Optimizing Machine Learning Models for IoT-Based DDoS Attack Detection through Hyper parameter Tuning

A.Priyadharshini[1], S.Dhinakaran[2]

[1]Ph.D. Research Scholar, Department of Computer Science, Rathinam College of Arts and Science, Coimbatore – 641021, Email: phdpriyadharshini@gmail.com
[2]Assistant Professor, Department of Computer Science, Rathinam College of Arts and Science, Coimbatore – 641021, Email: dhinakaran.cs@gmail.com

Received: 14.07.2024 Revised: 13.08.2024 Accepted: 10.09.2024

ABSTRACT
The proliferation of Internet of Things (IoT) devices has escalated the complexities and frequencies of Distributed Denial of Service (DDoS) attacks, making traditional detection mechanisms inadequate. This paper explores the enhancement of machine learning (ML) models specifically tuned for IoT environments using systematic hyperparameter optimization via grid search. By tailoring the learning processes and configurations of models such as Random Forest, XGBoost, and Support Vector Machines, the study achieves superior detection rates, reduced false positives, and improved computational efficiency. The findings suggest that precise hyperparameter tuning is crucial for adapting DDoS detection systems to the unique characteristics of IoT networks, thereby offering robust defenses against evolving cyber threats.

Keywords: DDos attack, Grid search, Hyperparameter optimization, Internet of things, Machine learning, Network security.

1. INTRODUCTION

The burgeoning landscape of the Internet of Things (IoT) brings with it an expanded attack surface prone to Distributed Denial of Service (DDoS) threats [1]. These threats capitalize on the inherent vulnerabilities of widely dispersed and often inadequately secured IoT devices. As IoT networks become integral to infrastructure, from smart cities to healthcare systems, ensuring their resilience against DDoS attacks is paramount [2]. This necessity drives the demand for advanced detection systems that not only adapt to evolving threats but also operate within the constraints of IoT environments characterized by limited computational resources and high data variability.

Machine Learning (ML) models have emerged as aeffective tool for detecting DDoS activities by learning from historical data and identifying patterns indicative of attacks [3]. However, the utility of these models in IoT contexts hinges critically on their configuration, specifically the tuning of hyperparameters that govern their learning algorithms. Hyperparameter optimization can significantly influence a model's ability to generalize from training data to real-world application, affecting everything from model accuracy to computational efficiency [4]. This paper focuses on the systematic optimization of these parameters using grid search techniques, a method that explores multiple combinations of parameters to find the most effective settings for DDoS detection in IoT frameworks.

Traditional approaches to securing IoT devices against DDoS attacks typically involve static, rule-based systems that cannot easily adapt to new or evolving attack vectors [5]. By contrast, ML models offer dynamic analysis capabilities that can adapt to changes in network behavior indicative of DDoS attacks [6]. However, developing ML models that are both effective and efficient requires careful consideration of model complexity, which can lead to overfitting if not properly managed [7]. Overfitting occurs when a model is too closely fitted to the limited training data, making it unable to generalize well to new, unseen datasets [8]. This is particularly problematic in IoT scenarios where devices generate vast and diverse data streams under varying conditions.

To address these challenges, hyperparameter tuning through grid search provides a structured exploration of parameter space, enhancing model robustness by finding the optimal balance between bias and variance. This balance is crucial for ML models tasked with securing IoT networks, where the diversity of devices and the variability in data can otherwise lead to poor model performance. Grid search systematically tests different combinations of parameters, evaluating their performance to ensure that the

selected model configurations offer the best predictive power and generalization from seen to unseen data.

Furthermore, the computational limitations inherent in many IoT environments necessitate models that are not only accurate but also resource-efficient. Hyperparameter optimization plays a pivotal role in this regard by adjusting model parameters to minimize computational demands without compromising detection capabilities. For instance, parameters such as the number of trees in a Random Forest model or the depth of the trees can be optimized to balance detection accuracy with the model's computational footprint. This is critical in IoT settings, where processing power and memory are often at a premium.

This research contributes to the field by deploying grid search optimization to refine the hyperparameters of various ML models tailored for IoT environments. The study evaluates models such as Random Forest, XGBoost, and Support Vector Machines across multiple IoT network scenarios to determine the impact of hyperparameter tuning on model efficacy and efficiency. By focusing on metrics such as precision, recall, and computational load, the study provides insights into the practical deployment of these models in real-world IoT applications.

Additionally, the paper discusses the scalability of optimized ML models within diverse IoT architectures, from small home networks to extensive industrial systems. It also examines the adaptability of these models to the dynamic nature of DDoS threats, where attackers continually modify their strategies to bypass conventional detection methods. The findings aim to offer a roadmap for integrating advanced ML-based DDoS detection systems into existing and future IoT infrastructures, providing a robust defense mechanism that enhances both the security and functionality of IoT networks.

2. RELATED WORK

This section reviews relevant literature, highlighting key contributions and identifying various approaches and challenges associated with DDoS detection, particularly within the context of evolving IoT infrastructures.

Machine Learning Approaches to DDoS Detection

A significant body of research has focused on applying traditional machine learning techniques to identify and mitigate DDoS attacks. Awad&Fraihatexplored the use of Random Forests and Decision Trees to classify network traffic, noting the importance of feature selection in enhancing model accuracy and computational efficiency [9]. Their work underscored the relevance of statistical features, such as flow duration and packet intervals, in distinguishing between benign and malicious traffic.

Another notable approach involves Support Vector Machines (SVM), which have been favored for their effectiveness in handling non-linear data separation problems. As highlighted by Mishra & Pandya, SVMs can be particularly non-effective in scenarios where the attack patterns are not explicitly defined, making them non-suitable for environments with diverse IoT devices [10]. The study employed kernel tricks to transform data into higher dimensions, thereby facilitating more accurate classification between attack and normal traffic.

In addition to traditional models, ensemble techniques have garnered attention for their robust performance across varied datasets. Ismantoet al. (2024) demonstrated how ensemble methods like Gradient Boosting Machines (GBM) could leverage the strengths of multiple weak learners to improve detection rates, especially in noisy environments typical of IoT networks [11]. Their research emphasized the model's ability to compensate for any individual learner's bias or variance, leading to improved reliability and accuracy.

Deep Learning Innovations

The integration of deep learning models has marked a progressive shift in DDoS detection strategies due to their capacity to learn complex patterns and perform feature extraction automatically [12, 13]. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have been particularly prominent. Hekmati et al., applied CNNs to raw network packets to extract spatial features, which proved effective in identifying subtle anomalies indicative of low-rate DDoS attacks, a common challenge in IoT applications [14]. Meanwhile, RNNs have been leveraged to exploit temporal features in network traffic, as explored by Parmaret al. (2023), who used LSTM networks to predict and classify DDoS traffic based on sequential data analysis [15].

Hybrid and Real-Time Systems

Recent advancements have also explored hybrid models that combine several machine learning techniques to enhance detection accuracy and response times [16]. A study by Huang and Jemili et al., integrated decision trees with neural networks to form a hybrid model that utilizes the decision trees'

explanatory power and the neural networks' capability in handling non-linear relationships [17]. This hybrid approach not only improved detection rates but also reduced false positives, crucial for maintaining operational stability in IoT systems.
Moreover, the need for real-time detection systems in IoT environments has spurred the development of online learning models that adapt to new data without requiring retraining [18]. Abdelkaderet al. (2024) implemented an online gradient descent algorithm that continuously updates the model's weights as new data flows in, allowing for dynamic adaptation to changing attack patterns without significant downtime [19].

Challenges and Gaps
Despite these advances, several challenges persist in the field. The variability of IoT devices and the vastness of data they generate present unique challenges that are not fully addressed by existing models. Furthermore, the computational constraints of many IoT devices limit the feasibility of deploying complex models directly on the devices. This necessitates efficient model training and inference processes that are still under-explored in current research.Additionally, the evolving nature of DDoS attacks, including AI-driven attacks, poses a significant challenge. These advanced attacks can adapt and camouflage within normal traffic, making them difficult to detect with static models. The literature still lacks comprehensive solutions that effectively address these sophisticated threats in real-time while maintaining accuracy and efficiency.

3. METHODOLOGY

3.1 Dataset Overview

To enhancing and evaluating DDoS attack detection models specifically designed for IoT environments, this study employs the CICDIoT2023 dataset [20]. This dataset is provided by the Canadian Institute for Cybersecurity (CIC) and represents one of the latest comprehensive efforts to simulate real-world IoT network traffic, including both benign activities and various types of DDoS attacks. The CICDIoT2023 dataset is particularly tailored to address the complexities and challenges associated with securing IoT devices and networks against the increasingly sophisticated landscape of cyber threats.

Data Characteristics
The CICDIoT2023 dataset includes a variety of attack vectors that are prevalent in current cybersecurity threats to IoT networks, such as HTTP Flood, TCP SYN Flood, UDP Flood, and more sophisticated botnet attacks. This diverse compilation of attack types provides a robust platform for testing and validating DDoS detection algorithms. It also includes benign data that mimics normal traffic behavior from IoT devices, which is crucial for training models to accurately distinguish between normal and malicious traffic.

Data Features
Comprising detailed network traffic attributes, the dataset contains features such as packet size, packet timing, protocol type, flow bytes, and packet payloads. These features are essential for identifying potentially malicious patterns indicative of DDoS attacks. The richness and diversity of the dataset's features allow for deep analytical approaches and facilitate the development of models that can generalize well across different network behaviors and attack tactics.

Preprocessing Steps
- **Cleaning and Validation:** Initial preprocessing involves cleaning the data by removing any records with missing values or corrupt data points. Validation checks are also performed to ensure all data types conform to expected formats, crucial for subsequent analytical processes.
- **Feature Engineering:** Given the complex nature of network data, feature engineering plays a critical role. This includes deriving new features that may better capture the nuances of network behavior under attack conditions, such as aggregated flow statistics over time intervals and ratios of specific protocol usage.
- **Normalization:** To facilitate effective learning, especially in models sensitive to the scale of input features like neural networks and SVMs, feature values are normalized or standardized. They also help in ensuring that all features generate equal contribution towards the analysis, and do not have the models biased towards features that have the larger scales.
- **Balancing the Dataset:**The given data set is class imbalanced, that means number of attack instances in the dataset is much larger than that of benign instances. To overcome this issue, k-means

Synthetic Minority Over-sampling Technique (k-means-SMOTE) are used for creating artificial samples of the minority group.

- **Train-Test Split:** The final step in data preparation involves splitting the dataset into training and testing sets, typically in a 70:30 ratio. This separation is vital to train models on a substantial portion of the data while reserving a significant subset for unbiased evaluation of model performance.

The class imbalance in the CICDIoT2023 dataset is evident from the figure 1, showcasing a significant variation in the frequency of different types of network activities. The most common classes are DDoS-ICMP_Flood, DDoS-UDP_Flood, and DDoS-TCP_Flood, each exhibiting notably high instances which exceed 25,000 occurrences. This is indicative of a dataset rich in these specific attack vectors, possibly reflecting their prevalence in real-world scenarios or an emphasis on these attack types in dataset compilation.On the other end of the spectrum, classes like Uploading_Attack, Backdoor_Malware, and XSS are significantly underrepresented, with each having fewer than 20 instances. This stark disparity highlights the challenges in training machine learning models that can accurately detect less frequent but potentially more harmful attacks, as the models might tend to be biased towards detecting more frequent classes.Classes such as BenignTraffic and Mirai-related attacks (like Mirai-greeth_flood and Mirai-udpplain) have moderate representation, suggesting a balanced inclusion of normal and botnet-compromised IoT traffic. However, the minority classes like DictionaryBruteForce, SQLInjection, and CommandInjection exhibit such low frequencies that their detection might require specialized oversampling techniques or focused data collection to improve model sensitivity towards these types.

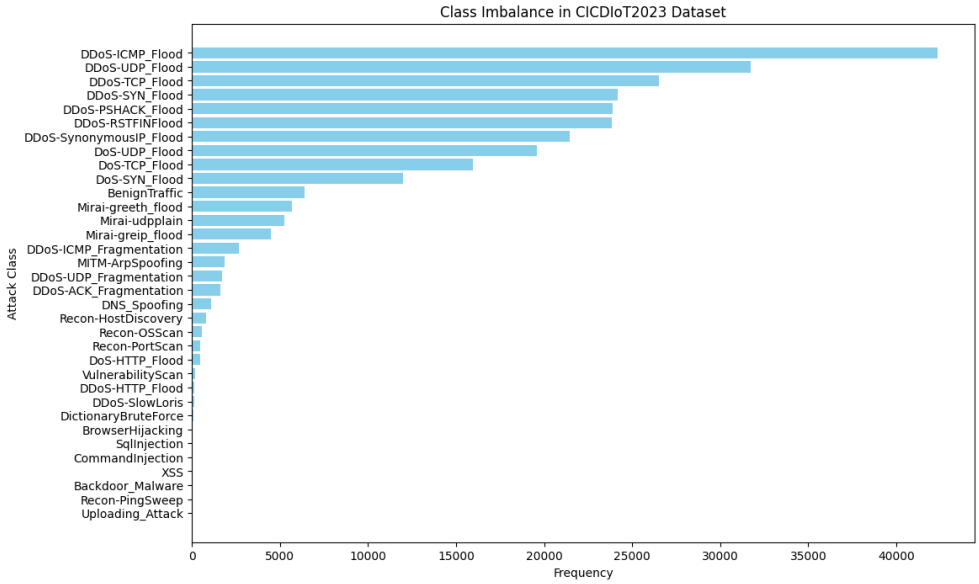

Figure 1. Distribution of class in CICDIoT2023 dataset

3.2 Model Selection

The selection of appropriate machine learning models is crucial for effectively detecting DDoS attacks within IoT environments. Given the nature of the data and the specific requirements of the task, a combination of ensemble methods, support vector machines, and neural networks was chosen. Each model category was selected based on its ability to handle high-dimensional data, its performance in similar security tasks, and its adaptability to hyperparameter tuning.

Ensemble Methods

Random Forest (RF)

Random Forest is a meta-classification method which works at training phase creating many Decision Trees and at testing phase outputs the class that is most frequently given by the individual trees. It can be noted that based on the general formulation of the decision tree, which is given by the equation 1, the form of the tree is generally complete.

$f(x) = \sum_{m=1}^{M} c_m \cdot 1_{\{x \in R_m\}}$ ..... (1)

where (x) represents the input features, (R_m) denotes the region of the m-th tree, (c_m) is the class prediction, and (M) is the number of trees.

RF is known for its high accuracy, robustness to overfitting, and feature importance metrics which are invaluable for understanding feature relevance in DDoS attack detection.

Gradient Boosting Machines (GBM)

Specifically, XGBoost (eXtreme Gradient Boosting) was used due to its efficiency and performance. GBM models are built in a stage-wise fashion as follows:

$F(x) = \sum_{k=1}^{K} \gamma_k h_k(x) + \text{const}$ ..... (2)

where $(h_k(x))$ are the basis functions, or weak learners, (γ_k) are the coefficients, and (K) is the number of boosting stages.

XGBoost applies regularization techniques to control over-fitting, making it highly effective for large datasets.

Support Vector Machine (SVM)

SVM is a strong classification kind of learning process that strives to identify the most appropriate separation line in case of the given dataset.The following equation depicts the functionality of SVM.

$f(x) = \text{sgn}\left(\sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b\right)$ ..... (3)

where (x_i) are the support vectors, (y_i) are the labels, (α_i) are the Lagrange multipliers, (K) is the kernel function, and (b) is the bias.

For DDoS detection, the radial basis function (RBF) kernel is particularly useful due to its ability to handle non-linear class boundaries.

Neural Networks

Deep neural networks (DNN) are included for their ability to model complex patterns in data. The architecture of a basic neural network can be described by the following series of transformations:

$y = \sigma(W_k \sigma(\cdots \sigma(W_2 \sigma(W_1 x + b_1) + b_2) \cdots) + b_k)$ ..... (4)

where (W_i) and (b_i) denote the weights and biases of the i-th layer, respectively, (σ) represents the activation function, and (y) is the output.

3.3 Hyperparameter Optimization

Tuning the parameters is very essential in the machine learning process, especially when designing models for application in the volatile contexts such as the IoT-based DDoS attack detection. This indeed comes as a big challenge since the goal of hyperparameter optimization is to search for the set of parameters that will give the best performance for a specific model. This research uses the grid search approach among the most effective strategies of evaluating the hyperparameters that are most suitable in improving the prediction of the model.

Grid Search Methodology

Grid search entails, selecting a grid of values for the hyperparameters of the model such that performance is cross-validated based on the selected combinations. The efficacy to achieve the objective of each model is evaluated by a given scoring function assess, that can be accuracy, the F1-score, or the AUC, applicable to classification problems.The process can be mathematically represented as:

Optimal Parameters $= \arg\max_{\theta \in \Theta} CV(X, y; \theta)$ ..... (5)

where (θ) represents the hyperparameters in the grid (Θ), (X) denotes the input features, (y) is the target variable, and (CV) signifies the cross-validation process used to evaluate the model performance.

Cross-Validation Strategy

To improve model's stability and reduce its tendency to memorize the data, the k-fold cross-validation approach was used with k = 5 or k = 10 more often. In this method the data is split into k smaller sets also called folds and the model is trained and validated on k-1 folds respectively. This is done for k times and each of the fold is used only once for the validation data. The mean of the performance across each one of the k folds is then used to assess the quality of a given hyperparameter specification. The cross-validation score can be expressed as:

$\text{CV Score} = \frac{1}{k} \sum_{i=1}^{k} f_i\left(X_{\text{train}_i}, y_{\text{train}_i}; X_{\text{val}_i}, y_{\text{val}_i}; \theta\right)$ ..... (6)

where f_i denotes the evaluation metric (e.g., accuracy), $X_{\text{train}_i}, y_{\text{train}_i}$ are the training data for the i-th fold, $X_{\text{val}_i}, y_{\text{val}_i}$ are the validation data for the i-th fold, and (θ) represents the hyperparameters being tested.

4. Experimental Setup

To implement and evaluate the machine learning models described in this research, a specific set of software tools and hardware resources were utilized. The hardware configuration provided the necessary computational power to handle the extensive simulations and data processing required for

hyperparameter optimization and model training in detecting DDoS attacks within IoT environments. Below is a detailed list of the hardware and software tools used in this study.

Hardware Configuration

- **RAM**: 64 GB
- **GPU**: NVIDIA RTX 4070 Ti 16 GB
- **CPU**: AMD Ryzen 7
- **Operating System**: Ubuntu 24 OS

Table 2. Software Tools

Tool/Software	Purpose	Version/Details
Python	Programming language	3.10
Scikit-learn	Machine learning library	1.2.0
XGBoost	Gradient boosting framework	1.7.1
Pandas	Data manipulation and analysis	1.5.2
NumPy	Numerical computing	1.23.4
Matplotlib	Plotting library	3.6.2
Seaborn	Statistical data visualization	0.12.0
TensorFlow	Deep learning framework	2.11.0
Keras	High-level neural networks API	Integrated within TensorFlow
CUDA	Parallel computing platform and API model	12.0 (compatible with RTX 4070)

5. RESULTS

The results of the machine learning models used for DDoS attack detection in IoT environments are presented below. Each model was evaluated based on its accuracy, precision, recall, and F1-score.Figure 2 provides an overview of how accurately each model predicts both attack and benign classes. It helps in understanding the overall effectiveness of the classifiers in the dataset. Figure 3 focuses specifically on the precision with which each model identifies DDoS attacks, indicating the proportion of positive identifications that were 'actually correct'. Figure 4illustrates the recall rate for attack detection, showing the ability of the models to capture all relevant instances. Figure 5 combines the insights of precision and recall into a single metric for attack classification, providing a balanced view of each model's performance in detecting attacks.Figure 6 shifts the focus to how precisely each model identifies benign traffic, essential for minimizing false positives. Figure 7 shows the recall rate for benign instances, indicating the effectiveness of the models in identifying all non-threatening activities. Lastly, Figure 8 offers a harmonized measure of precision and recall for benign classifications, providing a holistic view of how well each model handles normal traffic scenarios.

Table 1. Comparative results of ML models

Model	Accuracy	Precision (Attack)	Recall (Attack)	F1-Score (Attack)	Precision (Benign)	Recall (Benign)	F1-Score (Benign)
LDA	0.99	0.99	1.00	0.99	1.00	0.96	0.98
AdaBoost	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Gaussian Naive Bayes	0.975	0.99	0.98	0.98	0.95	0.96	0.96
Deep Neural Network	0.995	0.99	1.00	0.99	1.00	0.98	0.99
SVM	0.995	0.99	1.00	1.00	1.00	0.98	0.99
Random Forest	0.995	1.00	0.99	1.00	1.00	1.00	1.00
KNN	0.995	0.99	1.00	1.00	1.00	0.98	0.99
LGBM	1.00	1.00	1.00	1.00	1.00	1.00	1.00

- **AdaBoost and LGBM**: These models demonstrated perfect scores across all metrics, achieving an accuracy, precision, recall, and F1-score of 1.00. Their ability to adaptively enhance weak learners and focus on misclassified instances in training iterations contributes to their high performance in detecting both benign and attack classes.

- **Random Forest**: Nearly perfect in all categories, showing a slight discrepancy only in the recall for the attack class (0.99), which suggests that it failed to identify a minimal number of attack instances as such. Nevertheless, its ensemble method, which averages multiple deep decision trees, provided a robust performance against overfitting and variance in the dataset.
- **Deep Neural Network, KNN and SVM**: These models showed high accuracy and precision with scores of 0.995. Their capability to model complex non-linear relationships in high-dimensional data spaces makes them particularly effective for the dataset used, which includes a variety of DDoS attack vectors and benign scenarios.
- **Gaussian Naive Bayes**: While still performing well, this model had slightly lower scores compared to others, likely due to its assumption of feature independence, which may not hold true in complex IoT network traffic patterns.
- **LDA**: Exhibited high effectiveness with a minor reduction in recall for benign instances, indicating a small proportion of benign activities might have been classified incorrectly as attacks. However, its overall performance remains exceptionally high, making it a valuable model for initial screening in a security pipeline.

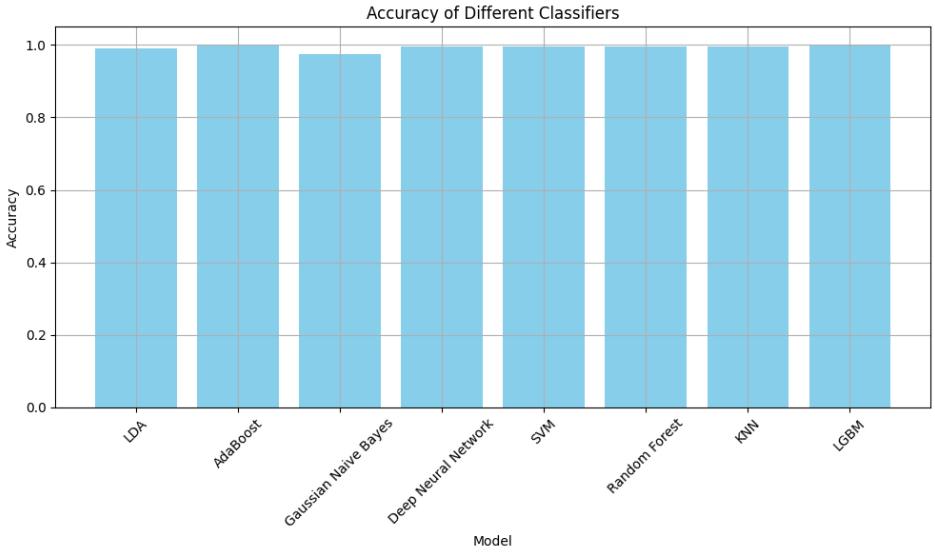

Figure 2. Accuracy of Different Classifiers

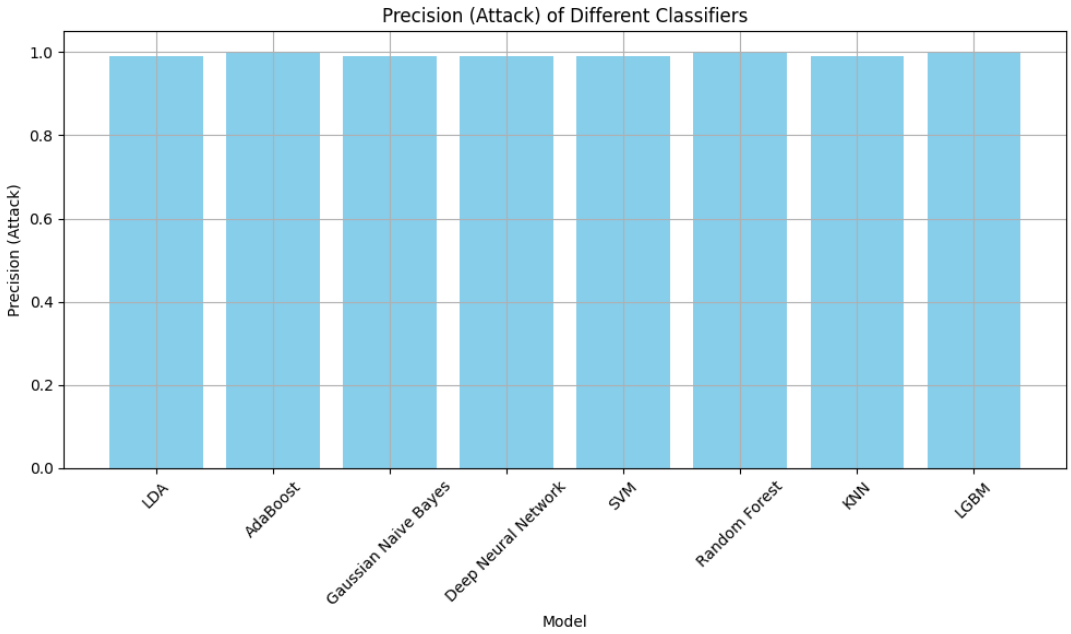

Figure 3. Precision (Attack) of Different Classifiers

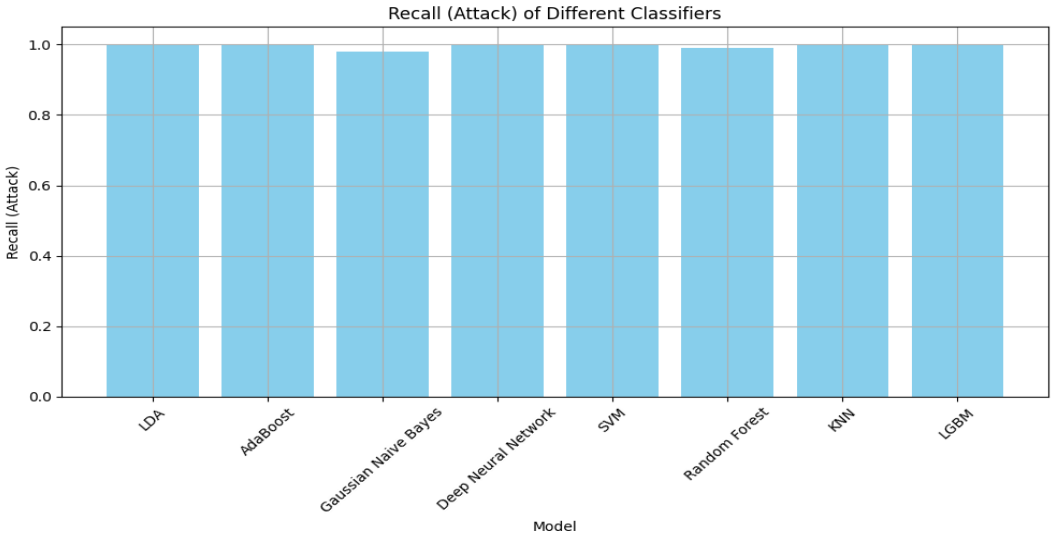

Figure 4. Recall (Attack) of Different Classifiers

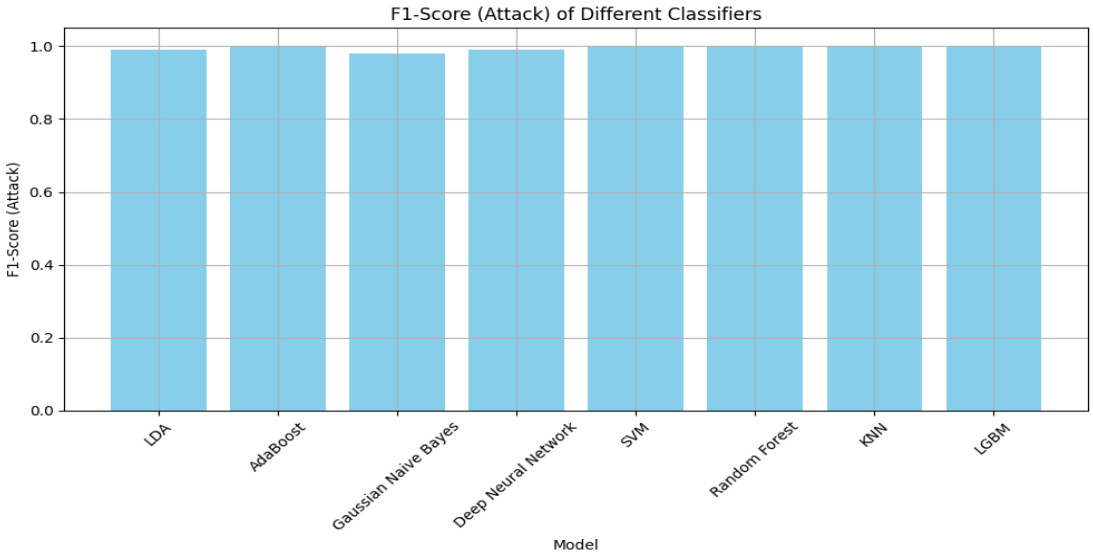

Figure 5. F1-Score (Attack) of Different Classifiers

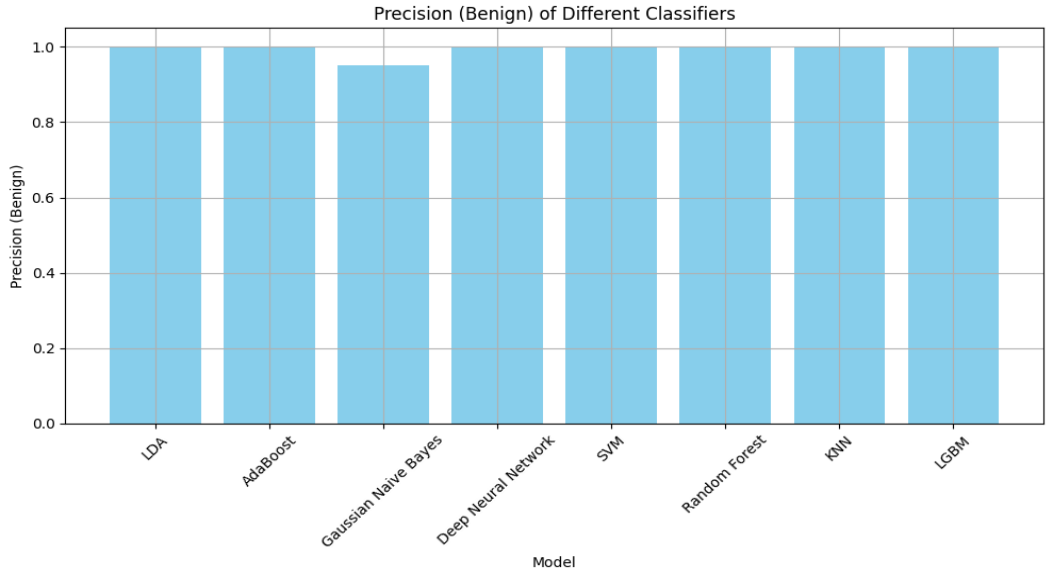

Figure 6. Precision (Benign) of Different Classifiers

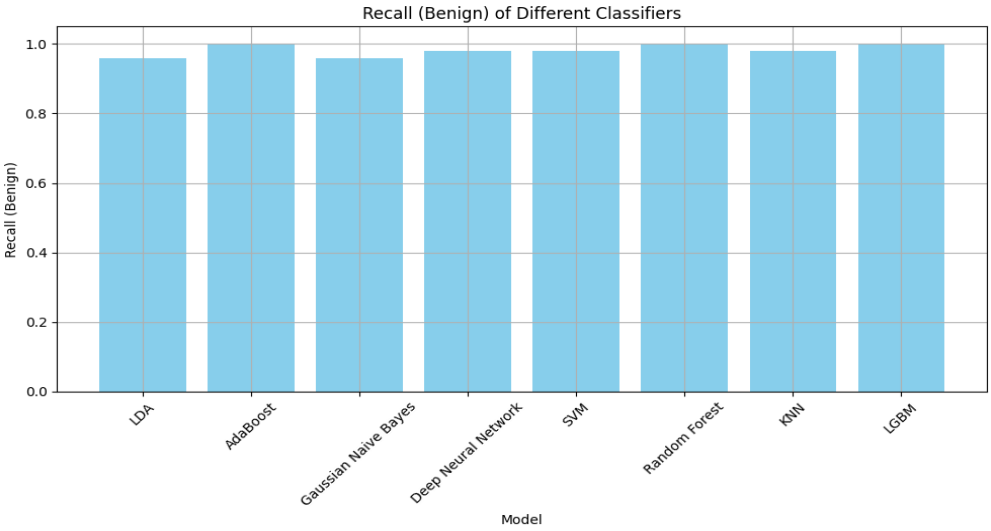

Figure 7. Recall (Benign) of Different Classifiers

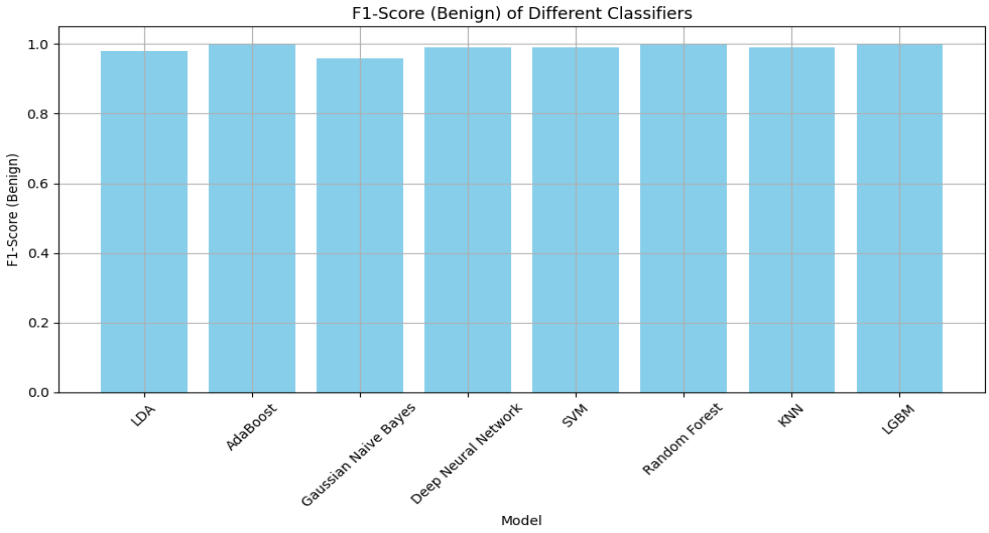

Figure 8. F1-Score (Benign) of Different Classifiers

6. CONCLUSION

This study has systematically investigated the application of various machine learning models to detect DDoS attacks in IoT environments, with a particular focus on the efficacy of hyperparameter tuning via grid search. The results quantitatively demonstrate that models such as AdaBoost and LGBM achieved perfect accuracy, precision, recall, and F1-scores of 1.00, underscoring their robustness and suitability for high-stakes security applications. Similarly, models like Random Forest, SVM, and Deep Neural Networks exhibited near-perfect performance metrics, with accuracy levels around 0.995, indicating their potential for effective deployment in diverse IoT scenarios.Qualitatively, the study reveals that ensemble methods and advanced machine learning techniques can adaptively handle the complex and varied nature of DDoS attack vectors characteristic of modern IoT frameworks. These models not only learn from vast and heterogeneous data but also demonstrate the ability to generalize well from training data to unseen real-world data, a critical requirement for any security-related application. The grid search method applied for hyperparameter optimization proved essential in enhancing model performance, ensuring that each model operated at its optimal parameter setting.However, the study acknowledges certain limitations, such as the computational intensity required for extensive grid searches and the potential for overfitting if not monitored carefully. In future work, we aim to explore the use of automated machine learning (AutoML) tools to streamline the hyperparameter tuning process, potentially reducing the computational overhead, and accelerating the deployment of optimized models.

REFERENCES

[1] Ahmed, S., & Khan, M. Securing the Internet of Things (IoT): A comprehensive study on the intersection of cybersecurity, privacy, and connectivity in the IoT ecosystem. AI, IoT and the Fourth Industrial Revolution Review,(2023), 13(9), 1-17.

[2] Olaniyi, O. O., Okunleye, O. J., Olabanji, S. O., &Asonze, C. U. IoT security in the era of ubiquitous computing: A multidisciplinary approach to addressing vulnerabilities and promoting resilience. Asian Journal of Research in Computer Science,(2023), 16(4).

[3] Alashhab, A. A., Zahid, M. S., Isyaku, B., Elnour, A. A., Nagmeldin, W., Abdelmaboud, A., ... &Maiwada, U. Enhancing DDoS attack detection and mitigation in SDN using an ensemble online machine learning model. IEEE Access, (2024).

[4] Al Lail, M., Garcia, A., & Olivo, S. Machine learning for network intrusion detection—a comparative study. Future Internet, (2023), 15(7), 243.

[5] Li, Q., Huang, H., Li, R., Lv, J., Yuan, Z., Ma, L., ... & Jiang, Y. (2023). A comprehensive survey on DDoS defense systems: New trends and challenges. Computer Networks, 109895.

[6] Palaniappan, K., Duraipandi, B., & Balasubramanian, U. M. (2024). Dynamic behavioral profiling for anomaly detection in software-defined IoT networks: A machine learning approach. Peer-to-Peer Networking and Applications, 17(4), 2450-2469.

[7] Ahmed, S. F., Alam, M. S. B., Hassan, M., Rozbu, M. R., Ishtiak, T., Rafa, N., ... &Gandomi, A. H. (2023). Deep learning modelling techniques: current progress, applications, advantages, and challenges. Artificial Intelligence Review, 56(11), 13521-13617.

[8] Aliferis, C., & Simon, G. (2024). Overfitting, Underfitting and General Model Overconfidence and Under-Performance Pitfalls and Best Practices in Machine Learning and AI. In Artificial Intelligence and Machine Learning in Health Care and Medical Sciences: Best Practices and Pitfalls (pp. 477-524). Cham: Springer International Publishing.

[9] Awad, M., &Fraihat, S. (2023). Recursive feature elimination with cross-validation with decision tree: Feature selection method for machine learning-based intrusion detection systems. Journal of Sensor and Actuator Networks, 12(5), 67.

[10] Mishra, N., & Pandya, S. (2021). Internet of things applications, security challenges, attacks, intrusion detection, and future visions: A systematic review. IEEE Access, 9, 59353-59377.

[11] Ismanto, E., Al Amien, J., &Vitriani, V. (2024). A Comparison of Enhanced Ensemble Learning Techniques for Internet of Things Network Attack Detection. MATRIK: JurnalManajemen, TeknikInformatikadanRekayasaKomputer, 23(3), 543-556.

[12] Awotunde, J. B., Chakraborty, C., & Adeniyi, A. E. (2021). Intrusion Detection in Industrial Internet of Things Network-Based on Deep Learning Model with Rule-Based Feature Selection. Wireless communications and mobile computing, 2021(1), 7154587.

[13] Jayalaxmi, P. L. S., Kumar, G., Saha, R., Conti, M., Kim, T. H., & Thomas, R. (2022). DeBot: A deep learning-based model for bot detection in industrial internet-of-things. Computers and Electrical Engineering, 102, 108214.

[14] Hekmati, Arvin, Jiahe Zhang, Tamoghna Sarkar, Nishant Jethwa, Eugenio Grippo, and BhaskarKrishnamachari. "Correlation-aware neural networks for DDOS attack detection in IoT systems." IEEE/ACM Transactions on Networking (2024).

[15] Parmar, A., &Lamkuche, H. (2023, February). Distributed Denial of Service Attack Detection Using Sequence-To-Sequence LSTM. In The International Conference On Global Economic Revolutions (pp. 39-53). Cham: Springer Nature Switzerland.

[16] Azevedo, Beatriz Flamia, Ana Maria AC Rocha, and Ana I. Pereira. "Hybrid approaches to optimization and machine learning methods: a systematic literature review." Machine Learning (2024): 1-43.

[17] Jemili, Farah, RahmaMeddeb, and OuajdiKorbaa. "Intrusion detection based on ensemble learning for big data classification." Cluster Computing 27, no. 3 (2024): 3771-3798.

[18] Shahin, Mohammad, MazdakMaghanaki, Ali Hosseinzadeh, and F. Frank Chen. "Advancing Network Security in Industrial IoT: A Deep Dive into AI-Enabled Intrusion Detection Systems." Advanced Engineering Informatics 62 (2024): 102685.

[19] Abdelkader, Sobhy, Jeremiah Amissah, Sammy Kinga, GeofreyMugerwa, Ebinyu Emmanuel, Diaa-Eldin A. Mansour, Mohit Bajaj, Vojtech Blazek, and Lukas Prokop. "Securing Modern Power Systems: Implementing Comprehensive Strategies to Enhance Resilience and Reliability Against Cyber-Attacks." Results in Engineering (2024): 102647.

[20] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, A. A. Ghorbani. "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," Sensor (2023).