Journal of Basics and Applied Sciences Research (JOBASR)
ISSN (print): 3026-9091, ISSN (online): 1597-9962
Volume 4(5) September 2026
DOI: https://dx.doi.org/10.4314/jobasr.v4i5.2

Network Intrusion Detection System Using Advanced Feature Selection and Optimized Deep Neural Networks for Attack Classification

Abubakar Abdulkadir[1*], Umar Iliyasu[2] & Zaharraddeen Sani[3]
[1]Department of Computer Science Education Isah Kaita College of Education P.M.B 5007, Dutsin-ma Katsina State
[2,3]Department Of Computer Science and IT Faculty of Computing and artificial Intelligence Federal University Dutsin-ma Katsina State, Nigeria
[*]Corresponding Author Email: aakado123@gmail.com

ABSTRACT
Recent advancements in network technologies have posed significant challenges to the performance of many existing Network Intrusion Detection Systems (NIDS). Most of these systems struggle to effectively integrate advanced feature selection methods, data imbalance handling techniques, and optimized deep neural network architectures, resulting in suboptimal classification performance. This research proposes novel Network Intrusion Detection System that integrates advanced feature selection, method (wrapper based techniques) data balancing method and an optimized deep neural network (DNN) architecture using bayesia optimization techniques to enhance detection accuracy and overall system robustness. The proposed NIDS was evaluated using two widely recognized benchmark datasets namely NSL-KDD and CIC-IDS2017—and compared against four intrusion detection models developed using state-of-the-art feature selection techniques as well as other existing benchmark models. Experimental results reveal that the proposed model significantly outperforms the compared systems across key evaluation metrics by achieving accuracy improvement of greater than 0.05 value,along side near perfect specifically 0.9674 on NSL-KDD and 0.9994 precision, recall, F1-score, demonstrating its superior capability in detecting and classifying network intrusions under diverse conditions.

Keywords:
IDS,
NIDS,
IoT,
Dos,
DDos

INTRODUCTION

In today's digital era, the rapid advancement of information and communication technologies has revolutionized how individuals, organizations, and governments operate. With the proliferation of network infrastructures and the exponential growth of connected devices—including computers, smartphones, servers, and Internet of Things (IoT) devices—global dependence on digital systems has increased dramatically. However, this technological expansion has also introduced new and complex cybersecurity challenges. The increasing interconnectivity and data exchange between devices have created a larger attack surface for malicious actors to exploit. As a result, securing modern network environments against evolving cyber threats has become an urgent and ongoing priority. Network Intrusion Detection Systems (NIDS) have emerged as a critical line of defense in safeguarding organizational networks against unauthorized access, data breaches, and malicious activities.

A NIDS functions as a surveillance mechanism that continuously monitors network traffic, analyzes patterns, and identifies signs of anomalous or suspicious behavior that may indicate a cyberattack. By doing so, it helps maintain the fundamental principles of cybersecurity integrity, confidentiality, and availability which are essential for the protection of sensitive data and the smooth operation of network-based services. However, despite their importance, traditional NIDS solutions have encountered significant challenges in adapting to the continuously evolving nature of cyberattacks. With the rapid increase in both the volume and complexity of network attacks, such as Denial of Service (DoS), Distributed Denial of Service (DDoS), IP spoofing, eavesdropping, SQL injection, and malware infections, traditional intrusion detection techniques often struggle to deliver the required accuracy and responsiveness. As noted by Snehal and Jadhav (2020), these legacy systems tend to produce high false positive rates, slow detection times,

and low classification accuracy, especially when confronted with large-scale, sophisticated, or previously unseen attack patterns. The primary limitation of these NIDS lies in their static rule-based or signature-based detection mechanisms. These systems depend on predefined patterns or signatures extracted from known attack behaviors. Consequently, they are effective only against previously identified threats but fail to recognize zero-day attacks or mutated malware that do not match existing patterns. This lack of adaptability severely limits their capability to detect new or evolving intrusions. Moreover, such systems require frequent manual updates to maintain their detection rules, which is both time-consuming and prone to human error. Another critical challenge is the high-dimensional and dynamic nature of network traffic data. Modern network environments generate massive volumes of data from multiple sources, including logs, routers, switches, and endpoint devices. Each data packet may contain numerous features representing IP addresses, ports, timestamps, protocols, payloads, and more. The complexity of this multidimensional data makes it extremely difficult for conventional algorithms to efficiently extract relevant features or distinguish between benign and malicious activities. The resulting computational overhead leads to slower detection speeds, missed attacks, and increased false alarms, ultimately reducing the reliability of the system. Additionally, the lack of integration between software and hardware security measures contributes to system vulnerabilities. In many cases, weak security designs, software bugs, or deliberate backdoors embedded in hardware components expose networks to potential exploitation. These vulnerabilities can bypass detection layers, allowing attackers to infiltrate systems unnoticed. Consequently, there is a growing need for intelligent intrusion detection frameworks that can adapt dynamically to changes in network behavior, analyze massive data volumes efficiently, and maintain low false alarm rates while ensuring real-time responsiveness. To overcome the inherent limitations of this traditional intrusion detection systems, researchers have increasingly turned to Machine Learning (ML) and Deep Learning (DL) methodologies. These data-driven approaches possess the ability to learn complex patterns, model non-linear relationships, and generalize from historical data to detect new or previously unseen attacks. Machine learning algorithms such as Support Vector Machines (SVM), Decision Trees (DT), K-Nearest Neighbor (KNN), and Random Forests (RF) have been successfully applied to classify normal and malicious network traffic with improved accuracy compared to signature-based systems. However, while ML-based models improve adaptability and automation, they still face challenges, particularly when dealing with large-scale and high-dimensional network datasets. Most ML models depend heavily on manual feature engineering, where the effectiveness of detection relies on the quality of selected features. Moreover, as the complexity of network environments grows, these models struggle to automatically extract meaningful representations from raw data, limiting their scalability and real-time performance in dynamic network environments. To address these shortcomings, Deep Learning (DL) has emerged as a promising solution. DL architectures such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Deep Neural Networks (DNN) are capable of learning hierarchical data representations directly from raw input without the need for extensive manual feature extraction. This allows for better detection of subtle, non-linear, and high-dimensional attack patterns. DL-based NIDS can thus offer improved accuracy, robustness, and scalability compared to traditional and ML-based systems. Nevertheless, deep learning models are not without their own limitations, including high computational cost, overfitting, slow convergence, and limited interpretability when applied to complex network datasets. Feature selection plays a vital role in enhancing the efficiency and effectiveness of deep learning-based intrusion detection systems. Since network datasets often contain redundant, irrelevant, or noisy features, applying advanced feature selection techniques helps in reducing expunge input dimensionality, improving model generalization, and minimizing training time. Among various optimization-based feature selection methods, metaheuristic algorithms have shown strong potential due to their ability to explore and exploit search spaces efficiently. Recent research has demonstrated the effectiveness of combining deep learning with optimization-based feature selection techniques. For instance, algorithms such as the Binary Particle Swarm Optimization (BPSO) and the Cuckoo Search Algorithm (CSA) have been used to identify the most informative subsets of features from large-scale network traffic data. These algorithms mimic natural behaviors—such as the swarm intelligence of birds (BPSO) or the brood parasitism of cuckoos (CSA)—to iteratively optimize the feature selection process. By selecting the most relevant attributes, they not only improve the accuracy of intrusion detection but also reduce computational costs and enhance real-time detection capabilities (Naveed et al., 2022; Raja & Pran, 2023). Such hybrid models that integrate optimization algorithms with deep neural architectures have been shown to outperform traditional systems and machine learning based systems in various metrics, including detection rate, precision, recall, and false positive rate. They offer a dynamic and adaptive approach that can handle both known and unknown attack types while maintaining scalability in complex and heterogeneous network environments.

However, this study seeks to design and develop a real-time Network Intrusion Detection System (NIDS) that

leverages advanced feature selection mechanisms, data balancing techniques and optimized deep learning architectures to achieve superior detection accuracy, faster processing speed, and improved classification of diverse network attacks. The proposed system will incorporate an improved Fruit Fly Optimization Algorithm with dimensional search control and memory-based mechanisms.for selecting the most discriminative features, thereby reducing input dimensionality and computational load. The optimized Deep Neural Network (DNN) will then be trained on the selected features to perform accurate, multi-class classification of network intrusions. By integrating adaptive learning mechanisms and efficient feature selection, the system aims to minimize false alarms, enhance generalization across multiple datasets, and ensure scalability for real-world deployment. Ultimately, the research contributes to the advancement of intelligent intrusion detection by bridging the gap between computational intelligence, feature optimization, and deep learning. The proposed model aspires to provide a robust, high-accuracy, and low-latency intrusion detection solution capable of addressing the increasing complexity of modern cyber threats and supporting secure, resilient network infrastructures in the digital age.

Zhao & Wang (2023) proposes transformer-based models for real-time NIDS; strong against adversarial threats also,Ding et al. (2022) proposes Hybrid architectures with synthetic augmentation; effective for imbalanced datasets. Alandjani (2024) – Lightweight quantized neural networks; scalable and efficient for IoT security.Jay (2023) proposes ANN on NSL-KDD for DDoS detection; 99.99% accuracy but limited to DDoS only. Waad (n.d.) proposes Hybrid CNN-Dense + Naïve Bayes model evaluated on UNSW-NB15; achieving 99.8% accuracy but lacks feature selection and cross-dataset generalization. Edosa et al. (2024) proposes an intrusion detection system achieving High accuracy but limited to few attack types/datasets; poor generalizability. Vanlalruata (2022) proposes a Deep learning model which was evaluated on CICIDS2017/2018 with L1/L2 regularization; high accuracy but high computational cost and imbalance issues.Vinayakumar et al. (2023) also proposes IoMT IDS with network flow + biometrics; achieving 99% accuracy but misclassifies non-IoMT data and needs cross-attack improvement. Zahra et al. (2023) APT-focused IDS with low power/e-waste design; promising but lacks standard dataset validation and performance metrics. João et al. (2023) proposes LSTM IDS evaluated on CICIDS2017; with low false rates but needs runtime and time complexity optimization. Lirim (2021) proposes Adaptive CNN IDS with semi-dynamic hyperparameter tuning; achieving 95.6% on UNSW-NB15; future work on zero-day attacks and feature reduction. Yusuf et al. (2022) proposes ANN + GridSearchCV IDS evaluated on CICIDS2017; achieving near-perfect accuracy but limited attack diversity. Nadir et al. (2023) proposes Firefly optimization + PNN which was evaluated on KDD99; outperforming traditional models but needs broader threat coverage.Alaeddine et al. (2020) proposes LSTM IDS evaluated on NSL-KDD; with 99.9% accuracy values; future work on diverse attack scenarios.Laghrissi et al. (2021) proposes LSTM + PCA/MI and evaluate on KDD99; realizing strong performance; suggests exploring LSTM variants as future work.Yesi et al. (2021) proposes Deep autoencoders + DNN IDS and evalauted on CSECIC-ID2018 and NSL-KDD; realzing good results; future work focust on other architectures. Bhuvaneshwari et al. (2021) proposes CNN + meta-heuristic feature selection evaluated on NSL-KDD; achieving 98.71% accuracy; future work focus on smaller datasets.Safi et al. (2022) proposes Deep CNN IDS and evaluated on IoTiD20; achieving 99.89% binary accuracy; needs multi-class robustness as future work.Alghamdi & Bellaiche (2023) proposes Deep ensemble (LSTM+CNN+ANN) with Lambda architecture IDS which was evalauted on IoT-23; 99.93% accuracy; suggests real-world validation as future work.Dhiaa et al. (2023) proposes ML-based IoT IDS with VGG-16, DenseNet, KNN, SVM; VGG-16 best (98%); used SMOTE for imbalance; ignores time/space complexity.

MATERIALS AND METHODS

Data Description and Preprocessing

The study utilizes two widely recognized benchmark datasets—NSL-KDD and CIC-IDS2017 to ensure a comprehensive and reliable evaluation of the proposed Network Intrusion Detection System (NIDS). These datasets were selected for their ability to represent diverse network traffic patterns and attack behaviors, thereby enhancing the assessment of the model's performance and generalization capability.The NSL-KDD dataset, introduced as an enhanced version of the original KDD'99 dataset, addresses key issues such as redundancy and class imbalance. It provides a more balanced and representative sample distribution, improving the credibility of model evaluation. The dataset contains 125,973 training records and 22,544 testing records and supports both binary and multi-class classification (DoS, R2L, U2R, and Probe). It is lightweight (approximately 20 MB), making it computationally efficient, and is publicly accessible via the University of New Brunswick repository.The CIC-IDS2017 dataset, developed by the Canadian Institute for Cybersecurity, mirrors contemporary real-world network conditions. Collected over seven days, it features normal and multiple attack types—such as DDoS, Port Scanning, Brute Force, Botnet, and Web Attacks—captured from realistic user interactions. Each record includes comprehensive flow-based and time-based features, making it suitable for both anomaly and signature-based

detection approaches.To prepare the data, Min–Max normalization was applied to standardize feature ranges, and SMOTE (Synthetic Minority Oversampling Technique) was used to correct class imbalances between attack and normal traffic. Collectively, these preprocessing steps and dataset choices ensure that the proposed NIDS is evaluated under diverse and realistic conditions, enhancing its robustness and adaptability.

Proposed Intrusion Detection Framework

The datasets were prepared and utilized to train the deep neural network (DNN) model for detecting attack and non-attack network packets. The data preparation process involved several key preprocessing steps to ensure quality and consistency. First, redundant entries were removed, and missing values were appropriately handled. Next, Min–Max normalization was applied to scale all feature values within a uniform range, ensuring fair contribution of each feature during model training. One-hot encoding was then employed to convert categorical features into numerical representations suitable for processing by the neural network. After preprocessing, the Enhanced Fruit Fly Optimization Algorithm (EFOA) was applied for dimensionality reduction to eliminate irrelevant or redundant features. The proposed algorithm integrates two improvement mechanisms Dimensional Search Control (DSC) and a Memory-Based Strategy (MBS) to enhance its performance and convergence efficiency.

The Dimensional Search Control mechanism directs and prioritizes the algorithm's search toward more promising regions of the solution space, thereby improving the balance between exploration and exploitation, accelerating convergence, and reducing the risk of getting trapped in local optima. Meanwhile, the Memory-Based Strategy enables the algorithm to store and recall previously successful search experiences, preventing it from revisiting poor solution regions and promoting more efficient optimization.

Following these stages, the study proceeded with model development, optimization, training, testing, validation, and benchmarking to comprehensively evaluate the proposed system's effectiveness.

DNN Model Development and Optimization

This section describes on how the deep neural network model was developed consisting of multiple layers and optimized using Bayesian optimization techniques.

Deep Neural Network Architecture

The core of the proposed NIDS is an optimized DNN consisting generally of:

i- Input Layer: Sized according to the selected feature set.

ii- Hidden Layers: Configured through hyperparameter tuning using backward propagation to balance depth and width, using Rectified Linear Unit (ReLU) activation functions for non-linearity.

iii- Output Layer: The output layer output the classification results, a softmax function is in cooperated in the output layer to classify network traffic into attack categories.

The proposed deep neural network (DNN) architecture consists of the following structure: an input layer with n neurons each corresponding to the selected features identified by the proposed feature selection algorithm. The input layer employs the ReLU (Rectified Linear Unit) activation function to introduce nonlinearity and enable the model to learn complex relationships within the data. Following the input layer, the network includes n hidden layers, each containing a specific number of neurons designed to progressively extract higher-level feature representations. The final output layer comprises multiple neurons corresponding to the target classes and utilizes the Softmax activation function for multi-class classification. In this architecture, each neuron receives input feature values, associated weights, and biases, and computes its output through the dot product of the input features and weights, followed by the addition of the bias term. The model is trained using backpropagation, which adjusts the network's parameters to minimize prediction errors. To evaluate the model's performance, the cross-entropy loss function is employed, as it effectively measures the divergence between the predicted probability distribution and the true class labels. The mathematical formulation of this loss function, along with the corresponding proposed neural network architecture diagram, is presented below. Mathematical expression for single neuron in the proposed architecture is given as:

$$Z = g\sum_{i=1}^{m} X_i\ W_i \dots\dots\dots\dots\dots\dots\dots\dots\dots Enq\ (1)$$

$$Z = g\left(W_o \sum_{i=1}^{m} X_i W_i\right) \dots\dots\dots\dots\dots\dots\dots Enq\ (2)$$

G represent non-linear activation function, z output of a single neuron, Wo bias value, xi single input feature to a single neuron, w represent weight of single connection. Which is equivalent to;

$$Z = W_{o,i} + X_1 W_{1,2} \dots\dots\dots\dots\dots X_m W_{m,2} \dots\dots\dots\dots\dots\dots Enq(3)$$

$$Y = g(W_o + X^T W) Where\ X = X_i\ and\ W = W_I \dots\dots Enq\ (4)$$

The above expression expresses the operation of a single neurons consisting of dot product of input feature Xi and weight Wi plus bias value Wo passed through ReLU activation function for non-linearity as shown in equation 1,2 and 3.

Equation 3 consists of Wo bias value, plus Weight vector represented by matrix W and Input feature value $X^{T.}$ Mathematical expression for the proposed architecture containsone input layer having input feature x1.. x2… x_m (number of neuron), and n hidden layers with output

$g(z_1^k), g(z_2^K)$…..$g(z_m^K)$ for first hidden layer and second layer $g(z_1^N)$,…..$g(z_m^N)$ and output layer consisting of neuron of $Y_{1.....}Y_m$.

First hidden and second layer mathematical expression:

$$Z_i^k = W_{o,i} + \sum_i^m X_{1-m} W_{1-m,i} \dots\dots\dots\dots\dots\dots Enq\ (5)$$

$$Z_M^N = W_{o,i} + \sum_i^m X_{1-m} w_{1-m,i.} \dots\dots\dots\dots\dots\dots Enq(6)$$

.Where Z^N_m represent output of first hidden layer expected to pass to the subsequent neuron in the next layer, Wo bias values, Xi-m input features from I neuron to m neuron, wi-m weight vector.

Stack multiple hidden layer neural network mathematical expression

$$Z_{k,i} = W_{o,i}^{(k)} + \sum_{J=1}^{NK-1} g\left(Z_j^{k-1} W_{i,j}^k\right) \dots\dots\dots\dots Enq(7)$$

The Single neuron mathematical expression output layers

$$Y_i = g(W_{o,i} + \sum_{j-1}^{m} \left(g((Z_i) W_{j,i}\right) \dots\dots\dots\dots Enq(8)$$

The activation function used for the hidden layers and output layers equations

Relu activation function= max (0,x)……...Enq (9)

Sigmoid activation function= $1/1+e^{-1}$.....Enq (10)

The expected loss function used to measure the correctness of the predicted sample is:

L= 1/N $\sum^N_{i=1}$ Y. log ($Y^{\wedge}_2$)……Enq (11)

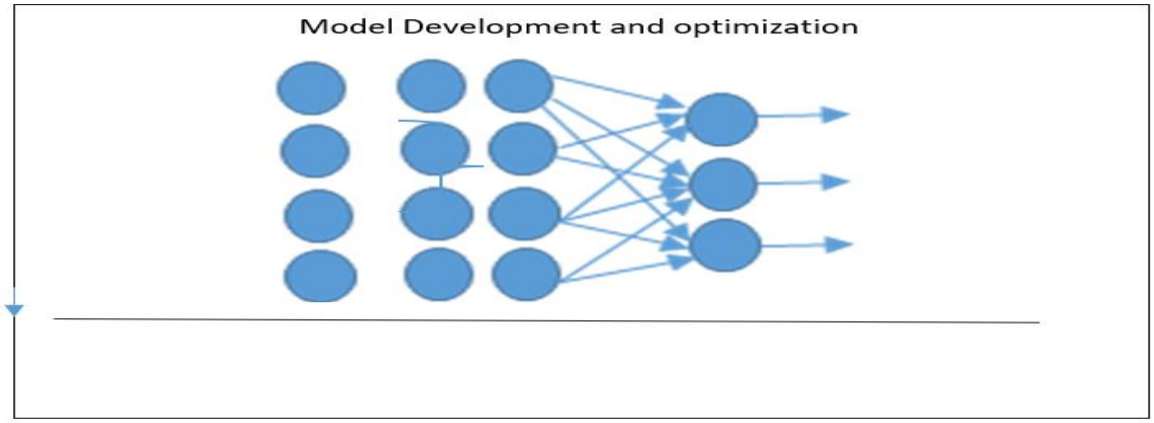

Figure 1 Deep Neural network Diagram consisting of two hidden layers and output layer.

Model Optimization

The proposed deep neural architecture is optimized using Bayesian optimization techniques. **Model Training and Validation**

The dataset was split into 70% training and 30% testing subsets using the Hold-Out method to ensure adequate model generalization.

i Training set (70–80%): Used to train the model

ii Testing set (20–30%): Used to evaluate final model performance

iii Validation set (10–15%): Optionally extracted from the training set for hyperparameter tuning

Model Evaluation

The system was evaluated using performance metrics such as accuracy, precision, recall, and F1-score, and the best-performing model was benchmarked against state-of-the-art models reported in the literature

Performance Metrics

The effectiveness of the proposed system **was assessed** using the following metrics:

i. Accuracy: The accuracy signify the proportion of correctly classified instances.

ii. Precision and Recall: By evaluating the models performance based on precision and recall the model's ability to minimize false positives and false negatives, respectively are assess.

iii. F1 Score: By assessing the performance of the model using F1SCORE which mean a harmonic mean of precision and recall, it highlight the model's overall classification performance.

iv. Latency refers to the time it takes for real-time detection in high-speed networks by measuring the latency of the proposed model means you are measuring the time taking for the model to response in the high speed network.

The above matric will be used to evaluate the models because they are most widely used performance metrics for evaluating the proposed models.

Implementation Requirements

The models were implemented in a Google Colab environment using Python programming language with the hardware specification **of** system RAM of 12.7GB, GPU RAM of 15GB with Window 10 OS. The preprocessing of the data as well as selecting features will be done by using libraries such as Scikit-Learn and Pycaret

RESULTS AND DISCUSSION

In this section, five distinct models were developed using five different meta-heuristic algorithms and evaluated on the NSL-KDD and CICIDS2017 datasets. Each model was trained and tested to assess its classification performance, while the deep learning model was further optimized using Bayesian optimization to obtain the most effective hyper-parameter configurations. The subsequent subsections present the detailed results of the model using each datasets, including performance metrics and the corresponding optimization convergence curves.

Results Using NSL-KDD datasets

Five intrusion detection models were designed, trained, and evaluated using Bayesian optimization over 20 experimental runs. Overall, the results reveal a clear performance hierarchy among the models.

The proposed model, which represents the first and most optimally configured architecture developed in this study, achieved the highest classification accuracy of 0.9674, accompanied by outstanding precision, recall, and F1-score values. Furthermore, the Receiver Operating Characteristic (ROC) analysis and the corresponding Area under the Curve (AUC) score demonstrated the model's strong discriminative capability, indicating its effectiveness in accurately distinguishing between malicious and legitimate network traffic. In addition to its predictive performance, the model exhibited relatively low training and inference times, reflecting a high level of computational efficiency when compared with the benchmark models. The combination of superior detection performance, strong generalization capability, and efficient computational characteristics highlights the proposed model as a reliable, scalable, and practical solution for network intrusion detection applications. The second model yielded a classification accuracy of 0.9561, representing a modest decrement relative to the highest-performing initial model. Its corresponding precision, recall, and F1-score values—0.471, 0.9670, and 0.9569, respectively were also slightly diminished, yet they remained commendably high, indicating sustained reliability in distinguishing between positive and negative instances. With respect to its discriminative ability, the model occupied an intermediate position within the comparative evaluation: it underperformed relative to the foremost model but distinctly outperformed the other contenders, thereby affirming its competitive standing. Moreover, the model exhibited commendable computational efficiency, reflected in its reduced training duration and rapid inference times. This characteristic renders it particularly suitable for real-time or resource-constrained applications, where balancing predictive accuracy with operational speed is essential. Collectively, these findings position the second model as a viable alternative, offering a pragmatic compromise between classification performance and computational economy. The third compared model achieved an accuracy of 0.9560, which is marginally lower than the first two models, yet still indicates strong overall classification capability. Its precision value of 0.9663 is moderately high, suggesting that when the model predicts an attack, it is highly likely to be correct. This reduces the number of false alarms generated, thereby lowering operational costs by minimizing the volume of junk alerts that security analysts must manually investigate. Additionally, the model's recall of 0.9459 demonstrates that it can detect nearly all real attacks occurring within the network, which is critical for minimizing the risk of undetected breaches. However, a significant concern arises from the model's reported perfect ROC and AUC values—an outcome that is practically unattainable in real-world intrusion detection scenarios. Such perfect scores often indicate data leakage or overfitting, where the model inadvertently memorizes dataset-specific artifacts such as particular IP addresses or duplicated samples, rather than learning generalized patterns of malicious behavior. Therefore, despite its strong performance metrics, the model's reliability in live deployment is questionable, as it may fail to generalize to novel or evolving attack strategies. In terms of computational efficiency, the model required slightly longer training and testing times compared to the first two models, indicating a moderate increase in computational cost. While this trade-off may be acceptable in environments where processing power is not a constraint, it should be considered when deploying the model in resource-limited or high-throughput network settings. Overall, while the model demonstrates promising detection capabilities, the perfect ROC/AUC score warrants further investigation to ensure that its performance is genuinely driven by meaningful learning rather than statistical artifacts. The fourth model, based on the Ant Lion Optimizer (ALO), achieved an accuracy of 0.9641, which demonstrates excellent overall classification performance, although it ranks slightly lower than the first three models. Its precision score of 0.9474 is moderately high, indicating that when the model flags an event as an attack, it is likely to be correct. This helps reduce the number of false positives, thereby

lowering operational costs by minimizing the volume of nuisance alerts that security analysts would otherwise need to investigate manually. However, this precision is somewhat lower compared to the first three models, meaning it may generate a slightly higher number of false alarms than its predecessors. The model's recall value of 0.9454 is strong, suggesting that it can detect the vast majority of actual attacks occurring within the network, which is essential for minimizing the risk of undetected intrusions and data breaches. Despite these promising metrics, a significant concern arises from the model's reported ROC and AUC values, which appear unusually high and may indicate potential data leakage or over fitting. Such perfect or near-perfect scores often suggest that the model has inadvertently memorized dataset-specific patterns such as particular IP addresses, port numbers, or duplicated traffic samples rather than learning the underlying characteristics of genuine malicious activity. This raises serious doubts about the model's ability to generalize effectively when deployed in live environments, where attack patterns evolve continuously and differ from training data. In terms of computational efficiency, the ALO-based model required longer training and testing times compared to the first three models, indicating a moderate increase in processing overhead. While this additional computational cost may be acceptable in environments with sufficient hardware resources, it could pose challenges in real-time or resource-constrained network settings where low-latency detection is critical. Overall, while the model exhibits strong detection capabilities and operational benefits, the suspiciously high ROC/AUC scores necessitate further scrutiny to ensure that its performance is driven by genuine learning rather than statistical artifacts or data contamination.

However, the final model implemented among the proposed designs yielded a lower accuracy than the comparative benchmark, underscoring certain limitations in its ability to effectively differentiate between attack and benign traffic. This reduced accuracy points to underlying deficiencies in the model's overall classification capability. Furthermore, when assessed in terms of precision and F1-score, the model underperformed relative to its counterparts, indicating a higher propensity for generating false alarms. The notable number of false positives, coupled with a diminished F1-score, suggests that the trade-off between precision and recall is unfavorably skewed, with precision-related shortcomings outweighing any gains achieved in recall. In addition, the model recorded lower Area under the Curve (AUC) and Receiver Operating Characteristic (ROC) values compared to the reference model, reflecting a reduced discriminative power and weaker predictive performance. With respect to computational efficiency, the model demonstrated moderate performance, achieving a marginally shorter testing time than the comparison model. However, it incurred a slightly longer training duration, with the exception of the SCMWOA model, which exhibited relatively higher training overhead. Overall, while the model offers certain operational efficiencies in inference, its predictive reliability and classification robustness remain areas requiring further refinement.

Results Using CICIDS2017

In this study, five distinct feature selection mechanisms were employed to develop deep learning-based intrusion detection models, all rigorously evaluated using the CICIDS2017 benchmark dataset to assess their classification performance, predictive ability, and computational efficiency. Among the proposed models, the variant incorporating the Enhanced Fruit Fly Optimization Algorithm (EFOA) demonstrated exceptional performance, achieving an outstanding classification accuracy of 0.9994, which underscores its strong discriminative capability and consistent predictive behavior across varied traffic patterns. The model further recorded precision, recall, and F1-score values of 0.9998, 0.9989, and their corresponding harmonic mean, respectively, highlighting its proficiency in accurately identifying malicious activities while maintaining a remarkably low false positive rate, thereby minimizing erroneous alerts that could undermine operational trust. In terms of predictive power, the EFOA-based model attained superior Area under the Curve (AUC) and Receiver Operating Characteristic (ROC) scores, reflecting its enhanced ability to distinguish between normal and anomalous network flows with high sensitivity and specificity. From a computational standpoint, the approach exhibited notable efficiency by achieving shorter training and testing durations relative to the other models under comparison, a particularly advantageous trait for time-sensitive deployment environments. Collectively, these findings affirm the reliability, scalability, and practical viability of the EFOA-enhanced deep learning model for real-world intrusion detection systems, positioning it as a compelling candidate for deployment in contemporary network security infrastructures where rapid and precise threat identification is paramount.

Under identical experimental conditions, the basic Fruit Fly Optimization Algorithm (FOA)-based model yielded a classification accuracy slightly lower than that of its enhanced counterpart, yet it still achieved near-perfect precision, recall, and F1-scores of 0.9991, 0.9988, and 0.9990, respectively. These impressive metrics underscore the model's robust capability to reliably distinguish between malicious and benign network traffic with a remarkably low incidence of misclassification. In terms of predictive power, the basic FOA-based model demonstrated exceptional discriminative ability, attaining perfect Area under the Curve (AUC) and Receiver

Operating Characteristic (ROC) values of 1.000, which indicate an ideal trade-off between true positive and false positive rates across all classification thresholds. Regarding computational efficiency, the model exhibited commendable performance, achieving training and testing times that were lower than those of all comparative models, with the sole exception of the proposed EFOA-based variant, which outpaced it marginally. These findings collectively suggest that while the basic FOA model may not surpass the enhanced version in overall accuracy, its near-optimal detection metrics, flawless predictive scoring, and favorable computational footprint render it a highly effective and practical solution for intrusion detection applications, particularly in scenarios where computational resources are constrained yet reliability remains paramount.

The Ant Lion Optimizer (ALO)-based model emerged as the third developed approach, achieving an accuracy value marginally lower than those of the first two comparative models. Despite this slight reduction in overall accuracy, the model demonstrated strong classification proficiency, with precision, recall, and F1-scores of 0.9993, 0.9987, and 0.9990, respectively metrics that collectively attest to its reliable capability in distinguishing between attack and normal traffic with minimal misclassification. In terms of predictive performance, the ALO-based model attained near perfect Area under the Curve (AUC) and Receiver Operating Characteristic (ROC) values of 1.000, indicating an optimal trade-off between sensitivity and specificity across all decision thresholds, thereby reflecting its outstanding discriminative power. Regarding computational efficiency, however, the model exhibited a mixed profile: while its testing time was slightly lower than that of four other models, suggesting reasonable inference speed, it incurred a higher training time compared to all competing approaches, which may pose considerations for deployment in environments where rapid model retraining is essential. Generally, the ALO-based model strikes a commendable balance between near-optimal detection metrics and perfect predictive scoring, though its increased training overhead warrants attention when prioritizing computational resources in real-world intrusion detection systems. The fourth model under evaluation achieved an accuracy that surpassed only the final comparative model, namely the SCMWOA-based approach, positioning it second-to-last in terms of overall classification performance. Nevertheless, the model demonstrated respectable detection capabilities, attaining precision, recall, and F1-score values of 0.9985, 0.9918, and 0.9951, respectively, metrics that indicate a reasonably effective ability to identify malicious traffic, even though with a slightly more pronounced trade-off between precision and recall relative to the top-performing models. In terms of predictive power, the model achieved perfect Area under the Curve (AUC) and Receiver Operating Characteristic (ROC) values of 1.000, signifying an ideal discriminative capacity between normal and attack traffic across all classification thresholds. However, from a computational efficiency standpoint, the model exhibited notable limitations, recording a testing time of 3.3469 seconds, which was considerably higher than most competing approaches. Additionally, its training time ranked among the highest across all models, surpassed only by the ALO-based model, which demonstrated even greater training overhead. These findings suggest that while the fourth model offers commendable predictive accuracy and flawless discriminative ability, its elevated computational demands particularly in terms of both training and inference durations may hinder its practicality for real-time intrusion detection applications, especially in resource-constrained or latency-sensitive environments. The final model, which exhibited the lowest performance among all compared approaches, achieved an accuracy that fell short of every other model under evaluation. Its precision, recall, and F1-score values were recorded at 0.9833, 0.9830, and 0.9906, respectively, while still reflecting a reasonable level of classification capability, indicate a more pronounced susceptibility to misclassification relative to its counterparts, particularly in accurately distinguishing between attack and benign traffic. Despite this relative decline in classification performance, the model demonstrated flawless predictive discriminative ability, attaining perfect Area under the Curve (AUC) and Receiver Operating Characteristic (ROC) values of 1.000, which underscore its optimal trade-off between true positive and false positive rates across all decision thresholds. From a computational efficiency perspective, however, the model presented a notably high training time of 115.125 seconds, substantially exceeding that of two compared models and raising concerns regarding its feasibility in environments requiring rapid model updates or iterative retraining. Conversely, its testing time of 2.9367 seconds was reasonably competitive, suggesting that once trained, the model can perform inference at a moderate pace suitable for certain deployment scenarios. Complete, while the model offers perfect predictive discrimination and acceptable inference speed, its inferior accuracy metrics and prohibitive training overhead significantly undermine its practical viability for real-world intrusion detection systems, particularly where both detection reliability and computational economy are critical considerations.

Performance Summary Table

The tables given below contains all values reported by the models in the experiments that compared the proposed models with other developed models. The tables are for NSL-KDD and CICIDS2017 datasets.

Table 1 Present Performance Summary table for NSL-KDD datasets Results

Performance Matric and Algorithm	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**AUC VALUE**	**Testing Time**	**Training Time**	**False Positive Rate**
ALO Based Model	0.9641	0.9474	0.9454	0.9636	0.9951	1.3139s	98.12s	0.054
BIFOA Based Model	0.9417	0.9201	0.9673	0.9431	0.9874	1.6858s	98.95s	0.033
Basic FOA Based Model	0.9560	0.9663	0.9459	0.9560	0.9874	1.6858s	98.95s	0.209
Scmwoa Based Model	0.9561	0.9572	0.9572	0.9561	0.9928	1.6934s	99.27s	
Proposed EFOA Based Model	**0.9674**	**0.9676**	**0.9673**	**0.9674**	**0.9933**	2.0905s	**97.20s**	0.059

Table 2 Present Performance Summary table for CICIDS2017 datasets Results

Performance Matrics and Models	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AUC VALUE**	**Testing Time**	**Training Time**	**False Positive Rate**
ALO Based Model	0.9989	0.9993	0.9987	0.9990	1.000	2.9284s	551.66s	0.001
BIFOA Based Model	0.9952	0.9985	0.9918	0.9951	0.9998	3.3469	235.48s	0.008
Basic FOA Based Model	0.9990	0.9991	0.9988	0.9990	1.000	2.7751s	131.99s	0.001
scmwoa Based Model	0.9905	0.9833	0.9830	0.9906	0.9997	2.7712s	115.125s	0.05
Proposed EFOA Based Model	**0.9994**	**0.9998**	**0.9989**	**0.9994**	**1.000**	**2.7712**	**112.32s**	**0.001**

Based on the performance metrics used to evaluate the five designed models, as presented in Tables 1 and 2, seven pie charts were developed to graphically illustrate the results. These charts depict the values attained by each model for the various performance metrics across the two datasets, providing a clear visual comparison of their performance.

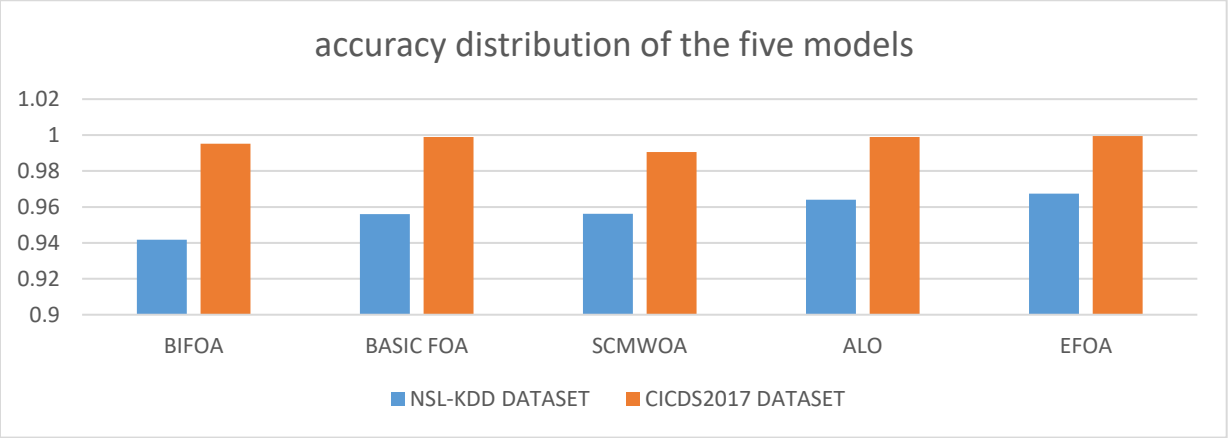

Figure 1: Accuracy values for five different models across two datasets.

Figure 1 illustrates the classification accuracy achieved by the five designed models alongside the proposed EFOA-based model across the two datasets. The results show that the proposed EFOA-based model attained the highest classification accuracy among all the evaluated models. This superior accuracy indicates the model's enhanced ability to correctly distinguish between anomalous and normal network packets while maintaining a low false positive rate, thereby demonstrating its effectiveness and reliability in intrusion detection

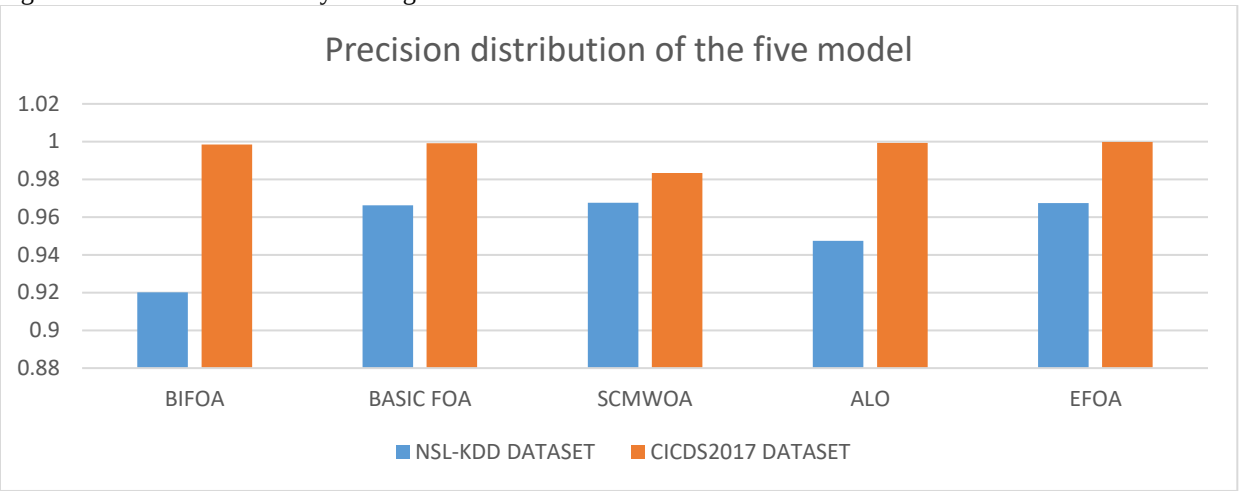

Figure 2 precision values for five different models across two datasets

Figure 2 illustrates the precision values of the five designed models alongside the proposed EFOA-based model across the two datasets. The results indicate that the proposed EFOA-based model achieved the highest precision, demonstrating its superior ability to minimize false positive predictions. This performance highlights its effectiveness and underscores its greater suitability for intrusion detection compared to the other evaluated models.

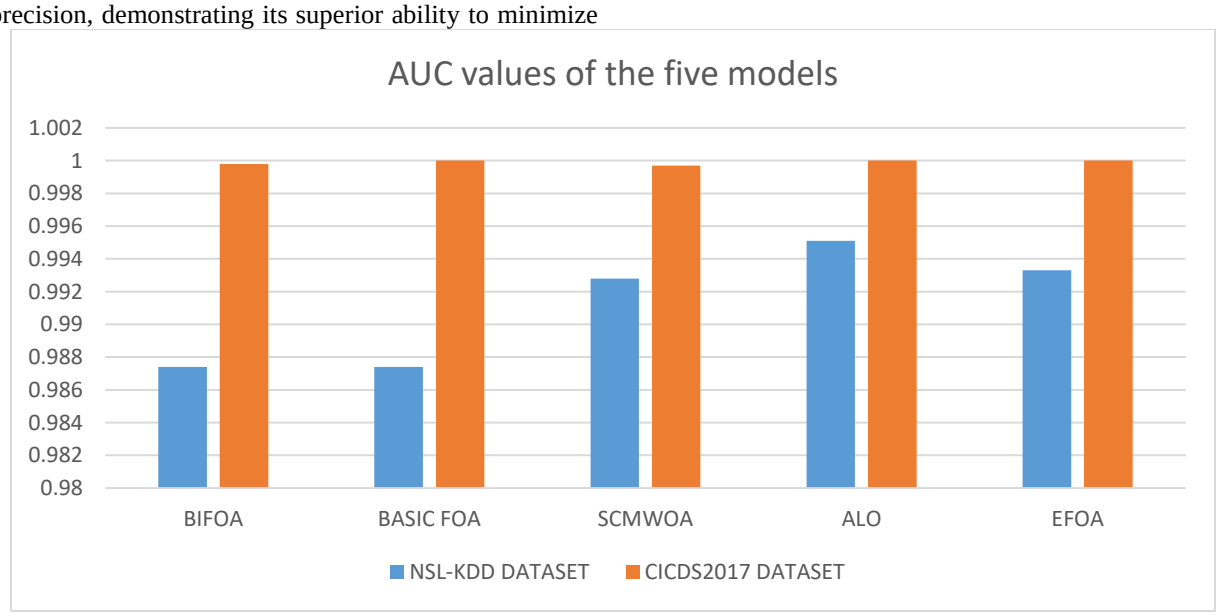

Figure 3 AUC Values of the five models

Figure 3 illustrates the Area Under the Curve (AUC) values of the five designed models alongside the proposed EFOA-based model across the two datasets. The results show that the proposed EFOA-based model achieved the highest predictive performance on the second dataset (CICIDS2017), as indicated by its superior AUC value. However, on the first dataset, the model recorded a lower AUC than some of the compared models, suggesting that while it performed exceptionally well on CICIDS2017, its predictive capability was comparatively lower on the first dataset.

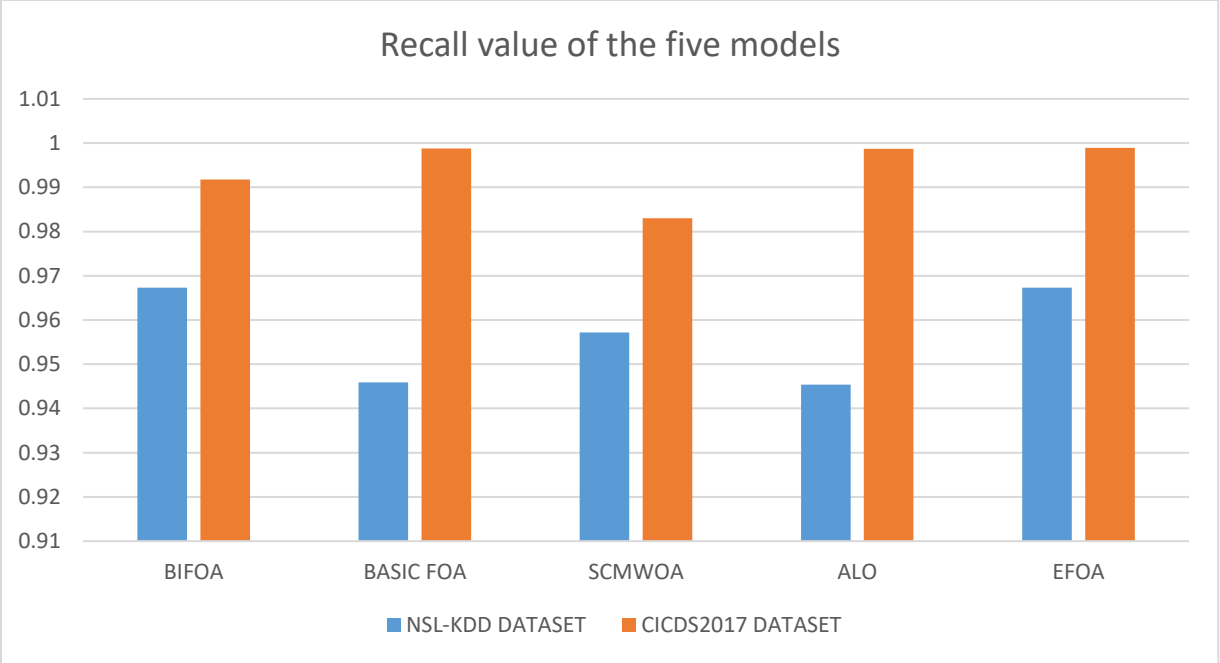

Figure 4 Recall Values of the five Models

Figure 4 illustrates the recall values of the five designed models alongside the proposed EFOA-based model across the two datasets. The results show that the proposed EFOA-based model achieved the highest recall values on both datasets, indicating its superior ability to correctly identify positive instances while minimizing false negative predictions compared to the other evaluated models.

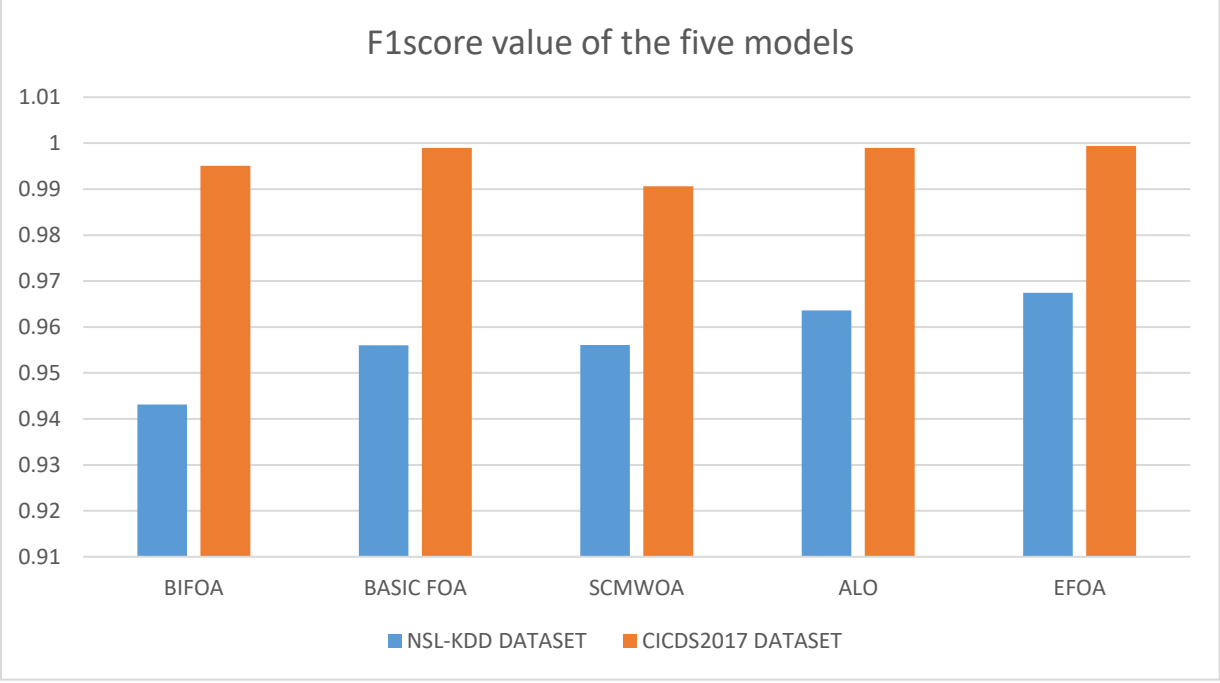

Figure 5 F1 score value of the five model

Figure 5 illustrates the F1-score values achieved by the five designed models alongside the proposed EFOA-based model across the two datasets. The results reveal that the proposed EFOA-based model attained the highest F1-score values, indicating its superior ability to achieve a balanced trade-off between precision and recall by effectively minimizing both false positive and false negative predictions compared to the other evaluated models.

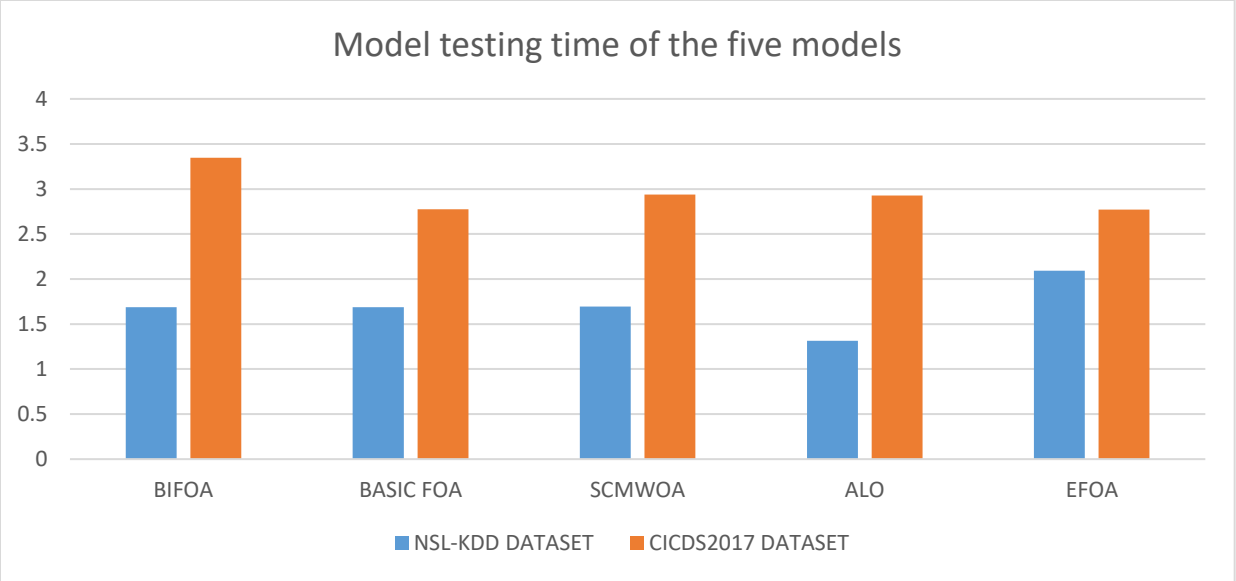

Figure 6 Testing time of the five model

Figure 6 illustrates the testing time of the five designed models alongside the proposed EFOA-based model across the two datasets. The results reveal that the proposed EFOA-based model achieved the shortest testing time on one of the datasets compared to the other evaluated models. This indicates that the model is capable of generating attack and non-attack predictions more rapidly, demonstrating its computational efficiency and suitability for real-time intrusion detection applications.

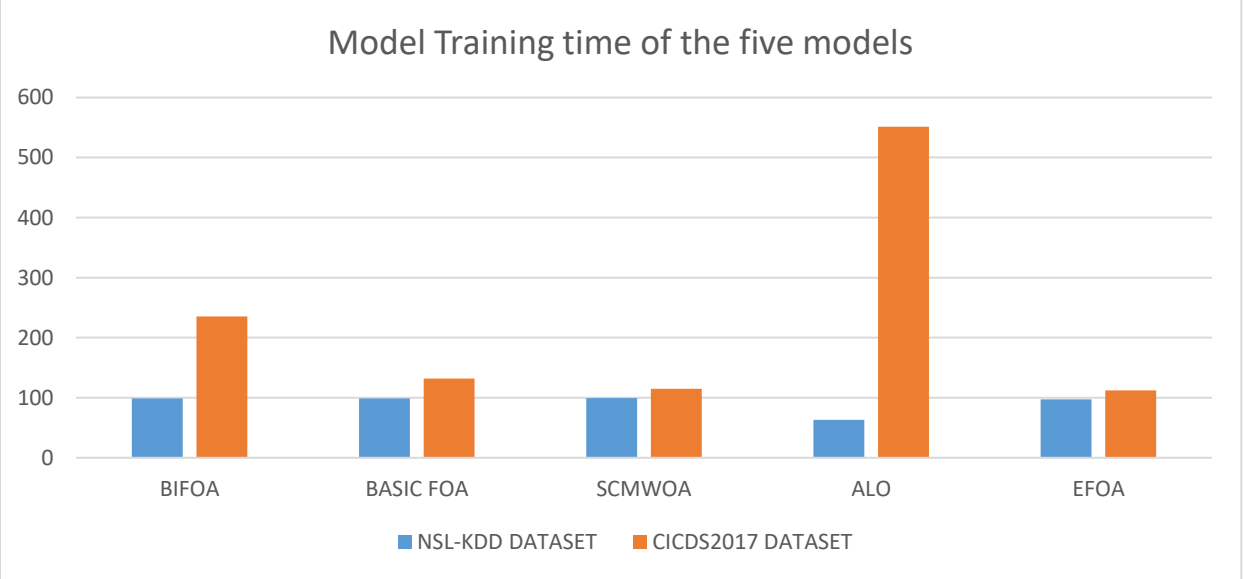

Figure 7Training time of the five Models

Figure 7 illustrates the training time of the five designed models alongside the proposed EFOA-based model across the two datasets. The results reveal that the proposed EFOA-based model achieved the shortest training time on the second dataset. On the first dataset, it recorded a lower training time than three of the compared models but a slightly higher training time than the ALO-based model. These findings indicate that the proposed model reduces computational cost while enabling faster training and more efficient deployment compared to most of the evaluated models.

Evaluation and Benchmarking

Confusion Matrix Comparison for Model trained using NSL-KDD

This section presents the performance of the best-performing model based on its optimization results and confusion matrix, which summarizes the numbers of true

positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The proposed model, which incorporates the Enhanced Fruit Fly Optimization Algorithm (Enhanced FOA) and was evaluated using the NSL-KDD dataset, is assessed using a combination of performance indicators, including computational efficiency, predictive capability, and standard classification performance metrics.

From an optimization perspective, the Bayesian optimization process enabled the model to converge to an optimal set of hyperparameters during the early stages of training, demonstrating efficient convergence behaviour. However, despite this rapid convergence, the proposed model achieved a lower validation accuracy than several of the models included in the comparative analysis. In contrast to the third model, which exhibited a higher validation accuracy and a lower misclassification rate, the proposed enhanced model was less successful in identifying hyper-parameter configurations that maximized classification performance. This behaviour is illustrated by the convergence curve shown in 8.

These findings emphasize that optimization performance should be evaluated not only by convergence speed but also by the quality of the resulting solution. Although the proposed model benefited from early convergence, its relatively lower accuracy, higher validation loss, and increased misclassification rate indicate that the selected hyper-parameters did not provide the best classification capability. As shown in Figure8, the confusion matrix recorded 13,032 true positives, 13,028 true negatives, 441 false positives, and 436 false negatives, corresponding to a false positive rate of 0.059. While these results demonstrate reasonable detection capability, they also reveal that rapid convergence alone is insufficient to guarantee superior classification performance. Therefore, effective optimization techniques should achieve both fast convergence and robust exploration of the search space to identify high-quality solutions that maximize intrusion detection accuracy while minimizing classification errors.

The figures 8 and 9presented below illustrate the confusion matrix and convergence curve of the proposed model, providing a visual representation of its classification performance and optimization behaviour.

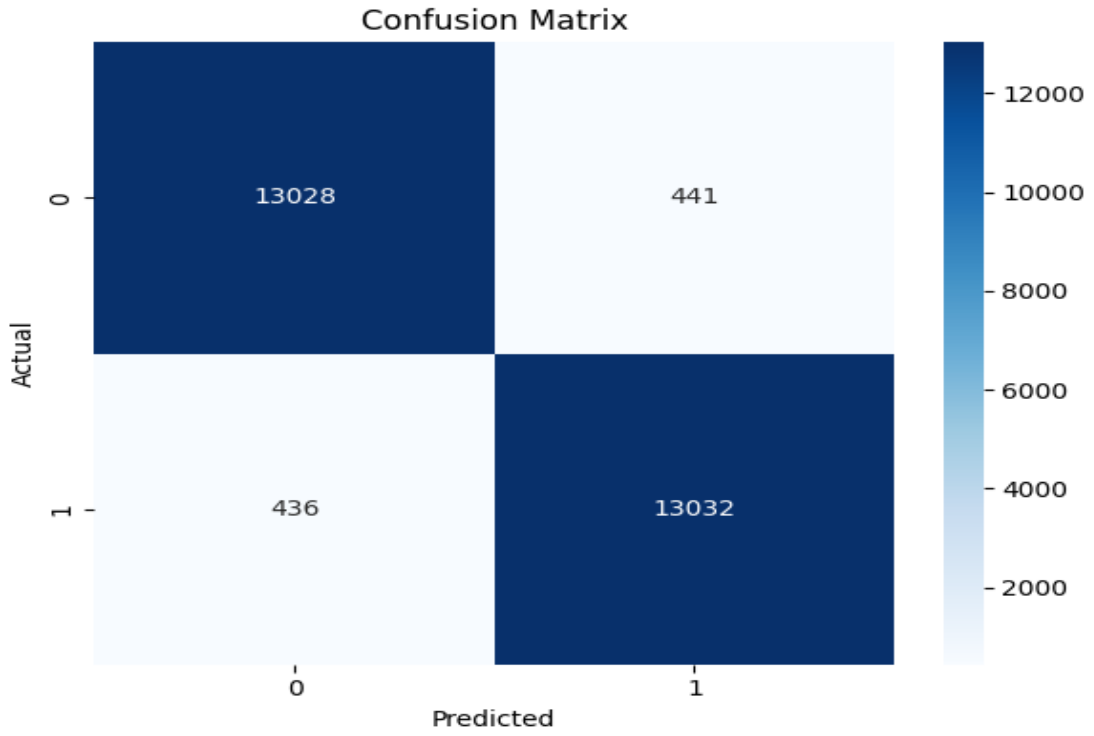

Figure 8 the confusion matrix of Enhanced FOA-DNN intrusion detection Model

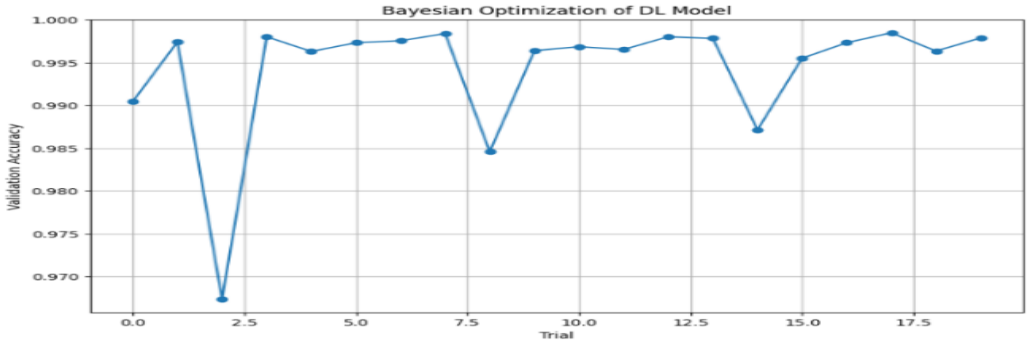

Figure 9 Convergence graph or curve of EFOA-DNN intrusion detection Model optimized using Bayesian optimization Technique

Confusion Matrix Comparison for Model trained using CICIDs2017

This section presents the performance of the best-performing model, focusing on its optimization results and confusion matrix, which summarizes the numbers of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The experimental evaluation of the Enhanced Fruit Fly Optimization Algorithm (Enhanced FOA) using the CICIDS2017 dataset was conducted based on a combination of performance indicators, including computational efficiency, predictive capability, and classification performance metrics.

The confusion matrix of the Enhanced FOA model provides further insight into its classification performance. The results indicate only a small number of misclassifications, with 27 attack instances incorrectly classified as normal traffic (false negatives) and 5 normal instances incorrectly identified as attacks (false positives). Although these errors are minimal, they suggest opportunities for further improvement, particularly in enhancing the model's ability to distinguish subtle variations in network traffic patterns.

From an optimization perspective, the convergence curve demonstrates that the Enhanced FOA model attained its highest validation accuracy during the early stages of training and maintained this level of performance consistently throughout the optimization process. Unlike the fluctuating convergence behaviour observed in the preceding models, the Enhanced FOA exhibited a stable convergence pattern, indicating improved optimization efficiency and reliability. Such stability minimizes unnecessary computational overhead associated with extended training while ensuring that the optimization process consistently converges toward high-quality solutions.

Overall, the Enhanced FOA model achieved superior classification accuracy, strong predictive performance, and balanced evaluation metrics while maintaining stable convergence throughout the optimization process. These characteristics demonstrate the effectiveness of the proposed optimization enhancements and highlight the model's suitability for practical network intrusion detection applications.

The figure 10 and 11presented below illustrate the confusion matrix and convergence curve of the Enhanced FOA model, providing a visual representation of its classification performance and optimization behavior.

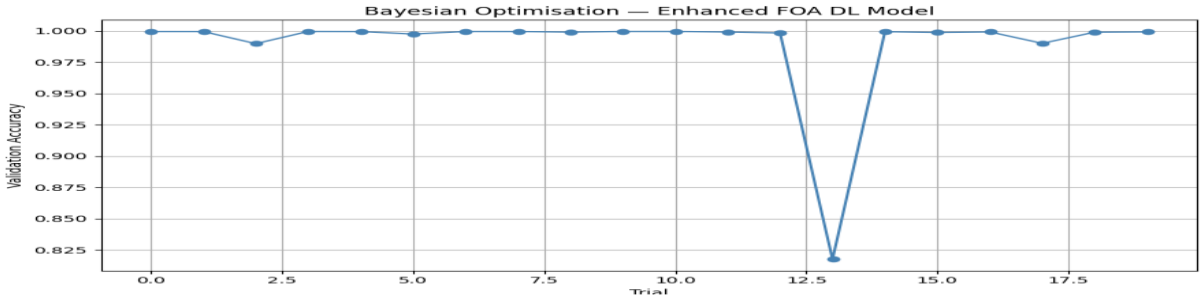

Figure 11: The convergence graph or curve of EFOA-DNN intrusion detection Model optimized using Bayesian optimization Technique

Comparison with Previous Works

In this section, the best-performing model, the proposed Enhanced FOA with integration of dimensional search control and a memory-based strategy to guide its search strategy is compared with state-of-the-art models reported in previous studies to evaluate its effectiveness for real-world deployment in production environments. The comparison is conducted using key performance indicators, including average accuracy, precision, recall, and F1-score,AUC, ROC values and Testing and Training time to provide a comprehensive assessment of its relative strengths.

The results presented in the table demonstrate that the proposed model, incorporating the Enhanced Fruit Fly Optimization Algorithm (FOA) for feature selection, exhibits superior performance compared to other state-of-the-art models in terms of classification accuracy. The proposed approach attained an accuracy rate approaching 100%, underscoring its high discriminative capability in correctly identifying both attack and non-attack packets. In contrast, the benchmark models achieved comparatively lower accuracy values, thereby validating the effectiveness and robustness of the proposed method. The graph given below demonstrate the accuracy distribution across different design models with the proposed model achieving superior performance.

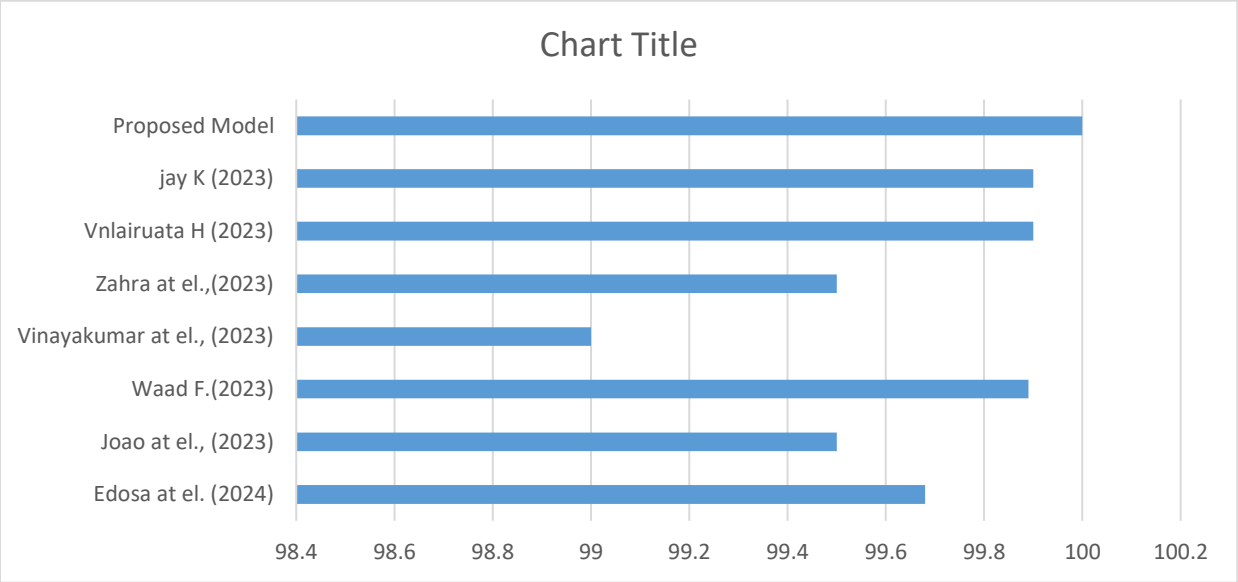

Figure. 12 Accuracy Distribution of the proposed model with state of the art models

Table3: Accuracy values of the proposed model and other state of the art models

S/NO	MODEL NAME	DATASETS USED	AVERAGE ACCURACY
1.	Edosa at el. (2024)	NSL-KDD, UNSW-NB15	99.68%
2.	Joao at el.,(2023)	CICID2017	99.5%
3.	Waad F.(2023)	CICID2017	98.0%
4.	Vinayakumar et al., (2023)	CICID2017	99.00%
5.	Zahra at el.,(2023)	NSL-KDD	99.50%
6.	(2023)	CICID2017	99.80%
7.	**Proposed Model**	CICID2017,NSL-KDD	96.74%-99.94%

CONCLUSION

The proposed model, designed using proposed EFOA feature selection technique, demonstrated remarkable performance and clearly outperformed previously reported state-of-the-art models. Beyond achieving superior average accuracy, the model exhibited strong consistency across other evaluation metrics, such as precision, recall, and F1-score, all of which attained nearly an average value of 100%. These results confirm the model's robustness in correctly distinguishing

between attack and non-attack instances with minimal misclassifications. One of the most notable strengths of the Enhanced FOA-based approach lies in its optimization process. By integrating dimensional search control and a memory-based strategy, the model effectively guided the optimization process toward the best optimal hyper parameters. This not only enhanced classification accuracy but also reduced the training time required to reach optimal solutions. As a result, the proposed model offered improved convergence stability compared with other approaches, ensuring smooth and reliable optimization performance throughout training. From a computational standpoint, the Enhanced FOA proved to be both efficient and resource-friendly. Its ability to achieve high performance with reduced computational overhead highlights its advantage over more complex feature selection techniques employed in previous studies. This makes it particularly suitable for deployment in real-world intrusion detection systems, where computational resources and training efficiency are critical.

In terms of misclassification rates, the confusion matrix analysis revealed that the Enhanced FOA-based model achieved the lowest rate of misclassified instances compared to other models. This performance translates directly into practical benefits, such as fewer false alarms, reduced analyst fatigue, and greater assurance that actual attacks are detected and addressed promptly. The combination of perfect detection metrics, low misclassification, and efficient optimization strongly positions the proposed EFOA model as a highly reliable and deployable solution for modern cybersecurity environments. The limitation of the model is that the model was not implemented, tested and trained in a real time network environment rather the model was created or implemented in a standalone form and its feature selection mechanism proposed in this work only used single meta-heuristic algorithm integrated with some mechanism to guide its search strategies. In future real time implementation of the model will be done to classified attacks instances in real time and hybrid of meta-heuristic algorithm will be used to create feature selection scheme in future taking computational complexity in to account.

REFERENCE

Snehal, T., & Jadhav, P. (2020). Addressing false positives in traditional network intrusion detection systems. . International Journal of Network Security, 22(4), 401–415.

Naveed, M., Fahim Arif ,F., Syed Muhammad Usman,S., M., Anwar A., Hadjouni,M., Elmannai, H. Hussain,S., Syed Sajid Ullah , and Umar, F., A (2022) Deep Learning-Based Framework for Feature Extraction and Classification of Intrusion Detection in Networks.https://doi.org/10.1155/2022/2215852

Zhao ,T, Yue, M. and Wang ,J. (2025) Robust Power System Stability Assessment Against Adversarial Machine Learning-Based Cyberattacks via Online Purification.

ZHAO R., MU Y., ZOU. L., AND WEN X.,(2022).*A Hybrid Intrusion Detection System Based on Feature Selection and Weighted Stacking Classifier*. Digital Object Identifier 10.1109/ACCESS.2022. 3186975.

Dagli, C., & Ashiku, L. (2021). Network intrusion detection system using deep learning. Procedia Computer Science, 185, 239-247 https://doi.org/10.1016/j.procs.2021.06.042.

Alandjani G. ,(2024) Optimizing Malware Detection for IoT and Edge Environments with QuAntization Awareness.Digital Object identifier 10.1109/Access.2024.3495635.

Vanlalruata, H., & Hussain, J. (2021). Dependable intrusion detection system using deep onvolutional neural network. Proceedings of the Complex Adaptive Systems Conference: Big Data, IoT, and AI for a Smarter Future, Malvern, Pennsylvania, June 16-18, 2021

Zahra, O., Raja, M., & Pran, R. (2023). Enhancing deep learning-based NIDS: A focus on imbalanced datasets and computational efficiency. Computers & Security, 119, 102754.

Joloudari J., H., , Marefat A. , Mohammad Ali Nematollahi M., A., , Solomon Sunday Oyelere,s., s. and Hussain S.,: Effective Class-Imbalance Learning Based on SMOTE and ConvolutionalNeural Networks. https://doi.org/10.3390/app13064006

Bhuvaneshwari, K. S., Venkatachalam, K., Hubálovský, S., Trojovský, P., & Prabu, P. (2022). Improved dragonfly optimizer for intrusion detection using deep clustering CNNPSO classifier. CMCComputers, Materials & Continua://doi.org/10.32604/cmc.2022.020769

EDoaa N. Mhawi D., N., , Ammar Aldallal A., &Hsassan S., Advanced Feature-Selection-Based hybrid Ensemble Learning Algorithms for Network Intrusion Detection Systemshttps://doi.org/10.3390/sym14071461