International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Energy Efficient Intrusion Detection System (IDS) and Feature Selection for IoT using DNN Model

Mr. B. Karthikeyan[1], Dr. K. Kamali[2]

[1]Assistant Professor, Dept. of English, Annamalai university, Annamalainagar, Tamilnadu
[2]Assistant Professor, Dept. of CSE, Annamalai university, Annamalainagar, Tamilnadu

Abstract
To reduce these risks and avoid system paralysis in, it is crucial to create novel methods for establishing strong security layers in Internet of Things (IoT) networks. The low processing power and storage capabilities of the majority of IoT devices mean that current attack protection techniques frequently need to be modified. An energy-efficient Intrusion Detection System (IDS) and feature selection (FS) for IoT using Deep and Convolutional Network (DNN), aims to balance security with the resource constraints of IoT devices. In first phase, Clustered Data Processing technique is applied for the feature selection of dataset. By identifying relationships between input attributes, this technique automatically creates self-organizing models of optimal complexity. In the next phase, DNN based detection model is applied by means of training and testing. Experimental results show that the proposed FS-DNN model outperforms the existing models with respect to accuracy and F1-score metrics.

Keywords: Internet of Things (IoT), Intrusion Detection System (IDS), Deep and Convolutional Network (DNN), Feature selection, Clustered Data Processing

1. Introduction

A global network of interconnected, communicative items is the vision of the Internet of Things (IoT), a technological paradigm shift. These days, practically each equipment in a vast array of industries uses has an internet connection. In order to gather and exchange data globally, 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 like smart cars and medical equipment due to their scale, diversity, and long-distance connectivity [1].

One effective security option is an Intrusion Detection System (IDS), which is designed to detect malicious activities or cyberattacks. An IDS can prevent potential harm by identifying variations between normal and abnormal network traffic or system usage patterns [3]. The three main categories of IDS detection methods are hybrid, anomaly-based, and signature-based. Signature-based intrusion detection systems use pattern-matching algorithms to find known attack signatures and, consequently, pre-existing threats. However, it cannot detect unknown or zero-day attacks. Anomaly-based intrusion detection systems, on the other hand, look at network traffic or usage patterns using statistical models, time-series studies, or machine learning techniques to find anomalies [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.

Machine Learning (ML) based intrusion detection systems employ artificial intelligence to find patterns in data and provide predictions without the need for explicit programming. By using intrusion datasets to train models, ML based intrusion detection systems can determine whether new, unseen traffic is valid or illegitimate [5]. One of the key benefits of ML-based IDS is its capacity to continuously improve detection accuracy without requiring model reconstruction, hence reducing false positives [6][7].

A cost-effective and energy-efficient intrusion detection system 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. Lightweight algorithms, edge computing, and ML techniques created for low-power devices are commonly used to reduce the computational load and energy consumption [8].

2. Related Works

ML is being used to detect anomalies in IoT systems and cyberattacks as a result of its growing viability. With the use of the CICIDS2017 dataset, this study [9] presents an improved anomaly-based Intrusion Detection DL Multiclass Classification Model (EIDM). Additionally, six network traffic characteristics are classified using four cutting-edge DL models.

A Teaching-Learning-Based Optimization-enabled IDS (TLBO-IDS) [10] offers efficient detection with low overhead. While maintaining maximum throughput and little communication and device overhead, the TLBO-IDS is capable of detecting a wide range of attack types, such as analysis, fuzzing, shellcode, worms, DoS, exploits, and backdoor intrusions. 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 Long Short-Term Memory (LSTM) to separate malicious from benign data.

3. Proposed Methodology

3.1 Feature Selection using Clustered Data Processing

A type of feedforward network for DL that is a member of the heuristic algorithm family is called Clustered Data Processing (CDP). By identifying relationships between input attributes, it automatically creates self-organizing models of optimal complexity. Without any outside input, it creates its internal structure. The following is the definition of the relationship between input variables x1 and x2:

$$z = b + \sum_{j=1}^{m} c_j x_j + \sum_{j=1}^{m} \sum_{k=1}^{m} d_{jk} x_j x_k + \ldots \quad (1)$$

where b, c, and d stand for the corresponding weights, and m stands for the variables in each neuron layer. In order to mimic the process of natural evolution, CDP uses inductive learning to infer the most intricate relationships between variables. In order to derive more complicated relationships, input connections are made simpler. To predict z, $m(m-1)/2$ pairs of input variables are used instead of n conventional input variables. When choosing the best qualities, CDP takes into account all likely input pairings, where "best" refers to the most significant correlations between input and output.

The CDP includes the subsequent steps:

Step 1: A single neuron receives two randomly selected properties. Determining the attribute pool, picking subgroups, evaluating performance, picking the best subset, and repeating the procedure are all

all included in this.

Step 2: By comparing the training set with each neuron's current state, the weights are estimated.

Step 3: For every neuron, probabilities are calculated using training and validation datasets.

Step 4: An objective criterion is used to select the most efficient neurons.

Step 5: After selecting validation error, the model is evaluated using bias error, validation error, and other criteria.

Step 6: Depending on the input factors, users have the option to change the neurons for each layer or have them calculated automatically.

Step 7: If there are validation issues or the maximum number of layers is reached, the process starts over from the beginning.

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.

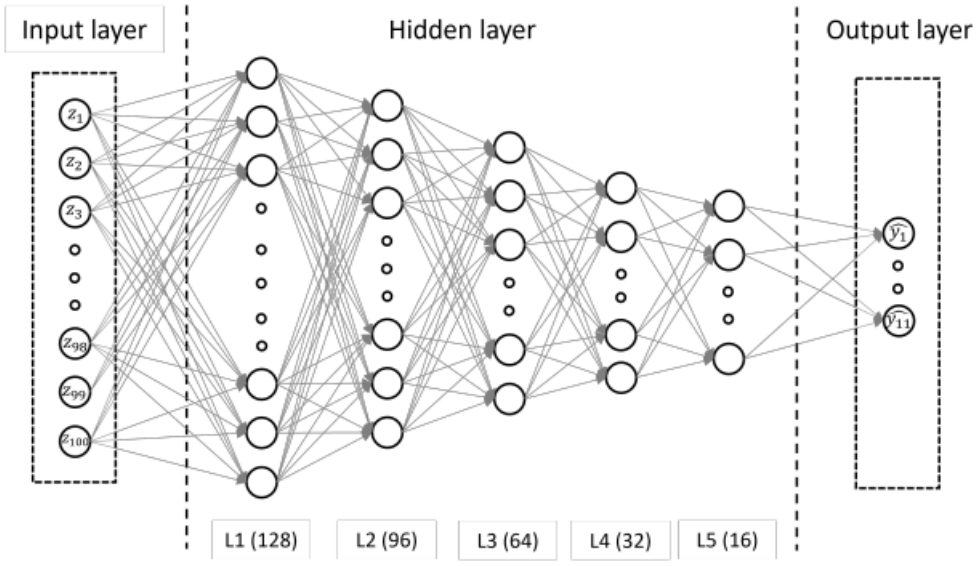

Figure 1. Architecture of the attack detection model

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:

$$H_i(x) = f(w_i^T x + c_i) \quad (2)$$

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:

$$f(x_j^r) = \max(0, x_j^r) \quad (3)$$

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:

$$\hat{y}_k = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_i}} \quad (4)$$

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..

3.4 The 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):

$$O = \gamma A + \delta E \quad (5)$$

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.

4. Experimental Results

The proposed energy efficient IDS using feature selection and DNN (FS-DNN) 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 AS-DNN 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

$$\text{Accuracy} = \frac{TN + TP}{FP + FN + TP + TN} \quad (6)$$

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

$$\text{F1-score} = 2x\frac{precisionXrecall}{precision + recall} \quad (7)$$

Here,

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN} \quad (8)$$

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

Techniques	Accuracy	F1-score
FS-DNN	97.75	96.55
DNN	95.64	94.2

ANN	94.70	92.18

Table 1 Comparison results of Accuracy and F1-score

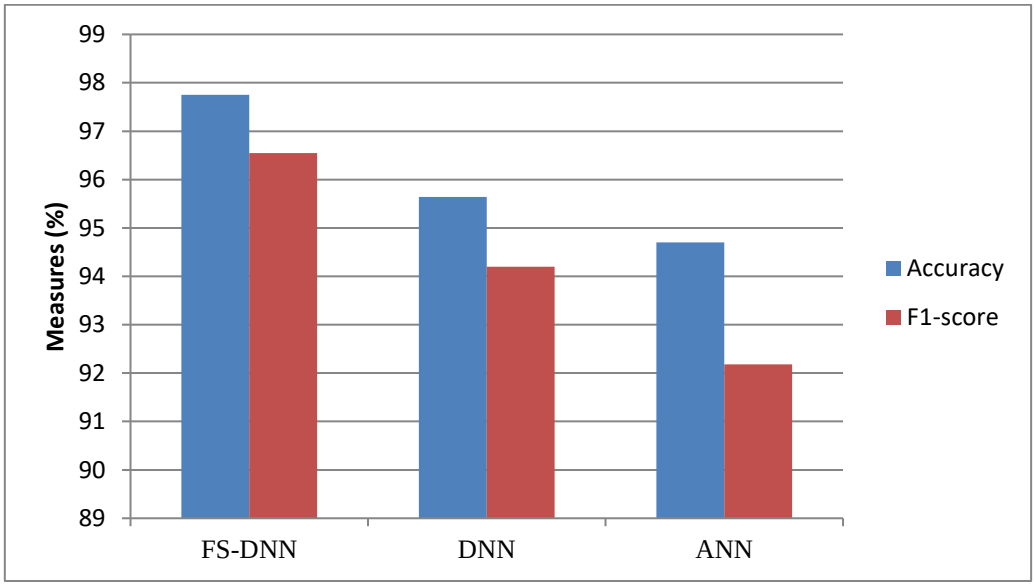

Figure 2 Comparison Results

As seen from Figure 6, the proposed FS-DNN classifier attains highest accuracy of 97.75% which is 3.12% and 2.15% higher than ANN and DNN, respectively. Similarly, the F1-score of AS-DNN is 96.55%, which is 4.5% and 2.4% when compared DNN and ANN classifiers, respectively.

5. Conclusion

This paper proposes an energy-efficient IDS and feature selection (FS) for IoT using DNN to provide security satisfying the resource constraints of IoT devices. In first phase, CDP technique is applied for the feature selection of dataset. By identifying relationships between input attributes, this technique automatically creates self-organizing models of optimal complexity. In the next phase, DNN based detection model is applied by means of training and testing. Experimental results show that the proposed FS-DNN model outperforms the existing DNN and ANN models with respect to accuracy and F1-score metrics

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