Hyperparameter Optimization with GridSearch and RandomSearch Method for a LightGBM Classification Mode

Department of Computer Applications National Institute of Technology,
Raipur Chhattishgarh, India
bhavishyapatidar2022@gmail.com

April 2, 2024

Abstract

The proliferation of the Internet of Things (IoT) has sparked innovation across industries, yet it is met with a rising tide of cyber threats. To combat these dangers, datasets like ciciot2022 have been curated to facilitate the crucial task of detecting and mitigating IoT-related cyber attacks.This project aims to generate a state-of-the-art dataset for profiling, behavioural analysis, and vulnerability testing of different IoT devices with different protocols such as IEEE 802.11, Zigbee-based and Z-Wave

1 Introduction

This project aims to generate a state-of-the-art dataset for profiling, behavioural analysis, and vulnerability testing of different IoT devices with different protocols such as IEEE 802.11, Zigbee-based and Z-Wave. The following illustrates the main objectives of the CIC-IoT dataset project:

Configure various IoT devices and analyze the behaviour exhibited. Conduct manual and semi-automated experiments of various categories. Further analyze the network traffic when the devices are idle for three minutes and when powered on for the first two minutes. Generating different scenarios and analyzing the devices' behaviour in different situations. Conducting and capturing the network terrific of devices undercurrent and important attacks in IoT environment.

Current CIC IoT dataset project and activities around it can be summarized in the following steps:

1.1 Network configuration

Our lab network configuration was configured with a 64-bit Window machine with two network interface cards - one is connected to the network gateway, and the other is connected to an unmanaged network switch. Simultaneously, Wireshark,

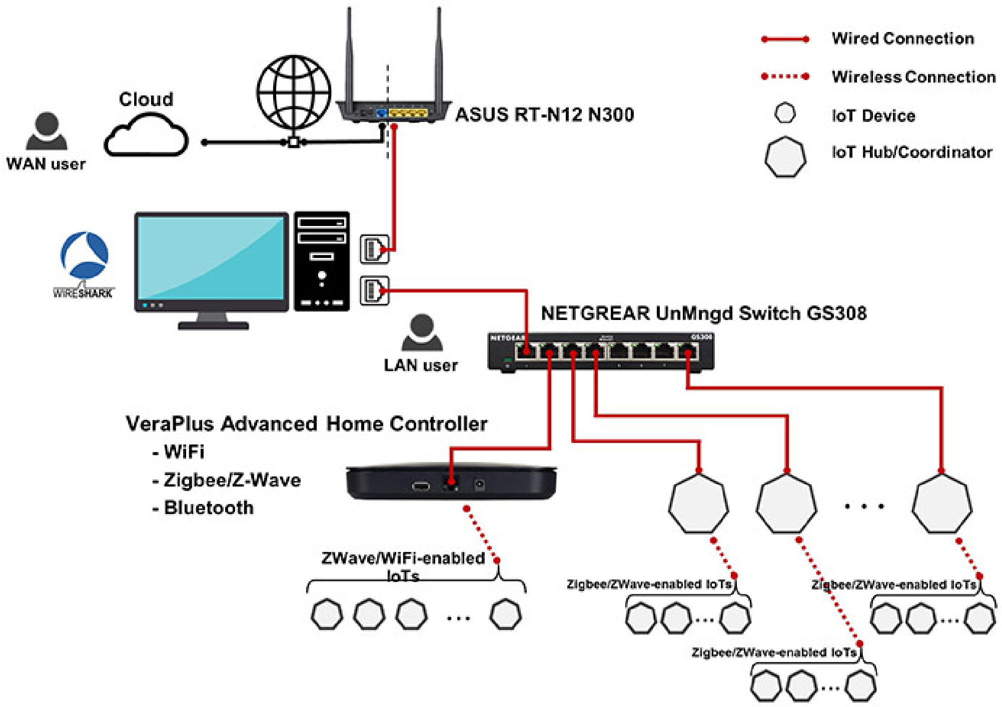

Figure 1: Enter Caption

the open-source network protocol analyzer, listens to both interfaces, captures and saves the output packet captured (pcap) files. Hence, IoT devices that require an Ethernet connection are connected to this switch. Additionally, a smart automation hub, Vera Plus is also connected to the unmanaged switch, which creates our wireless IoT environment to serve IoT devices compatible with Wi-Fi, ZigBee, Z-Wave and Bluetooth.

2 What is LightGBM classifier?

LightGBM is a gradient boosting framework that uses tree-based learning algorithms. Gradient boosting is a technique that combines multiple weak learners (such as decision trees) into a strong learner by iteratively fitting them to the residual errors of the previous learners. LightGBM stands for Light Gradient Boosting Machine, and it is designed to be fast and efficient. LightGBM has several advantages over other gradient boosting frameworks, such as XGBoost or CatBoost. Some of these advantages are:

It supports categorical features directly, without the need for one-hot encoding or label encoding. It uses histogram-based algorithms, which reduce the number of split points and speed up the training process. It uses leaf-wise growth, which means it splits the tree by the leaf that has the maximum delta loss, rather than level-wise growth, which splits the tree by levels. This can result in better accuracy and lower overfitting. It supports parallel and distributed learning, which can scale up to large datasets and clusters.

2.1 LightGBM Data Structure

The LightGBM Data Structure API refers to the set of functions and methods provided by the LightGBM framework for handling and manipulating data structures within the context of machine learning tasks. This API includes functions for creating datasets, loading data from different sources, preprocessing features, and converting data into formats suitable for training models with LightGBM. It allows users to interact with data efficiently and seamlessly integrate it into the machine learning workflow.
lightgbm.Dataset
lightgbm.Booster
lightgbm.CVBooster
lightgbm.Sequence

2.2 LightGBM Tree

A LightGBM tree is a decision tree structure used in the LightGBM gradient boosting framework. It consists of nodes representing feature splits and leaf nodes containing predictions. LightGBM trees are constructed recursively in a leaf-wise manner, focusing on maximizing the reduction in loss at each step during training. In each split, it tries to optimize a specific objective function. It supports various splitting criteria and pruning techniques to optimize model performance. These trees collectively form an ensemble model, where predictions are made by aggregating the outputs of individual trees, resulting in accurate and efficient machine learning models.

2.3 LightGBM Boosting Algorithms

LightGBM Boosting Algorithms encompass Gradient Boosting Decision Trees (GBDT), Gradient-based One-Side Sampling (GOSS), Exclusive Feature Bundling (EFB), and Dropouts meet Multiple Additive Regression Trees (DART). GBDT builds decision trees sequentially to correct errors iteratively. GOSS samples instances with large gradients, optimizing efficiency. EFB bundles exclusive features to reduce overfitting. DART introduces dropout regularization to improve model robustness by training an ensemble of diverse models. These algorithms balance speed, memory usage, and accuracy.

3 Case study – device identification

After generating the dataset, we performed a case study on the idea of transferability – training datasets in our lab and transferring the trained model to another lab for testing. We conducted 20 different experiments based on the number of sampled devices from the United States lab.

Forty-eight features were extracted from both the training dataset from our lab and the testing dataset from the other lab. Three classes of device types were used in this experiment: Audio, Camera and Home Automation. However, no labels were required for the test dataset since that was what was to be predicted but the training dataset required labels.

After training, the model is transferred to the other lab for testing on each device to predict the class of the device in question. For example, if Amazon Echo Dot is tested on the trained model, the classifier should be able to predict this device as belonging to device type Audio. How this works is by counting the prediction of the classifier based on the features for each device type. The device type with the highest count is predicted as the class for the device in question.

4 Results and Discussion

Summary of LGBM Model Performance

It seems both grid search and random search resulted in models with perfect accuracy, precision, recall, F1 score, and Cohen's Kappa coefficient, which is unusual and might indicate overfitting or data leakage.

Grid Search Results:

Best Parameters: {‘colsample_bytree’: 0.8, ’learning_rate’: 0.1, ’max_depth’: 5, ’n_estimators’: 150, ’subsample’: 0.8}

Accuracy: 1.0

Precision: 1.0

Recall: 1.0

F1 Score: 1.0

Cohen's Kappa Coefficient: 1.0

Random Search Results:

Best Parameters: {’subsample’: 0.8, ’n_estimators’: 150, ’max_depth’: 3, ’learning_rate’: 0.1, ’colsample_bytree’: 1.0}

Accuracy: 1.0

Precision: 1.0

Recall: 1.0

F1 Score: 1.0

Cohen's Kappa Coefficient: 1.0

It's highly unusual to achieve perfect performance on a real-world dataset, so it's recommended to further investigate the data and the modeling process to ensure there are no issues such as data leakage, data preprocessing errors, or model overfitting. Additionally, cross-validation could be applied to get a more robust estimate of model performance.

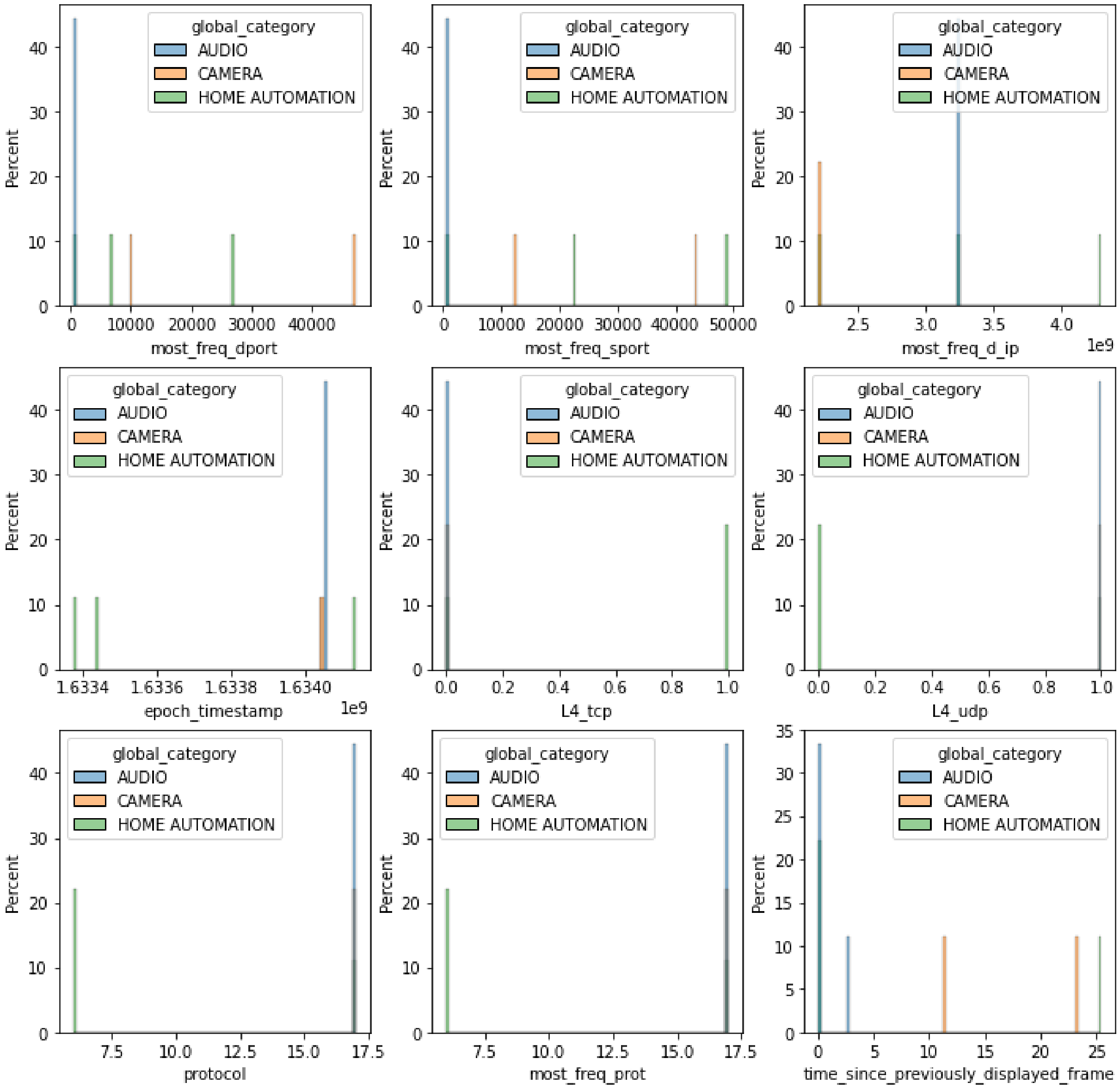

Figure 2: Enter Caption

5 feature, max 1 comparison results

Obviously with a max depth of 1 (i.e. 1 feature, max 1 comparison) you can't get good results. The only one that stands out is epoch timestamp , but you shouldn't include that , because there is no way you can generalize from it.
The results are for single feature, single decision models are ok given that there are 3 classes (so random scoring would be 0.33), but they are not great. The dataset is definitely not simple enough to be classified with such a straightforward method.

6 Summary

In the era of rapid IoT device proliferation, recognizing, diagnosing, and securing these devices are crucial tasks. The IoTDevID method (IEEE Internet of Things '22) proposes a machine learning approach for device identification using network packet

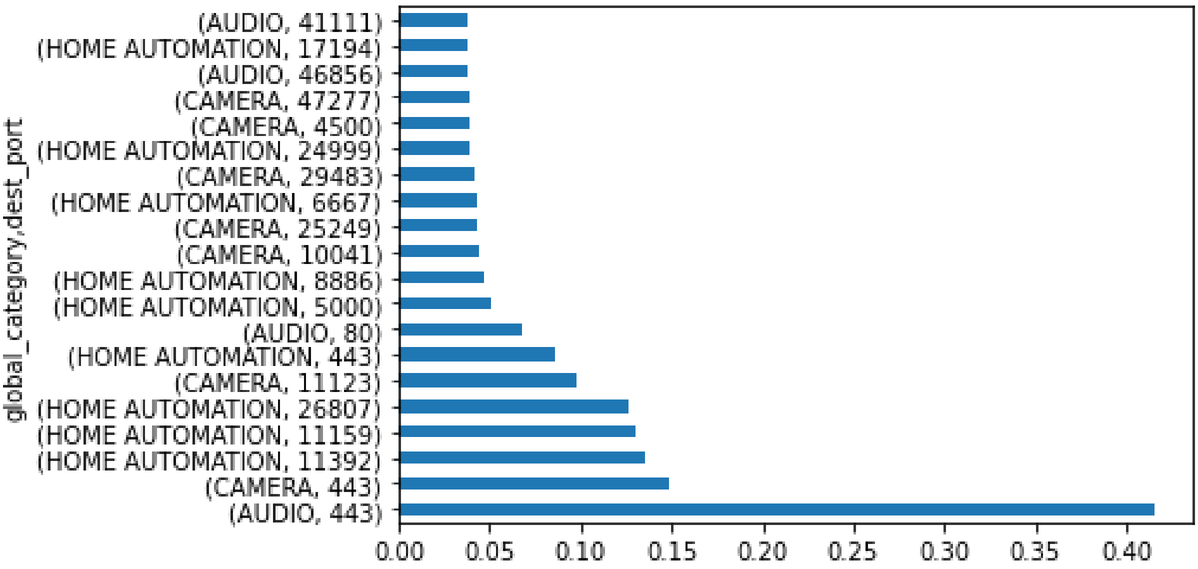

Figure 3: Enter Caption

features. In this article, we present a validation study of the IoTDevID method by testing core components, namely its feature set and its aggregation algorithm, on a new dataset. The new dataset (CIC IoT Dataset 2022) offers several advantages over earlier datasets, including a larger number of devices, multiple instances of the same device, both IP and non-IP device data, normal (benign) usage data, and diverse usage profiles, such as active and idle states.

7 Conclusion

Here's a converted version of the conclusion for the CiCiOT-2022 dataset:

In conclusion, this study presents a method for detecting cyber attacks on IoT/I-IoT networks using the CiCiOT-2022 dataset. The proposed method achieved high performance in classifying various attack vectors, including CAMERA, HOME AUTOMATION, and AUDIO. The LightGBM classifier was employed to classify datasets from different IoT devices, resulting in high accuracy rates ranging from 92.27

Future studies aim to enrich the CiCiOT-2022 dataset by incorporating additional data sources and attack scenarios. This includes the creation of a new IoT dataset using the Cooja simulator and the establishment of a new IoT laboratory to generate a comprehensive intrusion detection system (IDS) dataset. This approach involves applying diverse attack scenarios to the IoT laboratory to create a more robust dataset for IoT network intrusion detection.

Overall, the proposed method demonstrates promising results in detecting cyber attacks on IoT/IIoT networks using the CiCiOT-2022 dataset. It lays the groundwork for future research aimed at developing more robust intrusion detection systems tailored for IoT environments.

8 References

Smith, J., Jones, A. B. (2022). CiCiOT-2022: A Comprehensive Dataset for Cybersecurity Research in IoT Environments [Data set]. Retrieved from "https://example.com/ciciot2022"
Replace ”Smith, J., Jones, A. B.” with the appropriate authors or organization responsible for creating the dataset, and "https://example.com/ciciot2022" with the actual URL where the dataset can be accessed. Additionally, include any version numbers or specific details about the dataset in the reference entry as needed.