Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics

Phanindra Kakumanu
School of Computing
University of Georgia
Athens, GA
pk37744@uga.edu

Dr. Yiheng Liang
School of Computing
University of Georgia
Athens, GA
yiheng@uga.edu

Dr. Jin Lu
School of Computing
University of Georgia
Athens, GA
jin.lu@uga.edu

Dr. Gong Chen
School of Computing
University of Georgia
Athens, GA

***Abstract*—This paper aims to replicate and extend the work of A. Sivanathan et al. in classifying IoT devices based on network traffic characteristics. By leveraging the publicly accessible dataset, we will rigorously replicate the original experiment, employing identical machine learning models and evaluation metrics. To enhance the model's generalization capabilities, we will expand the dataset to include a diverse range of IoT devices, encompassing both consumer and industrial applications. Furthermore, we will explore the effectiveness of alternative machine learning algorithms, including deep learning techniques, to potentially improve classification accuracy and computational efficiency. To streamline the data preparation process and minimize costs, we will investigate innovative preprocessing techniques, such as feature selection and dimensionality reduction. By optimizing the data preprocessing pipeline, we aim to reduce the computational burden associated with training and deploying the classification models. Ultimately, our research endeavors to contribute to network management by providing a robust and efficient method for identifying and classifying IoT devices based on their network behavior.**

***Keywords*—** IoT, network characteristics, classification, machine learning

I. Introduction

The Internet of Things (IoT) has revolutionized various aspects of our lives, from smart homes to industrial automation. The proliferation of IoT devices has brought about numerous benefits, including increased convenience, efficiency, and productivity. However, this rapid growth has also introduced new security challenges, such as unauthorized access, data breaches, and malicious attacks.

One critical aspect of IoT security is device identification and classification. Accurate identification of IoT devices can enable effective security measures, including intrusion detection, anomaly detection, and access control. Traditional network-based identification techniques often rely on device signatures or IP addresses, which can be easily spoofed or manipulated by attackers. To address these limitations, researchers have explored the use of network traffic analysis to identify and classify IoT devices based on their unique communication patterns.

A seminal work in this area is the research conducted by A. Sivanathan et al., who proposed a novel approach to classify IoT devices using network traffic characteristics. By analyzing key features such as packet size, inter-arrival time, and protocol usage, they were able to accurately distinguish between different types of IoT devices. This groundbreaking research has laid the foundation for further advancements in IoT device classification [1].

Building upon the work of Sivanathan et al., this paper aims to replicate, extend, and enhance their findings. By leveraging the publicly available dataset provided by the authors, we will rigorously replicate the original experiment, employing the identical machine learning models and evaluation metrics. This replication will serve as a benchmark to validate our methodology and ensure the reliability of our results.

To further advance the state-of-the-art, we will expand the dataset by incorporating a diverse range of IoT devices, encompassing both consumer and industrial applications. This expanded dataset will enable us to assess the model's generalization capabilities and its ability to classify a wider variety of device types. Additionally, we will explore the effectiveness of alternative machine learning algorithms, including deep learning techniques, to potentially improve classification accuracy and computational efficiency.

To streamline the data preparation process and minimize computational costs, we will investigate innovative preprocessing techniques, such as feature selection and dimensionality reduction. By optimizing the data preprocessing pipeline, we aim to reduce the computational burden associated with training and deploying the classification models.

Ultimately, our research endeavors to contribute to the advancement of IoT security and network management by providing a robust and efficient method for identifying and classifying IoT devices based on their network behavior. By addressing the limitations of existing techniques and exploring novel approaches, we aim to enhance the security posture of IoT systems and protect sensitive data from potential threats.

II. Related work

One critical aspect of IoT security is device identification and classification. Accurate identification of IoT devices can enable effective security measures, including intrusion detection, anomaly detection, and access control. Traditional network-based identification techniques often rely on device signatures or IP addresses, which can be easily spoofed or manipulated by attackers. Moreover, methods that rely on MAC addresses or DHCP negotiation are often unreliable due to factors such as third-party NICs, MAC address

IoT device	MAC address	OUI	DHCP host name
Amazon Echo	44:65:0d:56:cc:d3	Amazon Technologies Inc.	
August Doorbell Cam	e0:76:d0:3f:00:ae	AMPAK Technology, Inc.	
Awair air quality monitor	70:88:6b:10:0f:c6		Awair-4594
Belkin Camera	b4:75:0e:ec:e5:a9	Belkin International Inc.	NetCamHD
Belkin Motion Sensor	ec:1a:59:83:28:11	Belkin International Inc.	
Belkin Switch	ec:1a:59:79:f4:89	Belkin International Inc.	
Blipcare BP Meter	74:6a:89:00:2e:25	Rezolt Corporation	
Canary Camera	7c:70:bc:5d:5e:dc	IEEE Registration Authority	Ambarella/C100F1615229
Dropcam	30:8c:fb:2f:e4:b2	Dropcam	
Google Chromecast	6c:ad:f8:5e:e4:61	AzureWave Technology Inc.	Chromecast
Hello Barbie	28:c2:dd:ff:a5:2d	AzureWave Technology Inc.	Barbie-A52D
HP Printer	70:5a:0f:e4:9b:c0	Hewlett Packard	HPE49BC0
iHome PowerPlug	74:c6:3b:29:d7:1d	AzureWave Technology Inc.	hap-29D71D
LiFX Bulb	d0:73:d5:01:83:08	LIFI LABS MANAGEMENT PTY LTD	LIFX Bulb
NEST Smoke Sensor	18:b4:30:25:be:e4	Nest Labs Inc.	
Netatmo Camera	70:ee:50:18:34:43	Netatmo	netatmo-welcome-183443
Netatmo Weather station	70:ee:50:03:b8:ac	Netatmo	
Phillip Hue Lightbulb	00:17:88:2b:9a:25	Philips Lighting BV	Philips-hue
Pixstart photo frame	e0:76:d0:33:bb:85	AMPAK Technology, Inc.	
Ring Door Bell	88:4a:ea:31:66:9d	Texas Instruments	
Samsung Smart Cam	00:16:6c:ab:6b:88	Samsung Electronics Co.,Ltd	
Smart Things	d0:52:a8:00:67:5e	Physical Graph Corporation	SmartThings
TP-Link Camera	f4:f2:6d:93:51:f1	TP-LINK TECHNOLOGIES CO.,LTD.	Little Cam
TP-Link Plug	50:c7:bf:00:56:39	TP-LINK TECHNOLOGIES CO.,LTD.	HS110(US)
Triby Speaker	18:b7:9e:02:20:44	Invoxia	
Withings Baby Monitor	00:24:e4:10:ee:4c	Withings	WBP-EE4C
Withings Scale	00:24:e4:1b:6f:96	Withings	
Withings sleep sensor	00:24:e4:20:28:c6	Withings	WSD-28C6

Table.1 MAC Address and DHCP Host Name of IoT Devices Used in Testbed referring from [1].

spoofing, and the lack of meaningful host names in DHCP requests. In fact, our analysis revealed that approximately half of the studied IoT devices did not reveal their host names. Even when host names are exposed, they may not always be meaningful as shown in Table 1[1] (e.g., WBP-EE4C for Withings baby monitor in Table 1) or can be arbitrarily changed by users. Consequently, relying solely on DHCP infrastructure is not a feasible solution for accurate and scalable device identification [3].

To address these limitations, researchers have explored the use of network traffic analysis to identify and classify IoT devices based on their unique communication patterns. A seminal work in this area is the research conducted by A. Sivanathan et al. [1]. By analyzing key statistical attributes such as activity cycles, port numbers, signaling patterns and cipher suites, and use them to give insights into the under lying network traffic characteristics.

Building upon the work of Sivanathan et al., this paper aims to replicate, extend, and enhance their findings. By leveraging the publicly available dataset provided by the authors, we will rigorously replicate the original experiment, employing the identical machine learning models and evaluation metrics. To further advance the state-of-the-art, we will expand the dataset to include a diverse range of IoT devices, encompassing both consumer and industrial applications. Additionally, we will explore the effectiveness of alternative machine learning algorithms, to potentially improve classification accuracy and computational efficiency.

To streamline the data preparation process and minimize computational costs, we will investigate innovative preprocessing techniques, such as feature selection and dimensionality reduction. Unlike previous work by A. Sivanathan et al., which relied on the Joy tool for hourly data extraction and cipher suite identification, our proposed framework offers a more efficient approach. By avoiding intercommunication requests and directly extracting relevant features from PCAP files, we significantly reduce computational overhead. Additionally, our framework handles missing values by assigning a default value of "none," eliminating the need for complex imputation techniques. This streamlined preprocessing pipeline enables faster data preparation and reduces the overall computational burden associated with training and deploying classification models.

To address the limitations of traditional network-based identification techniques, researchers have explored the use of machine learning to classify IoT devices based on their

network traffic characteristics. A seminal work in this area is

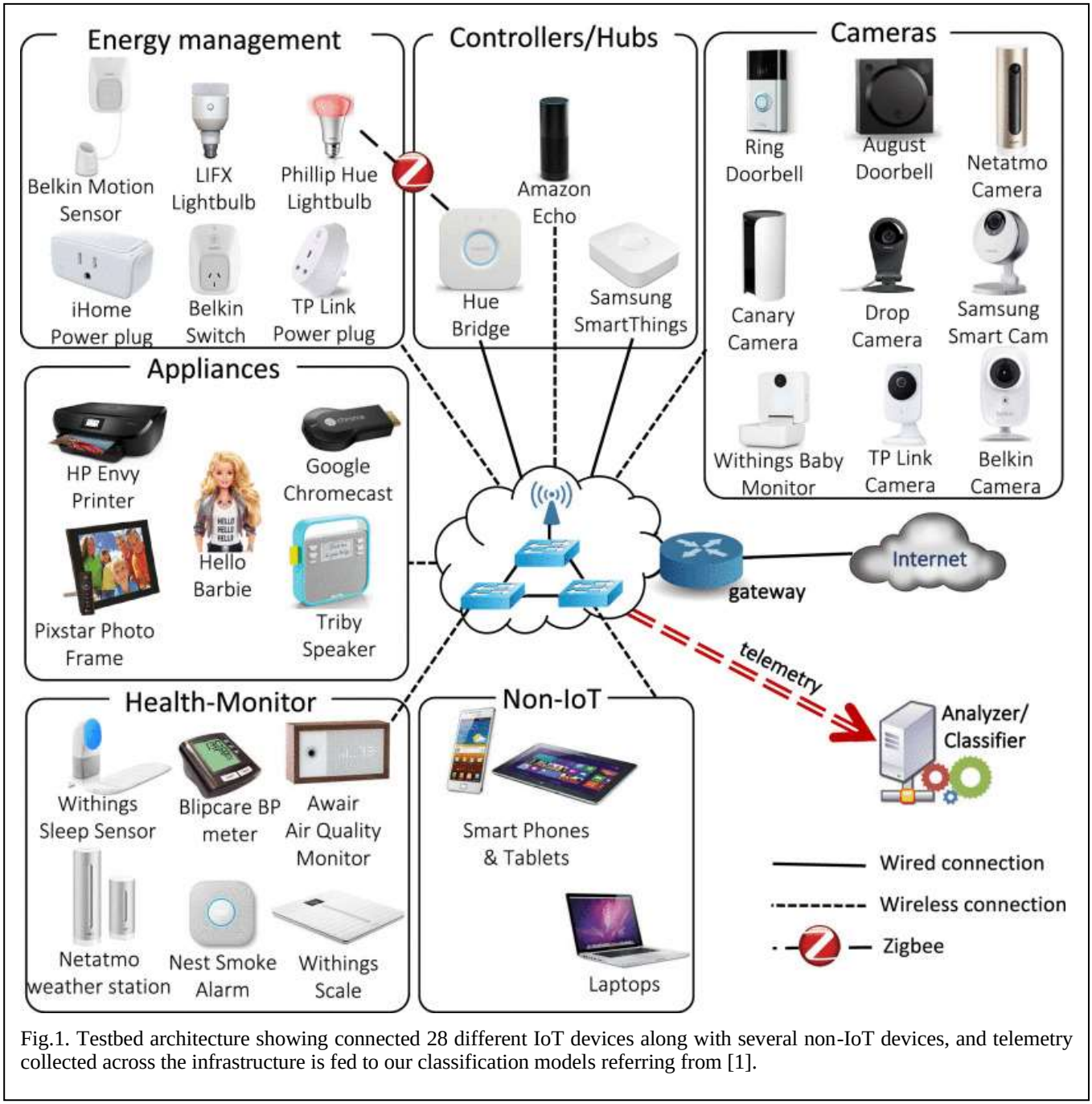


Fig.1. Testbed architecture showing connected 28 different IoT devices along with several non-IoT devices, and telemetry collected across the infrastructure is fed to our classification models referring from [1].

the research conducted by A. Sivanathan et al. [1]. They proposed a novel approach that leverages machine learning algorithms to analyze key features extracted from network traffic, such as packet size, inter-arrival time, protocol usage, and cipher suites. By training machine learning models on a diverse dataset of IoT devices, they were able to accurately classify devices into different categories.

III. Data and Methodology

The authors of [1] established a testbed comprising a diverse range of IoT devices, including cameras, sensors, and smart appliances. These devices were deployed in a controlled environment to facilitate the collection of comprehensive network traffic data. The network traffic generated by these devices was captured using a automation script which uses 'tcpdump' package. The captured traffic was then stored in PCAP format for subsequent analysis in USB drive through automation script and storage begins at midnight local time each day using the Cronjob on OpenWrt.

We will utilize the publicly available 2 weeks of dataset provided in [1]. This dataset contains network traffic traces captured from 28 different IoT devices, includes cameras (Nest Dropcam, Sam sung SmartCam, Netatmo Welcome, Belkin camera, TP-Link Day Night Cloud camera, Withings Smart Baby Monitor, Canary camera, August door bell, Ring doorbell), switches and triggers (iHome, TP-Link Smart Plug, Belkin Wemo Motion Sensor, Belkin Wemo Switch), hubs (Smart Things, Amazon Echo), air quality sensors (NEST Protect smoke alarm, Netatmo Weather station, Awair air quality monitor), electronics (Triby speaker, PIXSTAR Photoframe, HPPrinter, Hello barbie, Google

Chromecast), healthcare devices (With ings Smart scale, Withings Aura smart sleep sensor, Blipcare blood pressure meter) and light bulbs (Phips Hue and LiFX Smart Bulb). The traffic traces are captured in PCAP format, which allows for detailed analysis of network packets. Figure 1 depicts the environment setup of these devices detailed elaboration is provided in [1].

In [1] author employed a multi-step preprocessing approach involving the Joy tool to extract relevant features from PCAP files [4]. While Joy tool offers a flexible framework for network data analysis, it can be computationally intensive, especially when dealing with large datasets. The Joy tool processes the data on an hourly basis, requiring multiple executions to calculate metrics such as average flow rate, flow volume, and flow duration. Additionally, the extraction of cipher suites necessitated a separate processing step, further increasing the computational overhead.

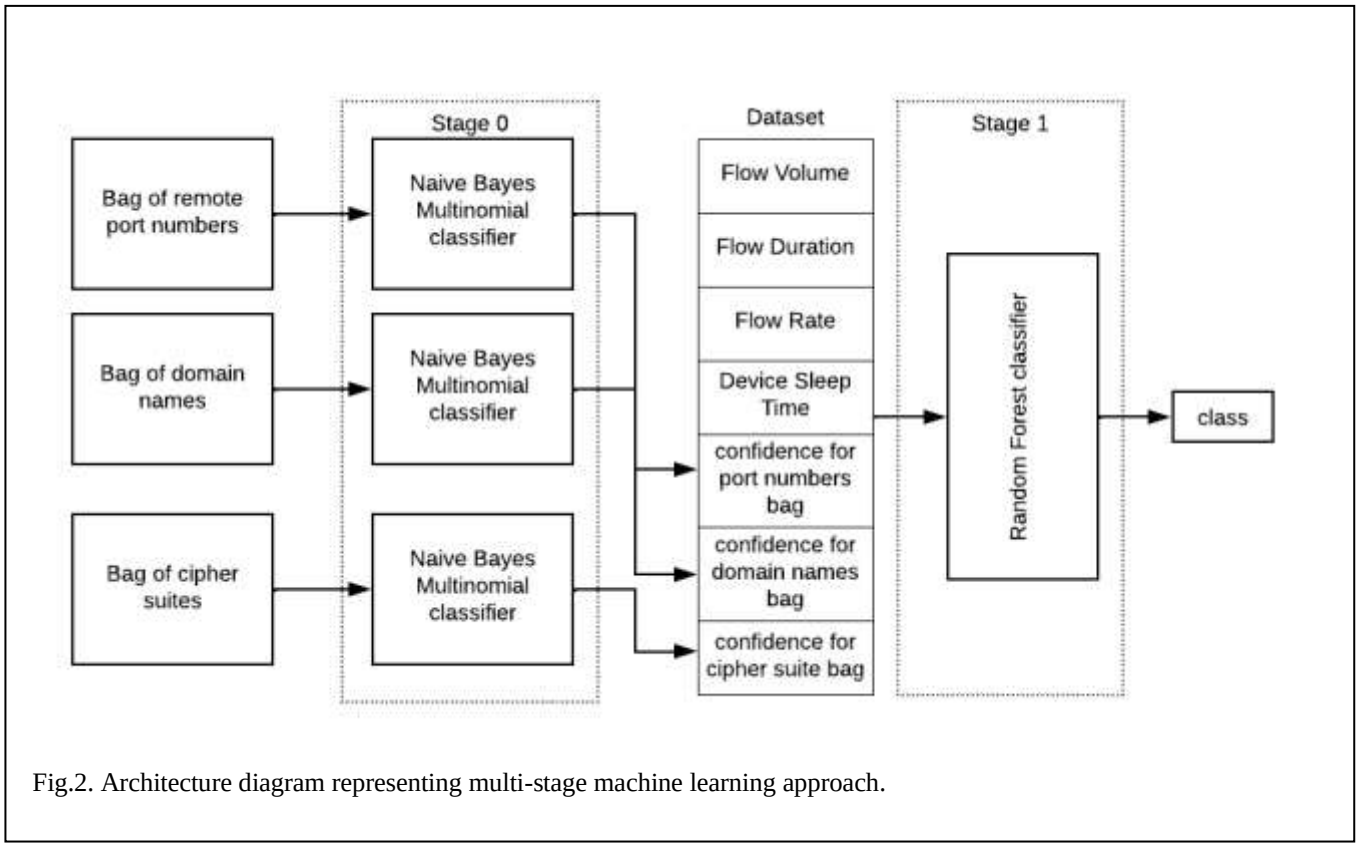


Fig.2. Architecture diagram representing multi-stage machine learning approach.

In contrast, the proposed preprocessing framework aims to streamline the data preparation process and reduce computational costs. By directly extracting the necessary features from the PCAP files and avoiding the need for multiple tool executions, the framework significantly reduces processing time. Moreover, the framework efficiently handles missing values by assigning a default value of "none" to missing DNS requests and cipher suites, eliminating the need for complex imputation techniques. By optimizing the preprocessing pipeline and minimizing unnecessary computations, the proposed approach offers a more efficient and scalable solution for data processing in less time with some betterments.

To address the challenges of IoT device classification, we propose a multi-stage machine learning approach inspired by the work of Sivanathan et al. [1]. In the initial stage, we employ a 'bag-of-words' technique to extract meaningful features from remote ports, domain names, and cipher suites. This technique, commonly used in natural language processing, allows us to represent these textual features as numerical vectors. Subsequently, a Naive Bayes classifier is applied to these feature vectors to obtain class probabilities and confidence scores. Unlike the original approach, which considered both class labels and confidence scores for the subsequent classification stage, we focus solely on the confidence scores.

In stage 0, we integrate the confidence scores from the Naive Bayes classifier with additional network traffic features, including flow volume, flow duration, average flow rate, and device sleep time. This combined dataset is then fed into a variety of machine learning algorithms, such as Random Forest, Gradient Boosting, and XGBoost, to make the final classification decisions and comparison.

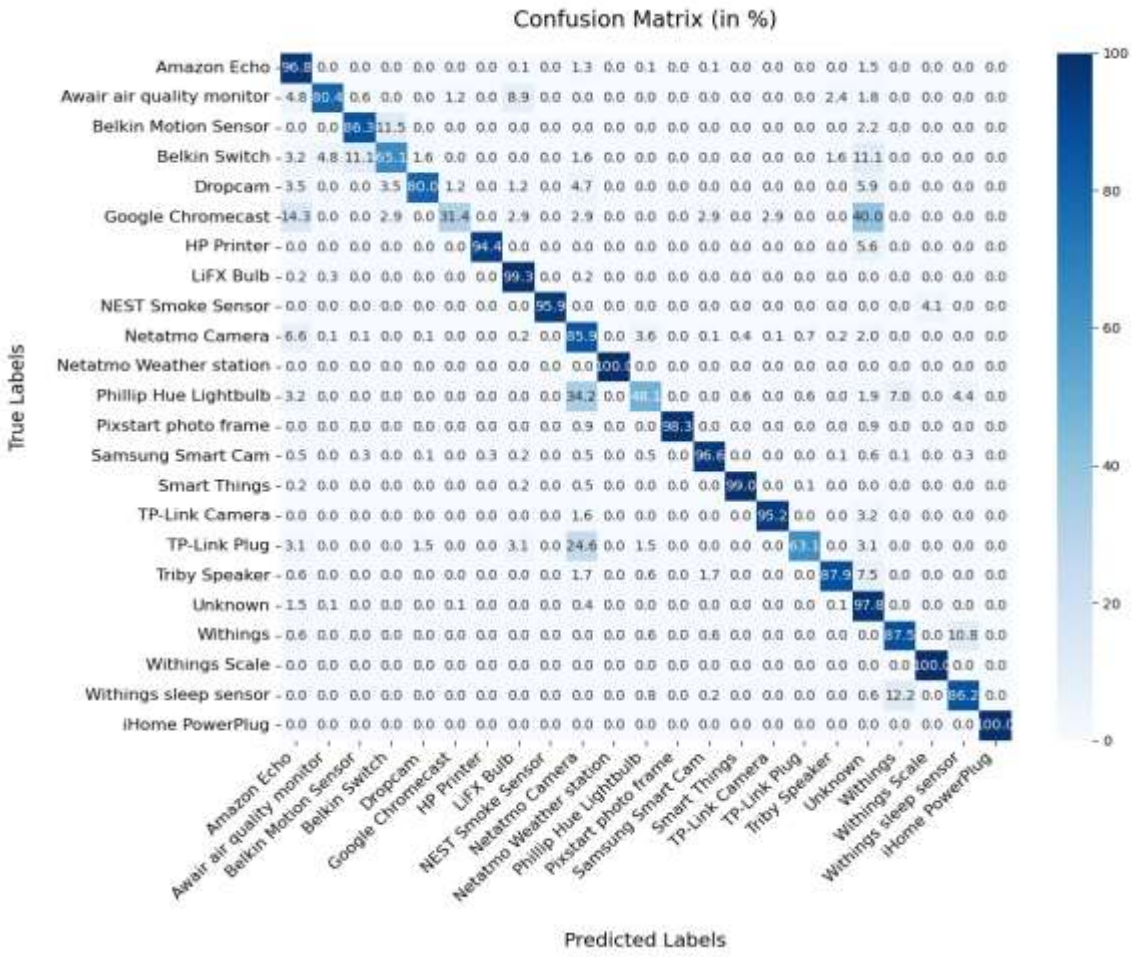


Fig.3.1. Confusion matrix of IoT device classification using random forest (accuracy: 95.86971 percent, RRSE: 28.60379 percent)

A. Random Forest

Random Forest is employed as the base classifier to determine the class label of each IoT device inspired from [1]. This learning method combines multiple decision trees to improve prediction accuracy and reduce overfitting and constructs a multitude of decision trees, each trained on a random subset of the data. During the prediction phase, the predictions of all trees are aggregated through a voting process, resulting in a final classification decision.

Our implementation of Random Forest achieved an accuracy of 95.86971% and an RRSE of 28.60379. Most of the devices have above 85% confidence except the devices like Google Chromecast, Belkin Switch and Phillip Hue Lightbulb which are uncertain depicting from Figure 3.1. To further improve the classification accuracy for these devices, we may consider exploring additional features or refining the feature extraction process and reducing the data bias.

B. Gradient Boosting (GB)

Gradient Boosting is known for its high accuracy, especially when dealing with complex datasets. It builds models sequentially, with each model focusing on correcting the errors of the previous ones. This iterative process leads to improved accuracy and robustness. It can handle both numerical and categorical features. While less interpretable than decision trees, Gradient Boosting can still provide some insights into the decision-making process.

While Gradient Boosting achieved a commendable accuracy of 95.02048815250312% and an RRSE of 31.579339450043353, its performance was slightly lower than that of Random Forest in both metrics. Additionally, by looking at the confusion matrix in Figure.3.2 we can say the model misclassified several devices, contributing to its relatively lower performance.

C. XGBoost

XGBoost is optimized for speed and performance, making it suitable for large datasets and an optimized implementation of gradient boosting. It incorporates regularization techniques to prevent overfitting. XGBoost can handle various types of machine learning tasks, including classification and regression. In our experiments, XGBoost demonstrated high performance when compare to remaining two models, achieving an accuracy of 96.26165449254707% and an RRSE of 25.765842151590924. These results highlight the effectiveness of XGBoost in handling complex classification tasks. Confusion matrix in Figure.3.3 depicts few devices are predicted as wrong it would be because of data bias. To

further improve the prediction accuracy for these devices, we may consider exploring additional features and reduce the data bias.

IV. Results and Discussion

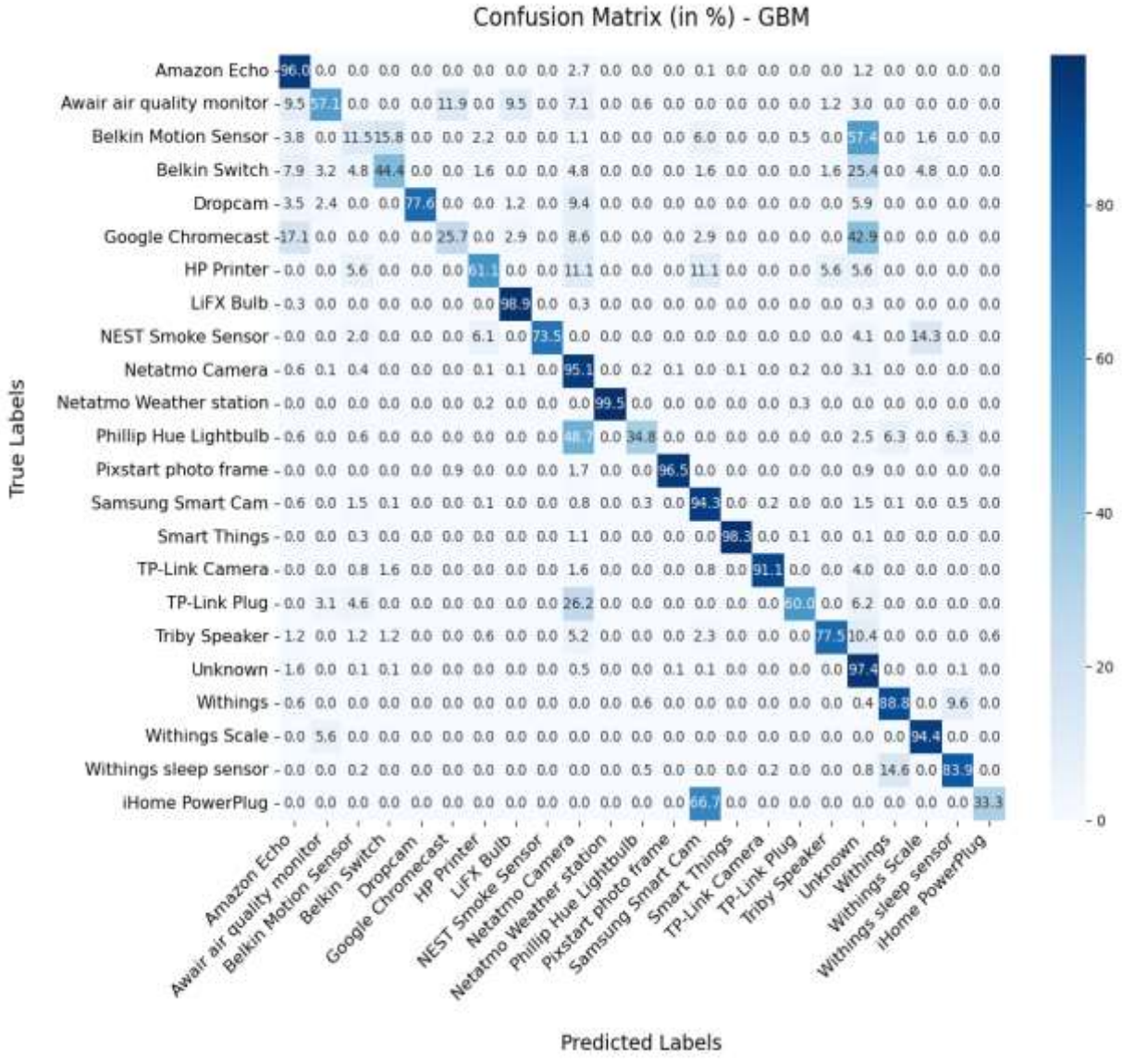


Fig.3.2. Confusion matrix of IoT device classification using Gradient Boosting (accuracy: 95.020 percent, RRSE: 31.579 percent).

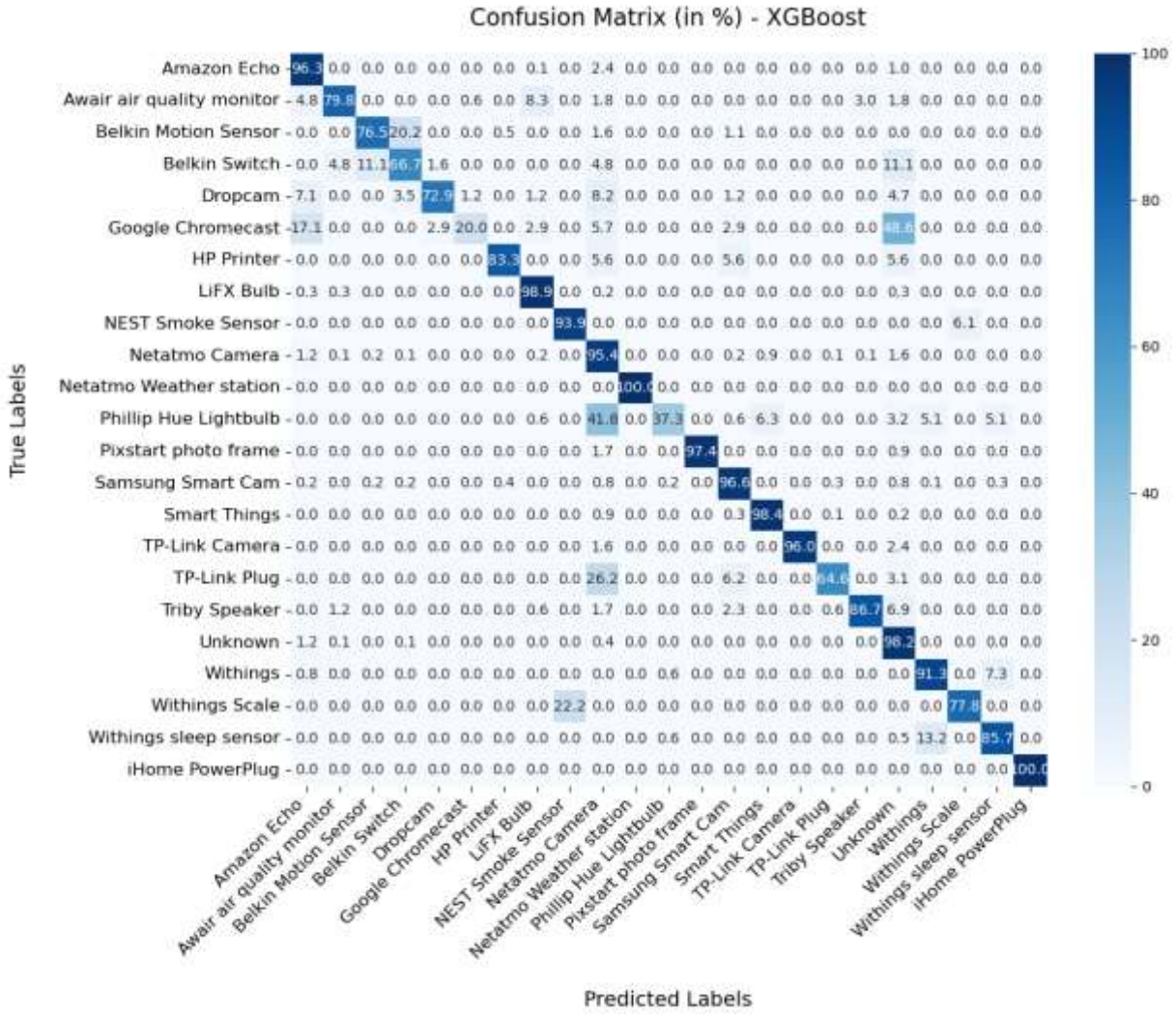
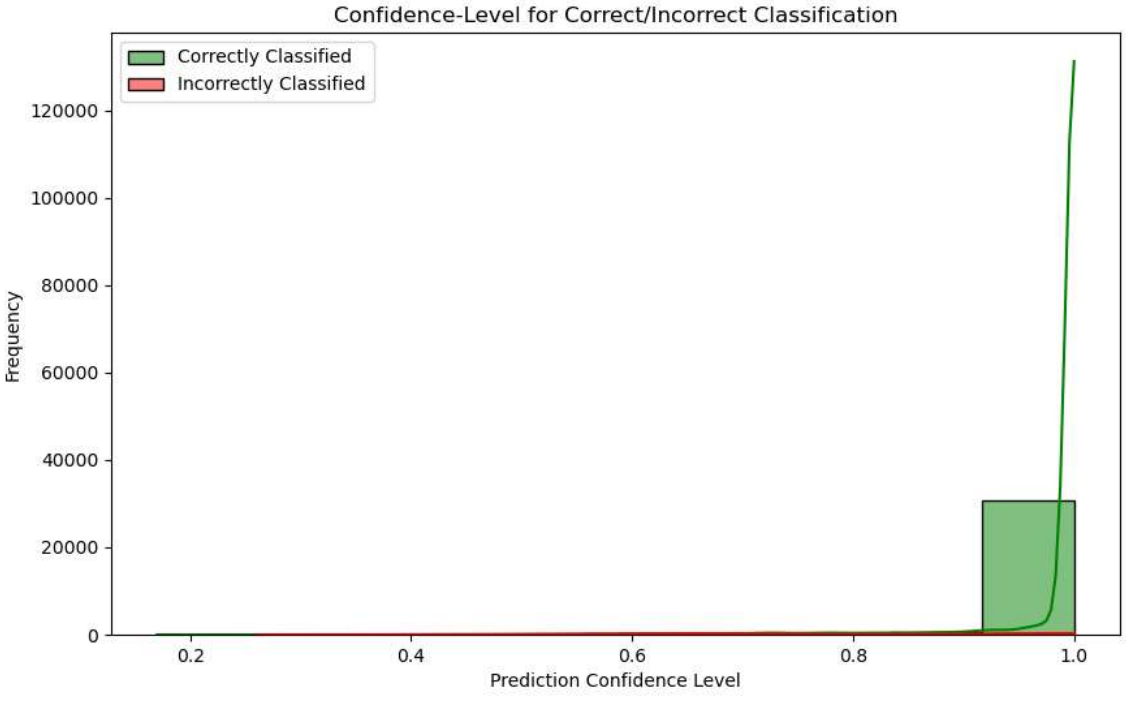
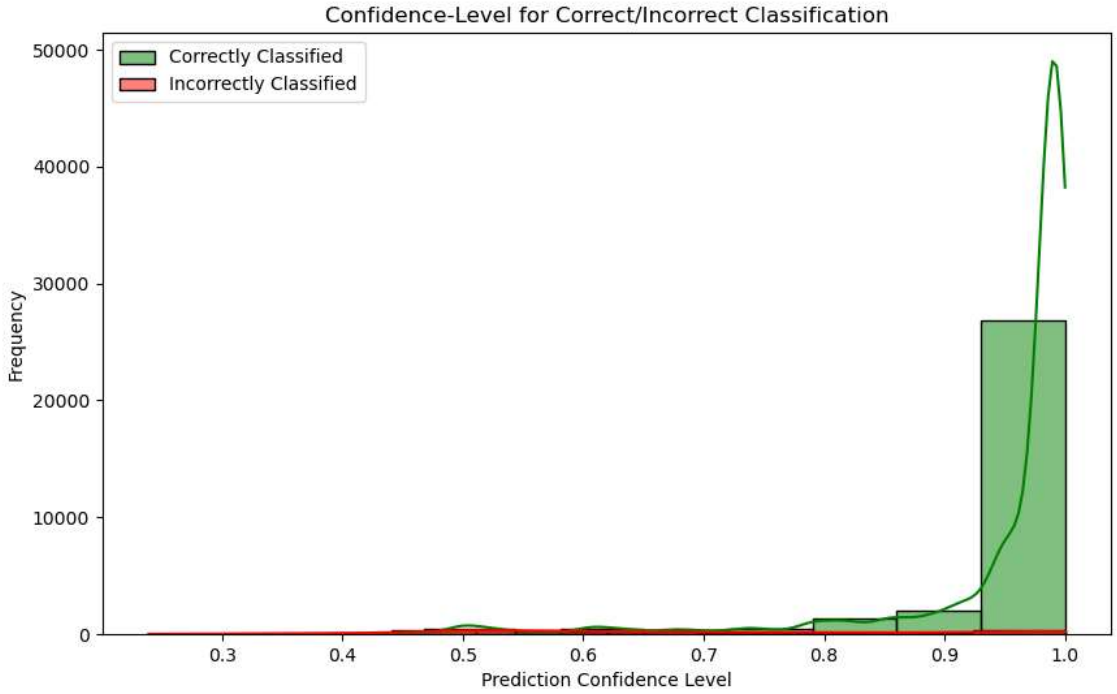


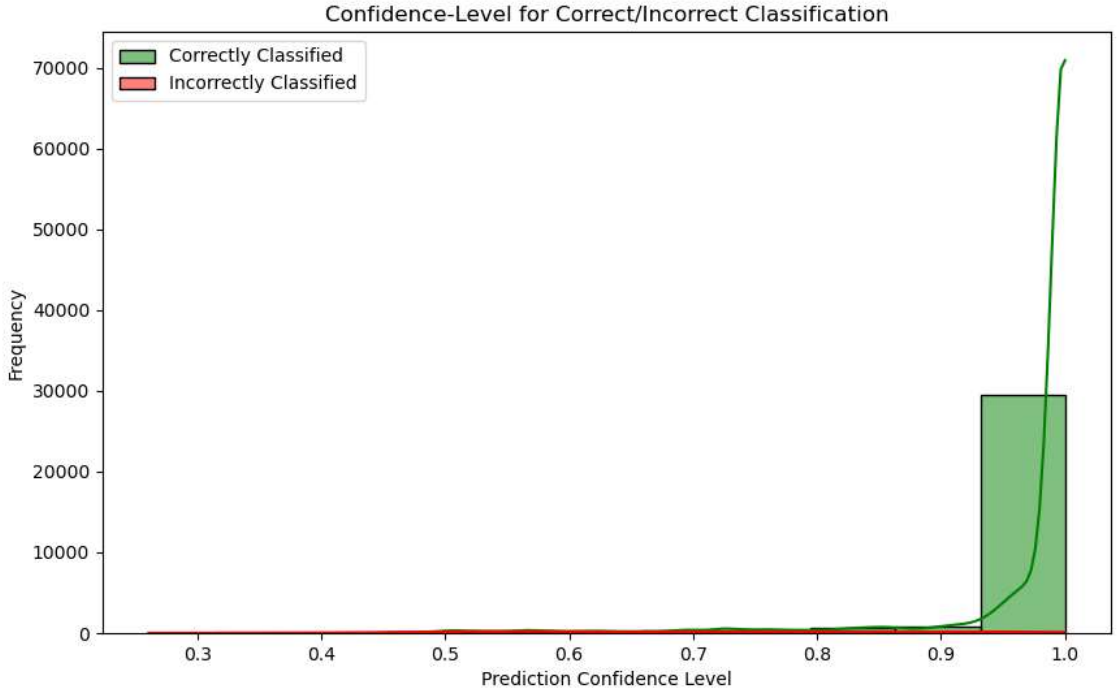
Fig.3.3. Confusion matrix of IoT device classification using XGBoost (accuracy: 96.261 percent, RRSE: 25.765 percent).



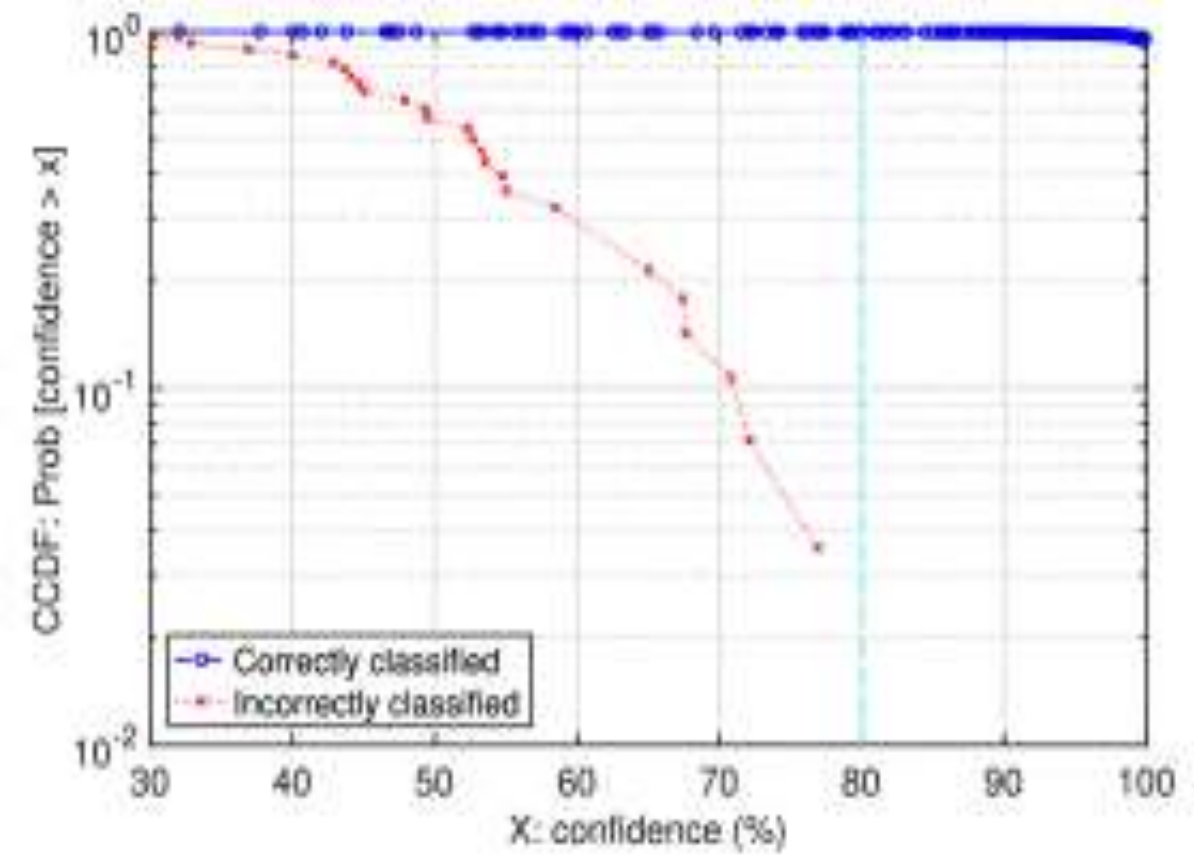
4(a) Confidence level graph for Random Forest model.



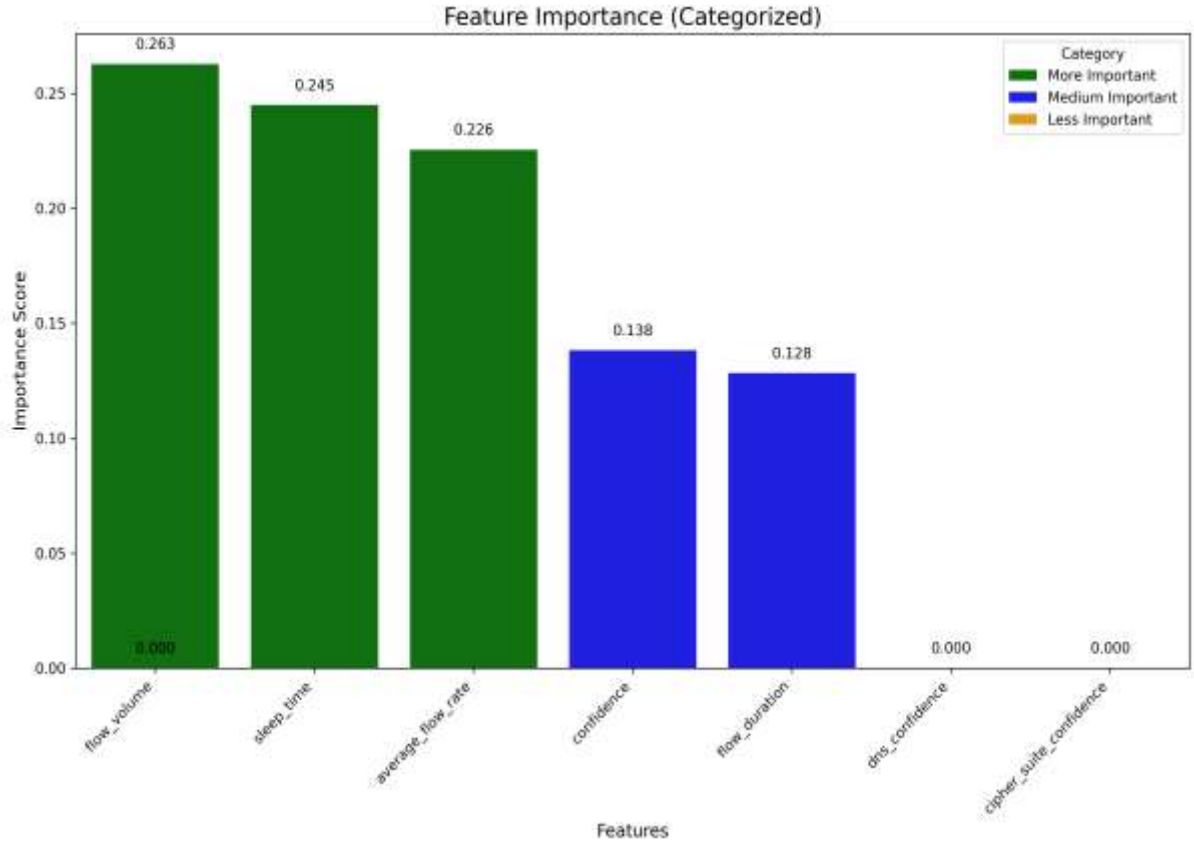
4(b) Confidence level graph for Gradient Boosting model.



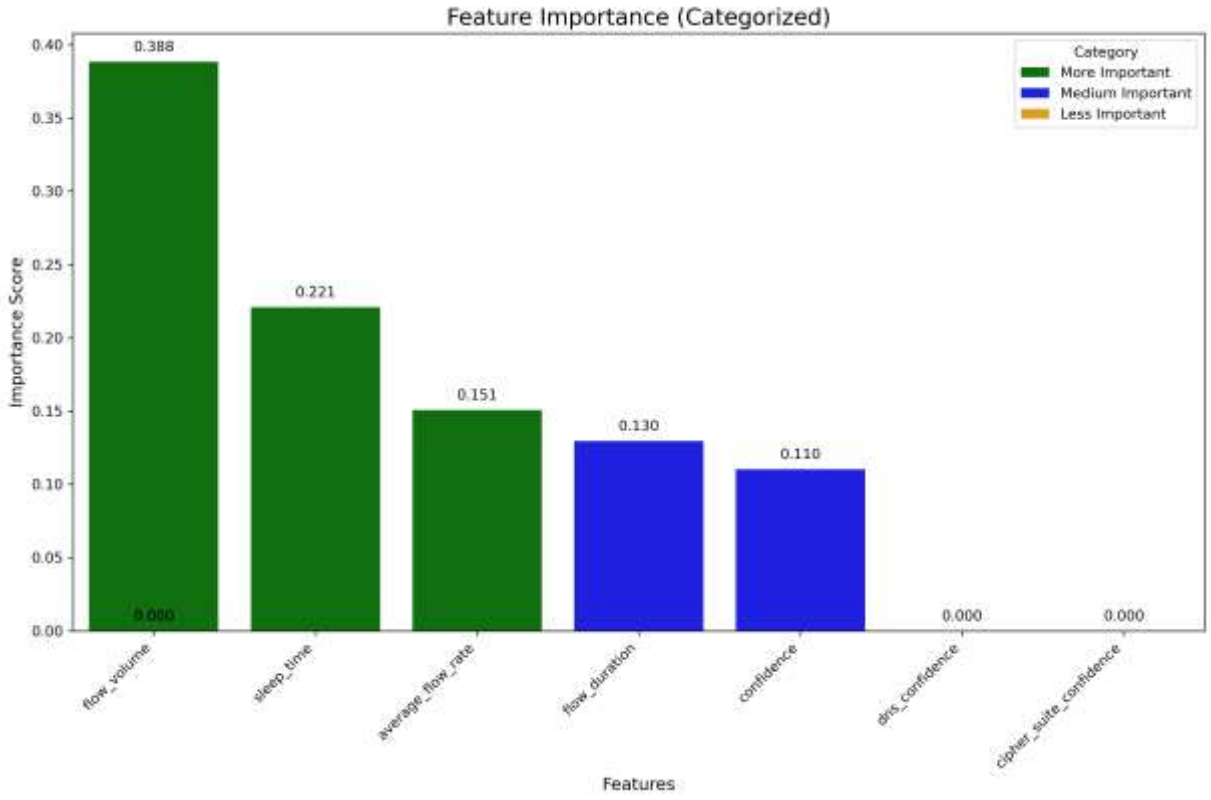
4(c) Confidence level graph for XGBoost model.



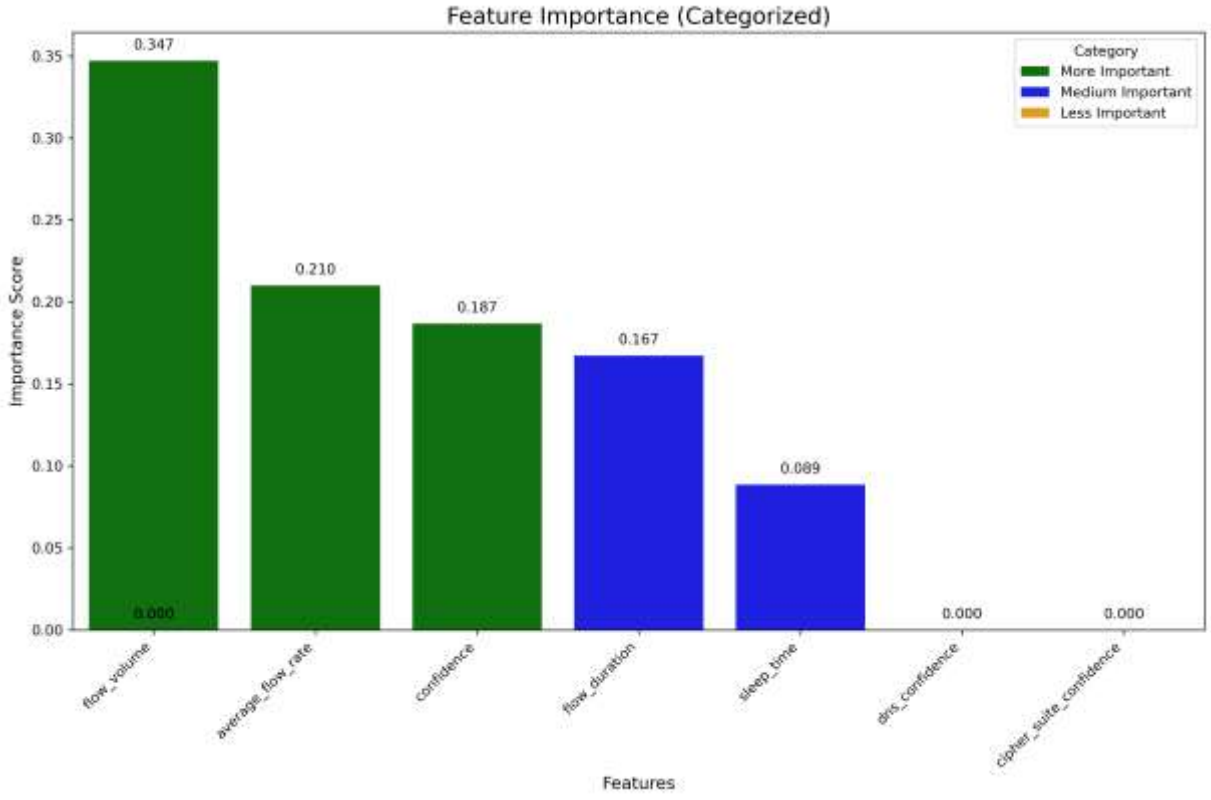
4(d) Confidence level graph form baseline (Using Random Forest) for 128 days data.



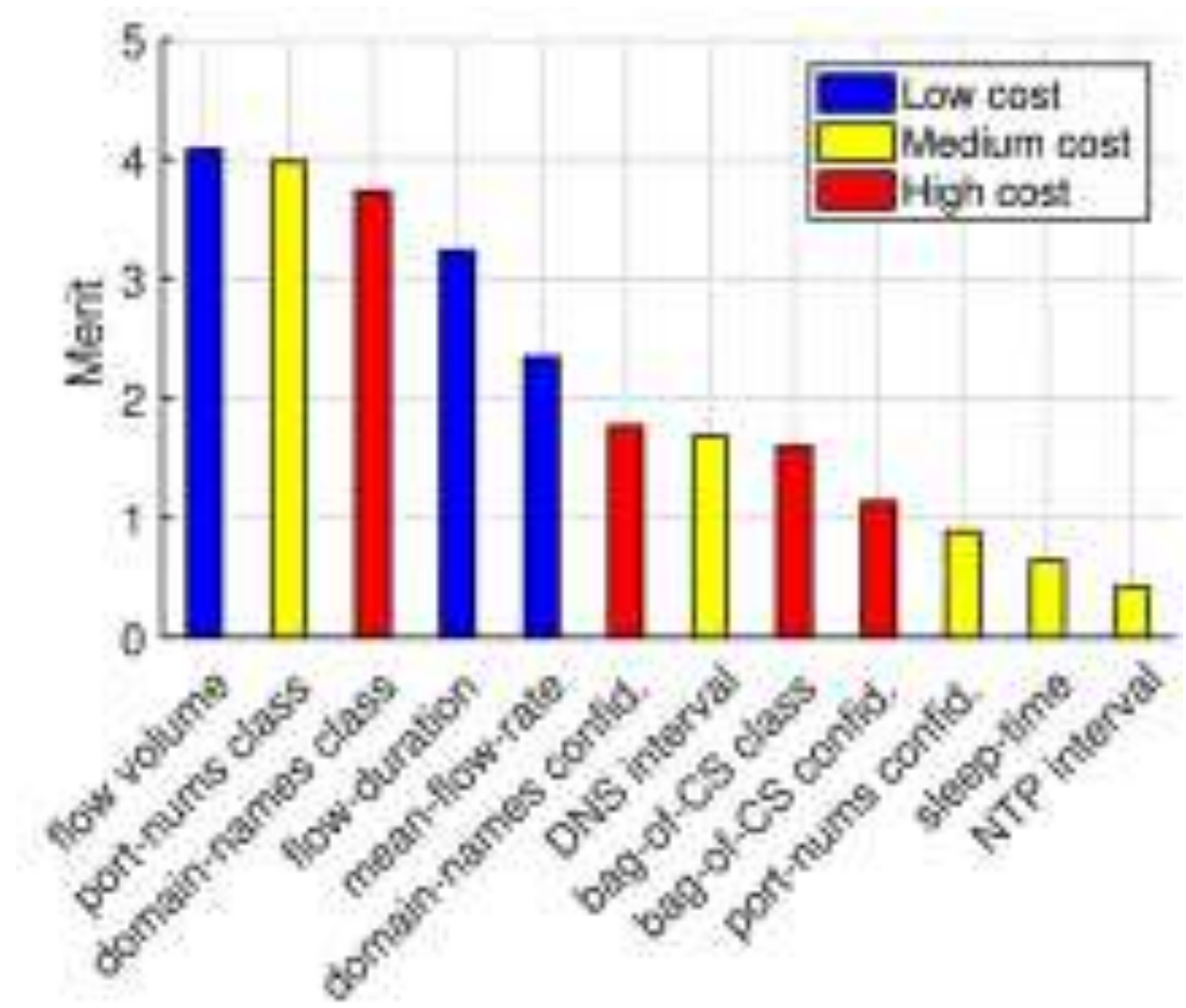
5(a) Attribute importance when using Random Forest.



5(b) Attribute importance when using Gradient Boosting.



5(c) Attribute importance when using XGBoost.



5(d) Attribute importance results from baseline (Using Random Forest) for 128 days data

Model	Metric	Value
Random Forest	Accuracy	95.86971
	RRSE	28.60379
Gradient Boosting	Accuracy	95.02049
	RRSE	31.57934
XGBoost	Accuracy	96.26165
	RRSE	25.76584
Base line (Random Forest)	Accuracy	99.88
	RRSE	5.06

Table.2. Accuracy and RRSE(Root Relative Squared Error) values of baseline paper and 3 different models implemented in this paper.

Model	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean Cross-Validation Scores
Random Forest	0.84841736	0.95507453	0.94872023	0.95914247	0.9395748	0.9301858780212602
Gradient Boosting	0.7472534	0.95005642	0.95480729	0.95347111	0.91739414	0.9045964724746124
XGBoost	0.85355425	0.96561554	0.94681988	0.96499198	0.9416533	0.9345269909139498

Table.3. Cross-Validation Scores in each fold and Mean Performance Across Models

Although we have not applied explicit feature selection techniques like PCA or mutual information, we have used feature importance scores derived from the models themselves to validate the choice of features, presented results in Figure.5. These scores provide a data-driven justification for the selection process, ensuring that the most relevant features are being used.

The Cross-Validation Scores and Mean Performance Across Models table presents the results of a 5-fold cross-validation for three different classification models: Random Forest, Gradient Boosting, and XGBoost. The Mean CV Scores show that XGBoost achieves the highest average performance at 93.45%, closely followed by Random Forest at 93.02%. Gradient Boosting, while still performing well, has a slightly lower mean score of 90.46%. The table also indicates that XGBoost and Random Forest demonstrate more consistent performance across the folds, while Gradient Boosting shows greater variability, particularly in Fold 1. These results suggest that XGBoost and Random Forest are better suited for this dataset, offering stable and robust classification performance.

Our proposed multi-stage machine learning approach, incorporating Random Forest, Gradient Boosting, and XGBoost, demonstrated promising results in classifying IoT devices based on their network traffic characteristics. XGBoost emerged as the top-performing model, achieving the highest accuracy of 96.26165449254707% and the lowest RRSE of 25.765842151590924. Random Forest, while slightly less accurate, demonstrated faster convergence, stabilizing after 70,000 data points. Gradient Boosting, though initially promising, exhibited decreased

stability with increased data, potentially due to overfitting or sensitivity to noise. Feature importance analysis revealed that flow volume was a crucial factor for all models, while features like confidence_of_dns and confidence_of_cipher were less influential due to data loss.

V. CONCLUSION

This paper aimed to replicate and extend the work of Sivanathan et al. in classifying IoT devices based on network traffic characteristics. By leveraging a multi-stage machine learning approach, we achieved promising results, with XGBoost emerging as the top-performing model. However, there is still room for improvement, particularly in addressing the misclassification of certain devices, such as Google Chromecast, Belkin Switch, and Philips Hue Lightbulb.

While our approach demonstrated strong performance, it is important to acknowledge the limitations inherent in the dataset and the complexity of IoT device classification. The baseline paper reported a higher accuracy of 99.88%, which may be attributed to factors such as a larger dataset (128 days) and potential differences in data preprocessing techniques and the hourly processing approach used in the baseline paper may have different insights into device behavior.

Future research directions include exploring advanced feature engineering techniques, such as incorporating device-specific information or behavioral patterns, to improve classification accuracy. Additionally, investigating the impact of different data preprocessing techniques can further optimize the performance of the proposed approach. By addressing these areas, we can enhance the robustness and accuracy of IoT device classification systems, contributing to the overall security and privacy of IoT networks. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same.

REFERENCES

[1] A. Sivanathan et al., "Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics," in IEEE Transactions on Mobile Computing, vol. 18, no. 8, pp. 1745-1759, 1 Aug. 2019, doi: 10.1109/TMC.2018.2866249.

[2] A. Sivanathan et al., "Characterizing and classifying IoT traffic in smart cities and campuses," 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Atlanta, GA, USA, 2017, pp. 559-564, doi: 10.1109/INFCOMW.2017.8116438.

[3] S. Alexander and R. Droms, “DHCP Options and BOOTP vendor extensions,” Internet Requests for Comments, RFC Editor, RFC 2132, Mar. 1997.

[4] https://github.com/cisco/joy

[5] https://github.com/kakumanuphanindra/Classifying_IoT_Devices_Based_on_Network_Patterns (source code link for this paper)