Security Measures at Network Layer of IoT Using Machine Learning Techniques

Ms. Varkha K. Jewani (Pragati V. Thawani) [1]
Dr. Prafulla E Ajmire[2*]
Dr. Mohammad Atique Mohammad Junaid[3*]
Dr. Zeba Atique Shaikh[4*]

[1]Research Scholar, Computer Science Dept., Sant Gadge Baba Amravati University, Amravati, India
[2] Supervisor, Computer Science Dept., Sant Gadge Baba Amravati University, Amravati, India*
[3] Professor & Head, Department of Computer Science, Faculty of Engineering & Technology, Sant Gadge Baba Amravati University, India.*
[4] Research Scholar, Computer Science & Engineering, Sant Gadge Baba Amravati University, India.*

Abstract— The world is changing dramatically from discrete systems to ubiquitous "things" with Internet connectivity that can share data and produce analytically-derived insights. The highly integrated global network structure known as the Internet of Things. But compared to other kinds of equipment, IoT devices are more susceptible to attacks because of their small size and constrained memory, processing power, battery life, and other resources. Security and privacy pose several challenges, making them the most essential considerations for Internet of Things applications. Through an analysis of security architecture and key technologies, including encryption, cryptographic techniques, and communication security, as well as a description of the challenges they present. Numerous security issues are addressed by machine learning (ML) techniques The benchmark dataset from CICIDS 2017 is used in this investigation. The Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv file from this dataset has been used to train and test the three-machine learning (ML) models: Linear Regression, Random Forest, and Decision Tree. There are 79 columns and 225745 rows in this CSV file. The identical set of data was used to evaluate all three techniques, and the results show that while linear regression is the fastest, it is also the least accurate. Random Forest is the slowest of the three but provides higher accuracy. In the end, decision trees are thought to be the most effective approach for these kinds of problems because they provide a reasonable balance between speed and accuracy. The purpose of hybrid algorithms is to generate more precise results. Fitting and residual graphs, among other important visualizations, are displayed by this machine learning model.

Keywords: Internet of Things, Security & Privacy issues, Machine Learning Classifiers, Hybrid Algorithm.

 https://doi.org/10.5281/zenodo.14093582

I.INTRODUCTION

Today, millions of individuals utilize the Internet for a variety of purposes depending on their needs and view it as a basic need. People utilize the Internet for a variety of things, including games, music, and movie amusement as well as meeting essential everyday necessities. This implies that a substantial portion of the population uses the Internet due to its widespread use and advantages. [1, 2]. The ability to plan and connect with people anywhere in the world over the Internet is another factor contributing to the growth of Internet users. Because it offers several advantages, the Internet of Things (IoT), a recently formed and rapidly expanding industry, enables machines and other items to connect and communicate with each other when the Internet is available [3]. This new technology's primary goals are to automate tasks and connect popular consumer devices to the Internet. Every object has specialized sensors attached to it that collect data from the real world and send it to the virtual one. Analysis is used to remove extraneous information from the data, which is then saved locally. Every object sends it's connected and acquired data to cloud storage, which gets it from local storage. The information acquired is then used to take the appropriate action. It is feasible to manage and control equipment and things remotely and use this information to retain records for later use, even when acting on it is not always necessary [4]. The four primary components of the Internet of Things are sensing, heterogeneous access, information processing, applications, and services. There are more elements added, such as privacy and security. Additionally, there will be corporate applications associated to the Internet of Things, including cyber-physical systems (CPS), cyber-transportation systems (CTS), and machine-to-machine (M2M) communications [5]. The terms "Internet" and "Things" were combined to create the term "Internet of Things," which is commonly abbreviated as "IoT." The standard Internet protocol suite (TCP/IP) is used by the Internet, a global network of interconnected computer networks that serves billions of people globally. Numerous electrical, wireless, and optical networking technologies connect millions of private, public, academic, corporate, and government networks on a local to global scale [6]. This implies that objects can be both living (such as people and animals like cows, calfs, dogs, pigeons, and rabbits) and non-living (such as chairs, refrigerators, tube lights, curtains, dishes, and other household or business equipment). Things are therefore now actual physical objects in this domain of matter [7].

Figure 1 represents IoT applications where it mentioned things that are actual physical objects in this domain also, it displays a few uses for the Internet of Things. Numerous sensors, each with a unique function, are dispersed across the city to manage a variety of tasks like waste management, traffic control, streetlight optimization, water conservation, energy expenditure monitoring, the creation of smart buildings, and more.

 https://doi.org/10.5281/zenodo.14093582

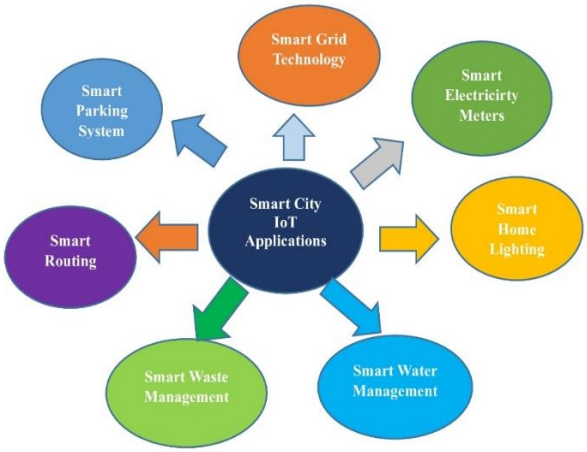

Fig. 1: IoT Applications

"An open and comprehensive network of intelligent objects that have the capacity to auto-organize, share information, data and resources, reacting and acting in face of situations and changes in the environment".

A combination of sensors, real objects, controllers, and actuators were used to compose the IoT components into an equation form [8]. The phrase "Internet of Things (IoT)" refers to a few ideas, including the expansion of the internet, the web as a physical environment, the use of widely distributed embedded devices, transmission, and actuation capabilities [9]. The 2020 conceptual framework states that the term "Internet of Things" (IoT) is expressed using a simple formula such as

IoT= Services+ Data+ Networks + Sensors [10]

The IoT four key technological enablers are: -

- RFID technology used for tagging the things.
- Sensor technology used for sensing the things.
- Smart technology used for thinking the things.
- Nanotechnology used for shrinking the things. [11]

There will be more challenging security issues with IoT. All of these "things" will have connections to this "internet," enabling them to speak with one another. By use of the mobile network, sensor network, conventional internet, and other networks, the Internet of Things increases the boundaries of the "internet". New privacy and security issues will arise as a result, and study into the integrity, authenticity, and confidentiality of IoT data will become more essential [12]. The initial IoT architecture did not include ambient intelligence or autonomous control. The concepts of IoT and autonomous control have been increasingly included into M2M research due to advancements in

 https://doi.org/10.5281/zenodo.14093582

distributed multi-agent control, cloud computing, and better network approaches. This has resulted in the transition of M2M into CPS. Distributed real-time control, intelligent zing interaction, interactive applications, cross-layer, and cross-domain optimization, etc. are some of the key areas of focus for CPS. New protocols and technologies need to be created to address the increasing demands for dependability, security, and privacy [13]. The volume of data produced by Internet of Things devices is so great that traditional techniques for gathering, storing, and analysing data may not work well in this context. In addition, one can use the sheer amount of data to spot patterns and behaviours, forecast outcomes, and carry out evaluations. The diversity of the data produced by IoT also creates new opportunities for the data processing techniques now in use. One of the computing paradigms that is most suitable in this case to provide embedded intelligence in the Internet of Things devices is Machine Learning (ML) [14]. Intelligent equipment and devices can benefit from machine learning (ML) to help them make inferences about their environment. It may also be described as a smart device's capacity to respond to knowledge and modify or automate a situation or behaviour; this is considered a crucial component of an Internet of Things solution. Furthermore, cross-layer architecture and optimal algorithms are needed to address the security and privacy concerns on the Internet of Things. For instance, IoT devices will require new types of specialized encryption and other methods to address security and privacy because of processing limits. Conversely, the sheer number of IoT devices creates additional challenges for security protocols. Discrete solutions are impossible for most complex security concerns. For example, while handling security issues like DDoS or penetration, there's a potential that false positives will happen, rendering the remedies ineffective against these attacks. Furthermore, it will undermine customer confidence, reducing the effectiveness of these solutions. A thorough security and privacy approach will address IoT security concerns by utilizing both current security solutions and the development of fresh, clever, dependable, evolutionary, and scalable techniques. Classification, regression, and density estimation are a few of the applications that have made use of machine learning techniques. Machine learning algorithms and techniques are used in many domains, including computer vision, fraud detection, bioinformatics, and virus detection, authentication, and speech recognition. Similarly, ML can be used by IoT to provide intelligent services. The use of machine learning to IoT networks for the purpose of providing security and privacy services is the main topic of discussion [15].

II. LITERATURE REVIEW

- Machine Learning-Based DDoS Detection for IoT: A Comprehensive Survey" (2021):
  This survey provides a comprehensive overview of machine learning-based DDoS detection

 https://doi.org/10.5281/zenodo.14093582

techniques in IoT. Decision Tree algorithms are discussed as a viable option and their performance compared to other methods. The review highlights the need for adaptive and lightweight models in the context of resource constrained IoT devices. [16]

- “Decision Tree-Based Approach for DDoS Attack Detection in IoT Networks” (2019):
  This research paper proposes a decision tree-based approach for DDoS detection in IoT networks. The authors’ present detailed analysis of how decision trees can effectively classify network traffic. They emphasize the simplicity and interpretability of decision tree models in the IoT context. [17]
- "An IoT-Based DDoS Detection System Using Ensemble Learning" (2020):
  This study focuses on ensemble learning methods for DDoS detection in IoT and includes decision tree algorithms as a key component. The review highlights the advantages of combining decision trees with other classifiers to improve accuracy and robustness in DDoS detection. [18]
- "DDoS Attack Detection in IoT: A Machine Learning Approach" (2017): This paper explores machine learning techniques for DDoS attack detection in IoT and provides an in-depth analysis of decision tree algorithms. It discusses how decision trees can be used to build lightweight and efficient models suitable for resource constrained IoT devices. [19].
- "Anomaly-Based Intrusion Detection in IoT Smart Home Networks Using Decision Trees" (2018): While not specific to DDoS attacks, this study focuses on intrusion detection in IoT environments using decision trees. It emphasizes the adaptability and ease of implementing decision tree models to detect anomalies in IoT network traffic, which can be applied to DDoS detection.[20]
- "DDoS Detection and Mitigation for IoT Systems: A Review" (2020): This review discusses the evolving landscape of DDoS attacks in IoT and various detection and mitigation techniques. Decision tree algorithms are highlighted as a promising approach, and their advantages, such as real-time monitoring and low computational overhead, are discussed.[21]
- "Machine Learning Techniques for DDoS Attack Detection in IoT" (2019): This paper delves into machine learning techniques for DDoS detection in IoT networks, with a focus on decision tree algorithms. It provides insights into the accuracy and scalability of decision tree-based models and their applicability to the dynamic nature of IoT [22].
- DDoS Detection in IoT Using Machine Learning Algorithms" (2018): This research explores machine learning algorithms for DDoS detection in IoT and includes decision trees as a key component of the study. The paper discusses the role of decision trees in classifying network traffic patterns and the importance of real-time monitoring.[23]

 https://doi.org/10.5281/zenodo.14093582

- A Review of Machine Learning Approaches for DDoS Detection in IoT" (2021): This review article provides a comprehensive analysis of machine learning approaches for DDoS detection in IoT, including decision tree algorithms. It discusses the need for accurate and lightweight models to protect IoT devices From DDoS attacks. [24].
- "Decision Trees for DDoS Attack Detection: A Comparative Study" (2019): This study conducts a comparative analysis of various decision tree-based models for DDoS attack detection, assessing their performance in IoT environments. It highlights the strengths and weaknesses of decision trees as compared to other classifiers. [25].

These literature sources collectively highlight the growing importance of decision tree algorithms in the context of DDoS attack detection within IoT networks. Decision trees offer a balance between accuracy and simplicity, making them a valuable tool in securing the ever-expanding IoT ecosystem against DDoS threats. Researchers and practitioners in the field can draw upon these studies to inform their own work on IoT network security.

III. METHODOLOGY

In this study, author design a framework for DDoS attacks. Classification and prediction based on existing datasets themachine learning method used. This framework includes: the following main steps.

i. The first step involves the selection of dataset forutilization.
ii. The second step involves the selection of tools andLanguage.
iii. The third step involves data pre-processing techniquesto handle irrelevant data from the dataset.
iv. Encoding is performed to convert symbolical data into numerical data.
v. In the fifth step, the data splitting is performed into train and test set for the model. In this step, user build and train proposed model. However, model optimization is also performed on the trained model interms of kernel scaling and kernel hyper-parameter tuning to improve model efficiency. When the model optimizes then it will generate output results from the model.

 https://doi.org/10.5281/zenodo.14093582

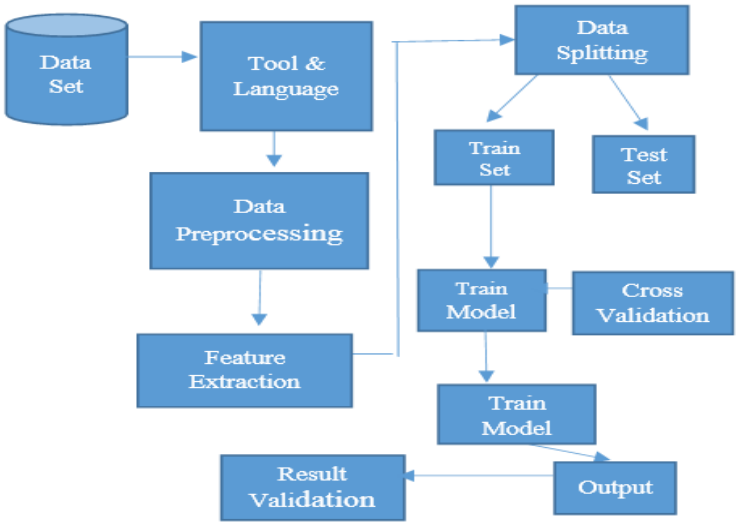

Fig. 2: Flow Diagram

IV. RESULTS AND DISCUSSION

- **Overview of Dataset**

This research represents all the outcomes of the models suggested to detect DDoS attack. All the results are presented step-by-step as figures. The CICIDS 2017 benchmark dataset was utilized for this. The ML (Machine Learning) Models Random Forest classifier, Hybrid1(Decision Tree + Linear Regression), Hybrid 2 (Decision Tree + Random Forest) were trained and tested using the Friday-Working Hours-Afternoon-DDos.pcap_ISCX.csv file from this dataset. There are 225745 rows and 79 columns in this csv file. Selected characteristics (columns) from the dataset were supplied to the ML models for training and testing, and their performance was evaluated in terms of Accuracy, F1-Score, Precision, Recall, Detection-Rate, False-Alarm-Rate, and the amount of time the model required to make a prediction. Each Model was tested with an increasing number of features; the first test included ten features, while the subsequent tests each added ten more features. The optimal model will be the one which uses the fewest features and requires the least False-Alarm-Rate for creating the accurate predictions. This allows us to measure the accuracy, recall, and precision, F1-Score, speed, and efficiency of each model. The proposed methodology is implemented in Python Jupyter notebook. JuptyterLab is the latest web based interactive development environment for notebooks, code, and data. Its flexible interface allows users to configure and arrange workflows in data science, scientific computing, computational journalism, and ML. A modular design invites extensions to expand and enrich functionality. The data is retrieved from DDoS evaluation dataset (CIC-DDoS2017). The most important feature sets to detect

https://doi.org/10.5281/zenodo.14093582

different types of DDoS attacks. The experiment shows the classification of the DDoS attack which will help in the prediction of the attack take place on IoT devices. The attacks are analysed, and polarity of the attacks is shown in the results. The DDoS attack are classified as clean traffic and attack traffic. The proposed methodology includes various ML classifiers: Linear Regression, Decision Tree, Random Forest, and Hybrid. All the ML classifiers are implemented using Python code. The study shows that Hybrid gives best accuracy as compared with other ML classifiers. Previous researcher has worked much on DT, RF & LR, NB, SVM but we have made the hybrid algorithm using DT, LR & RF. Better results are achieved when hybrid algorithm implemented in this research. Beginning the research with the data and then loaded and finally evaluate the results.

- **Percentage of classified data**

As per the below table results total number of traffic and percentage of the traffic are calculated. Following is the Summary of data.

TABLE I: Traffic Analysis

Total Traffic	Clean Traffic	Attack Traffic
225746	97719	128028
Percentage	43.26%	56.74%

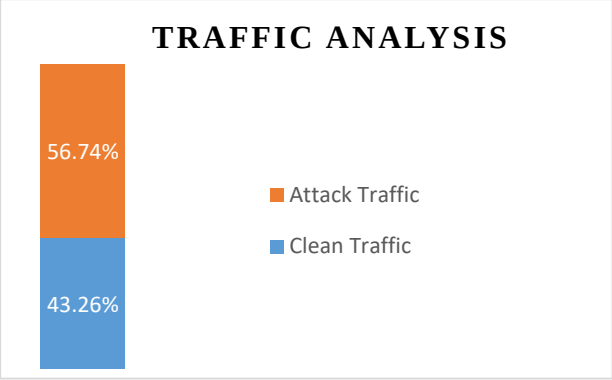

Fig. 3: Graphical output of Traffic Analysis

https://doi.org/10.5281/zenodo.14093582

ISSN: 0369-8963

- **Columns names**

The first step shows various columns in the data set. Following are the different columns of the DDoS attack which are listed below in Table 2. The result shows there are 79 columns that includes each detail about the data of traffic comes on router.

TABLE II: List of Columns

Sr.No	Column Name	Sr.No	Column Name	Sr.No	Column Name	Sr.No	Column Name
1.	Destination Port	11.	Bwd Packet Length Max	21.	Fwd IAT Total	31.	Fwd PSH Flags
2.	Flow Duration	12.	Bwd Packet Length Min	22.	Fwd IAT Mean	32.	Bwd PSH Flags
3.	Total Fwd Packets	13.	Bwd Packet Length Mean	23.	Fwd IAT Std	33.	Fwd URG Flags
4.	Total Backward Packets	14.	Bwd Packet Length Std	24.	Fwd IAT Max	34.	Bwd URG Flags
5.	Total length of Fwd Packets	15.	Flow Bytes/s	25.	Fwd IAT Min	35.	Fwd Header Length
6.	Total length of Bwd Packets	16.	Flow Packets/	26.	Bwd IAT Total	36.	Bwd Header Length
7.	Fwd Packet Length Max	17.	Flow IAT Mean	27.	Bwd IAT Mean	37.	Fwd Packets/s
8.	Fwd Packet Length Min	18.	Flow IAT Std	28.	Bwd IAT Std	38.	Bwd Packets/s
9.	Fwd Packet Length Mean	19.	Flow IAT Max	29.	Bwd IAT Max	39.	Min Packet Length
10.	Fwd Packet Length Std	20.	Flow IAT Min	30.	Bwd IAT Min	40.	Max Packet Length

https://doi.org/10.5281/zenodo.14093582

Sr.No	Column Name	Sr.No	Column Name	Sr.No	Column Name	Sr.No	Column Nar
41.	Packet Length Mean	51.	ECE Flag Count	61.	Bwd Avg Packets/Bulk	71.	Active Mean
42.	Packet Length Std	52.	Down/Up Ratio	62.	Bwd Avg Bulk Rate	72.	Active Std
43.	Packet Length Variance	53.	Average Packet Size	63.	Subflow Fwd Packets	73.	Active Max
44.	FIN Flag Count	54.	Avg Fwd Segment Size	64.	Subflow Fwd Bytes	74.	Active Min
45.	SYN Flag Count	55.	Avg Bwd Segme Size	65.	Subflow Bwd Packets	75.	Idle Mean
46.	RST Flag Count	56.	Fwd Header Length	66.	Subflow Bwd Bytes	76.	Idle Std
47.	PSH Flag Count Max	57.	Fwd Avg Bytes/Bulk	67.	Init_Win_ bytes_forward	77.	Idle Max
48.	ACk Flag Count	58.	Fwd Avg Packets/Bulk	68.	Init_Win_ bytes_backward	78.	Idle Min
49.	URG Flag Count	59.	Fwd Avg Bulk Rate	69.	act_data_ pkt_fwd	79.	Label
50.	CWE Flag Count	60.	Bwd Avg Bytes/Bulk	70.	min_seg_ size_forward		

- **Parameters for calculation-**

To classify BENIGN and DDoS attacks, the effectiveness of LR, DT, RF, Hybrid 1 (LR+DT), and Hybrid 2 (RF+DT) for DDoS detection is computed using 80% for training and 20% for testing. On the same training set, various classifier confusion matrices have been produced. An overview of a machine learning model's performance on a set of test data is provided via a confusion matrix as shown in table 3. It is a way to show how many instances, depending on the model's predictions, are accurate and inaccurate. It is frequently used to assess how well categorization models—which seek to assign a categorical label to each instance of input—perform. The matrix displays the number of instances produced by the model on the test data.

• **True positives (TP):** It happen when a positive data point is correctly predicted by the model.
• **True negatives (TN):** It happen when a negative data point is correctly predicted by the model.
• **False positives (FP):** It happen when the model makes an erroneous positive

 https://doi.org/10.5281/zenodo.14093582

data point prediction.

• **False negatives (FN):** It happen when a negative data point is miscalculated by the model.

TABLE III: Confusion Matrix

		Predicted Class	
		Normal	Attack
Actual Class	Normal	TP	FP
	Attack	FN	TN

- **Performance measure is calculated using following formulas-**

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$F1\ \text{Score} = \frac{2.\ \text{Precision XRecall}}{\text{Precision} + \text{Recall}}$$

Detection-rate = T P/ (T P + T N + F P + F N)

False-alarm-rate = F P/ (T N + F P)

- **Machine learning classifiers –** Linear Regression, Decision Tree, Random Forest, Hybrid 1(LR+DT), and Hybrid 2(RF+DT) machine learning classifiers are used to classify the data. Every classifier has a confusion matrix produced, and the metrics accuracy, precision, recall, and F1 score are all graphically displayed.

DDoS Attack Detection Framework Using Random Forest

Random Forest is the best suitable technique among the three presented in this study (Linear Regression, Random Forest, and Decision Tree) for classification issues of this nature. However, it has the drawback of being the slowest algorithm. Even with fewer training attributes, it achieves a very high level of accuracy. In this test, even with just 10 qualities provided for training and prediction, it makes Predictions with 99% accuracy. Since this algorithm's accuracy is already very high with less features, it doesn't really gain from having more features to train upon. The graph and table demonstrate that this algorithm's accuracy in this specific instance never falls below 99%, making it highly effective in DDoS detection. The number of columns and cumulative performance in terms of accuracy, precision, and other metrics are shown in Table 4. Recall, Time, False-Alarm-Rate, Detection-Rate, and F1-Score.

 https://doi.org/10.5281/zenodo.14093582

Figure 4 shows a graphic representation of the same. Figure 5 shows a graphic representation of the confusion matrix.

TABLE IV: Performance measure using Random Forest

No. of Columns used	Accuracy	Precision	Recall	F1-Score	Time (in second)	Detection-Rate	False-Alarm-Rate
10	**0.99975**	**0.99988**	**0.99968**	**0.99978**	**0.02054**	**0.56373**	**0.00015**
20	**0.99964**	**0.99980**	**0.99956**	**0.99968**	**0.02667**	**0.56366**	**0.00025**
30	**0.99971**	**0.99988**	**0.99960**	**0.99974**	**0.02498**	**0.56368**	**0.00015**
40	**0.99966**	**0.99984**	**0.99956**	**0.99970**	**0.02970**	**0.56366**	**0.00020**
50	**0.99971**	**0.99988**	**0.99960**	**0.99974**	**0.03305**	**0.56368**	**0.00015**
60	**0.99966**	**0.99980**	**0.99960**	**0.99970**	**0.03413**	**0.56368**	**0.00025**
70	**0.99988**	**1**	**0.99980**	**0.99990**	**0.03512**	**0.56379**	**0**
78	**0.99984**	**0.9999**	**0.99976**	**0.99986**	**0.03914**	**0.56377**	**0**

The output of Random Forest on 79 features calculated in table IV . As it can observed best Accuracy -99.98%, Precision -100%, Recall – 99.98%, & F1-Score – 99.99% at column position 70 and 78, where as less time taken at column position 10 is 0.02054 with good Detection-Rate 0.5637 with good False-Alarm-Rate - 0 at column position 70 and 78.

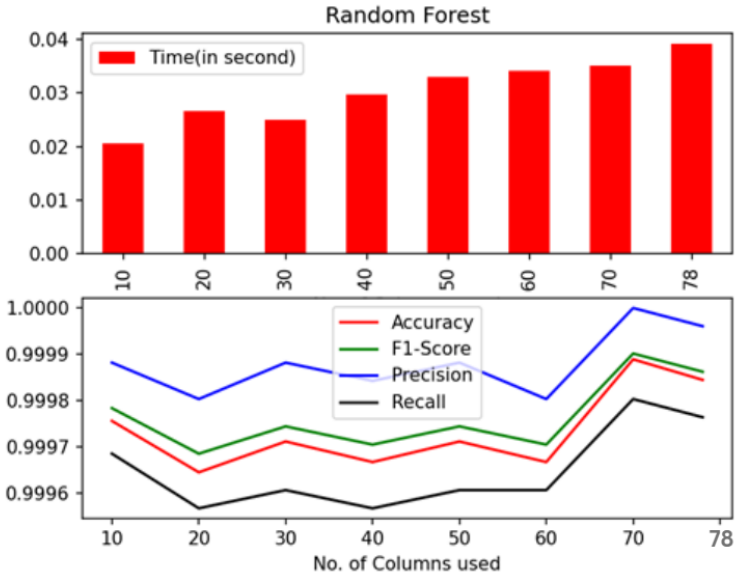

Fig. 4: Graphical Representation of Random Forest

https://doi.org/10.5281/zenodo.14093582

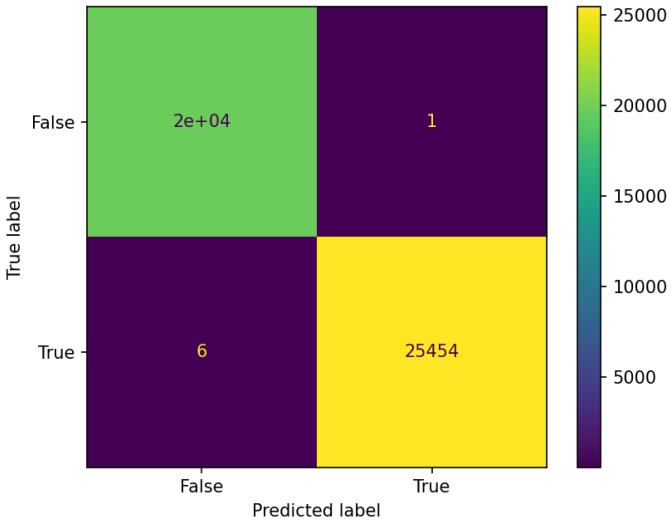

Fig. 5: Confusion Matrix of Random Forest

- **DDoS Attack Detection Framework Using Hybrid Algorithm (Proposed)**
  In this study, author created a DDoS attack framework. The machine learning technique was applied for classification and prediction using existing datasets.

  Hybrid Algorithm 1(Linear Regression + Decision Tree)

  Table 5 displays the number of columns and the cumulative performance in terms of Accuracy, Precision, and other metrics. Recall the F1-score, prediction time, Detection – Rate and False-Alarm -Rate. The same is displayed graphically in Figure 6. Confusion matrix is represented via graphically in Figure 7.

TABLE V: Performance measure using Hybrid Algorithm 1 (LR+DT)

No. of Columns used	Accuracy	Precision	Recall	F1-Score	Time (in second)	Detection-Rate	False-Alarm-Rate
10	0.99937	0.99921	0.99968	0.99945	0.00938	0.56375	0.00101
20	0.99727	0.99589	0.99929	0.99758	0.01970	0.56353	0.00533
30	0.99789	0.99698	0.99929	0.99813	0.02125	0.56353	0.00391
40	0.99847	0.99811	0.99917	0.99864	0.03713	0.56346	0.00243
50	0.99940	0.99917	0.99976	0.99946	0.03972	0.56379	0.00106
60	0.99944	0.99921	0.99980	0.99950	0.04210	0.56382	0.00101
70	0.99957	0.99941	0.99984	0.99962	0.04603	0.56384	0.00076
78	0.99946	0.99921	0.99984	0.99952	0.04312	0.56384	0.00101

https://doi.org/10.5281/zenodo.14093582

The output of Hybrid1 on 79 features calculated in table 5. As it can observed best Accuracy - 99.95%, Precision -99.94%, Recall – 99.98%, & F1-Score – 99.96% at column position 70 and 78, where as less time taken at column position 10 is 0.00938 with good Detection-Rate 0.5638 with good False-Alarm-Rate – 0.001 at column position 70 and 78.

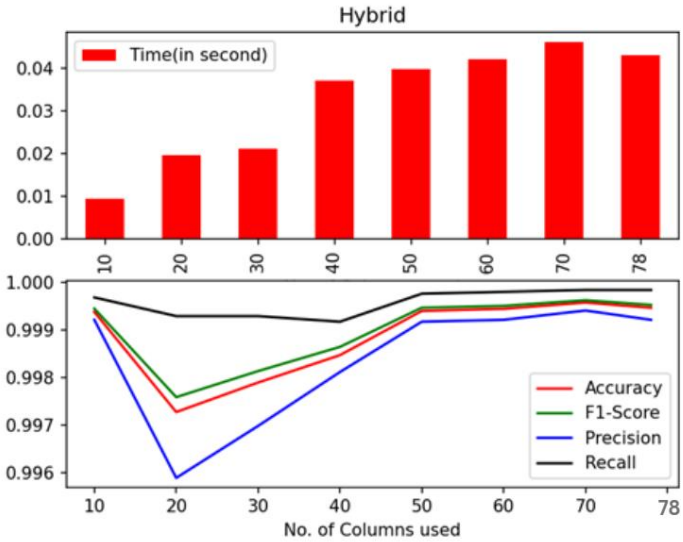

Fig. 6: Graphical Representation of Hybrid1 (LR+DT)

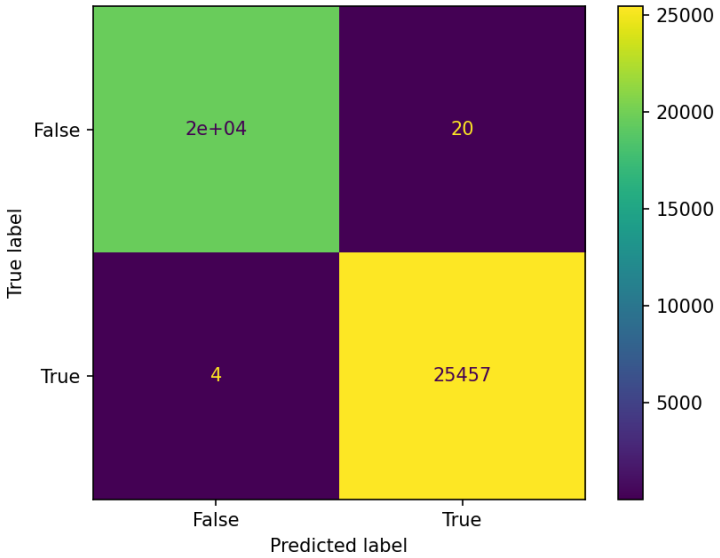

Fig.7: Confusion Matrix of Hybrid1 (LR+DT)

Hybrid Algorithm 2(Random Forest + Decision Tree)

Table 6. Displays the number of columns and the cumulative performance in terms of Accuracy, Precision, and other metrics. Recall, the F1-score, prediction time, Detection – Rate and False-

https://doi.org/10.5281/zenodo.14093582

Alarm-Rate. The same is displayed graphically in Figure 8. Confusion matrix is represented via graphically in figure 9.

TABLE VI: Performance measure using Hybrid Algorithm 2 (RF+DT)

No. of Columns used	Accuracy	Precision	Recall	F1-Score	Time (in second)	Detection-Rate	False-Alarm-Rate
10	**0.99971**	**0.99964**	**0.99984**	**0.99974**	**0.10452**	**0.56384**	**0.00045**
20	**0.99977**	**0.99972**	**0.99988**	**0.99980**	**0.07778**	**0.56386**	**0.00035**
30	**0.99973**	**0.99964**	**0.99988**	**0.99976**	**0.07020**	**0.56386**	**0.00045**
40	**0.99962**	**0.99956**	**0.99976**	**0.99966**	**0.08955**	**0.56379**	**0.00055**
50	**0.99986**	**0.99988**	**0.99988**	**0.99988**	**0.09316**	**0.56386**	**0.00015**
60	**0.99988**	**0.99988**	**0.99992**	**0.99990**	**0.10276**	**0.56388**	**0.00015**
70	**0.99988**	**0.99992**	**0.99988**	**0.99990**	**0.09967**	**0.563866**	**0.00010**
78	**0.99991**	**0.99992**	**0.99992**	**0.99992**	**0.12619**	**0.56388**	**0.00010**

The output of Hybrid2 algorithm on 79 features evaluated in table 6 . As it can observed best Accuracy -99.99%, Precision -99.99%, Recall – 99.99%, & F1-Score – 99.99% at column position 70 and 78, where as less time taken at column position 30 is 0.07 with good Detection-Rate 0.5638 with good False-Alarm-Rate – 0.0001 at column position 70 and 78.

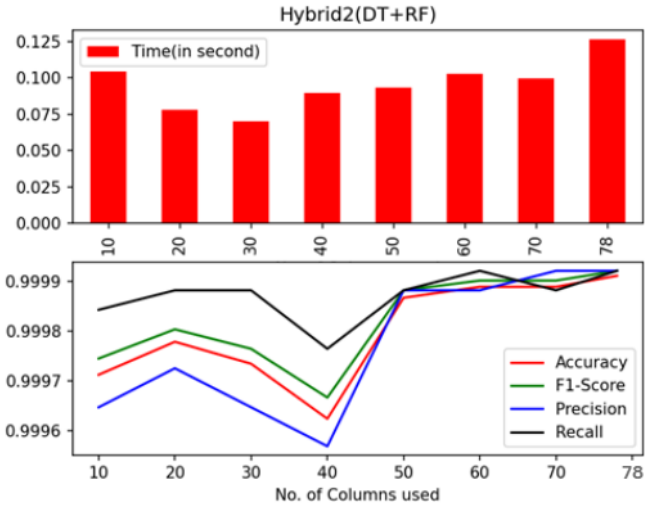

Fig. 8: Graphical Representation of Hybrid2 (RF+DT)

 https://doi.org/10.5281/zenodo.14093582

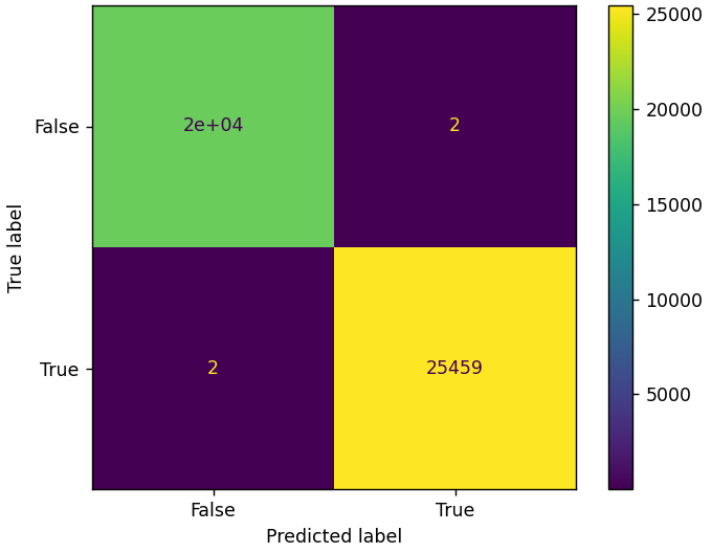

Fig. 9: Confusion Matrix of Hybrid2 (RF+DT)

Average Values Table

Table 7 represents the performance measure of ML classifiers using Average values with the help of all features of original dataset. The same is graphically represented via figure 10 Using Parameters -Accuracy, Precision, Recall, F1-Score and Figure 11 represents Graphical Representation of Average Values of ML Classifiers (Using Parameters –Time, Detection-Rate, False -Alarm-Rate)

TABLE VII: Performance measure using Average Values

ML Classifiers	Average Values						
	Accuracy	Precision	Recall	F1-Score	Time (in sec)	Detection-Rate	False – Alarm – Rate
Random Forest	0.9997	0.9998	0.9996	0.9997	0.0304	0.5637	0.0001
Hybrid 1 (DT+LR) Proposed	0.9988	0.9984	0.9995	0.9989	0.0323	0.5637	0.0020
Hybrid 2 (DT+RF) Proposed	0.9998	0.9997	0.9998	0.9998	0.0954	0.5638	0.0002

https://doi.org/10.5281/zenodo.14093582

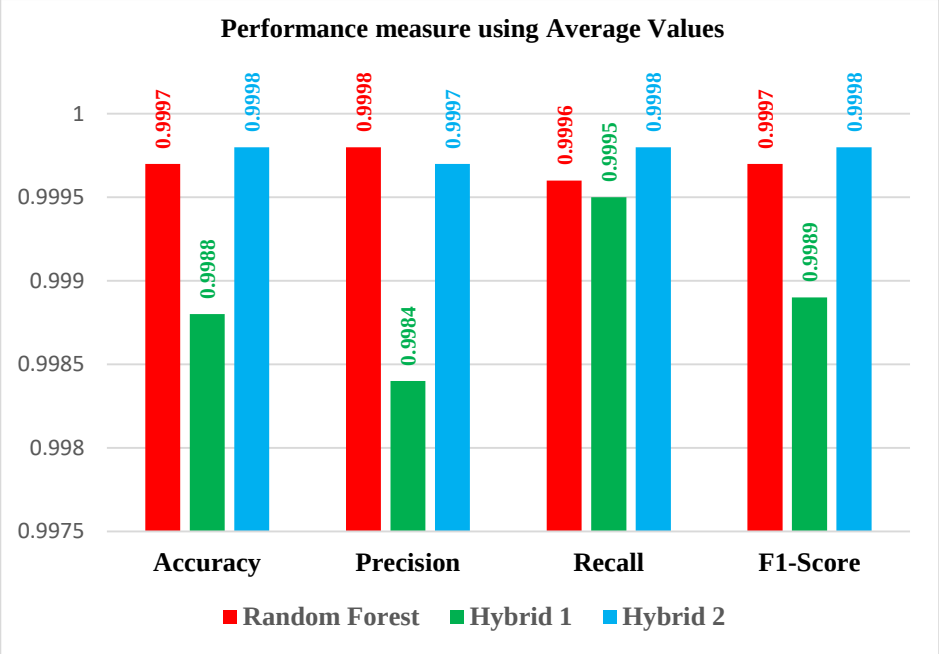

Fig. 10: Graphical Representation of Average Values of ML Classifiers (Using Parameters -Accuracy, Precision, Recall, F1-Score)

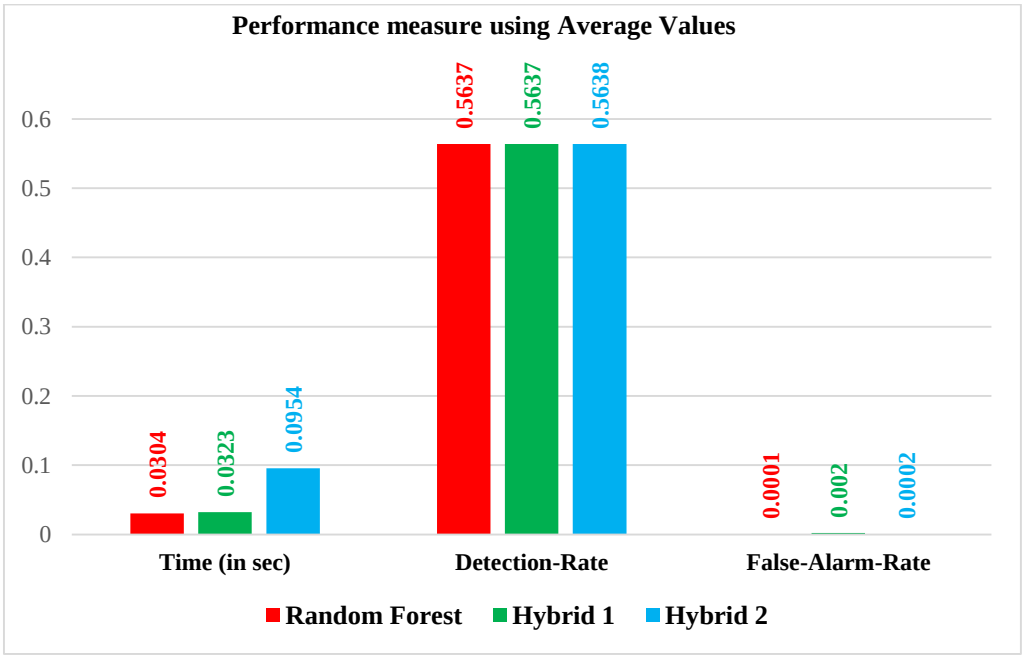

Fig. 11: Graphical Representation of Average Values of ML Classifiers (Using Parameters –Time, Detection-Rate, False-Alarm-Rate)

https://doi.org/10.5281/zenodo.14093582

- **Real Time detection of DDoS attack on Raspberry Pi using Hybrid Algorithm**

As the Internet of Things (IoT) ecosystem grows, the security of low-resource devices like Raspberry Pi is becoming more and more crucial. This study proposes and evaluates a detection system based on the hybrid method to handle the unique difficulty of detecting DDoS attack on Raspberry Pi. Decision trees are widely used on low power devices with constrained processing resources because of their efficiency, interoperability, and simplicity. Two basic phases make up the operation of the suggested detecting system. During the training stage, the decision tree model is trained on Raspberry Pi using features that are taken from regular network traffic. These factors include traffic patterns, system resource utilization, and network packet properties. By learning to differentiate between typical and abnormal behavior. During the detection phase, incoming traffic is classified as either normal or indicative of DDoS attack using the hybrid method, which is applied while continually monitoring real-time network traffic. The hybrid algorithms lightweight design guarantees effective execution on the Raspberry Pi's limited hardware, making it appropriate for real-time deployment. The study emphasizes how crucial it is to use lightweight machine learning approaches to safeguard IoT deployments to promote a more reliable and secure IoT ecosystem.

Framework to detect DDoS attack on Raspberry Pi using Hybrid Algorithm

In this study, we created a framework to detect DDoS attack on Raspberry Pi. The steps involved in Figure 12.

i. **Data Collection**: During regular, collect network traffic data from the Raspberry Pi. Created a diversified dataset by simulating several DDoS attack scenarios.
ii. **Feature Extraction**: Establish a list of characteristics that the algorithm will accept as input, packet sizes, rates, traffic patterns, and system resource usage.
iii. **Data Pre-Processing:** Preprocess and normalize the gathered data to guarantee consistency. and eliminate unnecessary information, organize partial or missing data points.
iv. **Training Phase:** Divide the preprocessed dataset into sets for testing. Utilizing the training data, teach the algorithm to recognize patterns of typical behavior.
v. **Algorithm Implementation:** Apply a lightweight machine learning library to the Raspberry Pi to implement the algorithm.
vi. **Real-time Monitoring**: In real-time, keep an eye on all incoming network traffic. Take features out of the traffic in real time and feed them into the model that has been trained.
vii. **Decision and Altering:** Sort incoming traffic using model as either typical or indicative of a DDoS attacks. When an attack is detected, an alerting mechanism is used to notify administrators to take corresponding action.

 https://doi.org/10.5281/zenodo.14093582

viii. **Adaptability and Dynamic Adjustment:** Establish systems that allow the sensitivity of the algorithm to be dynamically adjusted in response to shift network circumstances and resource availability.

ix. **Evaluation:** Use the testing set to regularly assess the detection systems performance. Analyze important parameters like false positive rates, recall, accuracy, and precision.

x. **Optimization:** Considering the Raspberry Pi's constrained resources, optimize the framework for efficiency. Examine the techniques to lower the number of false positives and false negatives.

xi. **Documentation and Reporting:** Include implementation specifies, parameter configurations, and any concerns in framework to evaluate and respond to DDoS attacks that are detected.

xii. **Integration with security Infrastructure:** Connect the Raspberry Pi's current security system to the DDoS detection framework. Verify if it is compatible with additional security policies and safeguards.

The Pseudo code to detect the attack on Raspberry Pi is written in Figure 13. The corresponding pin diagram showcased via Figure 14. The kit connectivity is represented in Figure 15. Figure 16 represents trigger of DDoS attack via RED LED. Whereas Normal traffic is depicted with GREEN LED in Figure 17. Table 8 represents the performance of Hybrid Algorithm on IoT Kit. The graphical output is represented via Figure 18.

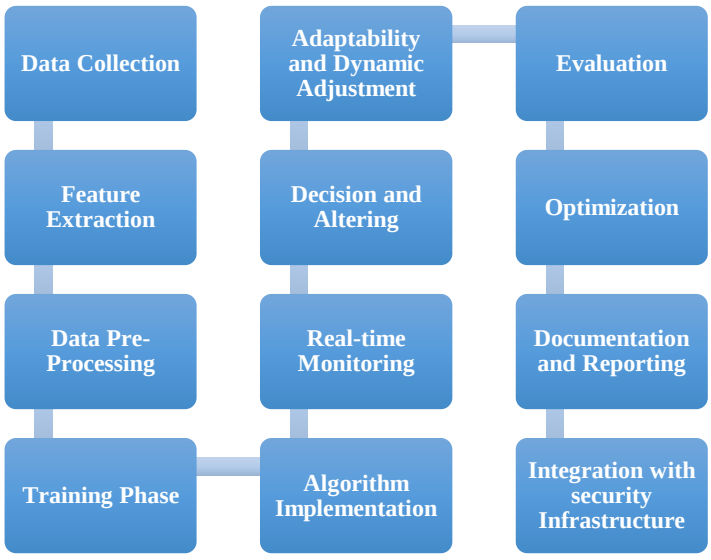

Fig. 12: Framework to detect DDoS on Raspberry Pi

https://doi.org/10.5281/zenodo.14093582

```
import RPi.GPIO as GPIO
import time
# Set up GPIO pins for LEDs
BENIGN_LED_PIN = 11  # Replace with the GPIO pin number connected to the
BENIGN LED-green
DDOS_LED_PIN = 12  # Replace with the GPIO pin number connected to the DI
LED-red
GPIO.setmode(GPIO.BOARD)
GPIO.setup(BENIGN_LED_PIN, GPIO.OUT)
GPIO.setup(DDOS_LED_PIN, GPIO.OUT)
for pred in y_pred_1:
        if pred == 0:  # BENIGN
            GPIO.output(BENIGN_LED_PIN, GPIO.HIGH)
            GPIO.output(DDOS_LED_PIN, GPIO.LOW)
        else:  # DDoS
            GPIO.output(BENIGN_LED_PIN, GPIO.LOW)
            GPIO.output(DDOS_LED_PIN, GPIO.HIGH)
        # Add a delay to observe LED status
        time.sleep(0.5)
        # Turn off LEDs
        GPIO.output(BENIGN_LED_PIN, GPIO.LOW)
        GPIO.output(DDOS_LED_PIN, GPIO.LOW)
        end=time.time()
```

Fig. 13: Pseudocode for DDoS Detection on Raspberry Pi

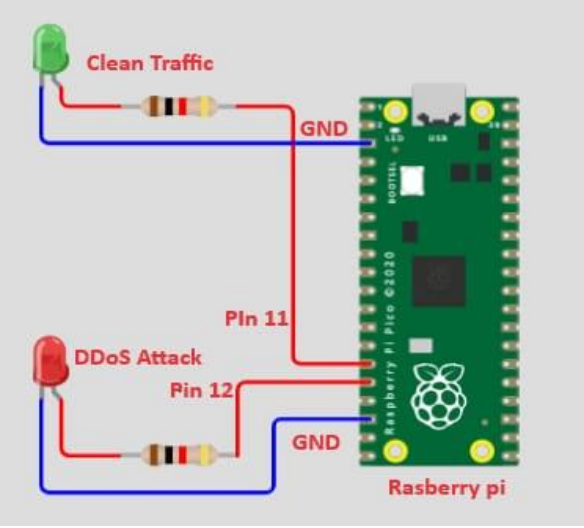

Fig. 14: Pin Diagram on Raspberry Pi

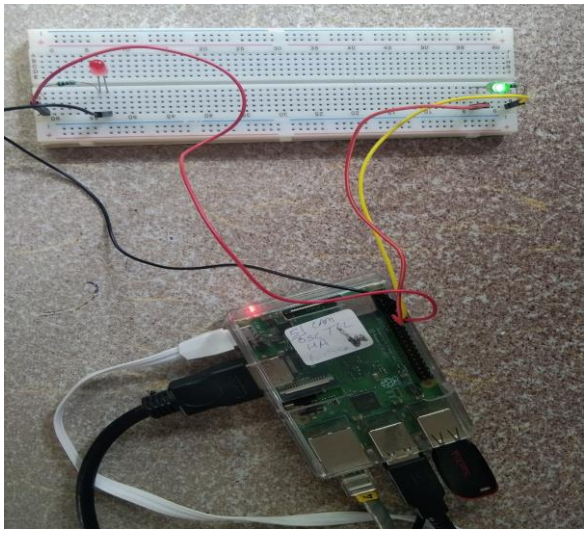

Fig. 15: Connectivity with Raspberry Pi

https://doi.org/10.5281/zenodo.14093582

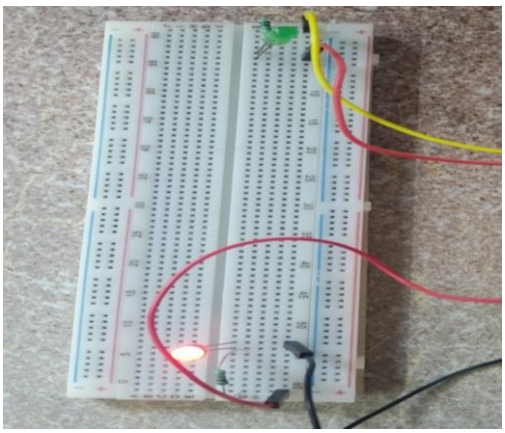

Fig. 16: Detection of DDoS Attack

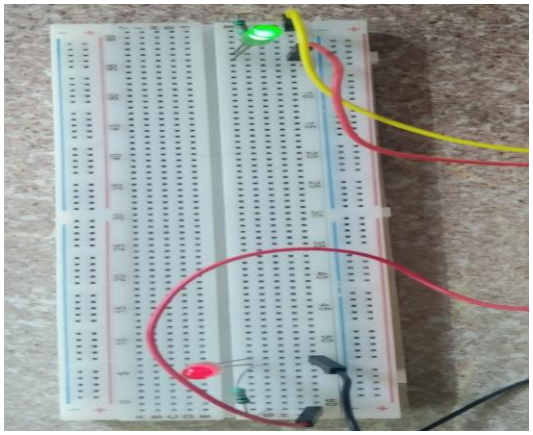

Fig. 17: Normal Traffic

Performance Evaluation using Hybrid 1 Algorithm -

The output of Hybrid1 on 78 features with 40,000 rows on IoT kit is calculated in table 8 . As it can observed best Accuracy , Precision, Recall & F1-Score with values 99.86% , 99.85 %, 99.82%,99.83% at column position 70, where as less time taken at column position 10 is 0.05293 with good detection_rate 0.4154 at column position 70&78 and false_alarm_rate 0.0004 at column position 10.The corrsponding time graph and confusion matrix reprsented via fig. 18 & 19.

TABLE VIII: Performance of Hybrid1 Algorithm on IoT Kit

No. of Columns used	Accuracy	Precision	Recall	F1-Score	Time (in second)	Detection-Rate	False-Alarm-Rate
10	0.99926	0.99941	0.99882	0.99912	0.05293	0.41573	0.00041
20	0.99853	0.99853	0.99794	0.99824	0.06960	0.41536	0.00104
30	0.99817	0.99736	0.99824	0.99780	0.09209	0.41548	0.00188
40	0.99841	0.99824	0.99794	0.99809	0.11124	0.41536	0.00125
50	0.99853	0.99853	0.99794	0.99824	0.13465	0.41536	0.00104
60	0.99829	0.99794	0.99794	0.99794	0.17706	0.41536	0.00146
70	0.99865	0.99853	0.99824	0.99838	0.17816	0.41548	0.00104
78	0.99804	0.99707	0.99824	0.99765	0.22046	0.41548	0.00208

https://doi.org/10.5281/zenodo.14093582

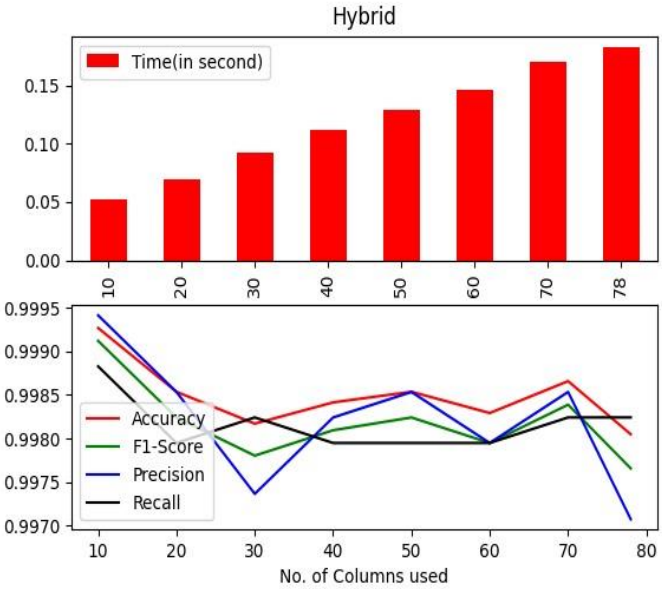

Fig. 18: Graph representing performance of Hybrid1 Algorithm on IoT Kit

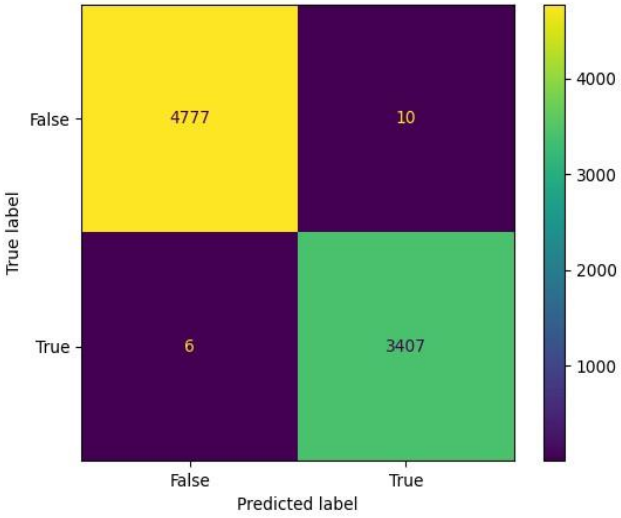

Fig. 19: Confusion matrix using Hybrid1Algorithm on IoT Kit

Performance Evaluation using Hybrid 2 Algorithm -

The output of Hybrid2 on 78 features with 40,000 rows on IoT kit is calculated in table 9 . As it can observed best Accuracy , Precision, Recall & F1-Score with values 99.93% , 99.97 %, 99.88%,99.92% at column position 78, where as less time taken at column position 10 is 0.1130 with good detection_rate 0.4158 at column position 70 and false_alarm_rate 0.0000 at column position 10.The corrsponding time graph and confusion matrix reprsented via fig. 20 & 21.

https://doi.org/10.5281/zenodo.14093582

TABLE IX: Performance of Hybrid2 Algorithm on IoT Kit

No. of Columns used	Accuracy	Precision	Recall	F1-Score	Time (in second)	Detection-Rate	False-Alarm - Rate
10	0.99963	1.00000	0.99912	0.99956	0.11306	0.41585	0.00000
20	0.99939	0.99970	0.99882	0.99926	0.12848	0.41532	0.00020
30	0.99963	0.99970	0.99941	0.99956	0.14768	0.41597	0.00020
40	0.99951	0.99970	0.99912	0.99941	0.17000	0.41585	0.00125
50	0.99914	0.99941	0.99853	0.99897	0.17768	0.41561	0.00041
60	0.99939	0.99970	0.99882	0.99926	0.19160	0.41573	0.00020
70	0.99951	0.99970	0.99912	0.99941	0.22018	0.41585	0.00020
78	0.99939	0.99970	0.99882	0.99926	0.22294	0.41573	0.00020

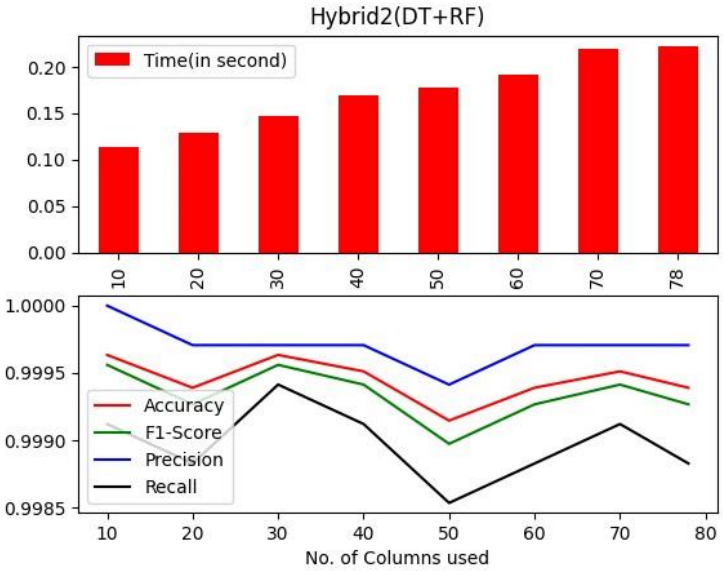

Fig. 20: Graph representing performance of Hybrid2 Algorithm on IoT Kit

https://doi.org/10.5281/zenodo.14093582

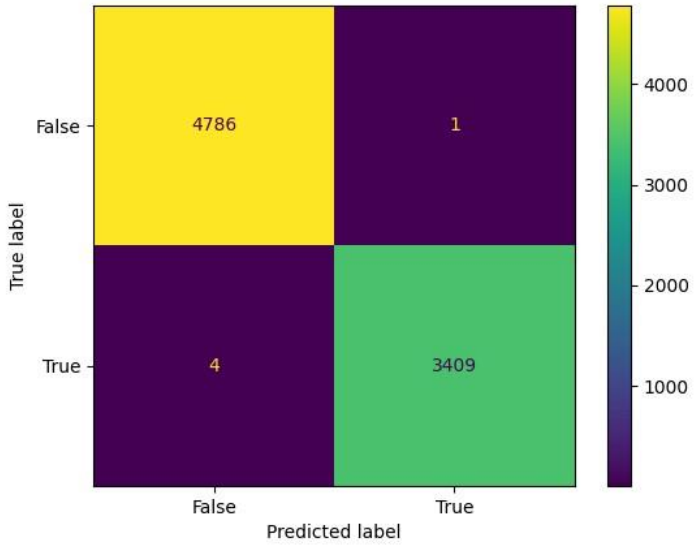

Fig. 21: Confusion matrix using Hybrid2 Algorithm on IoT Kit

V. CONCLUSION

Applying machine learning techniques to improve security measures at the network layer of the Internet of Things is a promising and practical way to address the constantly changing risks and obstacles in the networked world of gadgets. It permits a more adaptive, flexible, and effective method of defending IoT ecosystems from dynamic cyber security risks. But it's imperative that you approach the deployment with a thorough awareness of the advantages and difficulties posed by these technologies. IoT network security will be strengthened and improved by ongoing research, teamwork, and machine learning developments. In network situations, DDOS attack detection is more frequent, thus it's critical to be aware of the attacks that render network services inaccessible. Machine learning models can be used to train and test attack detection datasets to detect such an attack. The purpose of this research is to develop a machine learning model. The dataset under examination was incorporated into the well-known CICIDS 2017 dataset to conduct experiments for the study. More specifically, log files with benign, bot, and DDoS classes are taken into consideration on Friday afternoons. The identical data set was used to evaluate all three techniques, and the results show that while linear regression is the fastest, it is also the least accurate. Of the three, Random Forest is the slowest but also the most accurate. Ultimately, decision trees are considered the best method for these kinds of issues since they offer a fair mix between speed and accuracy. However, Hybrid Algorithms 1 (DT+LR) and Hybrid Algorithm 2 (DT+RF) were created to better understand the issue and increase security for IoT networks to improve security measures at the network layer. Testing revealed that Hybrid Algorithm 2 (DT+RF) is the strongest ML classifier out of all of them, justifying with Accuracy, Recall, F1 Score of 99.98%, and Precision with 99.97% with good false alarm rate of 0.0002. An IoT device is also used to test the algorithm and validate the results. In addition to being an ML model, it displays some significant

https://doi.org/10.5281/zenodo.14093582

visualizations, including fitting and residual plots. This demonstrates the model's significance and appropriateness for investigating the model for prediction.

-Competing Interests - Not Applicable
- Funding Information -Not Applicable
- Author contribution - Equal
- Data Availability Statement - Use of Dataset - CICIDS 2017
- Research Involving Human and /or Animals - Not Applicable
- Informed Consent - Yes

References

[1] Alaba, F. A., Othman, M., Hashem, I. A. T., Alotaibi, F.: "Internet of Things security: A survey". International Journal of Network and Computer Applications 88, 10-28. 2017.

[2] Baghli, R. B., Najm, E., Traverson, B.: "Towards a multi-leveled architecture for the Internet of Things". In: Proceedings of the 20th IEEE International Enterprise Distributed Object Computing Workshop (EDOCW), Vienna, Austria, September 5-9, 182-187.2016.

[3] Oppitz, M., Tomsu, P. "Internet of Things. In Inventing the Cloud Century", Springer: Cham, Switzerland, pp. 435–469, 2018.

[4] Zhang, D., Yang, L.T., Chen, M., Zhao, S., Guo, M. Zhang, Y. "Real-time locating systems using active RFID for Internet of Things". IEEE Syst. J. 1226–1235, 2016.

[5] Nagashree, R.N., Rao, V., Aswini, N. Near field communication. Int. J. Wirel. Microw. Technol. (IJWMT)vol 4, 20, 2014.

[6] Whitmore, A., Agarwal, A., Da Xu, L. "The Internet of Things—A survey of topics and trends. Inf. Syst. Front.", 17, 261–274, 2015.

[7] Kosmatos, E.A., Tselikas, N.D. and Boucouvalas, A.C. "Integrating RFIDs and Smart Objects into a Unified Internet of Things Architecture", Advances in Internet of Things: Scientific Research, 1, 5-12. http://dx.doi.org/10.4236/ait.2011.11002,2011.

[8] A. McEwen and H. Cassimally, "Designing the internet of things": John Wiley & Sons, 2013.

[9] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," Ad Hoc Networks, vol. 10, pp. 1497-1516, 2012.

[10] Atzori, A. Iera, and G. Morabito, "Understanding the Internet of Things: definition, potentials, and societal role of a fast-evolving paradigm," Ad Hoc Networks, vol. 56, pp. 122-140, 2017.

[11] Srivastava and T. Kelly, "The internet of things," International Telecommunication Union, Tech. Rep, vol. 7, 2005.

[12] Fadlullah, Z. M., Fouda, M. M., Kato, N., Takeuchi, A., Lwaski, N., Nozaki, Y.: Toward Intelligent Machine-to-Machine Communications in Smart Grid. IEEE Communications Magazine, Vol. 49, No. 4, 60-65. 2011.

[13] Chen, M., Wan, J., Li, F.: Machine-to-Machine Communications: Architectures, Standards, and Applications. KSII Transactions on Internet and Information Systems, Vol. 6, No. 2, 480-497. 2012.

[14] M. at. el, “Machine Learning for Internet of Things Data Analysis:A Survey ,” Journal of Digital Communications and Networks, Elsevier,vol. 1, pp. 1–56, February, 2018.

[15] D. B. J. Sen, “Internet of Things - Applications and Challenges in Technology and Standardization,”IEEE Transactions in Wireless Personal Communication, May 2011.

[16] [16] Ashton, K,“Internet of Things” Thing: In the Real-World Things Matter More than Ideas. RFID Journal.http://www.rfidjournal.com/articles/view?4986.2009.

[17] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications," IEEE Internet of Things Journal, 2017.

[18] B. N. Silva, M. Khan, and K. Han, "Internet of Things: A Comprehensive Review of Enabling Technologies, Architecture, and Challenges," IETE Technical Review, pp. 1-16, 2017.

[19] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," Ad Hoc Networks, vol. 10, pp. 1497-1516, 2012.

[20] I. Ali, S. Sabir, and Z. Ullah, "Internet of Things Security, Device Authentication and Access Control: A Review," International Journal of Computer Science and Information Security, vol. 14, p. 456, 2016.

[21] Yaqoob I., Ahmed E., Hashem I.A.T., Ahmed A.I.A., Gani A., Imran M., Guizani M. Internet of Things Architecture: Recent Advances, Taxonomy, Requirements, and Open Challenges. IEEE Wirel. Commun. ; 24:10–16. doi: 10.1109/MWC.2017.1600421. 2017.

[22] Pierre de LeussePanosPeriorellisTheoDimitrakos Srijith Krishnan Nair Self-Managed Security Cell, a "Security Model for the Internet of Things and services" DOI: 10.1109/AFIN.2009.15.2009.

[23] Hui Suoa, JiafuWana and CaifengZoua, JianqiLiua, "Security in the Internet of Things: A Review",International Conference on Computer Science and Electronics Engineering, pp. 649-651.2012.

[24] ChenQiang, Guang-ri Quan, Bai Yu and Liu Yang, "Research on Security Issues on the Internet ofThings", International Journal of Future Generation Communication and Networking, pp.1-9.2014.

[25] Kai Zhao and Lina Ge, "A Survey on the Internet of Things Security", IEEE, International Conferenceon Computational Intelligence and Security, pp. 663-667.2013.

 https://doi.org/10.5281/zenodo.14093582