL-KDD and UNSW datasets and criteria to assess the efficacy of quality measurements such as accuracy, precision recall, F1-score and model building time (MBT) is evaluated and analysed. Evaluation results are weighted against the various DL algorithms with the suggested model exhibiting better results than the other models by producing 0.96 accuracy, 0.956 precision, 0.955 recall and 0.9834s of MBT respectively. The proposed framework had proved its superiority in predicting the cloud attacks than the other existing frameworks.

Received: November 25, 2024 Revised: January 27, 2025 Accepted: February 24, 2025

Keywords: Internet of things (IoT); Machine Learning; Deep learning Algorithms; CAT-Swarm Optimization

1. Introduction

Recently, Internet of Things (IoT) has bright light of deployment in various fields such as health care, manufacturing, and automation. Due to its capability and applications, this technology is providing more comfort zones in every individual's life [1].

IoT is formed by combining the many embedded devices that are connected to the internet and can be managed remotely from every place [2]. These objects have ability to communicate with each other and used to collect the surrounding data and process the data into useful information. However, these networks are subjected to many vulnerabilities that may cause the leakage of user information and even threat to many user's lives [3-5].

Many attacks such as Denial of Services (DoS), data leakage, spoofing are the most common threats causes the blackouts in the networks that prevents the global time implementation of IoT in the real time scenario [6-8]. Conventional methods such as Intrusion detection systems (IDS), Signature based analogy detection (SBAD) and other intelligent mechanisms are used to detect the attacks in an IoT environment [9-11].

A DoS refers to the denial of service, where a server is inundated with malevolent communication. When numerous computers or compromised systems, known as bots, orchestrate DoS assaults on a single application, it leads to a

DDoS attack. Subsequently, the targeted network is flooded with packets originating from various locations globally. The landscape of DDoS attacks is evolving and broadening in terms of both volume and intricacy, owing to the proliferation of disruptive Internet technologies [12].

Mitigating DDoS attacks promptly upon detection is imperative due to the proliferation of internet usage, making internet-connected devices prime targets for cyber-attacks. Both industry and academia are exploring the integration of Dual Models namely Machine and Deep learning for DDoS identification, owing to their significant potential across various domains. While in ML, experts must select the features for classification, deep learning models undertake feature selection internally. DL methods like Artificial Neural Networks (ANN), Deep Neural Networks (DNN), and Recurrent Neural Networks (RNN) adeptly discern multiple patterns from extensive labelled datasets through layers of nonlinear transformations, thereby proving to be effective for DDoS detection. However, the DL provides the high prediction performance but suffers from the computational complexity, in which it fails to encounter the DDoS attacks with the stipulated time.

Motivated by this drawback, this paper proposes the secured framework based on deep neural network whose hyper-parameters are tuned by the cat optimization algorithm [13]. The idea of integrating the Cat optimization algorithm that increases the searching space that in turn increases the prediction performance with the less computational complexity. This paper puts forward novel ideas in the areas of:

1. The paper introduces the cat optimizer for tuning the hyper-parameters of gated units to achieve the highest performance with the low computational overhead.

2. The novel deep learning newly introduced feedforward networks, operating based on the fundamental concept of extreme learning machines, have been suggested.

3. Experimental assessment of the preliminary framework is carried out by implementing the NSK-KDD and UNSW -19 datasets and the effectiveness of the proposed prototype is evaluated and in contrast to Deep learning methods using key evaluation metrics. Results demonstrates the suggested framework excelled the various methodologies.

The subsequent sections of this document are ordered as pursues: Section-2 illustrates the related implementations from the different authors. Section 3 provides full details on the datasets utilized and the proposed framework put forward. Section 4 showcases experimental validation results, presents relevant performance metrics, and includes comparative analyses against existing methods. Section 5 culminates the paper, offering conclusions and exploring prospects for further improvement and expansion of this research.

2. Related Work

J. Li, Y. Li, et.al [14] (2023) integrated the two layered Convolutional Neural Networks (CNN) along with light weight cryptography techniques to provide the defensive characteristics in the IoT-Cloud-Edge Collaborative environment. Authors designed the intelligent systems to safeguard the cloud data against the Man-In-Middle attacks (MIM). The promising results are showcased with the accuracy of 96% and proves the performance of the proposed model can fit for an IoT-Cloud-Edge Environments.

Syed Mohamed et.al [15] (2023) introduced constructing an astute intrusion prevention framework designed to identify both network and application-driven assaults. Authors adopts Inter-Grading Normalization technique for pre-processing, Opposition-based learning Rat inspired Optimizer for feature selection and finally 2D- Array based CNN as the binary classifiers. The hybrid framework detects normal and abnormal traffics in the IoT environment and examined using datasets reliant on NetFlow. The suggested structure provides accuracy of 95.20% with the less false positive rate of 2.5%.

Susilo et al. (2023) [16] presented the smart intrusion prevention system using deep learning for IoT with the scalable IoT devices. This research study used Convolutional Neural Networks (CNN) and compared the results along with various machine-learning algorithms. The intensive analysis was performed by utilizing IoT-BoT datasets in which the CNN has produced the best accuracy of 91.27% with the training time of 27secs. Still the performance needs to be increased with reduced false alarm rate.

In [17], the authors presented the idea of introducing the distributed CNN for the IoT devices and LSTM for back-end clouds. This research article employs the IoT-BoT datasets and validates the capability of the model in detecting the malicious attacks such as botNets, Phishing and DoS in distributed IoT devices. The model produced the accuracy of 97.84% with the false alarm rate of 0.0001%. However, the proposed solution in this research fails to detect the emerging attacks, which still leaves IoT devices as vulnerable to many attacks.

Janardhna et.al [18] (2022) presented the quality of the various AI models using residing datasets for multiclass classification for an IoT-Cloud Environment. Convolutional Neural Networks (CNN), recurrent neural Networks (RNN), Naïve Bayes (NB), Decision Trees (DT) are used for the multi-class implementation. It is found hyper-parameter optimized RNN produced the highest accuracy 96% in predicting and detecting the security and privacy attack.

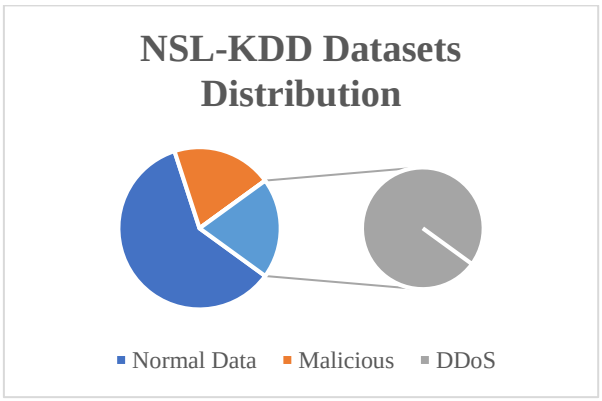
3. Proposed Model

This study introduces an innovative hybrid deep feedforward network enhanced through the utilization of the CAT algorithm, aimed at achieving improved predictive capabilities for predicting.

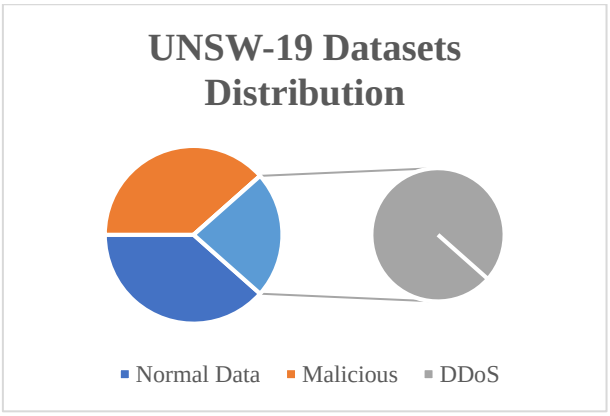
3.1 Materials and Methods

In this research, NSL-KDD and UNSW-19 datasets are employed for training the network. This research involves the utilization of two distinct datasets, i. eUNSW-NB15, and NSLKDD [19] are used for conducting the experiments The UNSW-NB15 dataset contains 49 features and 1 class label. A portion of the dataset is utilized for training and testing subsets - UNSW_NB15_Train and UNSW_NB15_Test. The training subset comprises 175,341 total instances, while the testing subset comprises 82,332 occurrences. Within the training subset, there exist 56,000 normal traffic instances and 119,341 attack traffic scenarios. In comparable ways, within the testing set, there are 37,000 standard traffic volume and 45,332 attack traffic examples. Holdout validation is performed by leveraging the full training and testing subsets as partitions. Cross-validation techniques only employ splitting of the training subset into folds.

Additionally, the NSL-KDD dataset containing 41 features and 1 class label is utilized to validate classifier performance. The KDD Train+ training subset and KDD Test+ testing subset from the NSL-KDD dataset are leveraged in this study. The KDD Train+ subset comprises 25,192 total instances, with 13,499 attack traffic occurrences and 11,743 normal traffic instances. The KDD Test+ subset is made up of 22,544 total instances, including 9,711 attack traffic examples and 12,833 normal traffic cases. Using these defined subset divisions allows avoidance of random sampling across the entire NSL-KDD dataset.



(a)



(b)

Figure 1. Datasets Distribution

3.2 Data Pre-processing

To start, the inputs are refined and normalized to transform attribute values into a finite numeric range. Min-max normalization is the technique employed, applying a linear transformation. This pre-processing produces new normalized dataset versions from the original raw data. After this initial pre-processing, the normalized data is passed into the feature extraction module.

3.3.1 Extreme Learning Machines – An Overview

In the third stage, extracted feature maps are used for training the deep feedforward learning model to classify the cardiac disorders. The suggested framework uses the principle of extreme learning machines suggested by G.B.Huang for the high speed and high accurate classification of different grades. This particular type of neural network employs a solitary hidden layer, wherein the hidden layer does not necessitate mandatory adjustment.

ELM leverages the kernel function to achieve high accuracy and optimize performance effectively. Its primary strengths lie in its ability to minimize training errors and enhance approximation quality. Through the utilization of auto-tuning for weight biases and activation functions that are non-zero, ELM is particularly suited for tasks such as classification and determining classification values. Further elaboration on the operational principles of ELM can be found in reference [19].

In this system, the 'L' neurons within the hidden layer must function using an activation function that is highly variable, such as the sigmoid function, while the output layer's activation function remains linear. In Extreme Learning Machines (ELM), there is no requirement for tuning the hidden layer. While hidden nodes are not deemed irrelevant, they do not require tuning, and their parameters may be pre-generated through a random process in advance.

Attending to the training set data, the output of the structure for an ELM is determined by equation (1)

$$f_L(x) = \sum_{i=1}^{L} \beta_i h_i(x) = h(x)\beta \quad (1)$$

Where x → input features from encoder-decoder

β → Provide the weight vector output, presented as follows

$$\beta = [\beta_1, \beta_2, \dots\dots\dots\dots\beta_L]^T \quad (2)$$

H(x)→ output hidden layer as delineated by the subsequent formula, is denoted as:

$$h(x) = [h_1(x), h_2(x), \dots\dots\dots\dots h_L(x) \quad (3)$$

To ascertain the target vector, denoted as Output vector O, the concealed strata are depicted by equation (4)

$$H = \begin{bmatrix} h(x_1) \\ h(x_2) \\ \vdots \\ h(x_N) \end{bmatrix} \quad (4)$$

The fundamental application of ELM employs the minimal nonlinear least squares techniques outlined in equation (5)

$$\beta' = H^* O = H^T (HH^T)^{-1} O \quad (5)$$

Where H∗ → inverse of H known as Moore−Penrose generalized inverse.

The equation can alternatively be expressed as follows:

$$\beta' = H^T (\frac{1}{C} HH^T)^{-1} O \quad (6)$$

Therefore, the determination of the outcome can be achieved by employing the aforementioned equation:

$$f_L(x) = h(x)\beta = h(x)\, H^T (\frac{1}{C} HH^T)^{-1} O \quad (7)$$

The identification of alien entities is determined through the utilization of mathematical formula (7), wherein thresholds are employed to ensure proficient categorization.

3.3.1 Improvisation in ELM

The Extreme Learning Machine (ELM) faces a significant limitation when dealing with extensive datasets, resulting in heightened computational burdens and diminished classification efficacy. Despite its recognized

efficiency during training and testing phases, a notable drawback lies in its suboptimal calibration of input weights and biases. Additionally, to fine-tune these weights optimally, ELM employs multiple hidden layers, deviating from conventional learning algorithms, potentially compromising detection accuracy.

To address the aforementioned limitation, a novel bio-inspired CAT technique is employed for fine-tuning the input weights and bias factors, leading to enhanced classification accuracy. The key benefits of employing BAT algorithms are delineated as pursues:

1. Superior efficacy contrasted to PSO, GA, and alternative heuristic methods.

2. Rapid and adaptable exploration within the search domain.

The operational principles of the CAT algorithm are elaborated upon in the preceding section.

3.4 CAT Swarm Optimization Algorithm

The fundamental principle behind the Cat Swarm Optimization (CSO) algorithm draws inspiration from the leisurely habits of cats, blending their tendencies to rest and to keenly observe their surroundings. Cats, known for their relaxed demeanor, possess an acute awareness even during periods of repose. When they detect a target of interest, they swiftly transition into action. This characteristic behavior serves as the basis for the CSO algorithm, which integrates these two primary modes of cat behavior. In the CSO algorithm, each cat embodies answer set characterized by its position, fitness value, and flag. M dimensions within the search space define the place. Fitness value reflects the efficacy of the solution set, while the flag categorizes cats into either seeking or tracing mode. Consequently, determining the number of cats participating in each iteration becomes crucial before initiating the algorithm. Throughout the iteration process, the best-performing cat is retained in memory, culminating in the selection of the final solution at the algorithm's conclusion. The workflow of the CAT Swarm Optimization algorithm, depicted in Figure 2, elucidates its operational framework. Further elaboration on the mechanics of seeking and tracing modes is provided in subsequent sections.

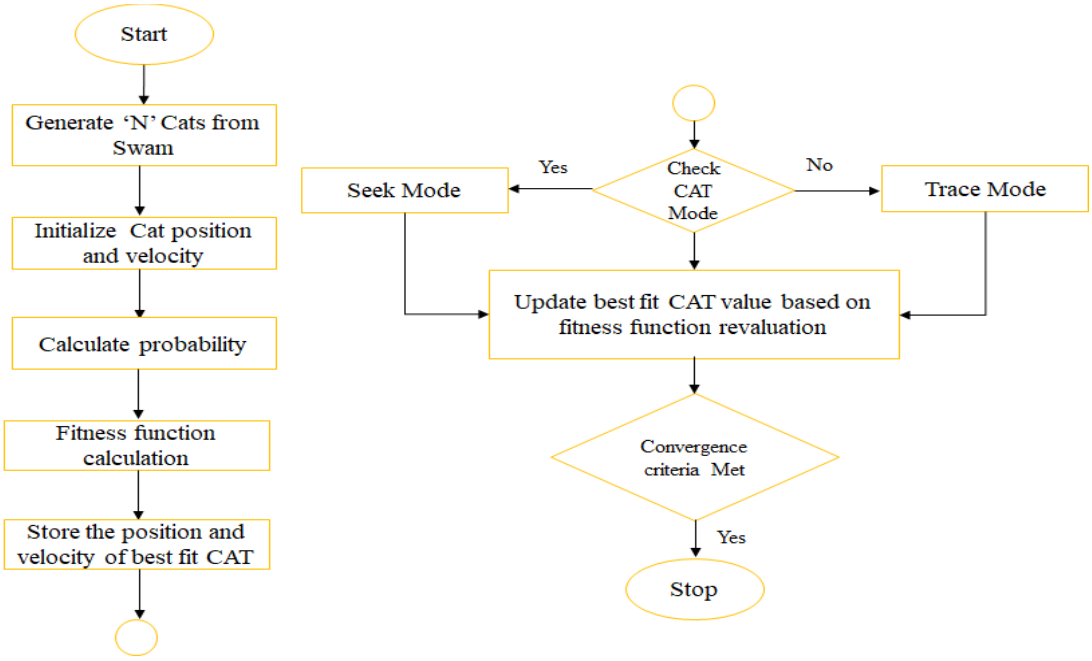


Figure 2. Comprehensive schematic illustration for the cat swarm Methodology

3.3.2.1 Seeking Modes

In the pursuit of mimicking the idleness of cats, four pivotal parameters are at play: the seeking memory pool (SMP), seeking range of the selected dimension (SRD), counts of dimension to change (CDC), and self-position considering (SPC). These parameters, govern the cat's behaviour. SMP delineates the magnitude of the seeking memory, dictating the number of potential positions a cat may explore. For instance, if SMP is 5, every cat would generate 5 new random positions, from which one would be chosen for further exploration. CDC determines the extent of dimension modification, ranging between 0 and 1. If, for example, CDC is set to 0.2 within a 5-dimensional space, four dimensions would undergo random modification while one remains constant. SRD serves as the mutative factor for selected dimensions, influencing the degree of mutation within the chosen parameters. SPC, a binary flag, determines the present whereabouts of a cat is eligible for consideration in the subsequent iteration. If set to true, one less candidate position is generated for each cat, as the current position is retained.

The seeking mode unfolds in the following steps:

1. Duplicate the current position of each cat, creating SMP copies.

2. Select CDC dimensions approximately for mutation in each duplicate, adjusting their values by adding or subtracting SRD.

$$x(new\_cat) = (1 + rand + SRD) * x(old\_cat) \quad (8)$$

Determine the Fitness function (FF) by calculating the new_cat's position (x(new_cat)) in relation to the old_cat's initial place (x(old_cat)) along with a random time interval (rand) between 0 and 1. Then, select the candidate position depends on the peak fitness function probability, as demonstrated as pursues (9)

$$P(i) = |(FF(i) - FF(b)|/(FF_{max} - FF_{min}) \quad (9)$$

In the given scenario, FF(i) represents the current cat's fitness, while FF(b) denotes the population count of cats. FFmax signifies the peak of the Fitness Function, whereas FFmin stands for the minimum value within the Fitness Function.

3.3.2.2 Tracing Modes

In this operational modality, cats mimic the tracing conduct. Initially, arbitrary velocity parameters are assigned to each dimension of a cat's location. Nonetheless, subsequent adjustments are imperative for updating velocity parameters. The locomotion of cats in this setting unfolds as such:

$$V(CAT) = V(CAT) + r * c\ (x(new\_cat) - x(old\_cat)) \quad (10)$$

3.3.3 Hyper parameter Optimized Using Cat Algorithm

In the suggested framework, the optimization of hyperparameters within fully connected layers is conducted using CSO. Given the pivotal role of these hyperparameters in influencing the performance of the network, their selection becomes imperative and contingent upon the specific application for which the CNN is deployed. The typical hyperparameter settings in ELM bearing substantial significance as delineated in 1st Tabular.

Table 1: Hyper parameter and its Functionalities

Sl.no	Hyper parameters	Functionalities
01	Learning Rate	An element employed to regulate the pace of training a network.
02	Epochs	Expresses the frequency at which the learning model adjusts its network based on the datasets.
03	Hidden Layers and Input weights	Establishes the operational sequence of the model.

In order to enhance result accuracy, it is imperative to optimize these hyperparameters. Algorithm-1 outlines the suggested approach inspired by CAT for hyperparameter optimization in ELM. CAT populations are chosen approximately, factoring in the quantity of inputs. The fitness function is revamped by adjusting the equation to reflect these modifications.

$$Fitness\ Function\ A = \{1 - mAMaximum(Accuracy)\} \quad (14)$$

The tabulated results in Table 3 delineate the fine-tuned hyperparameters achieved following the execution of the recommended optimization algorithm.

Sl.no	Algorithm-1: Full Procedure of CAT based hyper parameter optimization
01	Input: Bias Weights(β), Hidden layers count(η), Epochs count (μ), Learning Rate (α)
02	Commence Cat Swarm Population N and velocity of CATS be V
03	While n= 0 to N-1 where N is maximum iteration
04	Determine the likelihood and explore representatives utilizing Formula (8) and (9).

05	Determine the hyper parameters (β,η,μ,α)
06	Determine the fitness function utilizing Equation (14)
07	If (Fitness Function == Threshold attained)
08	Upgrade the latest Cats and keep the best ones saved.
09	else
10	Upgrade the Cat's parameters and Go to Step 04
11	End
12	End

Table 2: Optimized Hyper parameters used for training the network in ELM

Sl.no	Hyper parameters	Optimized Hyper parameters
01	Batch Size	05
02	No of epochs	100
03	Learning rate	0.001
04	Number of Optimized Iterations	20
05	No of search agents	15

4. Results and Discussion

4.1 Implementation Details

The proposed model is implemented using scikit-learn end-to-end python open-source platform. The multi-classification model was trained for 100 iterations. The CAT optimizer was used to optimize the loss functions in the networks that yields minimum loss during the iteration. The model was trained on i7CPU, 16GB RAM, NVIDIA K80 GPU with 2.5 GHZ Operating Frequency.

4.2 Performance Metrics

In this section, we have shown the superiority of the proposed model over the other deep learning models. To evaluate the performance of proposed architecture, metrics such as accuracy, sensitivity, specificity, recall and f1-score are calculated. Table 4 shows the mathematical expressions for calculating the metrics used for evaluating the proposed architecture

Table 3: Mathematical Expressions for the Performance Metrics' Calculation

SI.NO	Performance Metrics	Mathematical Expression
01	Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
02	Sensitivity or recall	$\frac{\text{TP}}{\text{TP+FN}}$ x100
03	Specificity	$\frac{TN}{TN+FP}$
04	Precision	$\frac{TN}{TP+FP}$
05	F1-Score	$2.\frac{Precison * Recall}{Precision + Recall}$

TP is True Positive Values, TN is True Negative Values, FP is False Positive and FN is false negative values.

4.3 Results and Discussion

Figure 1 illustrates the confusion matrix of the suggested structure for identifying DoS and Normal assaults. It is evident from the diagram that the proposed model achieved a 95% detection rate for DDoS attacks. To the addition, the performance metrics of the suggested structure have been computed and juxtaposed with those of various models, as depicted in the accompanying table.

Table 4: Quality metrics of the various methodology in identifying the normal scenario with the 70:30 ratio of training and testing data (NSL-KDD Datasets)

Algorithm	**Performance Metrics**				
	Accuracy	**Precision**	**Recall**	**Specificity**	**F1-Score**
SVM	78	77.6	77.6	0.23	77.6
RF	78.4	76.5	76.4	0.24	76.35
DT	75.5	74.3	74	0.26	74.4
ANN	68.5	67.5	67.3	0.334	67.4
ELM	87.4	86.5	86.4	0.114	86.4
Proposed Model	**98.4**	**97.6**	**97.3**	**0.001**	**97.5**

Table 5: Quality Measures of the different algorithms in identifying the DDoS hazard (NSL-KDD Datasets)

Algorithm	**Performance Metrics**				
	Accuracy	**Precision**	**Recall**	**Specificity**	**F1-Score**
SVM	78	77.6	77.6	0.23	77.6
RF	78.4	76.5	76.4	0.24	76.35
DT	75.5	74.3	74	0.26	74.4
ANN	68.5	67.5	67.3	0.334	67.4
ELM	87.4	86.5	86.4	0.114	86.4
Proposed Model	**98.4**	**97.6**	**97.3**	**0.001**	**97.5**

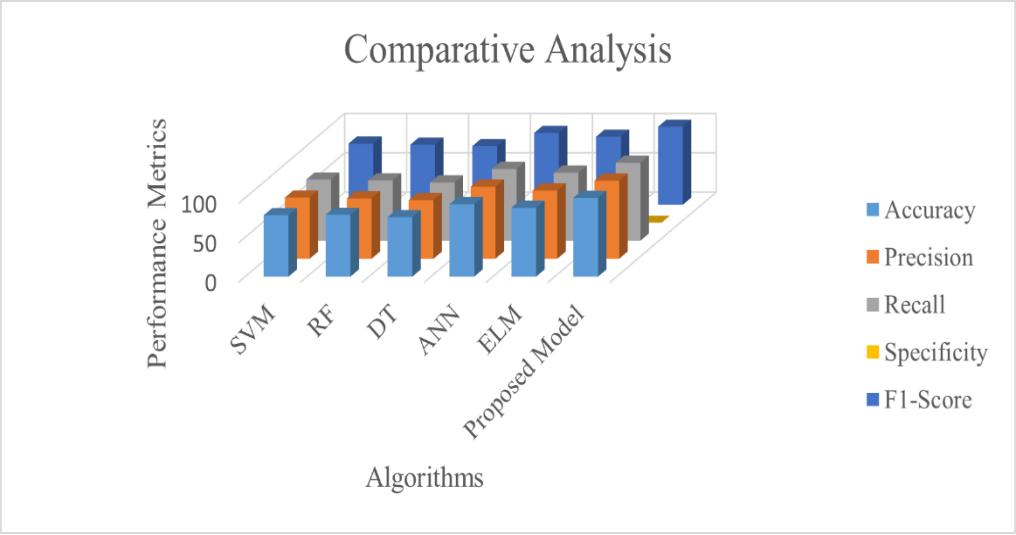


Figure 3. Correlative Study of various learning methodology in Normal Scenario (NSL-KDD Datasets)

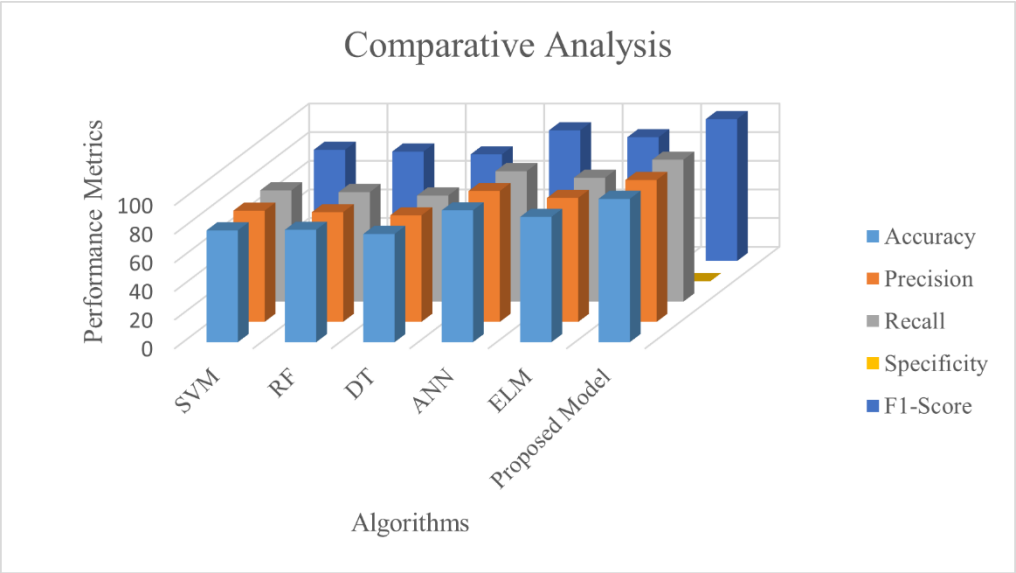


Figure 4. Correlative study of various framework in DDoS condition (NSL-KDD Datasets)

Table 6: Quality Measures of the various methodology in identifying the Normal Scenario (UNSW-19 Datasets)

Algorithm	**Performance Metrics**				
	Accuracy	**Precision**	**Recall**	**Specificity**	**F1-Score**
SVM	78	77.6	77.6	0.23	77.6
RF	78.4	76.5	76.4	0.24	76.35
DT	75.5	74.3	74	0.26	74.4
ANN	68.5	67.5	67.3	0.334	67.4
ELM	87.4	86.5	86.4	0.114	86.4
Proposed Model	**98.4**	**97.6**	**97.3**	**0.001**	**97.5**

Table 7: Quality Measures of the various methodology in identifying the DDoS attacks (UNSW-19 Datasets)

Algorithm	**Performance Metrics**				
	Accuracy	**Precision**	**Recall**	**Specificity**	**F1-Score**
SVM	82	81.3	80.5	0.200	81
RF	80.2	79.2	78.5	0.25	79
DT	81.4	80.3	79.5	0.21	78.4
ANN	92	91.3	90.2	0.110	90.4
ELM	88.5	87.5	86.4	0.114	86.4
Proposed Model	**98.4**	**97.6**	**97.3**	**0.001**	**97.5**

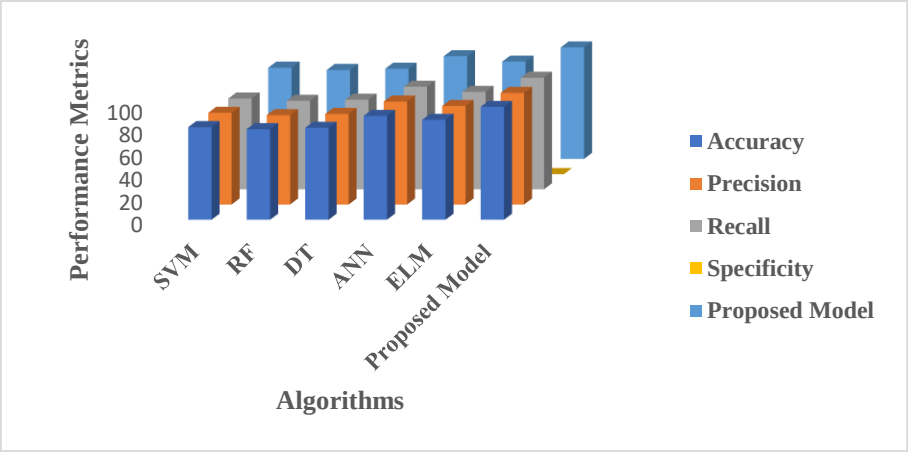


Figure 5. Correlative study of various Model for identifying the Normal Scenario (UNSW-19 Datasets)

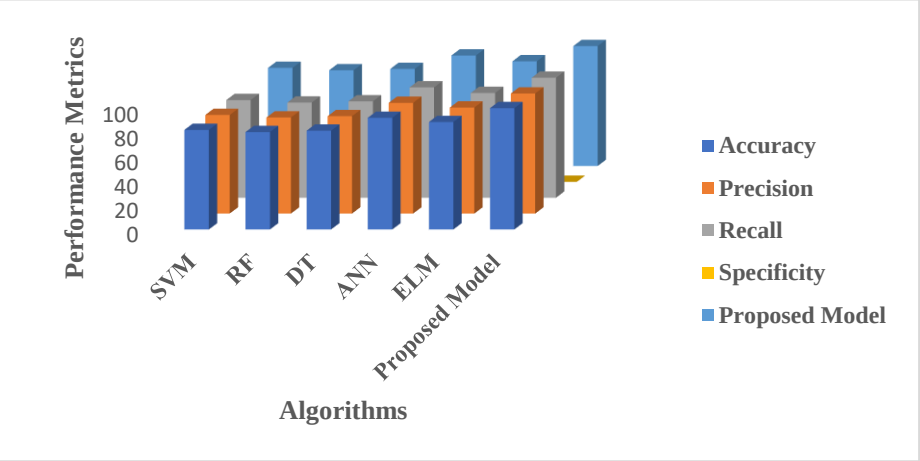


Figure 6. Correlative study of various Model for identifying the DDoS Attacks (UNSW-19 Datasets)

Figure (1 to 4) and Table (6 to 8) illustrates the authentication quality of the suggested framework in identifying the DDoS hazard and normal conditions. Figure 1 and table 6 depicts that the suggested framework has produced 95.5% in detecting the DDos attacks using both the datasets. The other learning models has produced the considerable promising performances but lesser than the proposed model. Though ELM and ANN have produced the better performance among the other machine learning models, it cannot able to match the performance of the suggested framework. CAT-FEED –FODS has outperformed the other learning models under the condition-1. With the Figure 3 and table 7 in detecting the foreign objects under the condition-2. In addition, the Figure 3 to table 8 depicts that the suggested framework is clear winner in identifying the DDoS hazard from the versatile datasets. From the results, it is clear that the CAT tuned Feed forward networks has played the significant role in detecting the DDoS attacks for the IoT-Cloud Environment.

5. Conclusion

This study proposed distinct machine learning model called FEED-CATS utilizing the feed forward networks whose hyper parameters are tuned by the cat swarm optimization algorithm. The FOD structure constructed for detecting the DDoS attacks in the IoT-Cloud Environment to demonstrate the superiority of the suggested framework, intensive analysis is examined by using the two different datasets such as NSL-KDD and UNSW datasets respectively. Additionally, the performance measures such as accuracy, precision, recall, specificity and F1-score are measured and contrasted with the various residing framework. Outcomes demonstrates that the suggested framework excelled the various machine learning model and highlight the extensive range of potential applications of the proposed FODS model to an effective detection of DDoS. AS the future scope, model can be bettered by considering the improvisation in the proposed model in handling the larger real time IoT datasets with multiple attacks.

Funding: “This research received no external funding”

Conflicts of Interest: “The authors declare no conflict of interest.”

References

[1] S. Huang, J. Su, S. Dai, C. Tai, and T. Lee, “Enhancement of wireless power transmission with foreign-object detection considerations,” in *Proc. 2017 IEEE 6th Global Conf. Consumer Electronics (GCCE)*, Nagoya, Japan, 2017, pp. 1-2.

[2] S. Fukuda, H. Nakano, Y. Murayama, T. Murakami, O. Kozakai, and K. Fujimaki, “A novel metal detector using the quality factor of the secondary coil for wireless power transfer systems,” in *Proc. 2012 IEEE MTT-S Int. Microwave Workshop Series on Innovative Wireless Power Transmission: Technologies, Systems, and Applications*, Kyoto, Japan, 2012, pp. 241-244.

[3] L. Xiang, Z. Zhu, J. Tian, and Y. Tian, “Foreign object detection in a wireless power transfer system using symmetrical coil sets,” *IEEE Access*, vol. 7, pp. 44622-44631, 2019.

[4] A. Azad, V. Kulyukin, and Z. Pantic, “Misalignment tolerant DWPT charger for EV roadways with integrated foreign object detection and driver feedback system,” in *Proc. 2019 IEEE Transportation Electrification Conf. and Expo (ITEC)*, Detroit, MI, USA, 2019, pp. 1-5.

[5] T. Sonnenberg, A. Stevens, A. Dayerizadeh, and S. Lukic, “Combined foreign object detection and live object protection in wireless power transfer systems via real-time thermal camera analysis,” in *Proc. 2019 IEEE Applied Power Electronics Conf. and Exposition (APEC)*, Anaheim, CA, USA, 2019, pp. 1547-1552.

[6] S. A. Nabi, P. Kalpana, N. S. Chandra, L. Smitha, K. Naresh, A. E. Ezugwu, and L. Abualigah, “Distributed private preserving learning based chaotic encryption framework for cognitive healthcare IoT systems,” *Informatics in Medicine Unlocked*, vol. 49, p. 101547, 2024. DOI: 10.1016/j.imu.2024.101547.

[7] S. Jeong, T. Lin, and M. M. Tentzeris, “A real-time range-adaptive impedance matching utilizing a machine learning strategy based on neural networks for wireless power transfer systems,” *IEEE Trans. Microwave Theory Techn.*, pp. 1-8, Sept. 2019. DOI: 10.1109/tmtt.2019.2938753.

[8] P. Kalpana, P. Srilatha, G. S. Krishna, A. Alkhayyat, and D. Mazumder, "Denial of service (DoS) attack detection using feed forward neural network in cloud environment," in *Proc. 2024 Int. Conf. Data Science and Network Security (ICDSNS)*, Tiptur, India, 2024, pp. 1-4. DOI: 10.1109/ICDSNS62112.2024.10691181.

[9] N. Prosen, M. Milanovič, and J. Domajnko, “On-line foreign object detection using double DD coils in an inductive wireless power transfer system,” *Sensors*, vol. 22, no. 4, p. 1637, 2022. DOI: 10.3390/s22041637.

[10] Y. Wang, H. Wang, and Z. Peng, "Rice diseases detection and classification using attention-based neural network and Bayesian optimization," *arXiv preprint arXiv: 2201.00893*, 2022.

[11] Y. Gong, Y. Otomo, and H. Igarashi, “Machine learning-based metal object detection for wireless power transfer using differential coils,” *J. Adv. Simulation Sci. Eng.*, vol. 9, no. 1, pp. 20-29, 2022. DOI: 10.15748/jasse.9.20.

[12] M. Wu, L. Guo, R. Chen, W. Du, J. Wang, M. Liu, X. Kong, and J. Tang, "Improved YOLOX foreign object detection algorithm for transmission lines," *Wireless Commun. Mobile Comput.*, vol. 2022, Article ID 5835693, pp. 1-10, 2022. DOI: 10.1155/2022/5835693.

[13] P. Kalpana, R. Anandan, A. G. Hussien, H. Migdady, and L. Abualigah, “Plant disease recognition using residual convolutional enlightened Swin transformer networks,” *Sci. Rep.*, vol. 14, p. 8660, 2024. DOI: 10.1038/s41598-024-56393-8.

[14] T. Munyer, D. Brinkman, X. Zhong, C. Huang, and I. Konstantzos, “Foreign object debris detection for airport pavement images based on self-supervised localization and vision transformer,” *arXiv preprint arXiv: 2210.16901*, 2022.

[15] R. J. Kuo and F. F. Nursyahid, “Foreign objects detection using deep learning techniques for graphic card assembly line,” *J. Intell. Manuf.*, 2022. DOI: 10.1007/s10845-022-01980-7.

[16] D. H. Nguyen, G. Tumen-Ulzii, T. Matsushima, and C. Adachi, “Performance analysis of a perovskite-based thing-to-thing optical wireless power transfer system,” *IEEE Photon. J.*, vol. 14, no. 1, pp. 1-8, 2022.

[17] A. Wu et al., “Mask R-CNN based object detection for intelligent wireless power transfer,” in *Proc. 2018 IEEE Globecom Workshops (GC Wkshps)*, Abu Dhabi, United Arab Emirates, 2018, pp. 1-5. DOI: 10.1109/GLOCOMW.2018.8644387.

[18] J. Cha, W. Lee, G. Choe, and Y. Kim, “A method of the improvement of wireless power transfer (WPT) system efficiency, compatibility, EMI reduction, and foreign object detection (FOD) for EV applications,” *SAE Tech. Paper 2020-01-0530*, 2020. DOI: 10.4271/2020-01-0530.

[19] B. Wang, S. Huang, and J. Qiu, “Parallel online sequential extreme learning machine based on MapReduce,” *Neurocomputing*, vol. 149, pp. 224-232, 2015.