National College of Ireland

Advanced Threat Detection in IoT Networks Using Hyperparameter-Tuned Machine Learning Models

MSc Research Project
MSc in Data Analytics

Tejasvi Mirle Nataraja
Student ID: 23217120

School of Computing
National College of Ireland

Supervisor: Vladimir Milosavljevic

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Tejasvi Mirle Nataraja
Student ID:	23217120
Programme:	MSc in Data Analytics **Year:** 2024-2025
Module:	Research Project
Supervisor:	Vladimir Milosavljevic
Submission Due Date:	January 29, 2025
Project Title:	**Advanced thread detection in the IoT networks using hyper parameter tuned machine learning models**.
Word Count:	……4000……**Page Count**……18……..

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: Tejasvi Mirle Nataraja

Date: 29-01-2025

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):	

Advanced Threat Detection in IoT Networks Using Hyperparameter-Tuned Machine Learning Models

Tejasvi Mirle Nataraja
23217120

Abstract

The use of IoT (Internet of Things) devices has brought several new forms of security risks which cause growth of malware that targets IoT networks. Current techniques for intrusion detection involve a IoT of interaction and may thus not be very efficient in handling new kind of threats.

The purpose of the research is to propose the IoT-2023 based efficient machine learning models for improving the identification of the illicit actions in the IoT networks. The proposed method combines feature selection and classification techniques which are useful for identifying the features that best characterize the malicious and benign network traffics. To get the best from the machine learning models developed hyperparameter tuning is utilized.

Machine learning classifiers have been developed such as Logistic regression, Decision tree, Decision stump, Random Forest, Naïve Bayes, K Nearest neighbours (KNN), Support Vector Machines (SVM), Multi Layer Perception (MLP), Gradient Boosting (GB) and Extreme Gradient Boosting (XGB). All these models work for have been made to work with binary and multi class classification tasks. In addition hyperparameter tuning is done for all the models. There is a significant improvement in the accuracy of the model.

Boosting algorithms like Gradient Boosting (GB) and Extreme gradient boosting (XGB) are working greatly with binary classification of the attack types with 95% accuracy and 95% F1 score each. Decision tree and random forest algorithm is capable to handle multi class classification in a better way with 84% accuracy each. Although hyperparameter tuning is responsible for model improvement, it has no compatibility with models like Naïve Bayes and Decision Stump models. With this accuracy of the model receded with complex classification tasks like 33% accuracy for decision stump and 67% accuracy for Naïve Bayes. With these studies we were able to reach the research objectives.

1 Introduction

It is the Internet of Things that has emerged as a technology that is transformative in nature which ensures the best communication among devices that are inter connected with a great impact on healthcare, transportation and smarthomes. Pinto Neto et al. (2023) said that this again has a challenging phase where IoT is vulnerable to cyber security threats. Due to wide interconnection in IoT networks and they are the source of very sensitive information and

data they may face various kinds of cyber security threats like Denial of Service, Distributed Denial of Service, Mirai, Spoofing and other kind of attacks.

There are many fields that this technology is impacting drastically. For instance, in health domain, patient can be tracked using IoT technology on a repeated and regular basis. In the area of transportation devices of Internet of Things have been able to detect and prevent occurrences of accident. IIoT (Industrial Internet of Things) solutions have also introduced different solutions such as high reliability and low latency automated monitoring and collaborative control. IoT solutions have also been extended to Education, Farming and Forestry. In the last decade, the society has seen a drastic shift of connections in the Internet of Things. In fact by connecting to IoT there is high potential of rise in the future adjacent areas. Thus, there is encouragement of generation and advancement in business thoughts and other concepts that require massive distributed system.
Efforts are on to make the secure and efficient operation of IoT devices. Extensive research has been done on generating datasets which include attacks against the IoT devices. Various organizations have contributed to the IoT attack dataset to enhance the security analytics of real IoT operations. The mimic of attacks are done using malicious IoT devices. Such legitimate datasets are generated by considering the range of attacks that can disrupt IoT operation. Different kinds of real IoT devices of various kind of brands are used for the purpose. These datasets are including attacks by malicious IoT devices.

The increase in dependency on Information Technology systems combined with the formation of complex attacks make the research worthful since there is a lack of accurate and efficient way of categorizing an attack. The concepts of eight type of attacking methods and binary classification like normal or attach call for methods that can cope with minority class and still are able to hold the accuracy in all cases. Moreover, empirical comparison of different types of machine learning algorithms would help to evaluate their applicability for practicing cybersecurity.

The project is going to be driven by the research question as follows.
How the machine learning models behave in identifying and classifying cyberattacks in IoT (Internet of Things) network among binary classification and multi-classification scenarios?

Key objectives to be met with this project is the Performance evaluation of machine learning models like Logistic regression, Decision tree, Decision stump, Random Forest, Naïve Bayes, K Nearest neighbours (KNN), Support Vector Machines (SVM), Multi Layer Perception (MLP), Gradient Boosting (GB) and Extreme Gradient Boosting (XGB) with respect to binary and multi-classification (Theresa and Ramli; 2023). In addition to that application of resampling techniques for class imbalance handling and optimization of models with the help of hyperparameter tuning. Provide the comparative analysis comprehensively for the evaluation metrics of the models like accuracy, precision, recall and F1 scores.

The contribution one could expect from this project are like multiple machine learning classification models are systematically evaluated for binary and multi-class attack detection in IoT networks. Techniques to address class imbalance like resampling and also the performance evaluation metrics fabricated for datasets that are imbalanced (Kumar, Rastogi and Ranga; 2024). Pros and cons of multiple machine learning classifier models for practical usage in networking security.

For documenting this project, a well-defined structure of the document has been adopted with sections like Literature Survey, Methodology of research, Evaluation, Results and Interpretation and Conclusion and Future Work. In Literature survey previous works related to the cyber-attacks and use of machine learning technologies have been analysed. In methodology of research**,** describing of the dataset, data cleaning process and feature engineering process have been addressed. In evaluation, results and interpretation section, evaluation metrics along with all results, visualizations and discussion on key observations and challenges found as part of execution of the project. In Conclusion and Future work section, the key findings and scope of future work that can be taken over this project has been addressed.
The evaluation metrics considered are accuracy, precision, recall, F1-score, specificity and geometric mean. The metric equations are as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad \ldots(1)$$

$$Precision = \frac{TP}{TP + FP} \quad \ldots(2)$$

$$Recall = \frac{TP}{TP + FN} \quad \ldots(3)$$

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision \times Recall} \quad \ldots(4)$$

$$Specificity = \frac{TN}{TN + FP} \quad \ldots(5)$$

$$Geometric\ Mean = \sqrt{specificity \times specificity} \quad \ldots(6)$$

Where TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative

2 Literature Survey

Internet of Things is becoming more and more important in many fields. It brings new features but also create big security problems. Pinto Neto et al. (2023) underlines the great

necessity to have good security rules because the number of IoT connections are growing especially in the areas like healthcare and transportation. This study introduced a dataset named CICIoT2023 dataset. This dataset not only fill the gaps in current resources but also offers significant information for creating sophisticated security analytics applications. The results emphasize the importance of strong security structures to protect the IoT environments from emerging threats.

In all the previous case studies, they are using a wide range of machine learning algorithms such as logistic regression, decision tree, KNN and Support Vector Machine for developing Intrusion detection system. All drawback which were available in past datasets like missing of real time data and others have been taken care by CICIoT2023 dataset. Hence it became a source of the research.

Kumar et al.(2024) in their has used the CICIoT2023 dataset in their analysis has worked on binary, 8 class and individual attack class classification. They made use of 5 machine learning algorithms like LR, SVM, Decision tree, Adaboost and Random Forest algorithms. Here random forest and decision tree models are best performing for binary classification. SVM model is least performer. But no attempt to use hyperparameter tuning is done here and comparison not evaluated with and without optimization methods (Kumar, Rastogi and Ranga; 2024). In the current project, emphasis is given on hyperparameter tuning for performance evaluation.

Decision tree and random forest models are the best working classifiers when a 8 class classification task is done (Theresa and Ramli; 2023). The project is nothing but the application of machine learning algorithms to identify Distributed Denial of Service (DDoS) attack on IoT networks. Hence the same dataset is vast and is able to give real-time experience for analysis.

3 Methodology of Research

This section best describes the steps taken to perform the project starting from data preparation to model building.

3.1 Case 1: Binary Classification

3.1.1 Data Preparation and Environment Setup:

Having decided to commence the execution of project the dataset to be chosen was quite challenging. For the current project we have used the CIC IoT dataset 2023 from the Canadian Institute of Cybersecurity website. Its free and opensource developed by researchers as part of their research (Neto et al.; 2023). Google Colab has been used to code and execute the code. All basic algorithms like matplotlib for visualization, seaborn for graphical representation of data, numpy and panda for statistical analysis has been imported. Throughout the project various runtime has been utilized based on the need of computation like CPU , T4 GPU which are free runtime versions available for all users of Google Colab. In need of highly computational tasks a paid subscription of A100 GPU, meaning 100 GPU units have been utilized. Python 3 is used as programming language. All the packages for

building the model from scikit-learn has been imported. For clean and efficient coding various function has been defined at the beginning of the project like function to check the missing values, function to create the confusion matrix, function to record the classification report and function to note down the time elapsed to run the notebook.

3.1.2 Exploratory Data Analysis:

Exploratory data analysis of the dataset has been performed to assess the characteristics of the dataset. The shape of the dataset has been found. There are 28,67,733 and 47 columns. There was no trace of missing values in any of the columns and missing types were checked. 46 columns have float type and one of the column has object datatype. The names of the variables are "Std", "ARP", "ack_flag_number", "Max", "Number", "DNS", "Tot sum", "Weight", "SMTP", "Duration", "Protocol Type", "Radius", "syn_flag_number", "psh_flag_number", "Drate", "Srate", "ICMP", "IPv", "ack_count", "IRC", "Variance", "TCP", "IAT", "fin_flag_number", "RST", "HTTP", "HTTPS", "cwr_flag_number", "Covariance", "DHCP", "AVG", "LLC", "Rate", "rst_flag_number", "Magnitude", "Urg", "Telnet", "Tot size", "SSH", "Header_Length", "label", "fin_count", "Min" and "syn_count". Among these “label” column is of object datatype as it contains string values with different type of attacks mentioned here. Descriptive statistics have been done to analyse the numerical values in the dataset. Now “label” column is assessed and found that it has 16 unique values with ‘BenignTraffic’ value is repeating the most with 1098195 times.

3.1.3 Data Preprocessing:

For the ease of processing, we are renamed the ‘label’ column to ‘Attack_type’. Some features like “Protocol Type” are renamed as “Protocol_Type”, “Tot sum” was changed as “Tot_sum” and “Tot size” was replaced as “Tot_size”. The 16 unique values in the “Attack_type” column are "Recon-PortScan", "MITM-ArpSpoofing", "XSS", "BenignTraffic", "Recon-HostDiscovery", "Recon-OSScan", "VulnerabilityScan", "BrowserHijacking", "SqlInjection", "Recon-PingSweep", "Uploading_Attack",

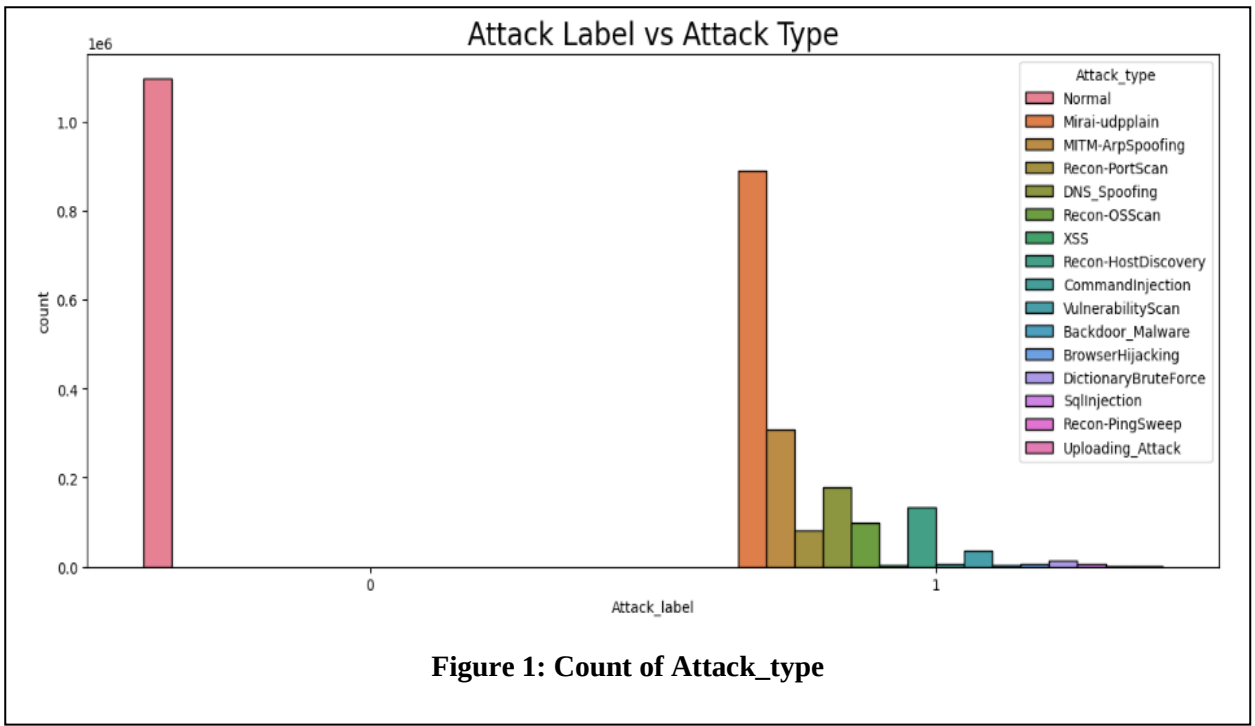

Figure 1: Count of Attack_type

"CommandInjection", "DNS_Spoofing", "Backdoor_Malware", "DictionaryBruteForce" and “Mirai-udpplain". Here "BenignTraffic" is the normal kind of attack and rest are the very malicious attacks. Here for simplicity, we are renaming "BenignTraffic" as “Normal”. Here new column “Attack_label” added to the dataset and now using lambda function of python we are assign value 0 wherever there is “Normal” in “Attack_label” with all other values assigned 1. From Figure 1, we can see that count of malicious attacks are greater when all of them are put together but count of normal attack is less. Mirai-udpplain and MITM ArpSpoofing are most frequent IoT attacks. XSS, Recon-PingSweep an Uploading_Attack are very less frequent. This shows a diverse dataset. From this we come to conclude that a need of class imbalance is necessary to avoid biased result and make sure that there is robust classification. As there is diverse attack categories, the dataset will be a ground for intrusion detection systems and minority classes has to be addressed properly.

From Figure 2, we can see that normal attack share the maximum area and among malicious attacks , “Mirai_udpplain” has a maximum share of 31.1%. Attacks like DNS_Spoofing with 10.7%, MITM_Arpspoofing with 6.24% and Recon-PortScan with 4.56%. Columns “Drate” and “DHCP” are dropped as they have null values. Label encoding of “Attack_label” column is done to convert categorical variables into numerical values. Now the unique values of

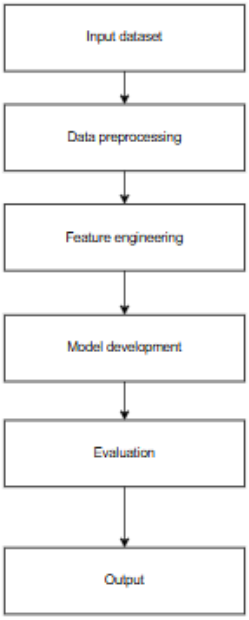

Figure 1.1 Flow Diagram of the Project

“Attack_label” column are "Recon-OSScan", "CommandInjection", "BrowserHijacking", "MITM-ArpSpoofing", "Backdoor_Malware", "Recon-PortScan", "Normal", "DictionaryBruteForce", "Recon-HostDiscovery", "SqlInjection", "Mirai-udpplain", "Uploading_Attack", "DNS_Spoofing", "VulnerabilityScan", "Recon-PingSweep" and "XSS". The dependent and independent variables are identified. “Attack_label” is the target variable. The dataset is split into test set and train set in the proportion of 80% and 20%. The next step is the class balancing by using random sampling technique. The class balance

before and after resampling of the abnormal class is 1415493 and 878693. Percentage of abnormal class is found to be 61.7%.

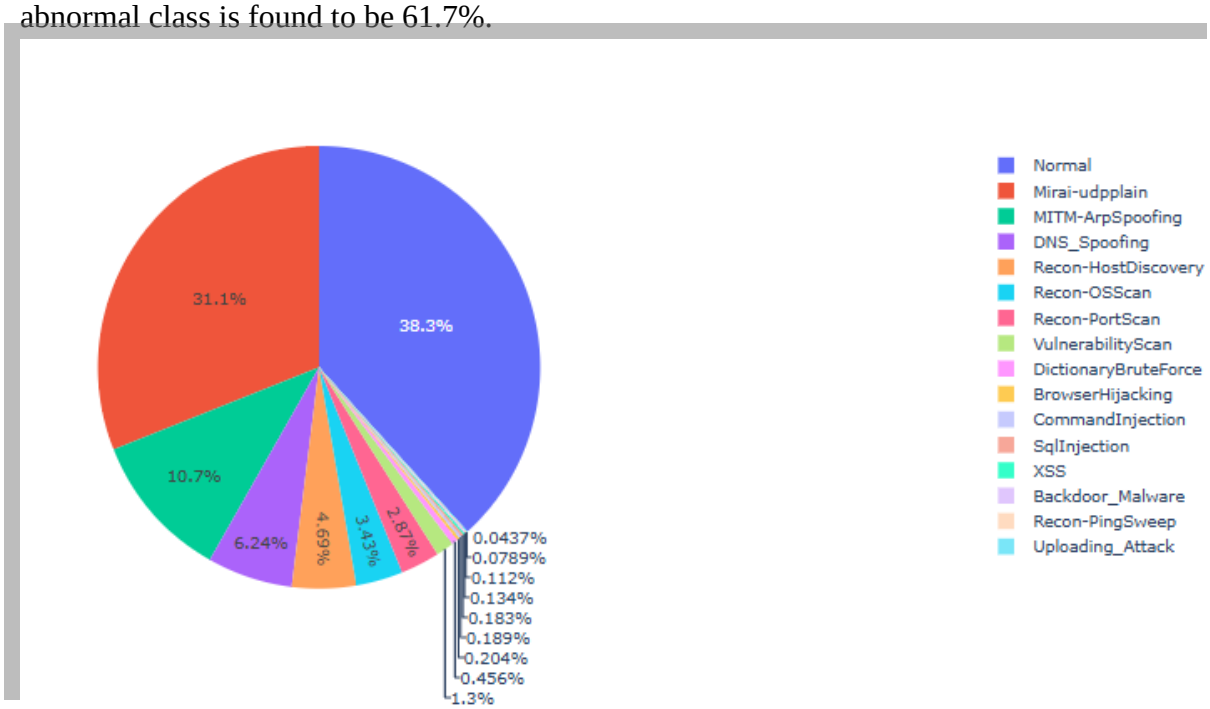

Figure 2 : Proportion if types of IoT attacks

3.1.4 Feature scaling:

Feature scaling is done as we are dealing with algorithms like logistic regression, Support Vector Mechanism and MSP neural network algorithm which perform well with data that are scaled. Without scaling the larger values will tend to dominate leading to inappropriate result. The flag like "is_data_scaled" is used to monitor how the scaling is impacting model accuracy.

3.1.5 Dataset Reduction:

From Figure 3, we see that an attempt to test the test set for accuracy has been done with different fraction of the dataset like 1%, 2%, 5%, 10%, 25%, 50%, 75% and 100%. Y axis represents the count of each category. Though there is no major difference in accuracy except in decimals it is considered safe to assume that even 1% of the dataset yields the same result as higher percentage of dataset considered for the evaluation. Hence for safer and system compatible calculation purpose, this project considered 1% of data from the dataset for the analysis.

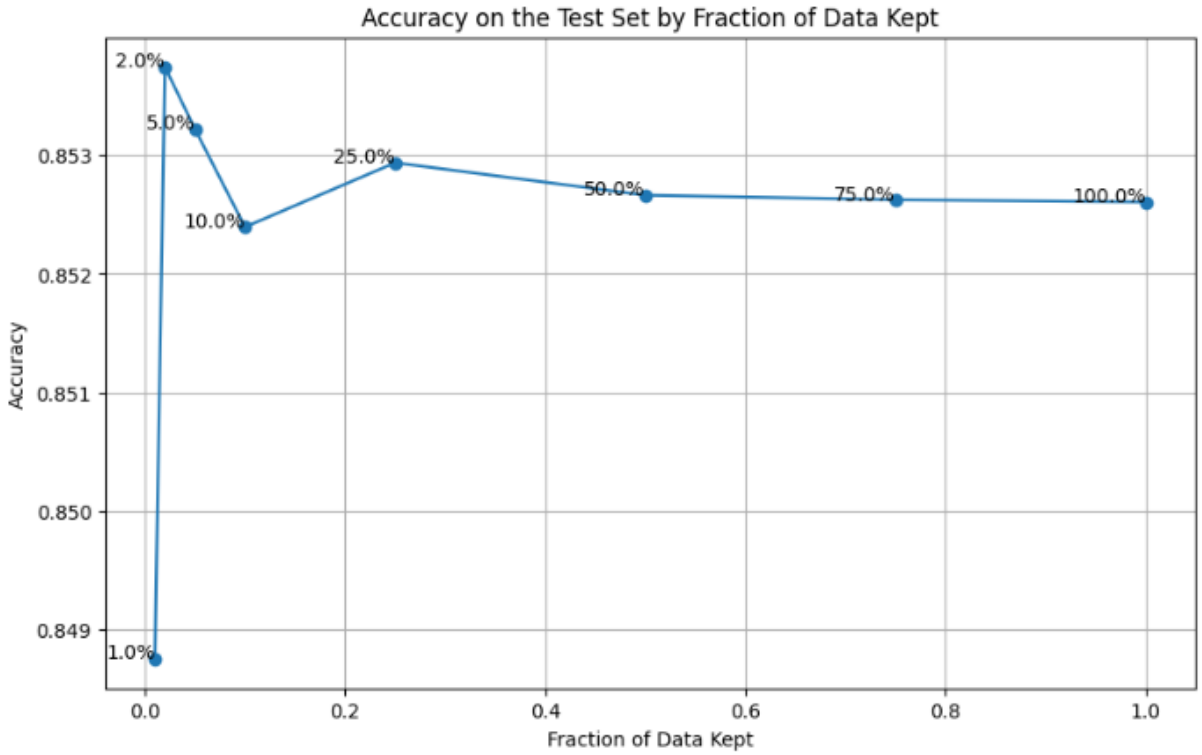

Figure 3: Test set accuracy for different fraction of dataset

3.1.6 Model building:

After all the data preparation, the model building is done for all the machine learning models like Logistic regression, Decision tree, Decision stump, Random Forest, Naïve Bayes, K Nearest neighbours (KNN), Support Vector Machines (SVM), Multi-Layer Perception (MLP), Gradient Boosting (GB) and Extreme Gradient Boosting (XGB). Various motive for there for choosing the classifier models. Logistic regression is suitable for linearly separable data and for easy interpretation. Decision tree and Random Forest algorithm are capable of handling nonlinear relationships. Naive Bayes algorithm is suitable to check with high dimentional dataset. KNN although best suitable for clustering activity, also has favourable reasons to work with classification tasks. Multi-Layer Perception (MLP) being a deep learning model capture complex and non-linear relationships. Support Vector Machine (SVM) is chosen as it works well when the dataset is high-dimentional, complex decision boundaries and better handling of small size of data samples. Boosting algorithms like Gradient Boosting and Extreme Gradient Boosting are best in capturing complex relationships. After this hyperparametr tuning is worked upon.

3.2 Case 2: Multi Classification

In this case all the preprocess steps remain same same except for label encoding process. Here the unique values of attack types in the target variable are mapped into 8 different

categories of attacks. The eight categories considered are Normal, DoS, Reconnaissance, spoofing, injection, malware, brute force and other. Later label encoding is done.

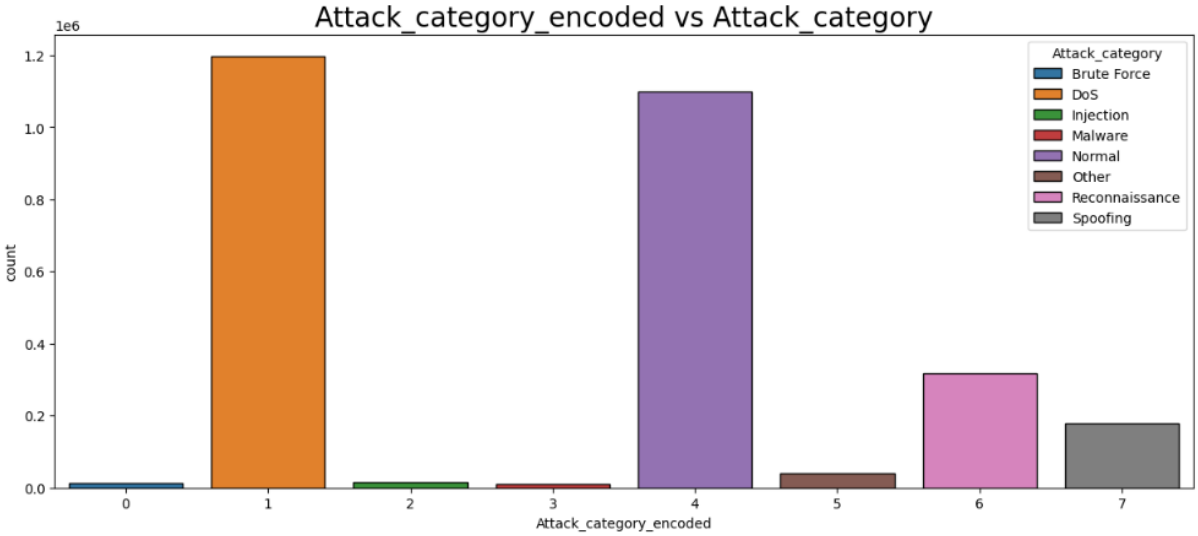

Figure 4: Frequency of attack category in muti classification

From Figure 4, the x-axis is an indication of attack categories which are encoded with numbers 0 to 7 and each number represents specific type of attack. It is clear that count of Denial-of-Service attack which is category 1, dominates as it is more frequent. As per the dataset, Denial of Service is the most common attack one can observe. Normal traffic which is category 4, is also one of the prominent categories which is one of the typical observations from a intrusion detection system. Category 6 and 7 which are Reconnaissance and Spoofing has considerable amount of counts. Following this kind of imbalance distribution methods like under sampling is the need for processing. Higher count of normal and DoS attack is also an indication of the real-world scenario. This dataset helps to train the models which can handle high and low frequency attack categories in order to achieve balanced performance.

Figure 5 is a pie chart which tell that normal attack has the highest share of 41.8% and among malicious attacks Mirai attack has the maximum share of 38.3%. This is the real mimic of the real cases as Mirai attacks are very common in such IoT devices. The mirai attack is a threat to IP cameras and routers and also other connecting devices. Least recognized threats are categorized under other category.

From figure 6 it is noted that there is a significant difference in accuracy in the range of 64 to 69%. This made us to chose 25% of the dataset for the analysis over 1% which yield least result. After building the model, hyperparameter tuning is done.

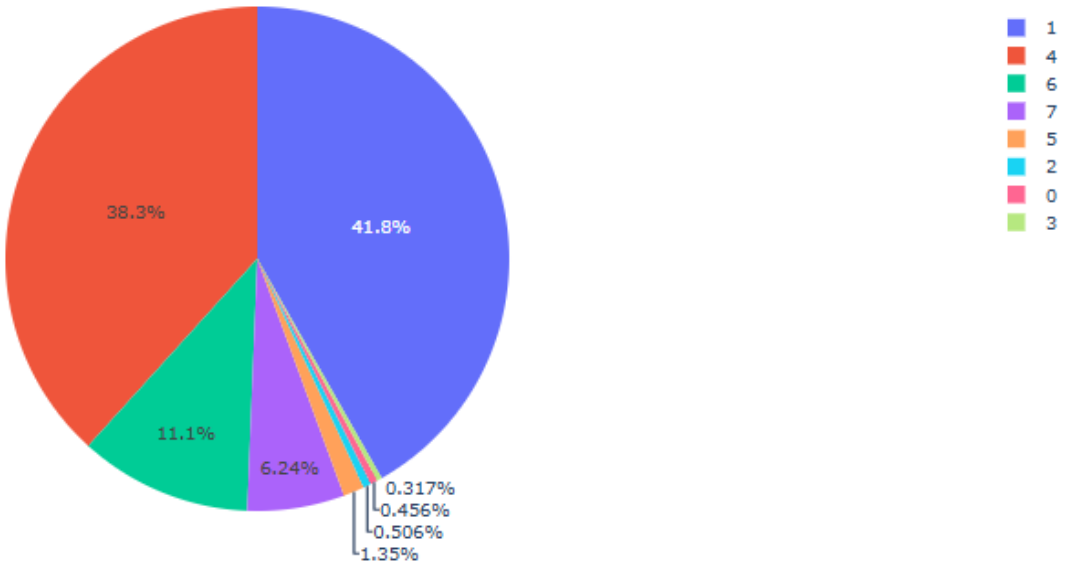

Figure 5: Pie chart of multi-classification of attack types

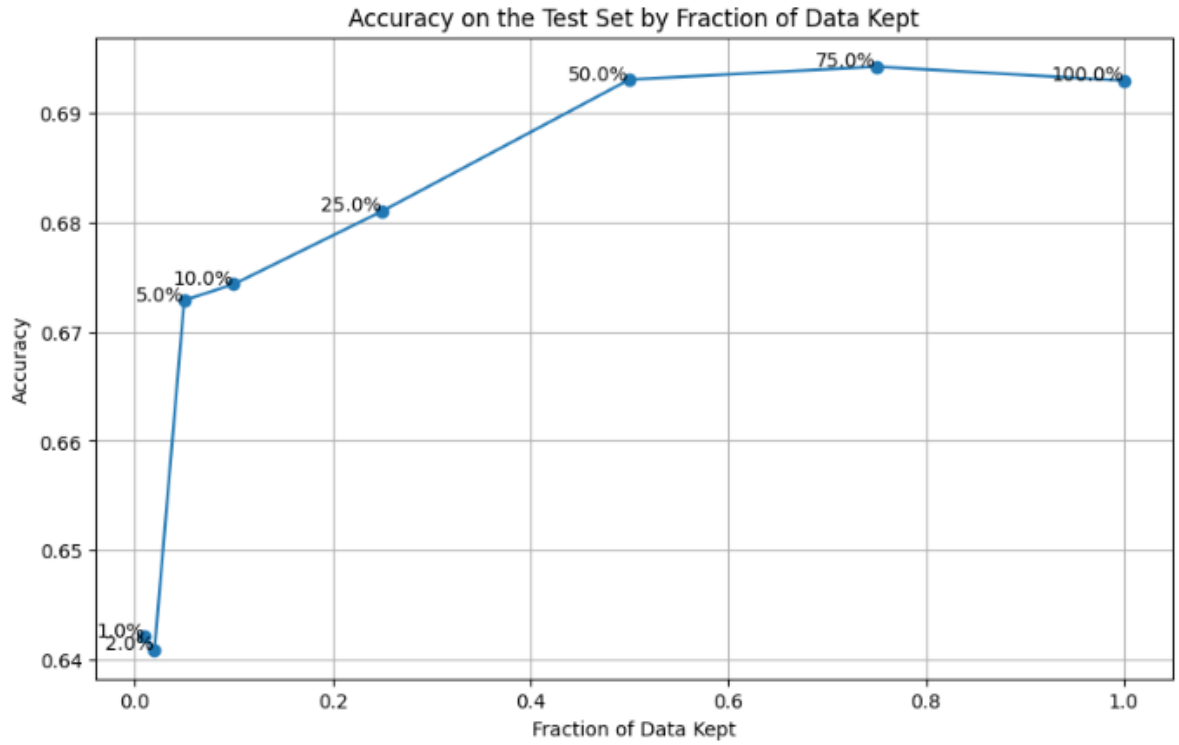

Figure 6: Accuracy of test set for different fraction of dataset in multiclass classification.

4 Evaluation, Results and Interpretation

This section discusses about the results with all evaluation metrics thoroughly discussed.

4.1 Case 1 – Binary classification

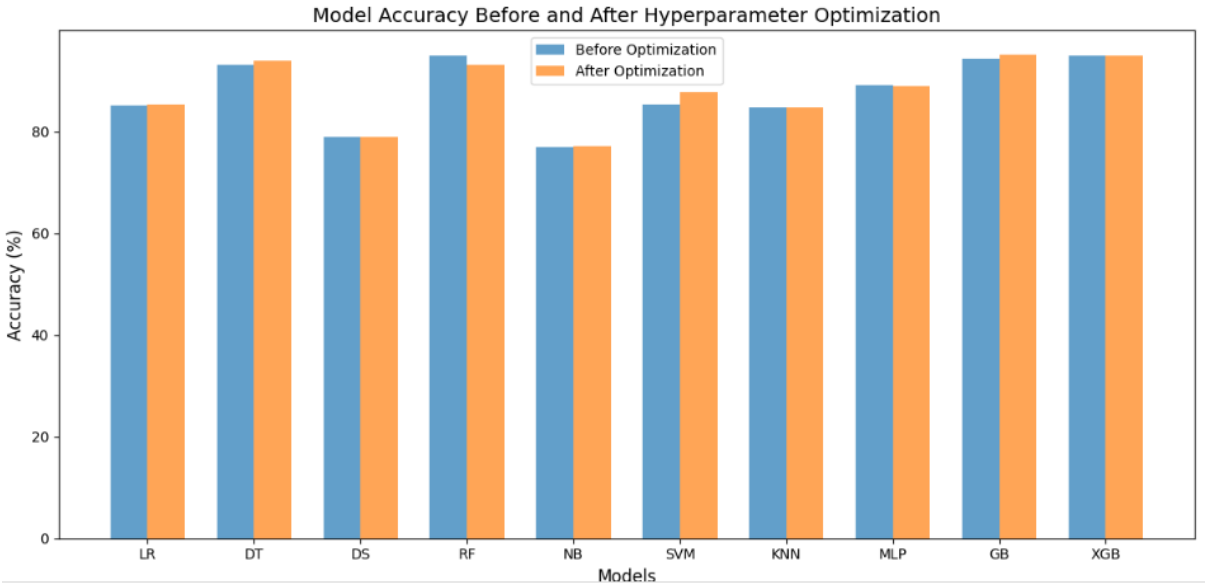

Figure 7: Accuracy comparison for binary classification

From figure 7, we observe that after hyperparameter tuning, Gradient Boosting (GB), Extreme Gradient Boosting (XGB) and Decision Tree (DT) with accuracy yielding 95.08%, 94.90% and 94.00% respectively. These models have higher prediction capacity. Among all the machine learning models, Support Vector Mechanism (SVM) exhibit significant improvement as the performance increase from 85.38% to 87.78% which accounts to 2.4% increase which is the highest. Here we can see that hyperparameter tuning is very effective in improving the performance of the model. But models like Random Forest (RF) and Multilayer perception (MLP) show reduction in performance with hyper parameter tuning. Decision Stump and Extreme Gradient Boosting models are consistent in its performance even after the hyperparameter tuning. The challenge faced during the execution is that Support Vector Mechanism (SVM) took higher time for executing the result as it is highly computation intensive. Since the dataset has more number of features, SVM model calculates the distance and transformation in a high dimentional space which is the key reason for higher execution time.

Table 1: Results for Binary Classification Before Hyper Parameter Tuning

Model	TN	FP	FN	TP	Accuracy	Sensitivity	Specificity	Geometric Mean	Precision	Recall	F1-Score
Logistic Regression	205810	13692	71591	282454	0.85	0.80	0.94	0.86	0.87	0.85	0.85

Decision Tree	204396	15106	24074	329971	0.93	0.93	0.93	0.93	0.93	0.93	0.93
Decision Stump	204075	15427	105922	248123	0.79	0.70	0.93	0.81	0.83	0.79	0.79
Random Forest	212651	6851	22780	331265	0.95	0.94	0.97	0.95	0.95	0.95	0.95
Naive Bayes	187790	31712	100370	253675	0.77	0.72	0.86	0.78	0.80	0.77	0.77
SVM	214090	5412	78435	275610	0.85	0.78	0.98	0.87	0.89	0.85	0.86
KNN	202434	17068	71137	282908	0.85	0.80	0.92	0.86	0.87	0.85	0.85
MLP	207503	11999	50833	303212	0.89	0.86	0.95	0.90	0.90	0.89	0.89
Gradient Boosting	211530	7972	25115	328930	0.94	0.93	0.96	0.95	0.94	0.94	0.94
XGBoost	212497	7005	26728	327317	0.94	0.92	0.97	0.95	0.94	0.94	0.94

From Table 1 and 2, we can understand that the ten machine learning classifier models considered was evaluated on binary classification work. Evaluation metrics like accuracy, sensitivity, geometric mean, precision, recall and F1-score were estimated before and after hyperparameter tuning to analyze prediction abilities of the models developed. Before hyperparameter tuning, Gradient Boosting and XGB models showed optimal performance with 94.23% and 94.12% accuracy respectively. Both the models have sensitivity of 93% to 95% and also specificity of 96% to 97% which indicates the best classification of the model. Random forest with accuracy of 94.83% is showing the best results in terms of all the metrics considered. Moderate performers are the LR, SVM and KNN models with balance in specificity and sensitivity. MLP, a deep learning model with 89.05% accuracy has the capacity to handle non linear patterns. Naïve Bayes and Decision Stump has low accuracy of 78.84% and 76.97%.

From Table 2, which shows the results of binary classification with hyper parameter tuning. Again the best models after optimization are GB and XGB models. The accuracy of decision tree model increased by 1%. SVM model showed the maximum improvement by 2%. LR and KNN models shows slight increase in accuracy post hyper parameter tuning. Naïve Bayes and Decision stump models shows stable results. Random Foresta and MLP model showed reduced accuracy post optimization. This could be due to overfitting.

Table 2: Results for Binary Classification After Hyper Parameter Tuning

Model	TN	FP	FN	TP	Accuracy	Sensitivity	Specificity	Geometric Mean	Precision	Recall	F1-Score
Logistic Regression	205658	13844	70101	283944	0.85	0.80	0.94	0.87	0.87	0.85	0.86
Decision Tree	211741	7761	26627	327418	0.94	0.92	0.96	0.94	0.94	0.94	0.94
Decision Stump	204075	15427	105922	248123	0.79	0.70	0.93	0.81	0.83	0.79	0.79
Random Forest	213263	6239	33416	320629	0.93	0.91	0.97	0.94	0.94	0.93	0.93
Naive Bayes	187788	31714	100087	253958	0.77	0.72	0.86	0.78	0.80	0.77	0.77
SVM	211212	8290	61815	292230	0.88	0.83	0.96	0.89	0.90	0.88	0.88
KNN	206574	12928	75144	278901	0.85	0.79	0.94	0.86	0.87	0.85	0.85
MLP	207414	12088	51578	302467	0.89	0.85	0.94	0.90	0.90	0.89	0.89
Gradient Boosting	212906	6596	21644	332401	0.95	0.94	0.97	0.95	0.95	0.95	0.95
XGBoost	213024	6478	22750	331295	0.95	0.94	0.97	0.95	0.95	0.95	0.95

4.2 Case 2 – Multi classification

Table 3 and 4 has information about the results of multi classification before and after hyperparameter tuning. So, before hyper parameter tuning, top performing decision tree model has the best accuracy of 84.15% with precision of 90.67% which is high, recall of 84.15%, F1-score of 86.73%. The geometric mean is relatively high with value 87.35%. This indicates the robust performance. Random Forest model with accuracy of 84.08% with high precision of 92.70% and F1-score of 87.64%. Moderately performing MLP model has accuracy of 72.69% with F1-score of 78.74%. It was able to capture non linear relationships. SVM and KNN models with 65.97% and 64.06% accuracy indicate that the model has the ability to take care of multiclass classification. Low performing models here we see is that Naïve Bayes and Decision Stump.

After hyper parameter tuning, there is spike in accuracy of decision tree model again with 85.26% accuracy. F1-score increased to 87.98%. Random Forest shows the dip in accuracy and F1-score. MLP models accuracy remained stable with 72.86%. There is drastic

improvement in accuracy of SVM model by 5%. LR model also showed improvement from 67.99% to 68.65%. Again decision stump and naïve bayes models remained stable with minimal changes. KNN model also improved by 2% post tuning work.

Table 3: Results for Multi Classification Before Hyper Parameter Tuning

Model	Accuracy	Geometric Mean	Precision	Recall	F1-Score
Logistic Regression	0.68	0.86	0.68	0.75	0.76
Decision Tree	0.84	0.91	0.84	0.87	0.87
Decision Stump	0.33	0.40	0.33	0.35	0.36
Random Forest	0.84	0.93	0.84	0.88	0.88
Naive Bayes	0.61	0.67	0.61	0.59	0.64
SVM	0.66	0.86	0.66	0.74	0.75
KNN	0.64	0.84	0.64	0.72	0.73
MLP	0.73	0.88	0.73	0.79	0.80

Table 4: Results for Multi Classification After Hyper Parameter Tuning

Model	Accuracy	Geometric Mean	Precision	Recall	F1-Score
Logistic Regression	0.69	0.85	0.69	0.75	0.76
Decision Tree	0.85	0.92	0.85	0.88	0.89
Decision Stump	0.33	0.40	0.33	0.35	0.36
Random Forest	0.78	0.92	0.78	0.84	0.85
Naive Bayes	0.67	0.73	0.67	0.67	0.70
SVM	0.70	0.87	0.70	0.77	0.78
KNN	0.66	0.84	0.66	0.73	0.75

MLP	0.73	0.87	0.73	0.79	0.80

Table 5 clearly specify that as the complexity of the task of classification has an inversely proportional effect on the accuracy. All the models accuracy decreased drastically from binary classification tasks to multiclass classification tasks. Decision tree model is again the best model to be considered when the classification task is compared. Models like decision stump prove to be inefficient with varying complexity.

Table 5: Comparison of Accuracy for Binary and Multiclass

Model	Accuracy (Binary)	Accuracy (Multiclass)
Logistic Regression	0.8536	0.6865
Decision Tree	0.94	0.8526
Decision Stump	0.7884	0.3329
Random Forest	0.9309	0.7828
Naive Bayes	0.7702	0.6708
SVM	0.8778	0.7025
KNN	0.8464	0.661
MLP	0.889	0.7286

5 Conclusion and Future Work

This project assessed how various machine learning algorithms perform with binary and multiclass classification tasks, with or without hyperparameter tuning. The observations show that Gradient Boosting (GB) and XGBoost algorithms which are tree-based ensemble methods have excelled well in binary classification with high measurements like Accuracy, Sensitivity, Specificity and F1-score. These models showed that they are capable to handle class imbalance and complex data patterns which simulates real time use cases.

Among multiclass classification models, Decision tree and Random Forest models are best performing. They are having high F1-score and accuracy. The project has illustrated that hyperparameter tuning enhanced abilities of SVM and Logistic regression (LR) stressing the importance of optimization of parameters. Decision stump and naïve bayes models remained stable in all cases which is an indication had they have limited capabilities with respect complex data.

As a result, the observations tell us how tree-based models and hyperparameter tuning can optimize the performance of the machine learning classifiers. Although tree-based models have favour for binary and multiclass classification tasks, SVM and MLP models showed the improvement in accuracy by hyper parameter tuning, making them the possible choice. Also the increase in complexity of classification task has a inverse effect on accuracy of the model.

By keeping this project as benchmark, one can improve the result in future using improved version of class imbalance like SMOTE. Advanced feature extraction process also has the

chance to improve the efficiency. Automating hyperparameter tasks with Bayesian optimization is one thing one can work upon.

Discussions of the Project:

When complexity is taken into consideration, decision tree is a single level decision tree. But decision tree is a multi-level which make it possible to capture complex pattern in the dataset. Also decision tree is more capable to capture non linear relation ship between the features which is a common aspect in the Internet of Things dataset. Decision fits better due to complex level. Hence when multi class model is built the accuracy fell to 33% in decision stump. But in binary classification, there is only tangible reduction of around 18% in reduction. Most of the literatures have not included hyperparameter tuning with no proper optimization techniques. But by this implementation we are able to achieve significant improvement in the value of F1 score and accuracy majorly in boosting algorithms used in the project. Use of both binary and multi classification is adding to capture complex pattern. Usage of diverse machine learning models from basic classification models to deep learning model along with boosting. This results in the comparative analysis among the models. Testing the various proportion of dataset before model building to focus on right model building. In case of binary classification, the dataset seem be highly imbalanced and biased towards major class as it is only 2%. Another possibility is that 2% of dataset chosen is subjected to few variations and noise in the dataset. Another reason may be over fitting. But there is decrease in accuracy with 100% of the dataset because of increase in variability and complexity. Here various noise level the dataset is subjected to. The model fail to capture the diverse characteristics of the dataset. Also minor classes may not be well represented with 100% data included. In case of multi classification the scenario is reverse. With clear distinguished data, with increase in size of dataset, learning ability of the model increases and model is able to distinguish among multiple categories. The model learns infrequent attack types in larger dataset. As per the definition, **True Positive** will occur when the model correctly predict a positive instance as positive. For real example a patient has cancer which is the actual condition and the model prediction is also cancer. This case is true positive and there is correct prediction by the model. **False Negative** occurs when the model predicts a negative instance, but actual instance is positive. Let us again consider a medical use case where the patient has cancer which is the actual condition and the model predict that there is no cancer to the patient. Although F1 and accuracy are the evaluation metrics, which is used to assess the performance of the model. We can say that F1 score is given more consideration when dataset is imbalanced. Accuracy is given more importance when dataset is balanced.
If the dataset has higher percentage of major class, then accuracy is more as per major class. But a model with higher F1 score, give focus on predicting of minor classes. Also F1 score focus mainly on positive class. Accuracy is sensitive to correct predictions. So depending on the context, correct evaluation metrics are chosen.

References

Pinto Neto, E.C., Dadkhah, S., Ghorbani, A.A., Ferreira, R., Zohourian, A. and Lu, R. (2023) 'CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment', *Sensors*: 1-26.

Kumar, A.G., Rastogi, A. and Ranga, V. (2024). 'Evaluation of Different Machine Learning Classifiers on New IoT Dataset CICIoT2023'. *Proceedings of the International Conference on Intelligent Systems for Cybersecurity (ISCS)*, 1–6.

Thereza, N. and Ramli, K., (2023). 'Development of Intrusion Detection Models for IoT Networks Utilizing CICIoT2023 Dataset'. *Proceedings of the 3rd International Conference on Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS)*, 66–72.

Prasad, S., Sharma, I. and Rajendraprasad, D, (2024). 'Federated Learning Models for Intrusion Detection in Industrial IoT Networks'. *7th International Conference on Circuit Power and Computing Technologies (ICCPCT)*, 1260–1265.

Gyawali, S, Huang, J. and Jiang, Y, (2024). 'Leveraging Explainable AI for Actionable Insights in IoT Intrusion Detection'. *19th Annual System of Systems Engineering Conference (SoSE)*, 92–97

Huynh, N. S., De La Cruz, S. and Perez-Pons, A., (2023). 'Denial-of-Service (DoS) Attack Detection Using Edge Machine Learning'. *2023 International Conference on Machine Learning and Applications (ICMLA)*, 1741-1745.

ElSayed, Z., Elsayed, N. and Bay, S., (2024). 'A Novel Zero-Trust Machine Learning Green Architecture for Healthcare IoT Cybersecurity: Review, Analysis, and Implementation'. *SoutheastCon 2024*, 686-692.