

# ENHANCING DDOS DETECTION IN IOT SYSTEMS THROUGH BOOSTING TECHNIQUES

POORNIMA@PRIYANKA.R<sup>1</sup>, KARUPPURAJA.S<sup>2</sup>, MOHAN RAJ.P<sup>3</sup>, LAVANESH.T.M<sup>4</sup>

<sup>1</sup>Assistant Professor 1, Department of Computer Science and Engineering,  
K.L.N. College of Engineering, Sivagangai, India

<sup>2,3,4</sup>Student, Department of Computer Science and Engineering,  
K.L.N. College of Engineering, Sivagangai, India

\*\*\*

**Abstract**-Distributed Denial of Service (DDoS) attacks pose in both traditional networks and in Internet of Things (IoT), including home networks. This paper proposes the application of DDoS traffic detection model leveraging boosting algorithms to address this critical issue. With the IoT devices, attackers exploit vulnerabilities of botnet networks, amplifying the impact of DDoS attacks. Machine learning techniques algorithms is used for DDoS detection, categorized into supervised and unsupervised approaches. Despite advancements in Machine Learning (ML) and deep learning, DDoS attacks remain a significant threat to the integrity and availability of the Internet. Our proposed model utilizes boosting learning classification algorithms to analyze network data and identify malicious traffic patterns. Evaluation of the detection model relies on publicly available datasets, ensuring robustness and generalizability. The primary objective of this project is to develop effective algorithms for identifying and mitigating DDoS attacks within networks. Additionally, as social networks continue to grow exponentially, detecting attacks within these platforms presents a complex challenge. To address this, our research involves the development and comparison of four distinct machine and deep learning algorithms for DDoS detection. Through this research, we aim to contribute to the ongoing efforts to safeguard network against DDoS attacks, thereby enhancing the security of the Internet ecosystem.

**KeyWords**-Distributed Denial of Service attacks, Internet of Things, Boosting algorithms, Machine learning techniques, Four distinct machine and deep learning algorithms

## 1. INTRODUCTION

IoT devices are increasingly targeted by attackers due to vulnerabilities in their software and hardware, leading to the creation of botnets for DDoS attacks. As IoT devices in homes and organizations, their minimal security measures make them easy targets. Stakeholders, including users and service providers, have a vested interest in preventing IoT devices from being exploited. Efforts are underway to

develop datasets and machine learning models for effective attack detection and feature selection.

Distributed Denial of Service threats involves deploying machine learning-based detection frameworks, particularly for botnet attacks prevalent in IoT environments. The Botnets, controlled by a single entity, can consist of thousands of compromised devices operating covertly. Machine Learning techniques and Deep Learning, offer solutions for network detection in IoT systems. However, the IoT devices require efficient feature selection and lightweight detection systems to detect high-dimensional traffic data and sophisticated attacks. UNWS dataset aim to provide labeled data for supervised learning, essential for accurate detection in IoT networks.

The previous Researchers developed a framework for predicting DDoS attack types using machine learning, employing Random Forest and XG-Boost algorithms. They utilized the UNWS-np-15 dataset, obtained from GitHub, and implemented the framework in Python. After model deployment, performance is evaluated using confusion matrices. Results indicated Random Forest achieved Precision and Recall of approximately 89%, with an average Accuracy of around 89%. XG-Boost yielded Precision and Recall of about 90%, with an average Accuracy of approximately 90%. Comparatively, their model significantly outperformed existing research, which reported defect determination accuracies of approximately 85% and 79%.

This system utilizes the UNWS dataset sourced from its repository as input. Data preprocessing is conducted to handle missing values and encode input data labels, essential for accurate predictions. The dataset is then divided into training and testing sets based on a specified ratio, with the majority allocated to training and a smaller portion to testing. Training data is used to evaluate the model, while testing data is employed for prediction. Classification algorithms, including both machine learning

(e.g., MLP, KNN, RF, AdaBoost) and deep learning methods (e.g., RNN), are implemented. The results showcase accuracy, precision, recall. The output shows the graphical comparisons are drawn to evaluate the results of the algorithms.

## 2. LITERATURE REVIEW

Distributed Denial of Service attacks pose a significant threat to computing resources and users. While researchers have combined classification algorithms with distributed computing to counter these attacks, their solutions are often static and unable to keep up with evolving attack tactics. To address this, we propose a dynamic DDoS detection system comprising classification algorithms, a distributed infrastructure, and fuzzy logic. Our system dynamically selects from Naive Bayes, Decision Trees (Entropy and Gini), and Random Forest to detect diverse DDoS patterns. Through evaluation, we validate the effectiveness of our fuzzy logic system in selecting algorithms based on traffic status, noting a trade-off between accuracy and delay. Additionally, we assess the impact of the distributed system on algorithmic delays. Overall, our approach offers adaptability in combating dynamic DDoS attacks within a concise framework. advancement in the ongoing battle against cyber threats. By harnessing the synergistic power of classification algorithms, distributed computing, and fuzzy logic, we offer a formidable defense mechanism capable of thwarting even the most cunning and elusive DDoS attacks. As cyber adversaries continue to evolve their tactics, our dynamic approach ensures that defenders remain one step ahead, safeguarding the availability and integrity of critical computing resources.

The advancements in hardware have significantly expanded the data parallelism capabilities for neural network training. The increasing batch size, a straightforward method implemented to leverage next-generation hardware, is explored in this study. We aim to characterize the impact of batch size on training time, measured by the steps needed to achieve a target out-of-sample error. The investigation spans various training algorithms, models, and datasets, revealing substantial variation across workloads.

In literature regarding batch size's effect on model quality are attributed to differences in meta parameter tuning and compute budgets at different batch sizes. Importantly, there is no evidence suggests that larger batch sizes compromise out-of-sample performance. Implications for accelerating neural network training in the future are discussed. Our study offers insights supported by a vast

dataset of 71,638,836 loss measurements across 168,160 models and 35 workloads, available for further analysis.

Distributed Denial of Service (DDoS) flooding attacks pose severe threats to online service availability by inundating victims with massive volumes of traffic, disrupting normal communication or causing complete system crashes.

Detecting these attacks promptly is crucial, as delays can exacerbate damage and necessitate manual intervention. In this paper, we introduce HADEC, a Hadoop-based Live DDoS Detection framework designed to efficiently analyze flooding attacks using MapReduce and HDFS.

We develop a counter-based DDoS detection algorithm for four major flooding attack types (TCP-SYN, HTTP GET, UDP, and ICMP) in MapReduce. Additionally, we deploy a testbed to assess the performance of HADEC in real-time DDoS detection scenarios. These experiments demonstrate that HADEC can effectively process and detect DDoS attacks within reasonable timeframes.

## 3. MODULES

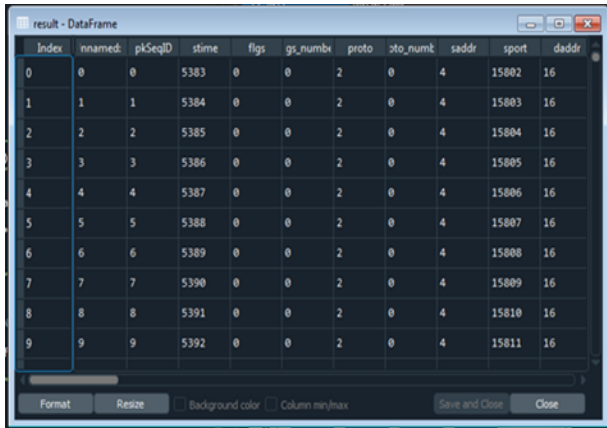
### 3.1) DATA SELECTION

The input data utilized in our study was sourced from a dataset repository, with a particular focus on the UNWS dataset. This dataset served as the foundation for our analysis, which primarily centered on the process of identifying malicious traffic. The UNWS dataset encompasses a diverse range of network traffic, including Internet of Things (IoT) data, normal traffic flows, and various cyber-attack traffic flows, notably those associated with botnet attacks.

To ensure the accuracy and effectiveness of our analysis, we employed a realistic test bed to develop the dataset, incorporating informative features that reflect real-world network behavior. Additionally, in our pursuit of enhancing the performance of deep learning models and prediction accuracy, we expanded the feature set by extracting and incorporating additional relevant features. Furthermore, to facilitate improved model performance and enhance the interpretability of results, the extracted features underwent labeling, including classification into attack flow, categories, and subcategories.

This comprehensive approach to data selection, feature extraction, and labeling underscores our commitment to developing robust and effective predictive models for detecting and mitigating cyber threats in network environments. This process helps in reducing noise, biases, and irrelevant information, thus improving the quality and

reliability of the results obtained from data analysis or research. This process aims to ensure that the data collected aligns with the objectives and requirements of the project, providing accurate, reliable, and meaningful insights or results.



Index	Unnamed	pkSeqID	stime	flgs	gs_numbr	proto	pto_numtr	saddr	sport	daddr
0	0	0	5383	0	0	2	0	4	15802	16
1	1	1	5384	0	0	2	0	4	15803	16
2	2	2	5385	0	0	2	0	4	15804	16
3	3	3	5386	0	0	2	0	4	15805	16
4	4	4	5387	0	0	2	0	4	15806	16
5	5	5	5388	0	0	2	0	4	15807	16
6	6	6	5389	0	0	2	0	4	15808	16
7	7	7	5390	0	0	2	0	4	15809	16
8	8	8	5391	0	0	2	0	4	15810	16
9	9	9	5392	0	0	2	0	4	15811	16

Figure-1: Data Selection

### 3.2) DATA PREPROCESSING

Data pre-processing collects the datasets for machine learning tasks by eliminating unwanted data and transforming it into a suitable structure.

This process involves various operations aimed at enhancing the dataset's quality and efficiency.

Firstly, data transformation operations are applied to reshape the dataset into a format conducive to machine learning algorithms.

Additionally, cleaning the dataset involves removing irrelevant or corrupted data that could compromise accuracy.

This includes addressing missing data, where null or NaN values are replaced with 0 to maintain dataset integrity. Furthermore, missing and duplicate values are systematically removed to ensure data cleanliness and consistency.

Another key aspect of pre-processing is encoding categorical data, which involves converting variables with a finite set of label values into numerical format, as many machine learning algorithms require numerical input and output variables.

By meticulously executing these pre-processing steps, we enhance the dataset's usability and facilitate more effective machine learning model training and evaluation

```
-----Checking Missing Values-----
Unnamed: 0    0
pkSeqID      0
stime        0
flgs         0
flgs_number  0
proto        0
proto_number 0
saddr        0
sport        0
daddr        0
dtype: int64
```

```
-----Before Label Encoding-----
Unnamed: 0  pkSeqID  stime  ...  attack  category  subcategory
0  1650261  1650261  1.528103e+09  ...  1  DDoS  HTTP
1  1650262  1650262  1.528103e+09  ...  1  DDoS  HTTP
2  1650263  1650263  1.528103e+09  ...  1  DDoS  HTTP
3  1650264  1650264  1.528103e+09  ...  1  DDoS  HTTP
4  1650265  1650265  1.528103e+09  ...  1  DDoS  HTTP
5  1650266  1650266  1.528103e+09  ...  1  DDoS  HTTP
6  1650267  1650267  1.528103e+09  ...  1  DDoS  HTTP
7  1650268  1650268  1.528103e+09  ...  1  DDoS  HTTP
8  1650269  1650269  1.528103e+09  ...  1  DDoS  HTTP
9  1650270  1650270  1.528103e+09  ...  1  DDoS  HTTP
[10 rows x 47 columns]
```

```
-----After Label Encoding-----
Unnamed: 0  pkSeqID  stime  ...  attack  category  subcategory
0  0  0  5383  ...  1  0  0
1  1  1  5384  ...  1  0  0
2  2  2  5385  ...  1  0  0
3  3  3  5386  ...  1  0  0
4  4  4  5387  ...  1  0  0
5  5  5  5388  ...  1  0  0
6  6  6  5389  ...  1  0  0
7  7  7  5390  ...  1  0  0
8  8  8  5391  ...  1  0  0
9  9  9  5392  ...  1  0  0
[10 rows x 47 columns]
```

Figure-2: Data Preprocessing

### 3.3) CLASSIFICATION

In This process, we implement various machine and deep learning algorithms to address the task at hand, including Multilayer Perceptron (MLP), K-Nearest Neighbors (KNN), Random Forest (RF), AdaBoost, and Recurrent Neural Networks (RNN). Each of these algorithms serves a distinct purpose and offers unique advantages in solving different types of problems. Starting with Multilayer Perceptron (MLP), it belongs to the class of feedforward artificial neural networks (ANNs).

The term MLP is sometimes used broadly to refer to any feedforward ANN, but more strictly, it denotes networks comprising multiple layers of perceptron's. These networks consist of an input layer, one or more hidden layers, and an output layer.

MLPs are adept at learning complex patterns and relationships in data, making them suitable for a wide range of applications, including classification and regression tasks. K-Nearest Neighbors (KNN) is a supervised machine learning algorithm used for both classification and regression tasks.

The algorithm operates by assigning a new data point to the class that is most common among its k nearest neighbors, where k is a user-defined parameter.

KNN is a non-parametric algorithm, meaning it does not make any assumptions about the underlying data distribution. It is simple to implement and understand, making it a popular choice for various classification tasks. Random Forest (RF) is a classification algorithm comprised of multiple decision trees.

It leverages the technique of bagging (bootstrap aggregating) and feature randomness to construct an ensemble of trees. Each tree in the forest is built using a random subset of the training data and a random subset of features, reducing the correlation between individual trees and improving the overall accuracy of predictions' is known for its robustness against overfitting and its ability to handle large datasets with high dimensionality.

Adaptive Boosting is an ensemble boosting classifier that combines multiple weak learners to create a strong learner with improved accuracy. The algorithm works iteratively, adjusting the weights of misclassified data points at each iteration to focus on the most challenging instances. By iteratively improving the model's performance, AdaBoost can achieve high accuracy even with simple base learners.

Recurrent Neural Networks are a type of neural network architecture designed to handle sequential data. Unlike feedforward networks, RNNs have connections that form directed cycles, allowing information to persist over time.

This recurrent structure enables RNNs to model temporal dependencies in data, making them suitable for tasks such as sequence classification (e.g., sentiment analysis, video classification) and sequence labeling (e.g., part-of-speech tagging, named entity recognition).

Overall, each of these machine and deep learning algorithms offers unique capabilities and is well-suited for specific types of problems.

By understanding their characteristics and applying them judiciously, we can leverage their strengths to effectively tackle various tasks in machine learning and data analysis

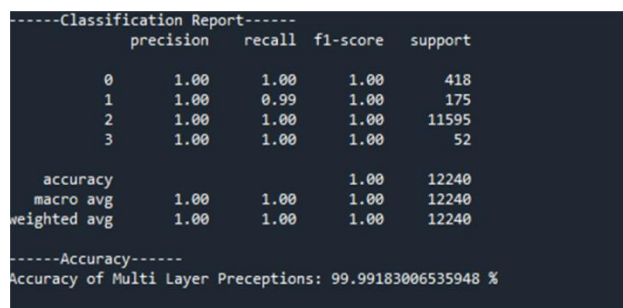


Figure-3:Multi-Layer Perceptron (MLP)

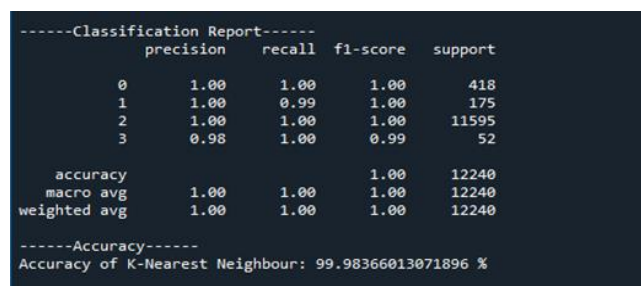


Figure-4:K-Nearest Neighbor (KNN)

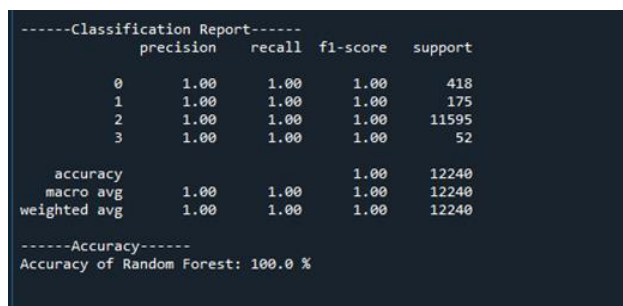


Figure-5:Random Forest (RF)

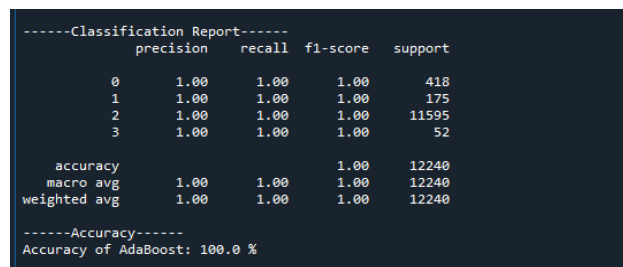


Figure-6:Ada boost

```

-----Classification Report-----
              precision    recall  f1-score   support

     0           1.00         1.00         1.00         418
     1           1.00         1.00         1.00         175
     2           1.00         1.00         1.00       11595
     3           1.00         1.00         1.00          52

 accuracy          1.00         1.00         1.00       12240
 macro avg          1.00         1.00         1.00       12240
 weighted avg          1.00         1.00         1.00       12240

-----Accuracy-----
Accuracy of AdaBoost: 100.0 %
    
```

```

None
Epoch 1/5
37/37 [=====] - 5s 29ms/step - loss: 2.0646 - accuracy: 0.0306
Epoch 2/5
37/37 [=====] - 1s 27ms/step - loss: 1.9252 - accuracy: 0.0311
Epoch 3/5
37/37 [=====] - 1s 28ms/step - loss: 1.9252 - accuracy: 0.0311
Epoch 4/5
37/37 [=====] - 1s 26ms/step - loss: 1.9252 - accuracy: 0.0311
Epoch 5/5
37/37 [=====] - 1s 28ms/step - loss: 1.9252 - accuracy: 0.0311
574/574 [=====] - 2s 2ms/step - loss: 1.9252 - accuracy: 0.0311
-----Accuracy-----
Accuracy of RNN: 61.10021725296974 %
    
```

Figure-7: Recurrent Neural Network (RNN)

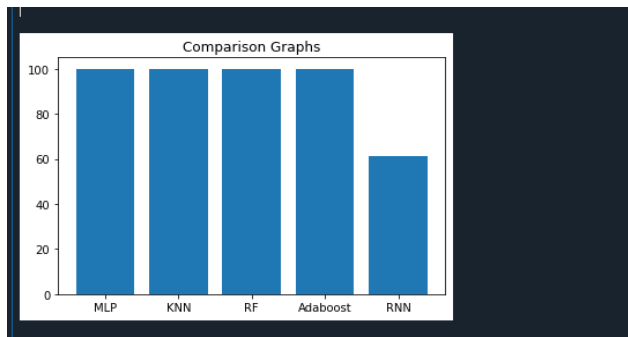


Figure-8: Comparison Graph

#### 4. CONCLUSION

Our study utilized the UNWS dataset to analyze IoT device behavior. We employed a variety of classification algorithms, including MLP, KNN, Random Forest, AdaBoost for traditional machine learning, and RNN for deep learning. We evaluated their performance based on accuracy and presented comparison graphs to illustrate their effectiveness. This comprehensive analysis offers valuable insights into the strengths of each approach when applied to the UNWS dataset, aiding future research in this domain.

#### 5. FUTURE ENHANCEMENT

In the future, there is a desire to enhance detection performance by combining of machine learning or deep learning algorithms into a multi-layered model. This hybrid approach aims to improve accuracy and reliability in

detecting patterns and anomalies within datasets. While the study focused on analyzing a dataset from a single network, future research could explore datasets from both larger and smaller network areas to gain insights into algorithm performance across varying network scales. As systems increasingly utilize analytics and data networks for predictive purposes, the use of machine learning or deep learning techniques to make predictions will become more prevalent. Overall, the future holds potential for leveraging advanced techniques to optimize network performance and enhance security.

#### REFERENCES

- [1] International Energy Agency - IEA, «World Energy Balances 2019,» IEA Publications & Data, Paris, 2019.
- [2]Empress de Pesquisa Energetic, «Balance Energetic National 2018.
- [3]Empress de Pesquisa Energetic, «Balance Energetic National 2019: and base 2018,» Ministerial de Minas e Energies, Rio de Janeiro, 2019
- [4]Y.-C. Lin, M.-H. Hung, H.-C. Huang, C.-C. Chen, H.-C. Yang, Y.-S. Hsieh, and F.-T. Cheng, “Development of advanced manufacturing cloud of things (amcot)- a smart manufacturing platform,” IEEE Robotics and Automation Letters, vol. 2, no. 3, pp. 1809–1816, 2017.
- [5]J. Qiu, Z. Tian, C. Du, Q. Zuo, S. Su, and B. Fang, “A survey on access control in the age of internet of things,” IEEE Internet of Things Journal, 2020.
- [6]K. Lab. (2019) Amount of malware targeting smart devices more than doubled in. [Online].
- [7] J. Qiu, L. Du, D. Zhang, S. Su, and Z. Tian, “Nei-tte: Intelligent traffic time estimation based on fine-grained time derivation of road segments for smart city,” IEEE Transactions on Industrial Informatics, 2019.
- [8]J. P. Anderson, “Computer security threat monitoring and surveillance, 1980. Lastaccessed:Novmeber-30, 2008.
- [9] D. E. Denning, “An intrusion-detection model,” IEEE Transactions on software engineering, no. 2, pp. 222–232, 1987.
- [10] A. K. de Souza y C. E. de Farias, Bioethanol in Brazil: Status, Challenges and Perspectives to Improve the Production,» de Bioethanol Production from Food Crops, Academic Press, 2019, pp. 417-443