Detection of Flood and Brute force Attacks on IOT devices using Hybrid model approach

MSc Research Project
MSc Cybersecurity

Venkata Nithin Krishna Singamsetty
Student ID: x23263326

School of Computing
National College of Ireland

Supervisor: Michael Prior

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Venkata Nithin Krishna Singamsetty …....…………………………………………………………………………………………

Student ID: x23263326 ……

Programme: MSc Cybersecurity ……………………………………………………………… **Year:** 2025 …………………………..

Module: MSc Practicum/Internship part 2 ………………………………………………………………………………………………………

Supervisor: Michael Prior ………………………………………………………………………………………………………

Submission Due Date: 11-12-2024 ……

Project Title: Detection of Flood and Brute force Attacks on IOT devices using hybrid model approach ……………………………………………………………………………………………………

Word Count: 6380 ……………………………………… **Page Count 21**……………………………….……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Venkata Nithin Krishna ……

Date: 11-12-2024 ……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Detection of Flood and Brute force Attacks on IOT devices using Hybrid model approach

Venkata Nithin Krishna
x23263326

Abstract

The usage of IOT devices has increased immensely all over the world. IoT devices now being great targets for cyberattacks like brute force efforts and flood attacks, this explosion in IoT adoption has also brought vulnerabilities. Strong detection methods are required since these threats expose security, dependability of IoT networks. In order to detect these assaults on IOT devices, this study suggests a hybrid machine learning model that combines the advantage of various methods to provide high accuracy and real-time responsiveness.

The research is on combining the supervised and unsupervised learning methods to check the system and network traffic which makes the system power to detect malicious activity. Comparing solo models to combined models the hybrid combination performs better in all the metrics evaluation.

This study also assesses the model's scalability and adaptability in dynamic IoT contexts, emphasizing how well it can identify intricate attack patterns and changing threats. By offering a reliable, effective, and flexible way to reduce the dangers associated with brute force and flood assaults, the research helps to improve IoT security and eventually ensure safer IOT deployments in crucial applications.

1 Introduction

Motivation:

The usage of internet of things has created a revolution in many industries which enabled seamless connectivity, intelligent, automated working in day-to-day tasks. But there are various disadvantages created by IOT devices by introducing security challenges like threats which made these devices a target for cyberattacks such as bruteforce and flood attacks. These attacks exploit the loopholes and hidden vulnerabilities of IOT devices which leads to data breach, system compromise, etc. This increase in number of attacks makes the necessity of development of efficient and robust techniques to protect IOT systems.

Background:

IOT security has become the utmost important field of research because of the critical roles how these devices play in industries like automotives, home appliances, healthcare and transportation. “Flood and brute force attacks” are one widely used attacks to IOT devices.

Flooding attacks are those that involve sending an enormous amount of network traffic with the intention of overwhelming bandwidth or processing capacity so that the IoT devices become unavailable for legitimate users. These are often used to precede more focused intrusions or as part of DDoS campaigns.

On the other hand, brute force attacks will exploit weak authentication mechanisms by continuous trying the all possible credentials or cryptographic keys until they guess correctly. This could compromise device access, allowing access to unauthorized control, data theft, or further exploitation of the IoT network.

Importance:

ML models has a pivotal role in detecting flood and brute force attacks on IOT devices. Hence there is a lot of importance in this research which involves various machine learning models. These models are chosen because:

- **Increasing IoT Security threats:** As IoT devices usage in homes, businesses, and many sectors, they became now targets for cyberattacks. By addressing vulnerabilities specific to IoT networks, research into detecting brute force and flood assaults improves overall security.
- **Adaptive Security Against Changing Dangers:** Cyberattacks are not always same, they keep on changing its methods as technology changes. IOT security is must in to resist from the new threats and save the sensitive data
- **Improving Detection Accuracy:** Traditional methods to detect iot threats is no longer safer and there raised a scope for machine learning models to improve the detections strategies.
- **Detection of Real time attacks:** Many devices function in real-time settings, such as smart homes, driverless cars, and healthcare monitoring. Machine learning research reduces response times and avoids possible damage by enabling realtime detection.
- **Safeguarding Private Information**: IoT devices frequently manage private or business information. Data breaches can result from flood assaults that overwhelm network security and brute force attacks that compromise login credentials. Research aids in the creation of reliable models that safeguard the integrity and confidentiality.

Research Question:

- How can hybrid machine learning model enhance the efficiency and accuracy in detection of flood and RTSP brute force attacks?

Objectives:

- To implement various ML models individually to detect flood , brute force attacks on IOT devices.
- To create an efficient hybrid machine learning model.
- To compare hybrid models with traditional methods.
- To verify the model using an actual IOT traffic dataset
- To find the challenges faced in detecting floods and brute force attacks using hybrid model.
- To assess the testing and training times of machine learning models.

Limitations

To handle the hybrid model and the real time IOT traffic is not an easy task and it has various limitations and restrictions

.

Resource Constraints of IoT Devices:

- **Limited Processing Power**: The deployment of sophisticated machine learning models may be hampered by IoT devices' frequently poor computing capacities.

- **Memory and Storage Limitations**: Many IoT devices may have memory and storage limits that are insufficient for large datasets and model parameters.

Real-Time Processing Challenges:

- Because the hybrid approach requires low latency and fast reaction times, it might be challenging to guarantee real-time operation, particularly when network traffic volume is high.
- There may be trade-offs between processing speed and detecting accuracy.

High False Positives and False Negatives:

- Even with hybrid models, distinguishing between legitimate traffic and malicious activities can still lead to false positives (incorrectly flagging benign activities) or false negatives (failing to detect an attack).

Data Availability and Quality:

- **Limited Datasets**: Access to real-world datasets containing IoT traffic data for flood and brute force attacks can be restricted or limited.
- **Data Imbalance**: Attack datasets are often imbalanced, with a lower frequency of attack samples compared to normal traffic, affecting model training and performance.

Scalability Issues:

- While IoT networks are growing, ensuring that the hybrid model scales efficiently to thousands or millions of devices can be challenging.
- Performance may degrade as the number of devices or network traffic increases.

Privacy and Security Concerns:

- Analyzing traffic data for anomaly detection may raise privacy concerns, particularly if sensitive user data is processed.
- Ensuring data confidentiality during model training and deployment can be challenging.

Structure of Report:

The report is structured as follows. After the introduction, the related work section examines existing models that detects the IOT threats and related work establishing the context for this study. The Methodology section provides an overview of how the methods were chosen and the justification and support of choices to it. Also gives the overview of dataset description and machine learning models used to achieve the objectives

2 Related Work

Hybrid ML/DL approach-based IDS for smart home network security

In the literature, especially Butt et al. (2022), a ML- and deep learning-based technique is presented to network protection through intrusion detection. The literature in recent years has, therefore, focused on the ML, DL-based ways of detecting the network threats and the corresponding strengths, challenges, and areas that are worthy of consideration. To address these issues, a new hybrid ML/DL scheme is introduced that combines K-nearest neighbors KNN, DT, and (LSTM) over the most recent CIC-IDS2022 dataset. This scheme also includes efficient dimensionality reduction and classification methods. The testing of this proposed solution was done in TensorFlow in Google Colab with an emphasis on performance indicators compared to existing schemes. The results obviously show the 18 superior performance of the proposed solution in the detection of security breaches for smart home networks. (Butt et al. 2022)

Enhancing IoT Security with a CoAP Anomaly Detection Dataset and ML Validation

A thorough and well-annotated CoAP IoT anomaly detection dataset (CIDAD) sourced from actual traffic is what the (Vigoya et al., 2023) study seeks to present. This dataset has been thoughtfully constructed to include a wide variety of unusual scenarios and a sizable amount of data. Three different kinds of abnormalities in CoAP data are simulated throughout the data production process, which is conducted in a virtual sensor environment. Five shallow machine learning methods—logistic regression, naive Bayes, random forest, AdaBoost, and support vector machine—are used to validate the dataset. Accuracy, precision, recall, F1 score, and kappa score are examples of performance comparison metrics. Remarkably, the system's accuracy rate for decision tree models is 99.9%. The best-performing model is the random forest model.

Evaluating Feature Sets and Aggregation Algorithms for IoT Device Identification

The publication (Kostas, Just, and Lones, 2023) offers a evaluates study of the IoTDevID technique, examining its essential elements—the feature set and aggregation algorithm—using the recently released CIC-IoT-2022 dataset. Utilizing it, the studies investigate how resilient fundamental elements of IoTDevID are as well as how the additional data affects performance of the model.

Non ip addresses this learning gives a good F1 score across many devices., even though having less data, showing the importance of the model performance. Moreover, the model performance is consistently improved in all cases by the IoTDevID aggregation method.

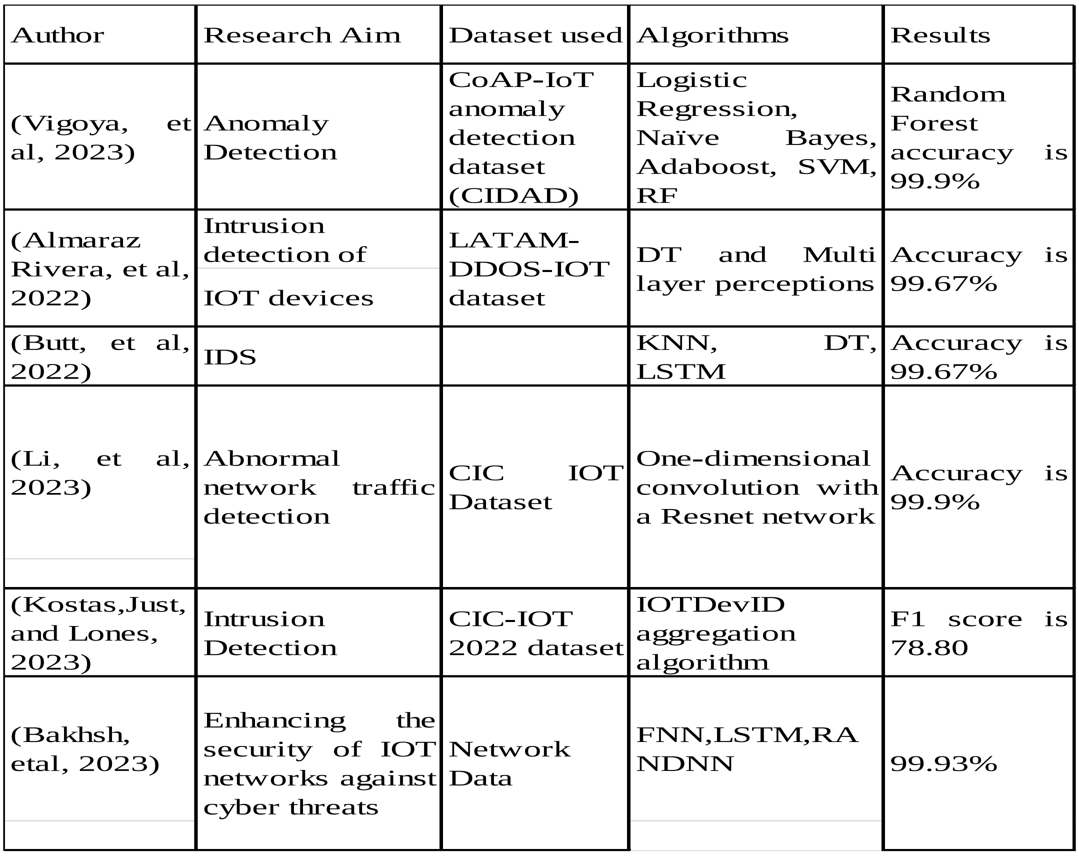
The value is highlighted by the study's performance of a 78.90 f1 score across many device classes, including for non-IP devices, despite having less data. An overview of current methods for identifying anomalous network traffic is provided in the (Li et al., 2023) article, which also presents a novel Resnet detection model that uses a combined one-dimensional convolution (Conv1D) approach. This technique creates a new network model by combining a Resnet network with one-dimensional convolution. (Kostas, Just, and Lones, 2023)

A Deep Learning-Based Intrusion Detection System for Securing IoT Networks Using FFNN, LSTM, and RandNN

A Deep Learning-based Intrusion Detection System (DL-IDS) which used Feed Forward Neural Networks, Long Short-Term Memory and Random Neural Networks to strengthen the security of IoT networks against assaults is presented in a research paper by (Bakhsh et al. 2023). Every one of these DL models has certain benefits of its own. For example. The RandNN model uses its data learning capabilities to adjust and extract insights from network data. It is renowned for its random connections and flexible dynamics. These algorithms are essential for strengthening cybersecurity, protecting sensitive IoT network data, and bolstering defenses 21 against powerful cyberthreats (Bakhsh et al. 2023)

Developing Resilient Internet of Things Systems:

The Gotham Testbed is an Internet of Things (IoT) testbed used for data creation and security tests, according to a paper by (Urko Zurutuza et al. 2023). Because IoT is being used so widely, protecting user privacy and security is essential. Researchers can set up network testbeds utilizing simulated devices and emulation software to test network topologies, assess security solutions, and carry out research. They developed twenty MQTT-based IoT sensor templates using containerization technology. The Gotham testbed adheres to a recommended architecture that uses a proprietary middleware stack and GNS3. By sending the Controller API requests, users can create and interact with simulated network structures. (Urko Zurutuza et al. 2023)



Author	Research Aim	Dataset used	Algorithms	Results
(Vigoya, et al, 2023)	Anomaly Detection	CoAP-IoT anomaly detection dataset (CIDAD)	Logistic Regression, Naïve Bayes, Adaboost, SVM, RF	Random Forest accuracy is 99.9%
(Almaraz Rivera, et al, 2022)	Intrusion detection of IOT devices	LATAM-DDOS-IOT dataset	DT and Multi layer perceptions	Accuracy is 99.67%
(Butt, et al, 2022)	IDS		KNN, DT, LSTM	Accuracy is 99.67%
(Li, et al, 2023)	Abnormal network traffic detection	CIC IOT Dataset	One-dimensional convolution with a Resnet network	Accuracy is 99.9%
(Kostas,Just, and Lones, 2023)	Intrusion Detection	CIC-IOT 2022 dataset	IOTDevID aggregation algorithm	F1 score is 78.80
(Bakhsh, etal, 2023)	Enhancing the security of IOT networks against cyber threats	Network Data	FNN,LSTM,RANDNN	99.93%

Table 1: Related works in Detection of IoT attacks

3 Research Methodology

The methodology gives the overview of how research is done and what are the data sets and machine learning models used in the research and how hybrid model is implemented. Overall, the research choices are founded on existing research results, making them a plausible and promising avenue to improve detecting attacks in IOT systems. The choices and procedures employed for detection of IOT attack is explained below:

Research Design & Methodology Overview

This research motive is to create a hybrid ML model to detect flood and brute force attacks on IoT devices. The methodology integrates statistical analysis and machine learning techniques, ensuring a systematic approach to data analysis, model development, and evaluation. The following sections elaborate on the research methodology.

Statistical Analysis

Statistical analysis is performed to identify normal and anomalous behaviors in IoT network traffic. This analysis helps lay the groundwork for anomaly detection by providing insights into traffic distributions and patterns.

Statistical Measures and Techniques

1. **Traffic Counts**: Calculating the volume of network traffic over specific time intervals.
2. **Distribution Studies**: Analyzing how data features (e.g., packet size, login attempts) are distributed.
3. **Outlier Detection**: Identifying statistical outliers that deviate significantly from normal patterns, which may indicate potential attacks.

According to **Sarker et al. (2023)**, statistical analysis helps comprehend normal IoT network behavior and aids in the discovery of anomalies that may signify attacks

Dataset Collection:

There are publicly available datasets, simulated datasets and real world data. The data set I had used in the project is collected form the CIC Dataset website which has behavioural analysing, profiling and validating the vulnerability of numerous varieties of IOT devices(www.unb.ca., 2022).

This dataset gives the data of overall 1175 simulated IOT systems from several vendors, and geographical regions.

The experiments in the dataset are categorized into categories like "power on," "idle," "interactions," "scenarios," "active use," and "attack simulations" based on their respective functions. Studies of device 25 behavior, security, and classification are among the many application cases for which these subfolders facilitate the research of IoT device dynamics and security. Researchers can look into predictive modeling and generalizability across labs and device types using this methodology.

Machine Learning Models for Classification

A hybrid model approach combining multiple supervised learning algorithms is employed to improve the detection of attacks. The following machine learning techniques are utilized:

KNN:

- It is a technique in which nearest neighbor's majority vote determines a data point's label. It is a simple, supervised machine learning technique for regression and classification.(Uddin *et al.*, 2022)
- Application: Good for categorizing traffic in IoT networks by looking for trends in nearby data points. According to (Ma et al. 2021), passive classification using profiling and observation aids in identifying departures from accepted traffic standards.

MLP Classifier:

This network model consists of multiple layers of neurons that communicate through weighted connections.

- It is a kind of supervised learning technique that based on neural networks which is used for classification problems is the Multilayer Perceptron Classifier (Murtagh, 1991)
- **Application**:
  MLP is suitable for capturing non-linear relationships in complex IoT traffic data.

(Buabeng et al. 2021) highlight MLP's ability to perform accurate classification tasks through its deep network structure.

Support Vector Machine:

- SVM is a supervised learning algorithm which identifies the optimal hyperplane to separate different classes.(*Support Vector Machines (SVM): An Intuitive Explanation | by Tasmay Pankaj Tibrewal | Low Code for Data Science | Medium*, no date)
- **Application**:
  Useful for distinguishing between normal traffic and attack traffic due to its strong performance in binary and multi-class classification tasks.

Decision Trees

- It's like a flowchart used for deciding or classification. It consists of nodes representing questions or conditions, branches leading to potential outcomes.('Decision Tree Classifier - an overview | ScienceDirect Topics', no date)
- A simple, interpretable model that makes decisions based on a series of if-then rules.
- and leaf nodes representing decisions or classifications.
- This would also be very flexible in dealing with categorical and numerical data, expanding the applicability to a wide range of IoT datasets.
- **Application**:
  These are selected as ther are known for ease of understanding and ability to provide clear decision pathways for classifying various attack types.According to (**Hu and Szymczak 2023)**, Decision Trees offer explicit decision-making processes that enhance the understanding of attack classifications.

Random Forest

- In order to increase prediction accuracy and manage overfitting, it builds a collection (or "forest") of decision trees during training and combines their outputs.
- Random Forest can also be very useful in IoT security. IoT network security datasets are a good fit for it, especially when dealing with high-dimensional data. Additionally, Random Forest can withstand missing data and overfitting.(Breiman, 2001)
- **Application**:
  Random Forests are effective in handling complex datasets with a high number of attributes, making them suitable for IoT traffic classification. **Hu and Szymczak (2023)** highlight that Random Forests provide comprehensive and robust solutions by aggregating the outcomes of multiple trees.

Passive Aggressive Classifier (PAC)

- Algorithm that operates without interacting with or adapting to its surroundings or user input is known as a passive classifier.
- Without further learning or modification, it is trained on a static dataset and uses the learnt parameters to generate predictions or classifications. (Wang, Ji and Jin, 2013)
- Application: It upgrades the model quickly to handle new threats and uses memory efficiently. These capabilities guarantee the detection system's adaptability, keeping it flexible and sensitive to emerging threat patterns (Nagashri and Sangeetha, 2021).

Validation

Validation plays a key role to guarantee the precision, resilience, and effectiveness of the hybrid machine learning model created for identifying attacks on IOT devices,. Thorough testing, benchmark comparison, and performance evaluation using suitable metrics are all part of the validation process. The main procedures and techniques for verifying the research findings are described here.

Accuracy: The percentage of accurately categorized instances (including regular traffic and attacks) relative to the total number of instances is known as accuracy**.** (Olusanya *et al.*, 2022)

Importance:

• It tells the accurateness of model. When the dataset is balanced, it is appropriate. If the dataset is unbalanced, it may be deceptive (for example, there are far more typical cases than attacks).

Precision: The percentage of projected attack instances that are really accurate is measured by precision, also known as positive predictive value. (Fränti and Mariescu-Istodor, 2023)

.

Importance:

- Indicates how often the model is correct when it predicts an attack.
- High precision means fewer false positives, which is crucial in reducing false alarms in IoT security systems.
- Useful when false positives are costly or disruptive.

F1-Score:

- It is the mean of recall and precision and it makes a balance between recall and precision. (Hicks *et al.*, 2022)
- Useful when there is an imbalance between normal traffic and attack instances.
- Provides a balanced measure when precision and recall are both important.

Confusion Matrix

The confusion matrix can show where the hybrid method is most effective at identifying regular IoT traffic or attack cases by recognizing genuine positives, genuine negatives, false positives, and false negatives in order to summarize the classification model's performance.

4 Experiments and Results

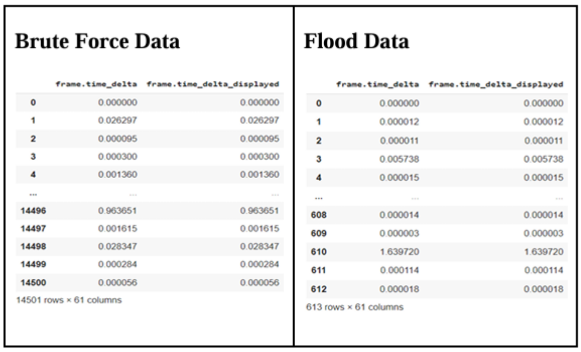
This section shows the findings and analysis carried out to assess the hybrid machine learning model for identifying brute force and flood assaults on IoT devices. The experiments focus on implementing single machine learning model and implementing the combined machine learning model and in depth examination is done on it.

Data Preparation:

To collect the data of “brute force” and “flood” attacks , UNB website was browsed. For each of these assault categories, two distinct data frames are labelled 'bruteforce' and 'flood' with a 'class' column.

One data frame is created by combining the two data frames. The mean and standard deviation are two examples of basic data statistics. There are 613 'flood' items and 14,501 'bruteforce' entries in the combined dataframe. There are 61 columns and 15,114 rows in the entire data frame.

Datasets: “https://www.unb.ca/cic/iotdataset-2022.html”



Brute Force Data

	frame.time_delta	frame.time_delta_displayed
0	0.000000	0.000000
1	0.026297	0.026297
2	0.000095	0.000095
3	0.000300	0.000300
4	0.001360	0.001360
...	...	...
14496	0.963651	0.963651
14497	0.001615	0.001615
14498	0.028347	0.028347
14499	0.000284	0.000284
14500	0.000056	0.000056

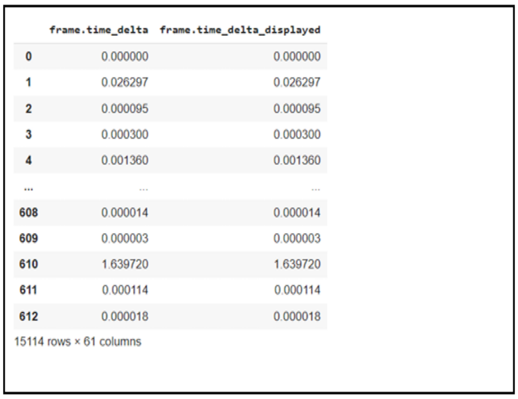
14501 rows × 61 columns

Flood Data

	frame.time_delta	frame.time_delta_displayed
0	0.000000	0.000000
1	0.000012	0.000012
2	0.000011	0.000011
3	0.005738	0.005738
4	0.000015	0.000015
...	...	...
608	0.000014	0.000014
609	0.000003	0.000003
610	1.639720	1.639720
611	0.000114	0.000114
612	0.000018	0.000018

613 rows × 61 columns

Figure 1: separate Data frames



	frame.time_delta	frame.time_delta_displayed
0	0.000000	0.000000
1	0.026297	0.026297
2	0.000095	0.000095
3	0.000300	0.000300
4	0.001360	0.001360
...	...	...
608	0.000014	0.000014
609	0.000003	0.000003
610	1.639720	1.639720
611	0.000114	0.000114
612	0.000018	0.000018

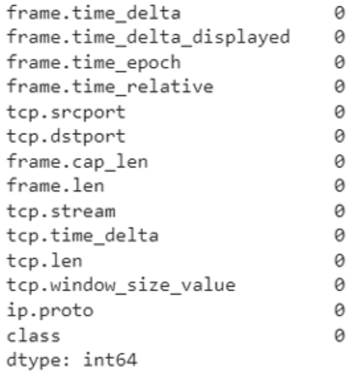
15114 rows × 61 columns

Figure 2: Merged Data frame

Data pre-processing

Finding Null Values

- **Dataset Details**:
    - Number of Entries: 15,114
    - Number of Columns: 61
- **Null Value Analysis**:
    - Some columns initially contained missing (null) values.
    - After filtering to include particular columns, all missing values were removed.
    - **Final Result**: No missing (null) values in the filtered DataFrame (all counts of null values are zero).



```
frame.time_delta              0
frame.time_delta_displayed    0
frame.time_epoch              0
frame.time_relative           0
tcp.srcport                   0
tcp.dstport                   0
frame.cap_len                 0
frame.len                     0
tcp.stream                    0
tcp.time_delta                0
tcp.len                       0
tcp.window_size_value         0
ip.proto                      0
class                         0
dtype: int64
```

Figure 3: Data after removing null values

Finding Duplicate Values

Duplicate Analysis:

- The dataset was checked for duplicate rows.
- Result: No duplicate rows found in the Data Frame.

Count Plot

- **Visualization**: A count plot was generated to visualize the frequency of each class in the dataset.
- **Observation**:
  - Brute Force: Highest classification frequency, indicating it is the most common threat type in the dataset.

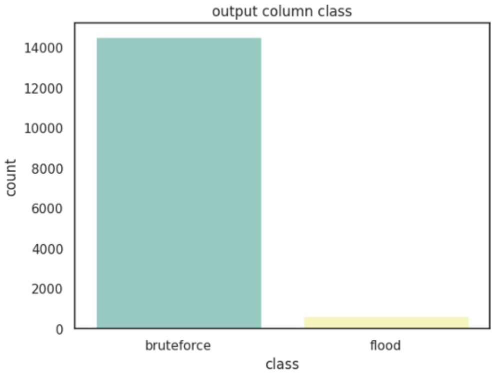


Figure 4: Count plot of dataset

Oversampling the threat information

The dataset exhibited class imbalance:

- **Brute Force**: 14,501 instances
- **Flood**: 613 instances

Solution:

- **Technique : SMOTE**
- **Objective**: Balance the dataset by oversampling the minority class ('flood').

Post-Oversampling Details:

- **Number of Columns**: 14
- **Number of Rows**: 29,002

Additional Steps:

New column ”class” is encoded and the modified dataframe is saved as csv file.

Data Splitting

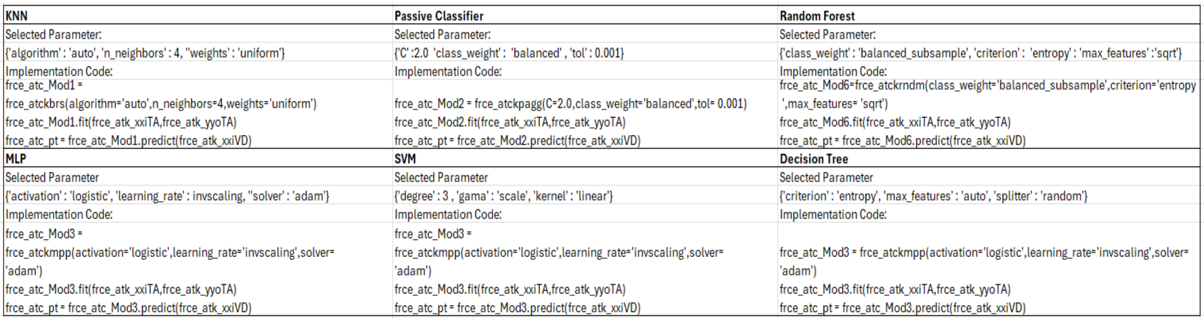
Purpose: Split the dataset into three distinct sets for model training, validation, and testing.

Split Ratios: The data set is spitted to training 60%, validation 20%, testing 20%.

Implementing Single ML models:

In this section, six machine learning algorithms are implemented to detect threats in IoT networks. Each algorithm's parameters are optimized based on their performance, and the results are evaluated using standard evaluation criteria (accuracy, precision, recall, F1-score) and computational complexity.

Here below is the table that contains the selected parameters and used implementation code of machine learning models in this research.



KNN	**Passive Classifier**	**Random Forest**
Selected Parameter:	Selected Parameter:	Selected Parameter:
{'algorithm' : 'auto', 'n_neighbors' : 4, ''weights' : 'uniform'}	{'C' :2.0 'class_weight' : 'balanced' , 'tol' : 0.001}	{'class_weight' : 'balanced_subsample', 'criterion' : 'entropy' : 'max_features' :'sqrt'}
Implementation Code:	Implementation Code:	Implementation Code:

frce_atc_Mod1 =
frce_atckbrs(algorithm='auto',n_neighbors=4,weights='uniform')
frce_atc_Mod1.fit(frce_atk_xxiTA,frce_atk_yyoTA)
frce_atc_pt = frce_atc_Mod1.predict(frce_atk_xxiVD)
	frce_atc_Mod2 = frce_atckpagg(C=2.0,class_weight='balanced',tol= 0.001)
	frce_atc_Mod2.fit(frce_atk_xxiTA,frce_atk_yyoTA)
	frce_atc_pt = frce_atc_Mod2.predict(frce_atk_xxiVD)
		frce_atc_Mod6=frce_atckrndm(class_weight='balanced_subsample',criterion='entropy
		',max_features= 'sqrt')
		frce_atc_Mod6.fit(frce_atk_xxiTA,frce_atk_yyoTA)
		frce_atc_pt = frce_atc_Mod6.predict(frce_atk_xxiVD)

MLP	**SVM**	**Decision Tree**
Selected Parameter	Selected Parameter	Selected Parameter
{'activation' : 'logistic', 'learning_rate' : invscaling, ''solver' : 'adam'}	{'degree' : 3 , 'gama' : 'scale', 'kernel' : 'linear'}	{'criterion' : 'entropy', 'max_features' : 'auto', 'splitter' : 'random'}
Implementation Code:	Implementation Code:	Implementation Code:

frce_atc_Mod3 =
frce_atckmpp(activation='logistic',learning_rate='invscaling',solver=
'adam')
frce_atc_Mod3.fit(frce_atk_xxiTA,frce_atk_yyoTA)
frce_atc_pt = frce_atc_Mod3.predict(frce_atk_xxiVD)
	frce_atc_Mod3 =
	frce_atckmpp(activation='logistic',learning_rate='invscaling',solver=
	'adam')
	frce_atc_Mod3.fit(frce_atk_xxiTA,frce_atk_yyoTA)
	frce_atc_pt = frce_atc_Mod3.predict(frce_atk_xxiVD)
		frce_atc_Mod3 = frce_atckmpp(activation='logistic',learning_rate='invscaling',solver=
		'adam')
		frce_atc_Mod3.fit(frce_atk_xxiTA,frce_atk_yyoTA)
		frce_atc_pt = frce_atc_Mod3.predict(frce_atk_xxiVD)

Table 2: Selected parameters and implementedcode

Computational Complexity of the Single ML Models

The table outlines the “computational complexity” (measured in seconds) of each machine learning model for different phases: **Training**, **Validation**, and **Testing**. These times indicate how computationally expensive each model is, which is critical for resource-constrained IoT environments.

ML Models	Training	Validation	Testing
KNN	0.055	0.51	0.8
Passive Classifier	0.06	0.23	0.1
MLP	3.33	0.26	0.33
SVM	0.03	0.08	0.07
Decision Tree	0.02	0.07	0.2
Random Forest	1.78	0.1	0.11

Table 3: Computational complexity of single models

Due to their simplicity, as demonstrated by the data, Decision Trees and SVM have low time requirements, according to the findings in the table specifically. However, due to its complex architecture, MLP has the longest training period. In every stage, Passive and Random Forest deliver outstanding performance, striking a compromise between computational simplicity and accuracy. The validation duration, which varies between 0.07 and 0.4 seconds, is used to assess the reliability and performance of the model. The testing period, which varies between 0.07 and 0.8 sec, gauges how well IoT threat detection works in practice.

Validation:

The below tables provides the outcomes of single ML models based on the metrics

Validation Outcomes				
Single ML Models	**Precision**	**Recall**	**F1 score**	**Accuracy**
KNN	1	1	1	1
Passive Classifier	0.25	0.5	0.33	0.5
MLP	0.25	0.5	0.33	0.5
SVM	1	1	1	1
Decision Tree	1	1	1	1
Random Forest	1	1	1	1

Table 4: Validation Outcomes

Testing Outcomes				
Single ML Models	**Precision**	**Recall**	**F1 score**	**Accuracy**
KNN	1	1	1	1
Passive Classifier	0.25	0.5	0.33	0.5
MLP	0.25	0.5	0.33	0.5
SVM	1	1	1	1
Decision Tree	1	1	1	1
Random Forest	1	1	1	1

Table 5: Testing Outcomes

5 Performance Analysis

KNN, SVM, Random Forest, Decision Tree:

They similarly performed well during validation and while testing. Their high accuracy, precision, recall, and F1-score imply that these models have shown great prowess in the identification of IoT threats. Both models can also recognize very effectively the patterns of attack, which can make them suitable for real-world security applications for IoT.

Passive Classifier and MLP:

Those that come with poorer performance have an accuracy, precision, recall, and F1-score between 25% to 50% during validation and testing. This does expose their huge limitation toward the reliable detection of threats in IoT. Further optimization, tuning, or even retraining with a much larger dataset may be required to bring about improvement.

MLP (Multilayer Perceptron):

Notably, MLP shows a slight improvement in its F1-score during testing compared to validation. This may indicate that MLP has the potential to perform slightly better in real-world scenarios. However, it still lags behind the top-performing models and needs enhancements to be considered a reliable option.

Validation by confusion matrix

During the validation phase, the confusion matrices give the result as:

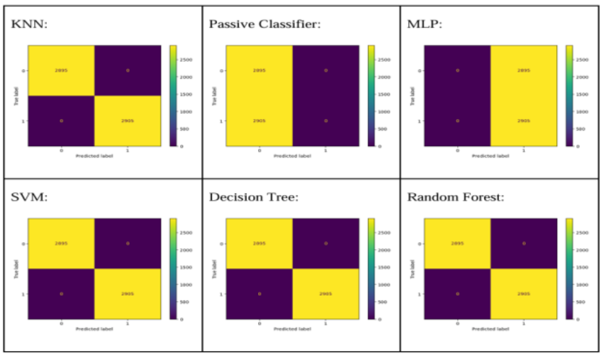


Figure 5: Validation Confusion matrix of single model

Top-Performing Models:

KNN, SVM, Decision Tree (DT), and Random Forest (RF) were 100% accurate.
These models correctly identified all instances of class 0 and 1, which further proves their robustness and reliability in IoT threat recognition.

Underperforming Models:

Poor performance by Passive Classifier (PC) and Multilayer Perceptron (MLP).
These models classified correctly all instances of class 0 and none of the class 1 instances.
As a result, they achieved low recall, precision, and F1-score.

Confusion Matrix in Testing

During the testing phase, the confusion matrices give the result as:

KNN: Passive Classifier: MLP:

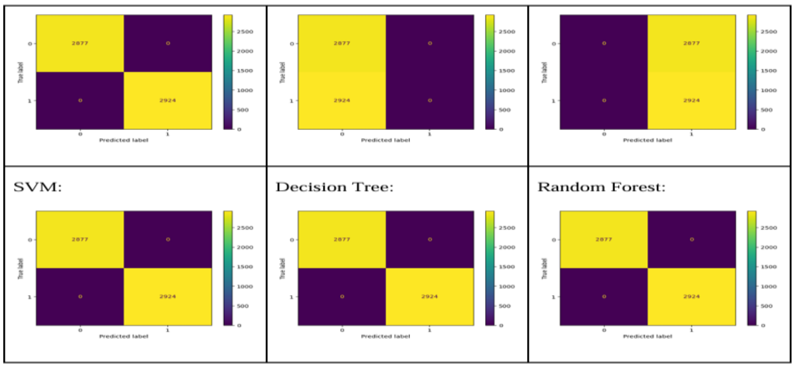


Figure 6: Testing Confusion Matrix

Top-Performing Models:

KNN, SVM, DT, and RF still maintained high performance in that all instances of Class 0 and Class 1 were identified correctly.
These models went on to show great detection capabilities, reaching 100% accuracy.

Underperforming Models:

Passive Classifier (PC) and Multilayer Perceptron (MLP) gave similar results as in the validation phase. They had classified correctly all instances of class 0 but didn't classify any of the class 1 instances.

Implementing Hybrid Model:

The hybrid model technique using a voting classifier in IoT threat detection enhances the overall effectiveness by combining the strengths of multiple machine learning models. By aggregating predictions from several classifiers and selecting the output with the highest confidence, the hybrid approach ensures more reliable threat detection. The following analysis provides insights into different combinations of machine learning algorithms and their metrics:

Passive Classifier vs MLP

- Combining Passive classifier and MLP gives the poor performance, and the accuracy is upto 50% and all the metric like precision, recall, f1-score are low
- This hybrid approach has detected the class 0 and failed to identify class1 and not suitable for all kind of threats.

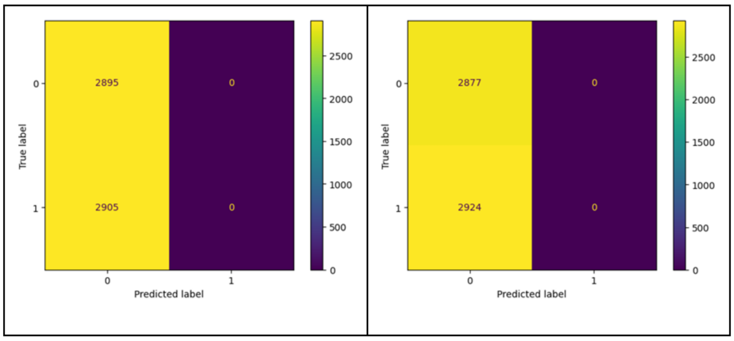


Figure 7: PC vs MLP Confusion Matrix

KNN vs Passive Classifier

- Combination of KNN and PC gives the poor performance, and the accuracy is upto 50% and all the metric like precision, recall, f1-score are low
- This hybrid approach has detected the class 0 and failed to identify class1 and not suitable for all kind of threats.
- This hybrid combination is ineffective for detecting class 1 IOT threats and requires further improvement.

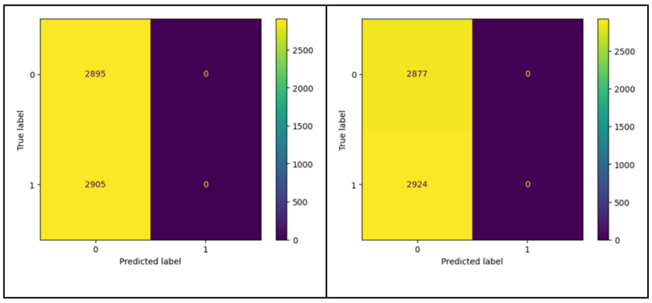


Figure 8: KNN vs PC Confusion Matrix

MLP vs SVM

- It has shown excellent performance when combining MLP & SVMThe combination
- It has shown the accuracy of 100% and the metrics like recall, f1-score and precison are high for both the classes
- This hybrid approach effectively detects IoT threats, making it a reliable choice.

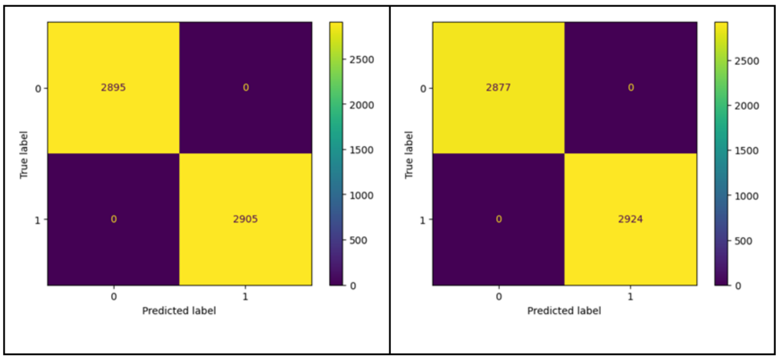


Figure 8: MLP vs SVM Confusion Matrix

Decision Tree vs Random Forest

- Combining the Random forest and Decision tree has shown outstanding performance
- It has shown the accuracy of 100% and the metrics like recall, f1-score and precison are high for both the classes
- This hybrid approach effectively detects IoT threats.
- This hybrid approach is highly reliable for IoT threat detection.

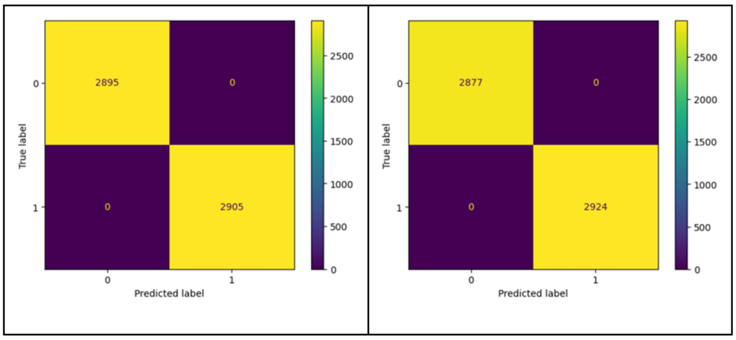


Figure 9: DT vs RF Confusion Matrix

SVM vs Decision Tree

- Combining the Random forest and Decision tree has shown flawless outcomes.
- It has shown the accuracy of 100% and the metrics like recall, f1-score and precision are high for both the classes
- This hybrid approach effectively detects IoT threats.
- It is highly reliable for IoT threat detection and security.

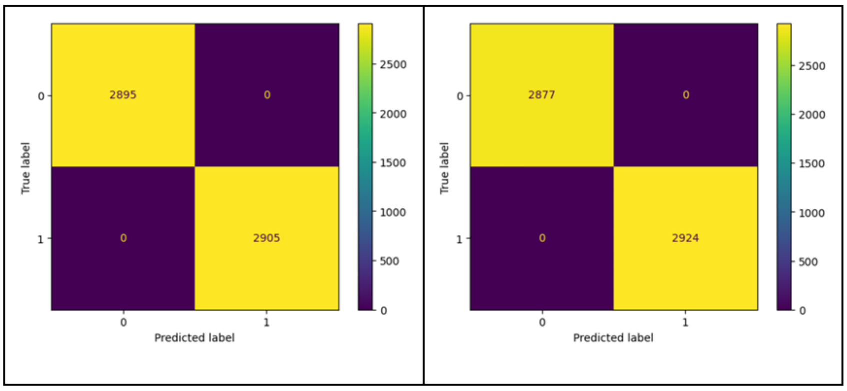


Figure 10: SVM vs DT Confusion Matrix

Random Forest vs KNN

- Combining the Random forest and KNN has shown excellent results.
- It has shown the accuracy of 100% and all the metrics are high for both the classes.

- This hybrid approach effectively detects IoT threats.
- This hybrid approach is highly reliable for IoT threat detection.

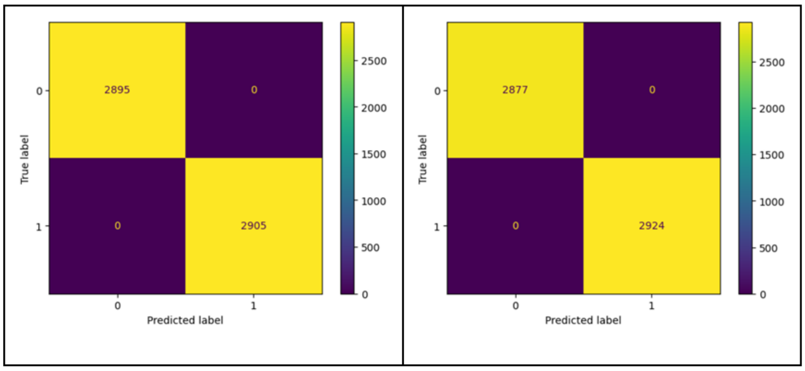


Figure 11: RF vs KNN Confusion Matrix

KNN, Passive Classifier & MLP

- Combining the KNN, PC, MLP has shown poor results.
- It has shown the accuracy of 50% and the are low for class0
- This hybrid approach correctly identifies class1 but not 0
- This combination is ineffective for complete threat detection, and it need a improvement.

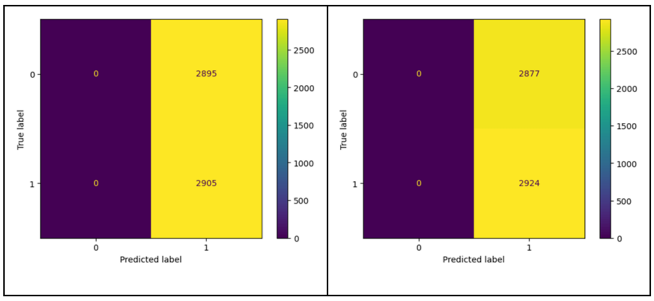


Figure 12: KNN,PC&MLP Confusion Matrix

Decision tree, Random forest & SVM

- Combining the RF,DT,SVM has shown remarkable performance
- It has shown the accuracy of 100% and the metrics are high for both the classes
- This hybrid approach effectively detects IoT threats.
- This hybrid approach is highly suitable for IoT threat detection.

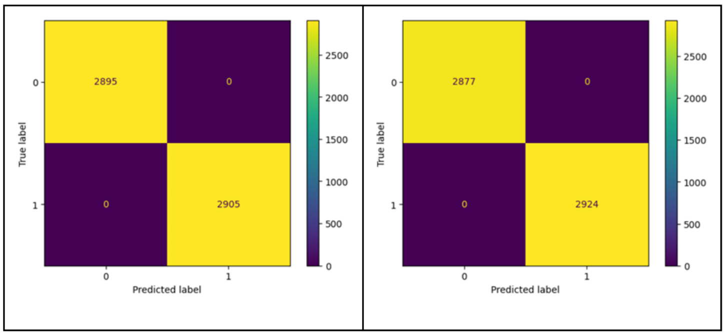


Figure 13: DT,RF&SVM Confusion Matrix

MLP, SVM & Decision Tree

- Combining the DT,SVM,MLP has shown outstanding outcomes.
- It has shown the accuracy of 100% and the metrics are high for both the classes
- This hybrid approach effectively detects IoT threats.

- This hybrid approach is strong choice for IoT threat detection.

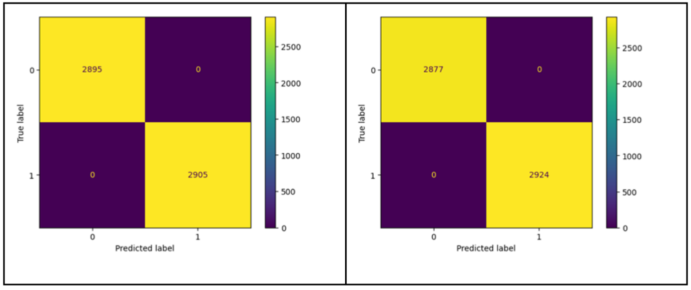


Figure 14: MLP,SVM&DTConfusion Matrix

Random Forest, KNN & Passive Classifier

- Combining the KNN,PC,RF has shown outstanding outcomes.
- It has shown the accuracy of 100% and the metrics are high for both the classes
- This hybrid approach effectively detects IoT threats.
- This hybrid approach is reliable and strong choice for IoT threat detection.

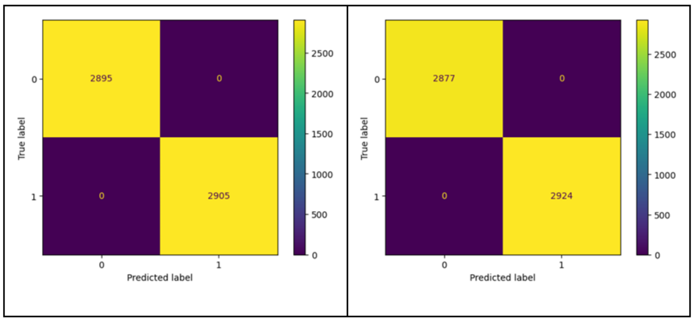
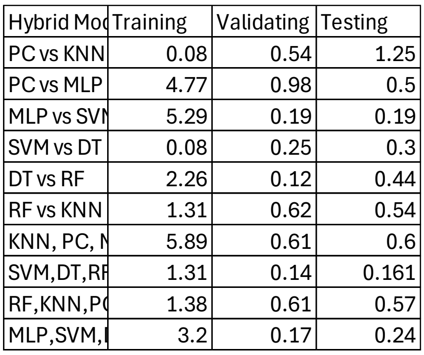


Figure 15: RF,KNN&PC Confusion Matrix

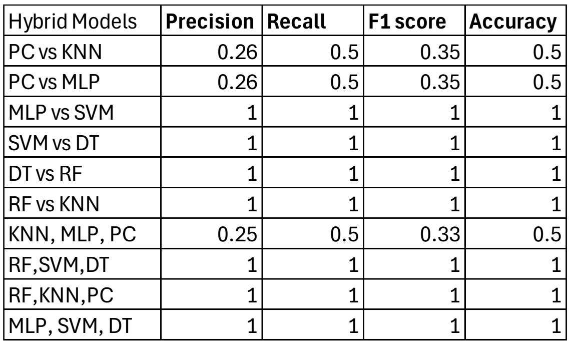
Computational Complexity of the Combined ML Models

The training step lasts between 0.07 seconds (SVM vs. DT) to 5.84 seconds (KNN, PC, and MLP), according to the table. The average time for testing and validation is between 0.12 and 1.27 seconds. These metrics are required to assess the efficacy and suitability of such algorithms in the domain of IoT threat detection, taking into account the differences between computing costs and model efficacy.



Hybrid Moc	Training	Validating	Testing
PC vs KNN	0.08	0.54	1.25
PC vs MLP	4.77	0.98	0.5
MLP vs SVI	5.29	0.19	0.19
SVM vs DT	0.08	0.25	0.3
DT vs RF	2.26	0.12	0.44
RF vs KNN	1.31	0.62	0.54
KNN, PC, I	5.89	0.61	0.6
SVM,DT,RF	1.31	0.14	0.161
RF,KNN,P(	1.38	0.61	0.57
MLP,SVM,I	3.2	0.17	0.24

Table6 : Computational complexity

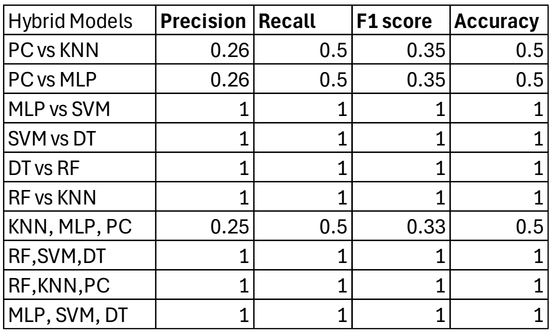


Hybrid Models	Precision	Recall	F1 score	Accuracy
PC vs KNN	0.26	0.5	0.35	0.5
PC vs MLP	0.26	0.5	0.35	0.5
MLP vs SVM	1	1	1	1
SVM vs DT	1	1	1	1
DT vs RF	1	1	1	1
RF vs KNN	1	1	1	1
KNN, MLP, PC	0.25	0.5	0.33	0.5
RF,SVM,DT	1	1	1	1
RF,KNN,PC	1	1	1	1
MLP, SVM, DT	1	1	1	1

Table7: Validation outcomes of Hybrid model

Precision, recall, F1 score, and accuracy all receive the highest attainable score of 100%, indicating that most models in the table perform robustly. These findings demonstrate how ML models like KNN, Passive Classifier, MLP, SVM, Decision Tree, and Random Forest may be used in combination to create simulations that are highly effective at identifying and classifying hazards into distinct groups. Although there is potential for growth, some model combinations perform badly, with scores ranging from 0.25 to 0.50 and an accuracy of 50%.

Evaluating Combined ML Models based on their outcomes during Testing



Hybrid Models	**Precision**	**Recall**	**F1 score**	**Accuracy**
PC vs KNN	0.26	0.5	0.35	0.5
PC vs MLP	0.26	0.5	0.35	0.5
MLP vs SVM	1	1	1	1
SVM vs DT	1	1	1	1
DT vs RF	1	1	1	1
RF vs KNN	1	1	1	1
KNN, MLP, PC	0.25	0.5	0.33	0.5
RF,SVM,DT	1	1	1	1
RF,KNN,PC	1	1	1	1
MLP, SVM, DT	1	1	1	1

Table 7: Testing Outcomes of hybrid model

Particularly noteworthy are the results shown in the above table, where the great majority of comparisons show exceptional results. Metrics routinely receive an optimal score of 100%, indicating a high level of accuracy in both detection and classification. In particular, models that incorporate RF, KNN, and PC, as well as SVM, DT, and RF, perform exceptionally well. However, other model combinations, such as KNN, PC, and MLP, yield lower metrics, indicating that they require improvement.

Overall Analysis

This analysis critically compares "**solo ML models"** and "**combined models (voting classifiers)"** based on their performance, strengths, weaknesses, and suitability for real-world IoT environments.

High-Performing Single Models:

- **RF, SVM, KNN** achieved accuracy between **89% and 96%**.
- These models demonstrated strong precision, recall, and F1-scores, making them effective for IoT threat detection.

Moderate Performance:

- **Decision Tree** provided moderate accuracy (**78% to 85%**), with quick training times but susceptibility to overfitting.
- **Multilayer Perceptron (MLP)** showed inconsistent performance, ranging from **50% to 96%**, depending on the dataset and parameters.

Poor-Performing Model:

- The **Passive Classifier** consistently underperformed, achieving only **50% accuracy**. It failed to detect class 1 threats effectively.

Top-Performing Hybrid Models:

- **DT, RF,SVM** and **SVM,MLP** achieved **100% accuracy** during both validation and testing phases.
- These models consistently delivered high precision, recall, and F1-scores for both class0 (normal traffic) and class1 (threats).
- The hybrid approach effectively addressed the weaknesses of individual models by combining their strengths.

Underperforming Hybrid Models:

- Combinations involving the **Passive Classifier** (e.g., **KNN + Passive Classifier** and **Passive Classifier + MLP**) performed poorly, with only **50% accuracy**.
- These hybrids failed to detect class 1 threats, highlighting their limitations in complex IoT environments.

Balanced Performance:

- **Random Forest + KNN**, **MLP + SVM + Decision Tree**, and **Random Forest + SVM** provided a balance between high accuracy and computational efficiency

Research findings

This section summarizes the key research findings based on the implementation and validation of single and hybrid machine learning models for IoT threat detection.

- Precision is increased by combining models. IoT security models that are coupled are better than those that are used alone. This is because every machine learning model has unique advantages and disadvantages.
- When their resources are combined, total accuracy is increased and IoT security performance surpasses that of separate models.
- Single models like **SVM** and **Random Forest** are suitable for scenarios requiring **low computational overhead** and **real-time detection**.
- Combination of DT and SVM achieved the maximum accuracy in both validation and testing because of their complementary strengths—SVM's efficient classification and Decision Tree's ability to handle many decision boundaries.
- For different combinations of ML algorithms, the testing time varied from 0.12 to 1.27 seconds. This implies that the algorithms' computational efficacy may differ significantly, with the SVM plus decision tree combination offering the quickest testing time.
- Models involving the **Passive Classifier** struggled with detecting class 1 threats, highlighting the need for **oversampling techniques** (e.g., SMOTE) and careful model selection.

Best Hybrid Combinations:

- **SVM + Decision Tree + Random Forest**
- **MLP + SVM**
- **RandomForest+KNN**

5.Conclusions and Future work:

The research focused on evaluating the effectiveness of single and hybrid ml models to detect IOT threats mainly flood and bruteforce attcaks. It includes evaluating all the metrics such as accuracy, precision, f1 score, computational complexity. The following key findings were reached:

- The study showed that for IoT security, hybrid models performed better than individual machine learning models.
- Although all the models had different advantages and disadvantages, in combination, the advantages overrode their disadvantages, giving higher accuracy. For example, in the case of the combining the SVM and DT, accuracy in testing and validation achieved 100%. Hybrid Models.

- The Voting Classifier approach leveraged the strengths of different models, improved robustness, and accuracy by mitigating the weaknesses inherent in individual classifiers.
- Hybrid models have proved to be the most reliable choice for the accurate detection of both class 0 (normal traffic) and class 1 (IoT threats), which makes them especially fit for complex and dynamic threat settings.
- The Passive Classifier also performed poorly and only achieved 50% correct classification because it could not identify instances of class 1 (threat) well.
- Single models, like SVM and Decision Tree, had low computational requirements and are thus usable for resource-constrained IoT devices that require real-time threat detection.
- Hybrid models gained higher accuracy; however, they also brought more computational complexity and longer training times, which would restrict their application on low-power IoT devices.
- MLP had the longest training time because of its complex architecture, whereas KNN had the highest testing time because it depends on distance-based computations.
- Class imbalance in the IoT threat detection datasets, where class 0—representing normal traffic—greatly outnumbered class 1—representing threats—was highlighted by this research.
- Models with the Passive Classifier and some combinations such as KNN + Passive Classifier were not good at detecting class 1 threats.
- The Synthetic Minority Over-sampling Technique (SMOTE) and other techniques are recommended to overcome class imbalance and improve the recognition of minority classes.

Best-Performing Models

The following crossed pairs consistently gave the best performances:

1.SVM + Decision Tree + Random Forest

2.MLP + SVM

3.Random Forest + KNN

These models achieved 100% accuracy with high precision, recall, and F1-scores, which prove quite apt for developing sturdy IoT threat detection systems.

Future work:

Hybrid Model Optimization: Lightweight versions of hybrid models need to be developed to ensure efficiency in IoT devices with constrained resources.

Class Imbalance Solutions: More advanced techniques, such as cost-sensitive learning and data augmentation, are explored for handling class imbalance more effectively.

Real-World Deployment:

Testing the models in real world IOT environments and evaluating their performance metrics. Increase the transparency of hybrid models to bring about trust and transparency in IoT security applications.

This research concludes by telling hybrid models of ml gives the best solution for IOT threats by high accuracy, robustness and overcomes the limitations of single models. Even it is intensive to implement hybrid models

This research concludes that hybrid models of machine learning assure the best solution for IoT threat detection owing to their high accuracy, robustness, and ability in overcoming single models' limitations. Despite being highly computationally demanding, these hybrid models guarantee dependability, making them indispensable in protecting IoT networks from advanced cyberthreats. Achieving a balance between performance, computational economy, and real-world application is crucial for the development of IoT security.

References

Butt, N., Shahid, A., Qureshi, K.N., Haider, S., Ibrahim, A.O., Binzagr, F. and Arshad, N., 2022. Intelligent Deep Learning for Anomaly-Based Intrusion Detection in IoT Smart Home Networks. Mathematics, 10(23), p.4598.

Bakhsh, S.A., Khan, M.A., Ahmed, F., Alshehri, M.S., Ali, H. and Ahmad, J., 2023. Enhancing IoT network security through deep learning-powered Intrusion Detection System. Internet of Things, 24, p.100936.

Sarker, I.H., Janicke, H., Maglaras, L. and Camtepe, S., 2023. Data-Driven Intelligence can Revolutionize Today's Cybersecurity World: A Position Paper. arXiv preprint arXiv:2308.05126.

www.unb.ca. (2022). IoT Dataset 2022 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. [online]

Breiman, L. (2001) ‘Random Forests’, *Machine Learning*, 45(1), pp. 5–32. Available at: https://doi.org/10.1023/A:1010933404324.

‘Decision Tree Classifier - an overview | ScienceDirect Topics’ (no date). Available at: https://www.sciencedirect.com/topics/computer-science/decision-tree-classifier (Accessed: 11 December 2024).

Fränti, P. and Mariescu-Istodor, R. (2023) ‘Soft precision and recall’, *Pattern Recognition Letters*, 167, pp. 115–121. Available at: https://doi.org/10.1016/j.patrec.2023.02.005.

Hicks, S.A. *et al.* (2022) ‘On evaluation metrics for medical applications of artificial intelligence’, *Scientific Reports*, 12, p. 5979. Available at: https://doi.org/10.1038/s41598-022-09954-8.

Murtagh, F. (1991) ‘Multilayer perceptrons for classification and regression’, *Neurocomputing*, 2(5), pp. 183–197. Available at: https://doi.org/10.1016/0925-2312(91)90023-5.

Olusanya, M.O. *et al.* (2022) ‘Accuracy of Machine Learning Classification Models for the Prediction of Type 2 Diabetes Mellitus: A Systematic Survey and Meta-Analysis Approach’, *International Journal of Environmental Research and Public Health*, 19(21), p. 14280. Available at: https://doi.org/10.3390/ijerph192114280.

Support Vector Machines (SVM): An Intuitive Explanation | by Tasmay Pankaj Tibrewal | Low Code for Data Science | Medium (no date). Available at: https://medium.com/low-code-for-advanced-data-science/support-vector-machines-svm-an-intuitive-explanation-b084d6238106 (Accessed: 11 December 2024).

Uddin, S. *et al.* (2022) ‘Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction’, *Scientific Reports*, 12(1), p. 6256. Available at: https://doi.org/10.1038/s41598-022-10358-x.

Wang, L., Ji, H.-B. and Jin, Y. (2013) ‘Fuzzy Passive–Aggressive classification: A robust and efficient algorithm for online classification problems’, *Information Sciences*, 220, pp. 46–63. Available at: https://doi.org/10.1016/j.ins.2012.06.023.