Volume (1) No (9), 2025
Journal of Global Research in Multidisciplinary Studies (JGRMS)
Review Paper/Research Paper
Available online at https://saanvipublications.com/journals/index.php/jgrms/index

Improvising Cybersecurity in IoT Networks Using Machine Learning for Intrusion Identification

Mr. Deepak Mehta
Assistant Professor
Department of Computer Sciences and Applications
Mandsaur University
Mandsaur
deepak.mehta@meu.edu.in

***Abstract*—Smart cities, industrial automation, healthcare, and the Internet of Things have all been profoundly affected by the exponential expansion of the IoT, which has enabled massive connection and intelligent decision-making. The ever-changing, varied, and sometimes under-resourced nature of IoT devices makes intrusion detection all the more important in protecting these networks from cyberattacks. The drawbacks of classic intrusion detection systems (IDS) include issues with scalability, high false positive rates, and the inability to identify sophisticated attacks such as zero-day vulnerabilities. To get over these problems, this study presents a robust intrusion detection model that optimizes features and leverages deep learning. To ensure high-quality input utilizing the UNSW-NB15 dataset, preprocessing procedures for data were utilized, such as normalization, resolving missing values, and balancing with SMOTE. Employed the Butterfly Optimization Algorithm (BOA) to boost computing efficiency and cut down on redundant features. Then, a Deep Neural Network (DNN) was trained with F1-score, recall, accuracy, and precision exceeding 98.04%. The suggested framework's better performance and scalability are brought to light through comparative study with existing frameworks. Lightweight adaptations for IoT devices with limited resources, real-time deployment, and evaluation across datasets the main areas of future development.**

Keywords—Intrusion Detection System (IDS), Internet of Things (IoT) Security, Deep Neural Network (DNN), Cybersecurity, Machine Learning, SMOTE

I. INTRODUCTION

The Internet of Things (IoT) has risen to the top of providing digital transformation due to the connectivity of hundreds of billions of gadgets in smart cities, homes, and companies [1]. IoT networks allow real-time coordination of automation and communication between devices, whether these devices are smart thermostats or smart health-care wearables, smart sensors in industries, or smart transportation. Supply chain networks provide the advantages of economies of scale, better decision making, reduction of operational costs, and customer flexibility [2][3] IoT solutions play a significant role in providing a safer and more intelligent environment that can respond to the needs of people in hazardous circumstances by integrating the ability to conduct continuous data gathering and evaluation.

With their advantages, however, come a larger attack surface that exposes IoT infrastructures to radically different threats to their cybersecurity. Nevertheless, the heterogeneous structural properties, as well as the outdated security schemes, of most systems in the last decade, make them more susceptible to malicious activities, because of the computational burden of IoT devices [4]. Cybercriminals target IoT networks due to these vulnerabilities; as a result, consumers cannot be confident that their sensitive data will be protected on these networks.

IDS has long been applied to monitoring network traffic, in order to identify out-of-the-ordinary behavior and cyber-attacks [5][6]. An IDS is a device that monitors traffic flow to determine whether a system is under attack or not and is therefore a crucial tool in the defense against cyber-attacks. Nevertheless, conventional IDS methods are slowly losing their utility to Internet of Things networks with varied and increasingly limiting requirements; as the responsibility of the IoT, defense temperatures increase and visibility decreases. Data breaches, malware infections, intrusion attempts, and DDoS (Denial-of-Service) attacks are just a few of the numerous risks that are becoming more widespread in IoT environments [7][8][9].

Conventional intrusion detection systems based on signatures will usually fare poorly in such environments as they are designed to mitigate against known attack patterns and not novel ones [10]. Moreover, the IoT devices, owing to resource limitations, are incapable of implementing expensive detection algorithms; therefore, the adaptive and low-cost security solution that could manage the dynamic characteristics of the IoT ecosystems is urgently requested [11][12]. More advanced systems that offer real-time protection proactively and not by simply relying on a static method of detection are therefore in demand [13].

Intrusion detection in IoT networks is a topic with many obstacles, and one potential solution is to use machine learning (ML) approaches [14][15]. ML models can process the large volumes of IoT traffic data, learn normal and abnormal behaviour patterns, and detect anomalies in real time [16]. Unlike more traditional solutions [17], ML-based IDSs are either dynamic or scalable and able to detect zero-day attacks without known signatures [18]. Intrusion detection systems that use ML can significantly enhance the cybersecurity of the IoT, and, therefore, the resiliency and confidence of increasingly connected ecosystems, which is why introducing machine learning to intrusion detection is an essential move toward building secure and sustainable IoT ecosystems [19][20][21].

A. Motivation and Contribution

The research is inspired by the fact that hacking on IoT networks, which form the backbone of the current digital

infrastructure is on the increase. High false positive rates, poor scalability, and inability to identify sophisticated attacks are just a few of the problems that plague traditional intrusion detection systems. With IoT generating vast, heterogeneous traffic, there is a need for intelligent, adaptive, and accurate detection mechanisms. This project seeks to construct a scalable intrusion detection model that improves cybersecurity and protects sensitive data in networked IoT environments. It does this by utilizing ML and DL techniques, feature selection, ensemble methodologies, and equally distributed datasets. A number of important advances in the area of network security are brought about by this study:

- An intrusion detection benchmark that is both demanding and representative was used to create and assess the model. The dataset used was the UNSW-NB15. The dataset includes a broad range of attack tactics as well as real-world traffic.
- Applied cleaning, handling of missing values, normalization, and balancing with SMOTE to improve data quality and ensure fair class representation.
- Integrated the Butterfly Optimization Algorithm (BOA) to eliminate redundant features, reduce computational cost, and enhance model efficiency.
- The proposed approach substantially improves detection performance and has been successfully used to detect intrusions in modern networks and the IoT.
- Developed a DNN-based intrusion detection system that achieved superior accuracy.
- Developed a comprehensive evaluation strategy by putting the model through its paces using numerous metrics to ensure a thorough and reliable study.

B. Justification and Novelty

The present intrusion detection system is generally hampered by class imbalance, feature redundancy, and the inability of these systems to detect different kinds of attacks; this study aims to overcome these shortcomings. The novelty of the proposed research is that the Butterfly Optimization Algorithm (BOA) is used together with a DNN model to select the most significant features and less significant features (the number of calculations is the least possible). In addition, the dataset has been balanced with SMOTE, implying that the minority attack classes are represented reasonably, improving the detection results in all categories. Using cutting-edge pre-processing methods, intelligent feature selection algorithms, and DL, this combination of techniques creates a novel and effective intrusion detection method that improves cybersecurity on IoT and modern networks.

C. Structure of the Paper

The following is the structure of the paper In Section II, relevant research is reviewed. In Section III, go over the evaluation process, model design, and measurements. The experimental results and comparison with the current approaches are offered in Section IV. Section V concludes the investigation and discusses potential next steps.

II. Literature Review

A number of prominent research studies on Intrusion Detection as a means of enhancing cybersecurity have been examined and discussed to inform and aid the creation of this work.

Ali et al. (2025) examine methods to detect and address malware injection attacks on the IoT by utilizing DL and ML techniques. Utilizing a pre-built dataset to simulate network traffic and different attack scenarios, three models, LSTM, Random Forest, and Support Vector Machine, were tested. RF (94%) was the most accurate and a strong and balanced performer. LSTM had 92% accuracy in sequential data analysis, again, SVM failed to manage its false positive rates at the cost of 85.7% accuracy. These results indicate that ML, specifically RF and LSTM, may help to improve the security of IoT. They open the path to stronger and more efficient frameworks to detect and mitigate MitM attacks [22].

Prasad et al. (2025) CSMCR is a new method that is introduced in this article for reducing unnecessary majority-class samples of datasets without affecting their consistency. To avoid employing oversampling or random under sampling, CSMCR evaluates feature-wise similarity to keep the retained majority instances diverse. By doing so, you can lessen the chances of data loss and steer clear of tedious, repetitive chores. Improved feature extraction and classification performance using a hybrid DL model that fused the Regents and FBNet architectures (). Experimental findings on several IDS datasets suggest that a 1:1 ratio is ideal for preventing overfitting and enhancing model interpretability. Showing better accuracy and computational efficiency than SMOTE-based methods, the suggested model attained F1-scores of 0.9758 and 0.9275 on RT-IoT2022 and UNSW Bot-IoT, respectively. An important finding is that compared to traditional oversampling methods, CSMCR cut training time by 53% [23].

Kumar et al. (2024) describe that the principles of DL and ML could be utilised to enhance the procedure for identifying and categorizing network attacks. For instance, Decision Tree, Random Forest, SVM, KNN, MLP, and DNN were all evaluated on the UNSW-NB15 dataset. They show that these models can make IDS much more effective and accurate. The Decision Tree model came in second place since the Deep Neural Network was only slightly behind with an average accuracy of 94.91. They describe performance in terms of the respective strengths and weaknesses of the models. Based on this research, more powerful and viable network intrusion detection systems are created, resulting in additional steps related to cybersecurity development [24].

Balaji et al.(2024) create an IoT network attack detector using a GAN network that combines deep learning with other methods. The complex and dynamic hostile environment of IoT networks makes it impossible to train the model with sufficient data. The main reason for this is that combining intrusion samples with ordinary samples would result in a high number of false positives. Developed a decentralized, dynamic IDS to identify harmful activities. In order to detect harmful behaviors, pre-processing establishes threshold levels. The experimental findings show that compared to earlier algorithms, HDGAN performed better in terms of accuracy (98%), precision (98%), and FPR (95% lower) [25].

Musleh et al. (2023). The declared objective of this work is to provide research on IoT ML intrusion detection systems, with a particular emphasis on various ML-based feature extraction algorithms. This study compared a number of feature extractors, including image filters and TL models like VGG-16 and Dense Net. Furthermore, all of the feature extraction algorithms that were considered were evaluated using a variety of ML approaches, including stacked models, RF, KNN, SVM, and many more. The study used the IEEE Data port dataset to analyse all integrated models. The optimal

results were achieved by combining VGG-16 with stacking, which allowed for a maximum accuracy of 98.3% [26]

Fernando et al. (2023) supervised ML approaches for intrusion detection using gathered network traffic are being studied in this study. The updated balanced dataset (IDSAI) with intrusions generated in attack scenarios is shown here. This newly-provided dataset allows us to compare how well models generalize across datasets. Many supervisory algorithms have proven to be effective in identifying intruders, including XGBoost, Gradient Boosting, DT, RF, and Extra Trees. The algorithms can produce predictions with a selected set of ten intrusions (such as ARP spoofing, ICMP echo request Flood, TCP Null, and others) with a precision of up to 92% after training and evaluation with binary (intrusion or non-intrusion) and multiclass (ten distinct intrusions or non-intrusion) predictions. To the contrary, the Bot-IoT dataset may be effectively used by models trained on the IDSAI dataset to attain a 90% prediction accuracy [27].

Table I summarizes the present status of intrusion detection for Internet of Things (IoT) networks, including recent advances, datasets, important results, and difficulties

TABLE I. SUMMARY OF EXISTING RESEARCH ON MACHINE LEARNING-BASED INTRUSION DETECTION IN IOT NETWORKS

Author(s)	Methodology	Dataset	Key Findings	Limitations	Future Work
Al-Hubaishi & Hachana (2025)	Ensemble-based classification + dimension reduction	TON_IoT	Achieved 98.7% accuracy, 97.5% precision, 96.8% recall; proposed new feature optimization method	Tested only on TON_IoT; potential overfitting to dataset-specific patterns	Extend testing to heterogeneous IoT environments and real-time deployment scenarios
Prasad et al. (2025)	Prasad et al. (2025)	Prasad et al. (2025)	Prasad et al. (2025)	Prasad et al. (2025)	Prasad et al. (2025)
Ali & Al-Sharafi (2025)	LSTM, Random Forest, SVM	Simulated network traffic dataset	Random Forest highest accuracy (94%), LSTM good for sequential data; highlighted ML potential for MitM attacks	SVM had high false positives; limited dataset realism	Analyse real-time MitM detection in IoT networks and investigate hybrid models
Kumar et al. (2024)	Decision Tree, Random Forest, SVM, KNN, MLP, DNN	UNSW-NB15	DL models enhance IDS accuracy and robustness; avg. accuracy 94.91%	The dataset may not cover all IoT-specific attack scenarios	Integrate IoT-specific datasets; evaluate model scalability for large-scale IoT networks
Balaji et al. (2024)	Balaji et al. (2024)	Balaji et al. (2024)	Balaji et al. (2024)	Balaji et al. (2024)	Balaji et al. (2024)
Musleh et al. (2023)	Applying ML techniques (RF, KNN, SVM, stacked models) and feature extraction methods (VGG-16, DenseNet) simultaneously	IEEE Dataport	Best performance with VGG-16 + stacked model (98.3% accuracy)	High computational cost due to deep feature extraction; not tested in real-time IoT	Optimize for lightweight deployment; evaluate energy-efficient solutions for IoT devices
Fernando et al. (2023)	Supervised ML (XGBoost, Gradient Boosting, DT, RF, Extra Trees)	IDSAI, Bot-IoT	Database improves model generalizability; achieved 92% accuracy for multiclass intrusions and 94% accuracy for binary intrusions	Limited attack types and IoT-specific testing; dataset may not fully represent live IoT traffic	Expand dataset diversity; focus on adaptive models against emerging threats in IoT

III. RESEARCH METHODOLOGY

The method for finding intrusions in IoT networks is meant to make them safer by providing fair data representation, improved model training, and strong pre-processing (see Figure 1). The UNSW-NB15 dataset underwent multi-stage pre-processing to provide high-quality and consistent training data. Correcting missing values, duplicates, and outliers was done before using label encoding to transform category variables into numerical form. To enable more efficient training, the data was later normalized between 0 and 1 by minmax normalization. The Butterfly Optimization Algorithm (BOA) enabled us to select the features that maximized performance with the minimum redundancy and retained only the most useful features. This made the dataset skewed because the different attacks were not balanced. To resolve this, a technique called the SMOTE was employed. After initial processing, two subsets were created, one consisting of the training data (80% of the total) and the other of the testing data (20%). The DNN model, which can detect intrusion in IoT networks by seeing small patterns and correlations, was ultimately trained using this cleaned dataset. So, it was feasible to have a forecast that was easy to predict. Regular performance measurements were used to ensure successful Intrusion Detection for better cybersecurity. These metrics included recall, accuracy, precision, F1-score, and ROC curves.

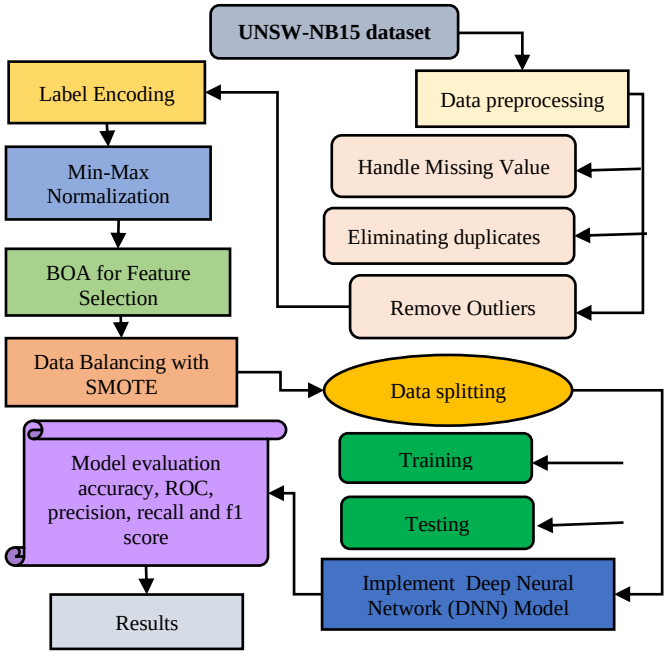

Fig. 1. Proposed Flowchart for Intrusion Detection in IoT Networks

The following is a sequential description of the suggested flowchart used to improve cybersecurity by Intrusion Detection.

A. Data Collection

The UNSW-NB15 dataset, which incorporates both realistic normal traffic and simulated attack behaviours, was created using an ACCS tool named IXIA Perfect Storm. The dataset includes 2,540,044 records, out of which 175,341 are used for training and 82,332 are tested. A total of 49 features, including reaction features (attack class, label), are included in each record. The dataset covers ten attack categories and distinguishes between normal and attack traffic. Data visualizations such as bar plots and heatmaps were used to examine attack distribution, feature correlations etc., are given below:

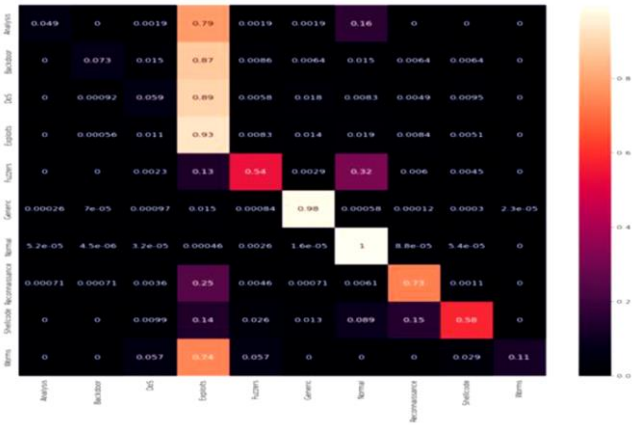

Fig. 2. Heatmap of UNSW-NB15 Dataset

The connection matrix heatmap in Figure 2 demonstrates the associations between several features, with values ranging from 0 to 1. In the graphic representation, lighter shades denote stronger correlations, whereas darker shades denote weaker or nonexistent ones. Some metrics, including "Dst_host_srv_serror_rate" and "Serror_rate" (~0.99) and "Dst_host_serror_rate" and "Serror_rate" (~0.95), demonstrate extremely strong correlations, indicating possible redundancy among them. In contrast, most other feature pairs display low correlation values near zero, reflecting minimal linear dependency. This analysis is particularly useful for detecting multicollinearity, helping to identify features that may require removal or dimensionality reduction during preprocessing to enhance model efficiency and performance.

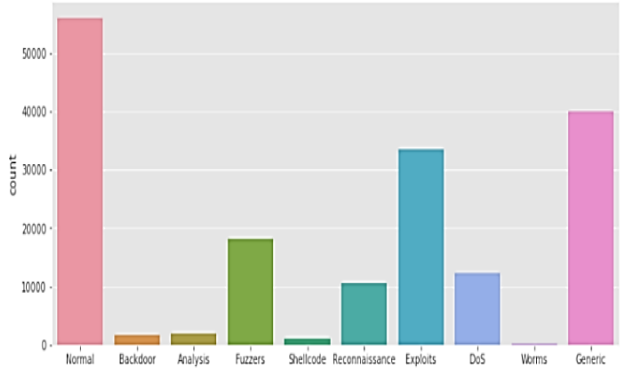

Fig. 3. Visualization of the Distribution of the Attack Class

This data collection is probably associated with cybersecurity, since Figure 3 is a bar chart showing the distribution of various kinds of network traffic. A few examples of the traffic classifications shown on the x-axis are Generic, Exploits, DOS, Backdoor, Analysis, Fizzers, Shellcode, Reconnaissance, and Fuzzers. The number of each class is shown on the y-axis. The Normal traffic is the most frequent, with a count of over 50,000. Generic and Exploits also feature rather frequently with over 30,000 counts. Fizzers and DoS are less common but still of high significance, with categories such as Worms, Shellcode, Backdoor, and Analysis having a very low count, meaning that this type of traffic in this data set.

B. Data Pre-Processing

The data preparation and pre-processing of data sets improve data quality, integrity, and consistency. It was cleaned up by taking the gaps in values, the duplications and the outliers. Also, data transformation and normalization were performed. The summary of the most critical preprocessing steps is shown below:

- **Managing missing values:** Ensuring data quality and the dependability of subsequent studies or model development requires careful management of missing values, which is an essential part of data cleaning. The data set's characteristics and the kind of missing data dictate the appropriate method(s) to use.
- **Eliminating duplicates:** To remove these duplicates, you can even take advantage of the built-in features of apps such as Excel or Google Sheets by selecting data, and using the Data tab, which offers the option of clicking the Remove Duplicates button and removing the duplicate rows or entries.
- **Remove Outliers:** An "outlier removal" is a method for cleaning up a dataset by eliminating exceedingly unusual individual data points.

C. Label Encoding

Using Labels for Storage ML data preparation technique for numerical encoding of categorical variables. This must be changed because of the numerical character of the majority of the ML algorithms. One method for making numerical data usable in a ML system out of categorical data is label encoding.

D. Min-Max Normalization

The minmax technique was used to normalize records to make sure that the values are all in the range of 0 to 1. This was to both optimize the performance of the classifiers used and to mitigate the effect of outliers [2]. It was normalized using the following mathematical Equation (1):

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

The original feature value is represented by X, the normalized value by X' , and the minimum and maximum feature values are X_{min} and X_{max}, Respectively.

E. Butterfly Optimization Algorithm (BOA) for Feature Selection

The goal of ML feature selection is to choose the most interesting variables to enhance accuracy, on cost and interpretation. Butterfly Optimization Algorithm (BOA) is a foraging and mating algorithm based on butterflies and widely applied in feature selection. Applying BOA to network traffic analysis allows one to prioritise features for optimal detection performance. Maximum packet length, total forward IAT,

mean forward flow IAT, forward URG flags, average forward segment size, starting forward win bytes, active max, and idle max are all part of forward IAT.

F. Data Balancing using SMOTE

Data balancing involves making adjustments to a dataset's class distribution to make it more representative of the classes. When one class (the majority class) has a lot more samples than the other classes (the minority classes), this characteristic is especially helpful in ML classification challenges. The performance of prediction models could be negatively impacted by these imbalances, which could cause them to be biased towards the dominant class. The SMOTE is the go-to method for this issue since it improves model performance by combining more minority group data, which in turn makes the representation of different classes more equitable.

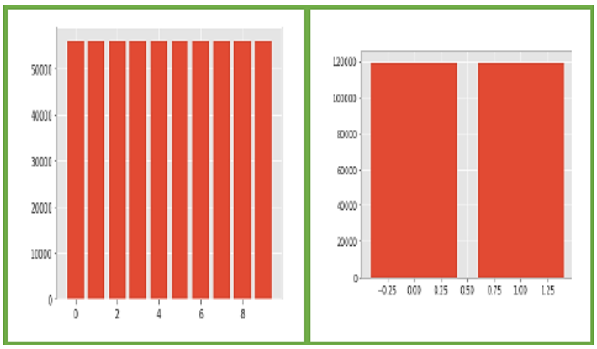

Fig. 4. Frequency Distribution After Resampling Data for Attack-Class(left) and Label(right)

Figure 4 shows two side-by-side bar charts. The left chart represents a distribution in ten categories (0 9), the frequencies here are almost equal, each bar is slightly above 50,000, indicating a well-balanced dataset in these classes. The chart on the right illustrates a binary distribution with two categories, both having almost identical frequencies of approximately 120,000 each. Together, these plots indicate that the dataset is well-balanced in terms of class distribution, both for the multi-class labels (0–9) and the binary classification labels.

G. Data Splitting

A dataset was comprised of two parts: a training set and a testing set. 80% of the data was set aside for the model's training and parameter estimation, while 20% was kept for the model's testing and performance evaluation.

H. Proposed Deep Neural Network (DNN) Model

The DNN algorithm is well-known in the academic community as a DL tool. A CNN has completely connected input, hidden, and output layers [28]. Each neuron is linked to all the neurons in the layer below it, but not to any neurons in the layer above or below it. The output is affected by an activation function after each network layer, which enhances the impact of network learning. In this way, DNN is similar to a huge perceptron that combines several perceptions. Consider the following Equation for the forward propagation calculation of the ith layer (2):

$$x^{(i+1)} = \sigma\left(W^{(i)}x^{(i)} + b^{(i)}\right) \quad (2)$$

When the input value is x, the bias vector is b, and the weight coefficient matrix is w [29]. A well-known hidden layer activation function is the ReLU Equation (3). A SoftMax activation is commonly used in the output layer of multi-class classification models.

$$\sigma(x) = max(0, x) \quad (3)$$

The structure of the network is determined by the loss function, which optimizes the network's backpropagation by assessing the output loss of training samples. The following formula describes the most common loss function used in classification tasks: cross-entropy Equation (4):

$$C = -\frac{1}{N}\sum_x [ylog(p) + (1-y)\log(1-p)] \quad (4)$$

There are three variables: p for the positive class's predicted probability, N for the number of training samples, and y for the true class label (1 or 0). The activation layer (ReLU), batch size (32), Adam optimiser (0.001), and number of epochs (4,000) were the settings that were set up for the DNN. For binary classification, used cross-entropy loss; to avoid overfitting, used dropout = 0.2. Optimal training was guaranteed by early stopping based on validation loss and enhanced stability during initialization.

I. Evaluation Metrics

An integral aspect of developing a successful ML model is doing model evaluations. Various measures can be used for evaluation. To measure how well the system worked, used the following metrics. Four possible permutations of anticipated and measured values are presented in the tabular form The results were marked as follows TP for correctly classified attacks, TN for correctly categorised innocuous traffic, FN for misclassified attacks, and FP for misclassified benign traffic. Below, go over the following matrix, which includes recall, accuracy, precision, and F1-score:

Accuracy: Accuracy, defined as the proportion of observations that are properly predicted relative to the total number of observations, is an essential performance indicator [16]. It was. Here it is: Equation (5)-

$$Accuracy = \frac{\text{TP+TN}}{\text{TP+Fp+TN+FN}} \quad (5)$$

Precision: Precision is a measure of how many positive values out of a total of positive values are actually expected to be true. The accuracy with which the classifier predicts positive classifications is denoted as Equation (6)-

$$Precision = \frac{\text{TP}}{\text{TP+FP}} \quad (6)$$

Recall: Recall is a measure that looks at how many positive values out of a total of positive values were correctly anticipated. The formula for it in mathematics is Equation (7)-

$$Recall = \frac{\text{TP}}{TP+FN} \quad (7)$$

F1 score: An F1 score is a harmonic mean of a classification problem's recall and precision. The formula for it in mathematics is Equation (8)-

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (8)$$

ROC-AUC Curve: The ROC curve indicates how well a classification task is doing at different threshold levels. AUC is a separability quantifier, unlike ROC, which is a curve of probability. It demonstrates the ability of the model to differentiate in categories. The positions of TPR and FPR are indicated along the y-axis of the ROC curve and x-axis respectively as represented in Equation (9-10).

$$FPR = \frac{FP}{TP+FP} \quad (9)$$

$$TPR = \frac{TP}{TP+FP} \quad (10)$$

A general notion of the model's efficacy is conveyed by these evaluation metrics.

IV. Results and Discussion

This section shows how the experiments were set up and what the results were for training and testing the suggested model based on the given needs Hardware requirements Equipped with 2 TB of RAM, an Intel Core i9-9820X 3.30 GHz processor, and the Ubuntu 20.04.1 LTS operating system. By the Jupyter notebook conda, the Python scripts have been coded. The findings of training the model on UNSW-NB15 are presented in Table II. The model's performance evaluation was conducted using the primary performance metrics—F1-score, recall, accuracy, and precision—and the results are displayed in Table II. The experimental results showed that the suggested DNN intrusion detection system for the IoT network worked better than any other alternatives. The model achieved an amazing precision of 98.73% in relation to appropriate categorization of normal as well as attack traffic. The DNN model is 97.73% precise and 97.94% accurate, which is a satisfactory trade-off between minimizing false-positives and the intrusion detection rate. Furthermore, the model's efficacy in bolstering cyberspace safety in the IoT environment is demonstrated by the 97.82% of the F1-score, which further demonstrates the model's power and validity.

TABLE II. Experiment Results of Proposed Models for Intrusion Detection in IoT Networks on UNSW-NB15 dataset

Performance matrix	Deep Neural Network (DNN) Model
Accuracy	98.73
Precision	97.73
Recall	97.94
F1-score	97.82

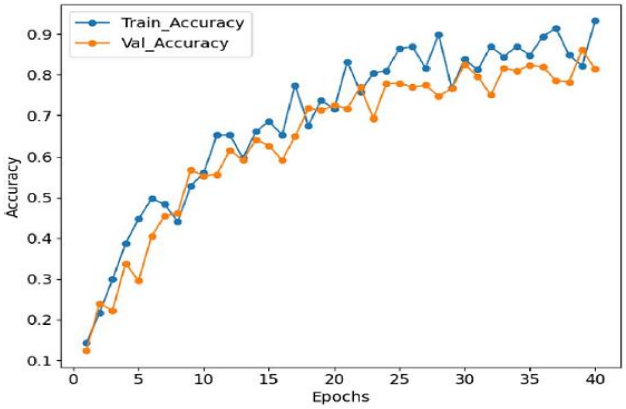

Fig. 5. Accuracy Curves for the DNN Model

Figure 5 shows the training and validation accuracy graphs, which show how the suggested model performed after 40 epochs. The blue line represents training accuracy from an initial value of around 0.1 to an end value of about 0.9, while the orange line reflects validation accuracy. Both lines exhibit a propensity to continuously grow. The two curves are very close, which means that the model is not overfitting significantly since validation accuracy always follows the training accuracy. At the final stage of the training process, both accuracies exceed 0.85 and it can be concluded that strong generalization capability is achieved and the model can be considered effective in intrusion detection.

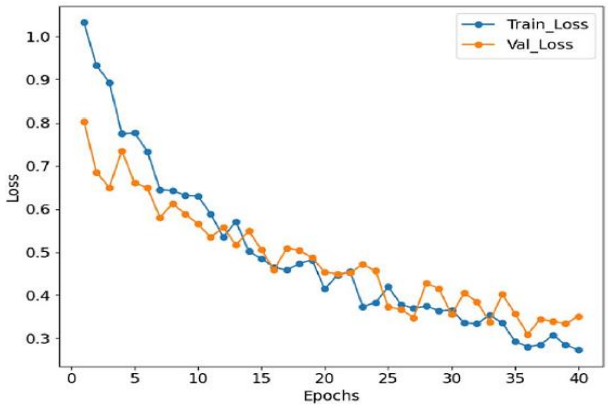

Fig. 6. Loss Curves for the DNN Model

Figure 6 shows the loss curves for training and validation, which show that the model learnt across 40 epochs. A training loss of around 1.0 and a validation loss of about 0.8 are shown by the blue and orange lines, respectively, and these reflect rather significant initial training and validation losses. Consistent losses as training progresses indicate effective learning and reduced error rates. In the last epochs, training loss decreases to less than 0.3 and validation loss levels off at a comparable range, with high-level consistency between the two. The model's robustness and reliability for intrusion detection are confirmed by the near convergence of the curves, which show that it generalizes well without considerable overfitting.

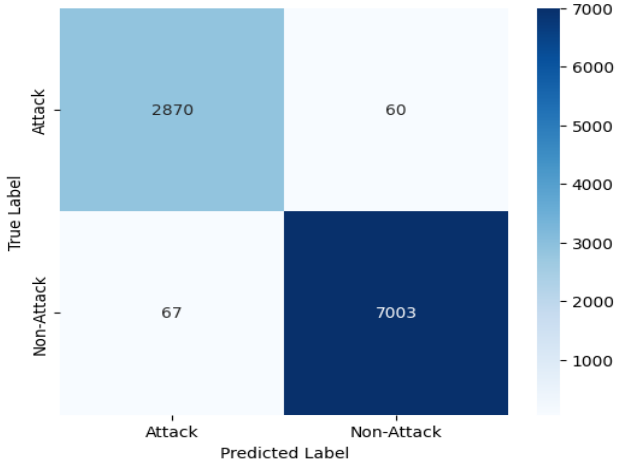

Fig. 7. Confusion Matrix for DNN

A confusion matrix, which illustrates the efficiency of a categorization model, is displayed in Figure 7. One category is Attack, while the other is Non-Attack. The predicted labels are shown in the columns, while the actual labels are displayed in the rows. The top-left cell shows that 2,870 instances were correctly predicted as "Attack," representing true positives. The bottom-right cell shows that 7,003 instances were correctly predicted as "Non-Attack," representing true negatives. The model erred in two ways: first, it falsely classed sixty "Attack" cases as "non-attack" (false negatives); second, it falsely identified sixty-seven "non-attack" instances as "Attack" (false positives).

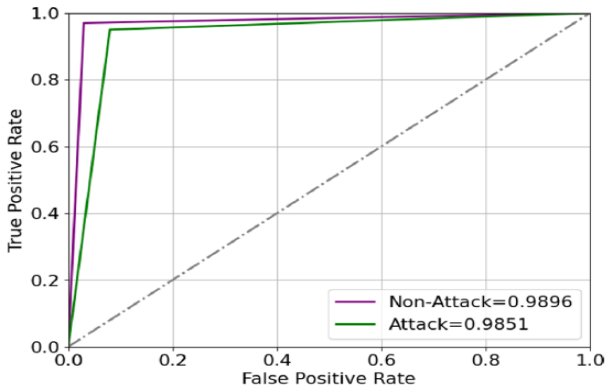

Fig. 8. ROC Analysis of the DNN Model

Figure 8 shows the suggested model's ROC curve performance for attack and non-attack classification. The two curves lie near the top-left corner implying a high level of detection with a few false positives. The non-attack AUC is 0.9896 and attack is 0.9851, so both classes can be considered as having excellent discriminative power. The model's efficacy, reliability, and robustness are shown by its high AUC values, which allow for a high TP rate and a very low FP rate.

A. Comparative Analysis

The purpose of this section is to compare several ML models that have the potential to detect breaches in IoT networks. All four models—SVM, LSTM, ET, and DNN—have their aggregate accuracy, recall, precision and F1-score shown in Table III Of all the models tested, the DNN model had the highest accuracy rate (98.73%) in identifying intrusions in IoT networks, as shown in Table III. Extra Trees (ET) was superior to both SVM performance and LSTM performance, with accuracy of 97.9%, 89% and 90.06% respectively. The effectiveness of DNN has been constantly higher than that of other models, which indicates its power and resilience, and this makes it possibly the most promising structure to implement as an intrusion detector in real-time IoT and as an additional defense measure against cybercrime.

TABLE III. PERFORMANCE COMPARISON OF DIFFERENT PREDICTIVE MODELS OF INTRUSION DETECTION IN IOT NETWORKS USING THE UNSW-NB15 DATASET

Models	Accuracy	Precision	Recall	F1-Score
SVM[30]	89	91	86	85
LSTM[31]	90.06	92.18	75.28	82.87
ET[32]	97.9	97.71	97.73	97.73
DNN	98.73	97.73	97.94	97.82

The proposed DNN model outperforms all of the competition with an astounding accuracy of 98.73%, making it a very practical choice. The accuracy also guarantees that it can properly identify the network traffic to the relevant category and thus, minimize the frequency of false identifications, and deliver a better trusted intrusion detection technology.. The DNN is a superior solution to augmenting cybersecurity with the UNSW-NB15 dataset because it achieves this level of accuracy. This type of high accuracy indicates the strength of the model when faced with various attack patterns. This fact further positions DNN as a promising approach for creating next-gen intrusion detection systems.

V. CONCLUSION AND FUTURE STUDY

Traditional intrusion detection systems aren't always up to the task of preventing the increasingly complex and ever-changing threats that target the IoT. The proposed framework overcame this issue by combining Butterfly Optimization Algorithm (BOA) with a Deep Neural Network (DNN), which allows selecting features effectively and classifying them efficiently. The optimized model achieved remarkable results in testing on the UNSW-NB15 dataset, boasting a recall of 97.94, an F1-score of 97.82, a precision of 97.73, and an accuracy of 98.73. These metrics reflect the ability of the framework to detect a broad spectrum of attacks with the greatest degree of precision and generate false positives at the lowest possible rate. Not only did the models, which included the BOA-dNN, show high performance to increase the classification capacity of the models, but they could also be calculated and have been shown to scale to large-scale IoT systems. These findings present the framework as a reliable and flexible method to enhance the cybersecurity of IoT and further contribute significantly to the research on smart intrusion detection.

Further research can be carried out to confirm the performance on heterogeneous data, design light versions of the model that can be executed on IoT devices with resource limitations and incorporate privacy-protection features such as federated learning. Live IoT network deployment and testing will also be necessary in real-time to test resilience against emerging and zero-day threats. Further, efficiency and detection rates can also be optimized using DL-based hybrid optimization methods.

REFERENCES

[1] K. Seetharaman, “Incorporating the Internet of Things (IoT) for Smart Cities: Applications, Challenges, and Emerging Trends,” *Asian J. Comput. Sci. Eng.*, vol. 08, no. 01, pp. 8–14, 2023, doi: 10.22377/ajcse.v8i01.199.

[2] Y. A. Qadri, A. Nauman, Y. Bin Zikria, A. V. Vasilakos, and S. W. Kim, “The Future of Healthcare Internet of Things: A Survey of Emerging Technologies,” *IEEE Commun. Surv. Tutorials*, vol. 22, no. 2, pp. 1121–1167, 2020, doi: 10.1109/COMST.2020.2973314.

[3] N. Yang, L. Wang, G. Geraci, M. Elkashlan, J. Yuan, and M. Di Renzo, “Safeguarding 5G wireless communication networks using physical layer security,” *IEEE Commun. Mag.*, vol. 53, no. 4, pp. 20–27, Apr. 2015, doi: 10.1109/MCOM.2015.7081071.

[4] S. B. Shah, B. Boddu, N. Prajapati, and S. A. Pahune, “AI-Powered Advanced Intrusion Detection for Securing Cloud Environments Against Network Attacks,” in *2025 Global Conference in Emerging Technology (GINOTECH)*, IEEE, May 2025, pp. 1–7. doi: 10.1109/GINOTECH63460.2025.11076673.

[5] N. K. Prajapati, “Federated Learning for Privacy-Preserving Cybersecurity: A Review on Secure Threat Detection,” *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 5, no. 4, pp. 520–528, Apr. 2025, doi: 10.48175/IJARSCT-25168.

[6] D. Patel, “Leveraging Blockchain and AI Framework for Enhancing Intrusion Prevention and Detection in Cybersecurity,” *Tech. Int. J. Eng. Res.*, vol. 10, no. 6, pp. 853–858, 2023, doi: 10.56975/tijer.v10i6.158517.

[7] D. D. Rao, A. A. Waoo, M. P. Singh, P. K. Pareek, S. Kamal, and S. V. Pandit, “Strategizing IoT Network Layer Security Through Advanced Intrusion Detection Systems and AI-Driven Threat Analysis,” *J. Intell. Syst. Internet Things*, vol. 24, no. 2, pp. 195–207, 2024, doi: 10.54216/JISIoT.120215.

[8] H. Mliki, A. Kaceam, and L. Chaari, “A Comprehensive Survey on Intrusion Detection based Machine Learning for IoT Networks,” *ICST Trans. Secur. Saf.*, vol. 8, no. 29, p. 171246, Nov. 2021, doi: 10.4108/eai . 6-10-2021.171246.

[9] Z. Chen *et al.*, “Machine Learning-Enabled IoT Security: Open Issues and Challenges Under Advanced Persistent Threats,” *ACM Comput. Surv.*, vol. 55, no. 5, May 2023, doi: 10.1145/3530812.

[10] S. A. Pahune, P. Matapurkar, S. Mathur, and H. Sinha, “Generative Adversarial Networks for Improving Detection of Network

Intrusions in IoT Environments," *2025 4th Int. Conf. Distrib. Comput. Electr. Circuits Electron.*, pp. 1–6, 2025, doi: 10.1109/ICDCECE65353.2025.

[11] V. Shah, "Analyzing Traffic Behavior in IoT-Cloud Systems : A Review of Analytical Frameworks," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 9, no. 3, pp. 877–885, 2023, doi: https://doi.org/10.32628/IJSRCSEIT.

[12] V. Shah, "Traffic Intelligence In Iot And Cloud Networks: Tools For Monitoring, Security, And Optimization," *Int. J. Recent Technol. Sci. Manag.*, vol. 9, no. 5, 2024, doi: 10.10206/IJRTSM.2025894735.

[13] I. H. Sarker, A. S. M. Kayes, S. Badsha, H. Alqahtani, P. Watters, and A. Ng, "Cybersecurity data science: an overview from machine learning perspective," *J. Big Data*, 2020, doi: 10.1186/s40537-020-00318-5.

[14] N. Patel, "AI-Powered Intrusion Detection and Prevention Systems in 5G Networks," in *2024 9th International Conference on Communication and Electronics Systems (ICCES)*, IEEE, Dec. 2024, pp. 834–841. doi: 10.1109/ICCES63552.2024.10859892.

[15] N. Prajapati, "The Role of Machine Learning in Big Data Analytics: Tools, Techniques, and Applications," *ESP J. Eng. Technol. Adv.*, vol. 5, no. 2, 2025, doi: 10.56472/25832646/JETA-V5I2P103.

[16] S. Mohammadi, H. Mirvaziri, M. Ghazizadeh-Ahsaee, and H. Karimipour, "Cyber intrusion detection by combined feature selection algorithm," *J. Inf. Secur. Appl.*, vol. 44, pp. 80–88, Feb. 2019, doi: 10.1016/j.jisa.2018.11.007.

[17] A. R. Bilipelli, "AI-Driven Intrusion Detection Systems for Large-Scale Cybersecurity Networks Data Analysis : A Comparative Study," *TIJER – Int. Res. J.*, vol. 11, no. 12, pp. 922–928, 2024.

[18] S. R. Janmejaya Mishra, Dr. Heena Kousar, Mrs.Sonali Hitesh Raut, Mr. Arivumani Samson Subramanian, Dr. Kadli Nanjundeshwara, Mrs. Kavitha Datchanamoorthy, Mr. Gowtham Semalaiappan, "AI-based network intrusion detection device," Design Registration No. 6426644 (UK IPO), 2025

[19] R. Q. Majumder, "Machine Learning for Predictive Analytics : Trends and Future Directions," *Int. J. Innov. Sci. Res. Technol.*, vol. 10, no. 4, 2025.

[20] B. Yadav, D. D. Rao, Y. Mandiga, N. S. Gill, P. Gulia, and P. K. Pareek, "Systematic Analysis of threats, Machine Learning solutions and Challenges for Securing IoT environment," *J. Cybersecurity Inf. Manag.*, vol. 14, no. 2, pp. 367–382, 2024, doi: 10.54216/JCIM.140227.

[21] H. Mistry, K. Shukla, and N. Patel, "Transforming Incident Responses, Automating Security Measures, and Revolutionizing Defence Strategies through AI-Powered Cybersecurity," *J. Emerg. Technol. Innov. Res.*, vol. 11, no. 3, p. 25, 2024.

[22] M. A. Ali and S. A. H. Al-Sharafi, "Intrusion detection in IoT networks using machine learning and deep learning approaches for MitM attack mitigation," *Discov. Internet Things*, vol. 5, no. 1, pp. 1–13, 2025.

[23] A. Prasad, W. Mohammad Alenazy, N. Ahmad, G. Ali, H. A. Abdallah, and S. Ahmad, "Optimizing IoT intrusion detection with cosine similarity-based dataset balancing and hybrid deep learning," *Sci. Rep.*, vol. 15, no. 1, pp. 1–24, 2025, doi: 10.1038/s41598-025-15631-3.

[24] G. Kumar, P. Gupta, G. K. Yadav, R. Verma, J. P. Bhati, and V. S. Bhakuni, "Evaluating the Effectiveness of Deep Learning Models in Network Intrusion Detection," in *2024 International Conference on Cybernation and Computation (CYBERCOM)*, 2024, pp. 766–771. doi: 10.1109/CYBERCOM63683.2024.10803243.

[25] S. Balaji, G. Dhanabalan, C. Umarani, and J. Naskath, "A GAN-based Hybrid Deep Learning Approach for Enhancing Intrusion Detection in IoT Networks," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 6, pp. 348–354, 2024, doi: 10.14569/IJACSA.2024.0150637.

[26] D. Musleh, M. Alotaibi, F. Alhaidari, A. Rahman, and R. M. Mohammad, "Intrusion Detection System Using Feature Extraction with Machine Learning Algorithms in IoT," *J. Sens. Actuator Networks*, vol. 12, no. 2, 2023, doi: 10.3390/jsan12020029.

[27] G.-P. Fernando, A.-A. H. Brayan, A. M. Florina, C.-B. Liliana, A.-M. Héctor-Gabriel, and T.-S. Reinel, "Enhancing Intrusion Detection in IoT Communications Through ML Model Generalization With a New Dataset (IDSAI)," *IEEE Access*, vol. 11, pp. 70542–70559, 2023, doi: 10.1109/ACCESS.2023.3292267.

[28] S. Nokhwal, P. Chilakalapudi, P. Donekal, S. Nokhwal, S. Pahune, and A. Chaudhary, "Accelerating Neural Network Training: A Brief Review," *ACM Int. Conf. Proceeding Ser.*, pp. 31–35, 2024, doi: 10.1145/3665065.3665071.

[29] S. Nokhwal, S. Nokhwal, S. Pahune, and A. Chaudhary, "Quantum Generative Adversarial Networks: Bridging Classical and Quantum Realms," in *2024 8th International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence (ISMSI)*, New York, NY, USA: ACM, Apr. 2024, pp. 105–109. doi: 10.1145/3665065.3665082.

[30] A. Khatib, M. Hamlich, and D. Hamad, "Machine Learning based Intrusion Detection for Cyber-Security in IoT Networks," *E3S Web Conf.*, vol. 297, pp. 1–7, 2021, doi: 10.1051/e3sconf/202129701057.

[31] S. A. Elsaid, E. Shehab, A. M. Mattar, A. T. Azar, and I. A. Hameed, "Hybrid intrusion detection models based on GWO optimized deep learning," *Discov. Appl. Sci.*, vol. 6, no. 10, p. 531, Oct. 2024, doi: 10.1007 /s42452-024-06209-1.

[32] M. A. Talukder *et al.*, "Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction," *J. Big Data*, vol. 11, no. 1, p. 33, Feb. 2024, doi: 10.1186/s40537-024-00886-w.

