International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 02 Feb. 2025, pp: 517-522 www.ijaem.net ISSN: 2395-5252

# Energy and Cost Effective Intrusion Detection System (IDS) for IoT using DNN-CFOA Model

[1]*Mr.B.Karthikeyan, [2*]Dr.K.Kamali
*Assistant Professor, Dept. of English, Annamalai university, Annamalainagar, Tamilnadu*
*Assistant Professor. , Dept. of CSE, Annamalai university, Annamalainagar, Tamilnadu*

Date of Submission: 15-02-2025 Date of Acceptance: 25-02-2025

**ABSTRACT**
Internet of Things (IoT) devices are vulnerable to a range of attacks that could compromise data or damage essential products. A cost-effective and energy-efficient Intrusion Detection System (IDS) solution for the IoT is needed to balance the security with the resource constraints of IoT devices. These IDS solutions optimize processing speed and energy consumption while maintaining robust threat detection. In this paper, Energy and Cost Effective IDS for IoT, using DNN-CFOA model is proposed. In this work, Deep and Convolutional Network (DNN) has been applied for the task of intrusion detection from patterns and Catch Fish Optimization Algorithm (CFOA) has been applied to optimize the weights of DNN model. Experimental results show that the proposed DNN-CFOA model outperforms the existing models with respect to accuracy and F1-score metrics.

**KEYWORDS:** Internet of Things (IoT), Intrusion Detection System (IDS), Cost effective, Deep Convolutional Network (DNN), Catch Fish Optimization Algorithm (CFOA)

## I. INTRODUCTION

A global network of interconnected, communicative items is the vision of IoT, a technological paradigm shift. According to recent figures, there are currently over 13.8 billion IoT devices in operation, and by 2025, that number is expected to rise to 30.9 billion. New uses are made feasible by this extensive network, which enables information sharing and remote control of smart devices. IoT networks are composed of intelligent physical objects that have been implanted with sensors, software, communication systems, and computational components.However, IoT devices are vulnerable to a range of attacks that could compromise data or damage essential products [1].

One effective security option is an Intrusion Detection System (IDS), which is designed to detect malicious activities or cyberattacks. Based on the kind of data source they employed,IDS can be categorized as either host-based or network-based.Because host-based IDS monitor data from the host system logs, including operating system and application logs, it can be helpful in detecting intrusions in sensitive files or programs. However, this type's reliance on host reliability and resource availability limits its ability to detect network threats [3]. Conversely, because network-based IDS operate independently of hosts, it can be used in a range of situations to detect network-based attacks. Nevertheless, it is limited to identifying attacks that take place within a certain network segment.

The three main categories of IDS detection methods are hybrid, anomaly-based, and signature-based [4].Even though anomaly-based intrusion detection systems are better at identifying emerging threats, they have disadvantages such as high false-positive rates and low explainability for reported anomalies. Hybrid intrusion detection systems (IDS) utilize the benefits of both approaches to provide a more comprehensive and effective intrusion detection system.

One of three methods is usually used by anomaly-based IDS: knowledge-based, statistics-based, or ML-based. The statistics-based technique can detect low-probability events as possible intrusions [5]. Nevertheless, this approach necessitates intricate mathematical expressions for variables that reflect user behavior.ML-based IDS employ Artificial Intelligence (AI) to find patterns in data and provide predictions without the need for explicit programming. By using intrusion datasets

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 02 Feb. 2025, pp: 517-522 www.ijaem.net ISSN: 2395-5252

to train models, ML-based intrusion detection systems can determine whether new, unseen traffic is valid or illegitimate [6]. ML-based intrusion detection systems use a variety of techniques, including clustering, association rules, decision trees, closest neighbour approaches, and DL.

Accurate test data classification is made possible by supervised learning, which builds predictive models using labelled training data. Nevertheless, this approach necessitates a large amount of labelled data, it is costly and labour-intensive to produce. Conversely, unsupervised learning employs unlabeled data and classifies inputs based on statistical features, making it suitable for scenarios when labelled data is unavailable [7].

A cost-effective and energy-efficient IDS for the IoT aims to balance security with the resource constraints of IoT devices. These IDS solutions optimize processing speed and energy consumption while maintaining robust threat detection [8]. Cost effectiveness is also achieved by lowering hardware requirements and utilizing software-based, scalable techniques. Because these IDS systems ensure real-time attack detection while preserving device performance and battery life, they are suitable for large, resource-constrained IoT installations.

## II. RELATED WORKS

It is difficult to develop an IDS for IoT networks because of the large amount of heterogeneous data generated, which makes real-time analysis tough. The inefficiency of traditional IDS methods emphasizes the need for sophisticated methods that use machine learning or deep learning. With a multifaceted classification strategy, this study [11] suggests a deep ensemble-based IDS that makes use of Lambda architecture. While multi-class classification uses a combination of LSTM, Convolutional Neural Network (CNN), and Artificial Neural Network (ANN) models to detect attack types, binary classification uses LSTM to separate malicious from benign data. The speed layer evaluates the model in real time, while the batch layer trains the model.

Traditional IDSs are not appropriate for edge-cloud systems, and the deployment of DL models close to devices is hampered by the massive amounts of IoT data and processing needs. This is addressed by a new edge-cloud-based IoT IDS [12], which uses distributed processing to train a Recurrent Neural Network (RNN) with Bidirectional LSTM, segment datasets, and pick attributes on time-series data. The model, which was tested on the BoT-IoT dataset, decreases the size of the dataset by 85% without sacrificing detection accuracy. In edge-cloud deployments, this scalable DL-based solution is perfect for managing high volumes of IoT data.

For local IoT gateways, Realguard [13] is a DNN-based network IDS that offers precise, real-time cyberattack detection with low processing requirements. Realguard outperforms rivals at 98.85% in identifying ten attack types with 99.57% accuracy thanks to an effective DNN model and a lightweight feature extraction mechanism. With a high packet processing rate of 10,600 packets per second, it functions well on gateways with limited resources, such as Raspberry Pi, and provides strong IoT network security.

### 2.1 Research Gaps

The existing works on IDS for IoT networks presents several gaps: Most studies emphasis on detecting a small set of IoT attack types, failing to address the wide and evolving range of potential attacks across diverse IoT applications and environments. Many approaches do not prioritize real-time intrusion detection, instead depending on offline datasets, which restrictsthe applicability of the system in actual IoT environments that necessitate dynamic and immediate threat responses. A significant number of studies utilize binary classifiers for distinguishing only between benign and malicious traffic, missing the capability to classify and recognize specific types of attacks, which is vital for effective IoT network security. Most of the existing IDS models do not combine ensemble-based methods, which combine multiple classifiers to enhance detection accuracy and precision, causing lower detection rates when compared with more advanced approaches.

## III. PROPOSED METHODOLOGY

### 3.1 Overview

### 3.2 DNN based detection

Hidden layers and neurons were gradually added to the DNN until the model met the requirements for maximizing the number of hidden nodes and successfully detecting attacks while consuming the least amount of resources. The five hidden layers in our improved model, which includes about 34,315 parameters, strike a balance between detection performance and simplicity. The architecture of the assault detection model is depicted in Figure 1, where each hidden layer is composed of neurons that are completely linked to those in the layer below.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 02 Feb. 2025, pp: 517-522 www.ijaem.net ISSN: 2395-5252

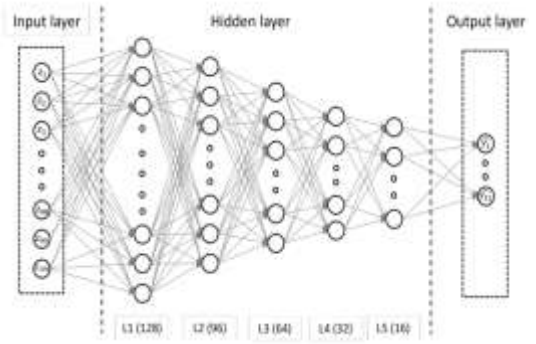


Figure 1. Architecture of the attack detection model

Over the levels, information will be processed forward. The feature extractor provides a normalized vector z ∈ Rm (m = 100) to the input layer, which consists of n neurons. Each hidden layer $H_i$'s computation, where the input vector $x \in R^{(d(i-1))}$ (from $H_{i-1}$ or z) is processed, is defined as follows:

$$H_i(x) = f(w_i^T x + c_i) \quad (1)$$

The activation function is denoted by 〖f:R〗 ^(d_(i-1) )→$R^{(d}_i$, while the weight matrix and bias vector are represented by $w_i$ and $c_i$, respectively. The values from layer $H_{i-1}$ are mapped to layer $H_i$ by the procedure $w_i^T x+c_i$. The ReLU activation function is used to improve convergence and address vanishing gradient problems. The output of each hidden layer or the element $j^{th}$ of vector $H_i(x)$ is calculated as follows:

$$f(x_j^r) = \max(0, x_j^r) \quad (2)$$

Additionally, this design necessitates classifying inputs into many attack categories. The softmax function is a widely utilized technique for these kinds of tasks in order to achieve this. By setting the final layer's size equal to the number of attack types, the softmax function calculates the likelihood that an input belongs to each class. This likelihood is provided by:

$$\hat{y}_k = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_i}} \quad (3)$$

where n is the number of attack types and y^ _k is the likelihood that the input vector x is a member of the kth attack class..

### 3.3 Optimization of Weights using CFOA

To optimize the weights of DNN, CFOA is applied. It includes the following steps:

#### 3.3.1 Initialization Process

As with many metaheuristic algorithms, the populations are produced at random during the CFOA initialization. N fishermen are taken into consideration in a D-dimensional search space, where the mathematical representation of the fishermen's population matrix is as follows:

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} x_{1,1}\, x_{1,2} \ldots x_{1,j} \ldots x_{1,D} \\ x_{2,1}\, x_{2,2} \ldots x_{2,j} \ldots x_{2,D} \\ \vdots \quad \vdots \quad \ddots \quad \vdots \quad \ddots \quad \vdots \\ x_{i,1}\, x_{i,2} \ldots x_{i,j} \ldots x_{i,D} \\ \vdots \quad \vdots \quad \ddots \quad \vdots \quad \ddots \quad \vdots \\ x_{N,1} x_{N,2} \cdots x_{N,j} \cdots x_{N,D} \end{bmatrix} \quad (4)$$

Every fisher represents a possible fix. The objective function of the problem can be evaluated using its values for the choice variables. For the fishermen, the objective function value is provided by:

$$F(X)=[F_1,F_2,\ldots F_N]=[f(X_1),f(X_2),\ldots f(X_N)] \quad (5)$$

In this case, the value of the objective function for the ith fisher is indicated by $F_i$ and $f(X_i)$. The position of the fisherman is indicated by each Xi, which is initialized using the following formula:

$$X_{i,j} = L_j + \text{rand} * (U_j - L_j), i = 1,2, \ldots N \quad (6)$$

where $U_j$ and $L_j$ represent the upper and lower bounds for the jth dimension in the problem space, and $X_{i,j}$ represents the $i^{th}$ fisher's location in the $j^{th}$ dimension. The two stages of metaheuristic algorithms are typically exploration and exploitation. The parameter W in CFOA controls the transition from exploration to exploitation. CFOA does exploration when W < 0.5 and exploitation otherwise. W is calculated using

$$W = \frac{\text{Currentiterationnumber}}{\text{max .iterationnumber}} \quad (7)$$

#### 3.3.2 Independent Search and Group Capture

During the exploration phase, CFOA employs two search strategies: group capture and independent search. A fisherman selects one of these search strategies to capture the fish in each iteration. The catch rate parameter α determines which of the two approaches should be used. The formula for estimating the value of α is

$$\alpha = (1 - \frac{3}{2}W)^{\frac{3}{2}W} \quad (8)$$

Fishermen choose to conduct independent searches when α is high (α > p). Fish surface to breathe as a result of the fishermen's disturbance of the water during the fishing process. By watching the ripples the fish makes, the fishermen are able to

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 02 Feb. 2025, pp: 517-522 www.ijaem.net ISSN: 2395-5252

determine the fish's location. Additionally, they modify their stance in response to other people's achievements. The following is the updating strategy for the fisherman's position:

$$X_{i,j}(t+1) = X_{i,j}(t) + \left(X_{r,j}(t+1) - Xi,jt+1*E+rand*wj*U \right) \quad (9)$$

The experience of the $i^{th}$ fisherman using $X_{r,j}(t)$ as a reference is represented by

$$E=(F_i-F_r)/(F_{max}-F_{min}) \quad (10)$$

Where D is the Euclidean distance between the reference point $X_r(t)$ and the $i^{th}$ fisher's position $X_i(t)$, $U=D*\sqrt{E}*(1-W)$. In contrast to $X_{i,j}(t)$, $X_{r,j}(t)$ is the location of a randomly selected fisherman. The current iteration's maximum and minimum fitness values are denoted by $F_{max}$ and $F_{min}$. A random unit vector in the D-dim space is denoted by $w_j$.

Fishermen switch to group capture when $\alpha \leq p$. In order to increase their fishing ability, fishermen often work together and use nets. A group of three or four fishermen work together. The following provides the update strategy:

$$X_{i,j}(t+1) = X_{i,j}(t) + \text{rand} * \left(C_e - X_{e,j}(t)\right) + (1-2W)^2 * r_1 \quad (11)$$

In this case, e is the group of three or four people, and Ce is the objective point of the group's encirclement. mean($X^{e(t)}$) is the average location of the group Xe(t), and $r_1$ is a random number between -1 and 1.

### 3.3.3 Collective Capture

In order to guide both free and concealed fish to a central region for encirclement and capture, all fishermen collaborate under a common search technique as the fishing continues. The distribution of fishermen is concentrated around the school of fish, with a gradual thinning of aggregation from the centre to the periphery, while the distribution range becomes more limited as it moves outward. The centre fishermen focus on capturing the school of fish while the perimeter fishermen handle any escaping fish. The most recent position of a fisherman is calculated using the algorithm below:

$$X_i(t+1) = X_G + \text{normrnd}\left(0, \frac{r_2*\varepsilon*|\text{mean}(X)-X_G|}{3}\right) \quad (12)$$

$$\varepsilon = \sqrt{\left(\frac{2*(1-W)}{(1-W)^2+1}\right)} \quad (13)$$

The position of the fisherman with the highest fitness value is denoted by XG. $|\cdot|$ specifies the usage of absolute values, and r2 indicates a random number between 1 and 3.

### 3.4 The CFOA fitness function

The goal of optimization is to increase the IDS's energy efficiency while maintaining or enhancing its accuracy. The purpose of objective O is calculated as the weighted sum of the ML model's energy expenditure per inference (E) and accuracy (A):

$$O = \gamma A + \delta E \quad (14)$$

where the trade-off between accuracy and energy efficiency is managed by the weights γ and δ. The proportion of correctly categorized examples to all samples is known as accuracy A. By adjusting the model's hyperparameters, the optimization process seeks to decrease O while guaranteeing that IDS achieves high accuracy without using excessive energy.

## IV. EXPERIMENTAL RESULTS

The proposed energy DNN-CFOA model has been implemented in Python 3.0 with Google Colab environment. The KDD cup dataset which contains 42 features with 494021 records, has been used in the experiments.

### 4.1 Classification Results

The performance of the DNN-CFOA model has been compared with the DNN and ANN classifiers, without applying any feature selection process. The classification performance is evaluated in terms of the following measures:

**Accuracy:** The ratio of successfully categorized data to total data

$$\text{Accuracy} = \frac{TN+TP}{FP+FN+TP+TN} \quad (15)$$

**F1-score**: The harmonic-mean of sensitivity and precision.

$$\text{F1-score} = 2x\frac{precisionXrecall}{precision+recall} \quad (16)$$

Here,

$$\text{Precision} = \frac{TP}{TP+FP}, \quad \text{Recall} = \frac{TP}{TP+FN} \quad (17)$$

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 02 Feb. 2025, pp: 517-522 www.ijaem.net ISSN: 2395-5252

Table 1 and Figure 2 show the comparison results of accuracy and F1-score for these 3 approaches.

| Techniques | Accuracy | F1-score |
|---|---|---|
| DNN-CFOA | 98.35 | 95.25 |
| DNN | 96.27 | 93.72 |
| ANN | 95.15 | 90.68 |

Table 1 Comparison results of Accuracy and F1-score

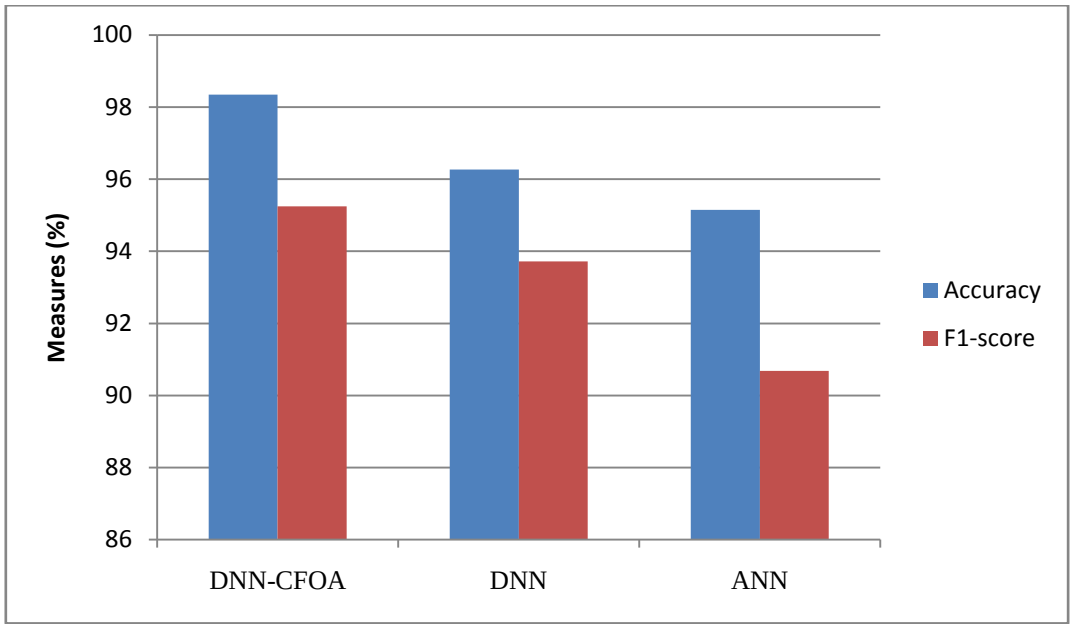


Figure 2 Comparison Results

As seen from Figure 6, the proposed FS-DNN classifier attains highest accuracy of 98.35% which is 3.25% and 2.11% higher than ANN and DNN classifiers, respectively. Similarly, the F1-score ofDNN-CFOA is 95.25%, which is 4.8% and 1.6% when compared to ANN and DNN classifiers, respectively.

## V. CONCLUSION

In this paper, energy and cost Effective IDS for IoT, using DNN-CFOA model is proposed. In this work, DNN model has been applied for the task of intrusion detection from patterns and CFOA has been applied to optimize the weights of DNN model.The KDD cup dataset which contains 42 features with 494021 records, has been used in the experiments. Experimental results show that the proposed DNN-CFOA model outperforms the existing models with respect to accuracy and F1-score metrics.

**Funding**: The authors didn't get any funding support for this research.

**Conflict of Interest:** The authors don't have any conflict of Interest

**Data Availability:** The authors didn't use any third party data materials in their research.

**Authors Contribution:** In this manuscript preparation author 1 prepared the concept and author 2 prepared the implementation part

## REFERENCES

[1]. V. Gotarane and R. Iyer, "Optimizing Energy-Efficient Machine Learning Algorithms for Real-Time Attack Detection in IoT Devices," J. Electrical Systems, vol. 20, no. 3, pp. 6912–6919, 2024.

[2]. Awajan, A. A Novel Deep Learning-Based Intrusion Detection System for IoT Networks. Computers 2023, 12, 34. https://doi.org/ 10.3390/computers12020034

[3]. S. Tsimenidis, T. Lagkas, and K. Rantos, "Deep Learning in IoT Intrusion Detection," Journal of Network and Systems Management, Springer, Volume 30, Issue 1, Article 8, 2022. DOI: 10.1007/s10922-021-09621-9.

[4]. S. Altamimi and Q. Abu Al-Haija, "Maximizing intrusion detection efficiency for IoT networks using extreme learning machine," Discover Internet of Things, vol. 4, no. 5, 2024. DOI: 10.1007/s43926-024-00060-x.

[5]. R. Khan, M. Kashif, R. H. Jhaveri, R. Raut, T. Saba, and S. A. Bahaj, "Deep Learning for Intrusion Detection and Security of Internet of Things (IoT): Current Analysis, Challenges, and Possible Solutions," Security and Communication Networks, 2022.

[6]. Deshmukh, A.; Ravulakollu, K. An Efficient CNN-Based Intrusion Detection System for IoT: Use Case Towards Cybersecurity. Technologies 2024, 12, 203. https://doi.org/ 10.3390/technologies12100203

[7]. Y. Otoum, D. Liu, and A. Nayak, "DL-IDS: A Deep Learning-Based Intrusion Detection Framework for Securing IoT," Transactions on Emerging Telecommunications Technologies, vol. 33, no. 3, Mar. 2022.

[8]. J. Corona, M. Antunes, and R. L. Aguiar, "Eco-Friendly Intrusion Detection: Evaluating Energy Costs of Learning," 2023 IEEE 10th World Forum on Internet of Things (WF-IoT), October 2023, doi: 10.1109/WF-IoT58464.2023.10539426.

[9]. O. Elnakib, E. Shaaban, M. Mahmoud, and K. Emara, "EIDM: Deep learning model for IoT intrusion detection systems," The Journal of Supercomputing, vol. 79, pp. 13241–13261, 2023. doi: 10.1007/s11227-023-05197-0.

[10]. A.Kaushik and H. Al-Raweshidy, "A novel intrusion detection system for internet of things devices and data," Wireless Networks, vol. 30, pp. 285–294, 2024. [Online]. Available: https://doi.org/10.1007/s11276-023-03435-0

[11]. R. Alghamdi and M. Bellaiche, "An ensemble deep learning-based IDS for IoT using Lambda architecture," Cybersecurity, vol. 6, no. 5, 2023. [Online]. Available: https://doi.org/10.1186/s42400-022-00133-w

[12]. A.Aldaej, T. A. Ahanger, and I. Ullah, "Deep Learning-Inspired IoT-IDS Mechanism for Edge Computing Environments," Sensors, vol. 23, no. 24, p. 9869, 2023. [Online]. Available: https://doi.org/10.3390/s23249869

[13]. Nguyen, X.-H.; Nguyen, X.-D.; Huynh, H.-H.; Le, K.-H. Realguard: A Lightweight Network Intrusion Detection System for IoT Gateways. Sensors 2022, 22, 432. https://doi.org/10.3390/s22020432