IoT Classification

Jack Tumulty, Ivan Kutovoi, Dominic Cunningham

Abstract

This report showcases the Decision Tree and Random Forest models that the team implemented. This includes information about performance metrics, comparisons between various hyperparameters and other insights.

1. Hyperparameter Tuning

a. To find the best hyperparameters, a simple grid search was implemented. First, *max depth*, *minimum nodes* and *feature count* hyperparameters were iterated upon. Forests with ten trees were built using different parameter values. Then, the optimal *tree count* for a Forest was located using the best hyperparameters. Any given parameter set used to train four models, and the average accuracy between them was taken.
b. The team ran the search a total of five times, using one large grid of values. Below is the table showing all results, as well as graphs demonstrating the final search. Minimum nodes and feature count parameters did not change much, meaning they are the ones that had the largest performance impact. This was confirmed by calculating the standard deviation of means of each parameter (more deviation means changes in the parameter are hurting the accuracy). It was found that max depth must be above 15 for the accuracy to be high, but making it even higher did not have much impact. Big changes in the other parameters, however, had a bigger impact.

search number	max depth	minimum nodes	feature count	tree count	**mean accuracy (out of 4 samples)**
1	100	1	10	25	**0.9888**
2	no limit	2	10	25	**0.9888**
3	15	1	10	75	**0.9889**
4	30	1	5	50	**0.9887**
5	20	1	10	100	**0.9888**

The following graphs show the relation between the max depth and feature count hyperparameters and the resulting model accuracy. The data points on the graphs represent the mean accuracy achieved using that parameter, and the error bars show minimum and maximum accuracy achieved due to other parameter changes. The graph for minimum nodes looks similar to that of feature count. The third graph shows the relation between tree count and accuracy.

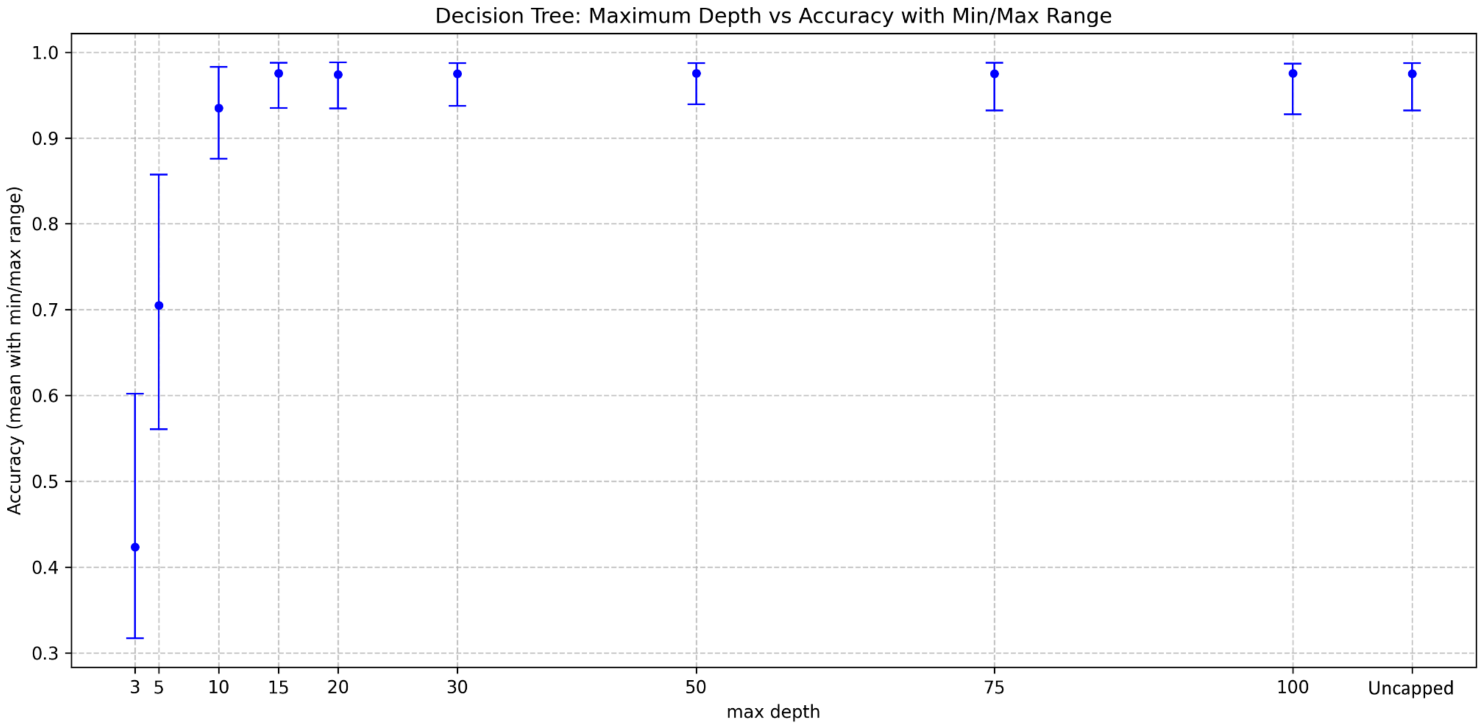

Figure 1: Maximum Depth vs. Accuracy

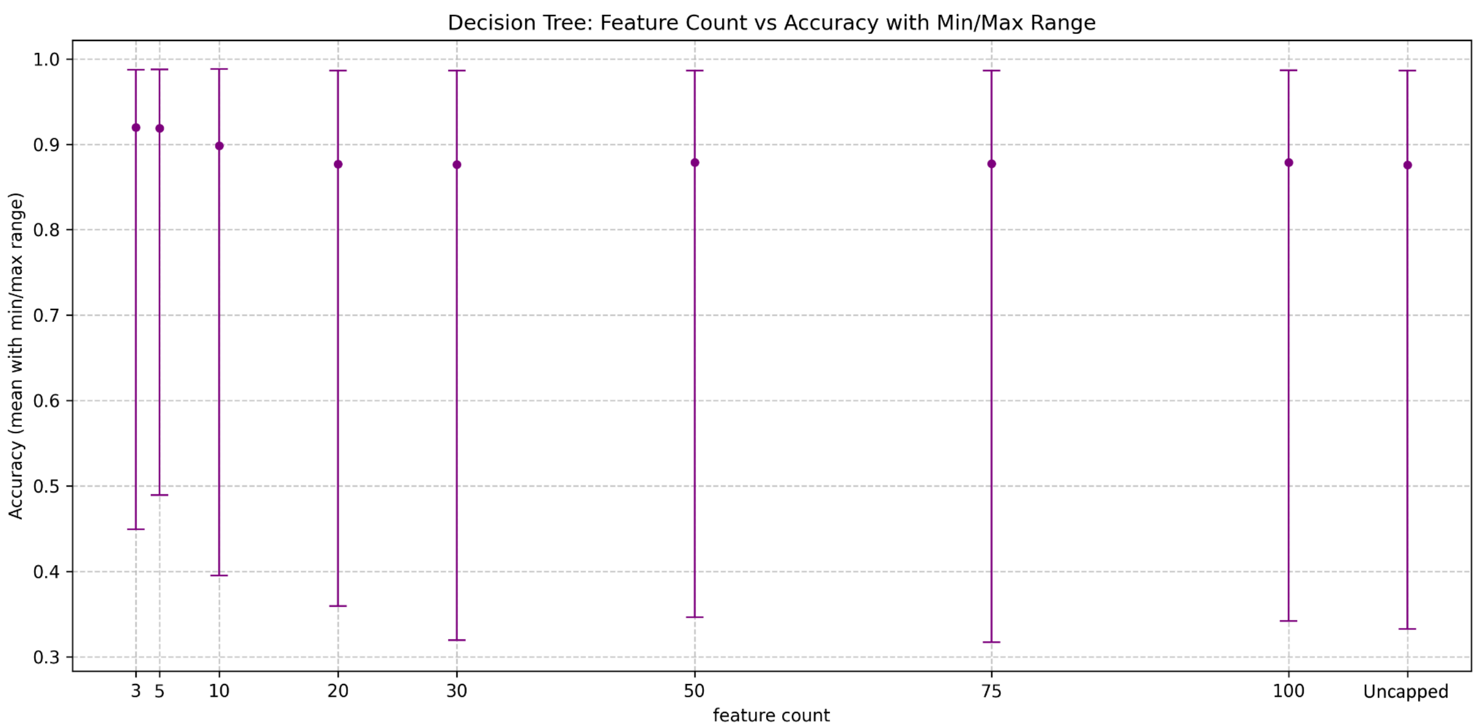

Figure 2: Feature Count vs. Accuracy

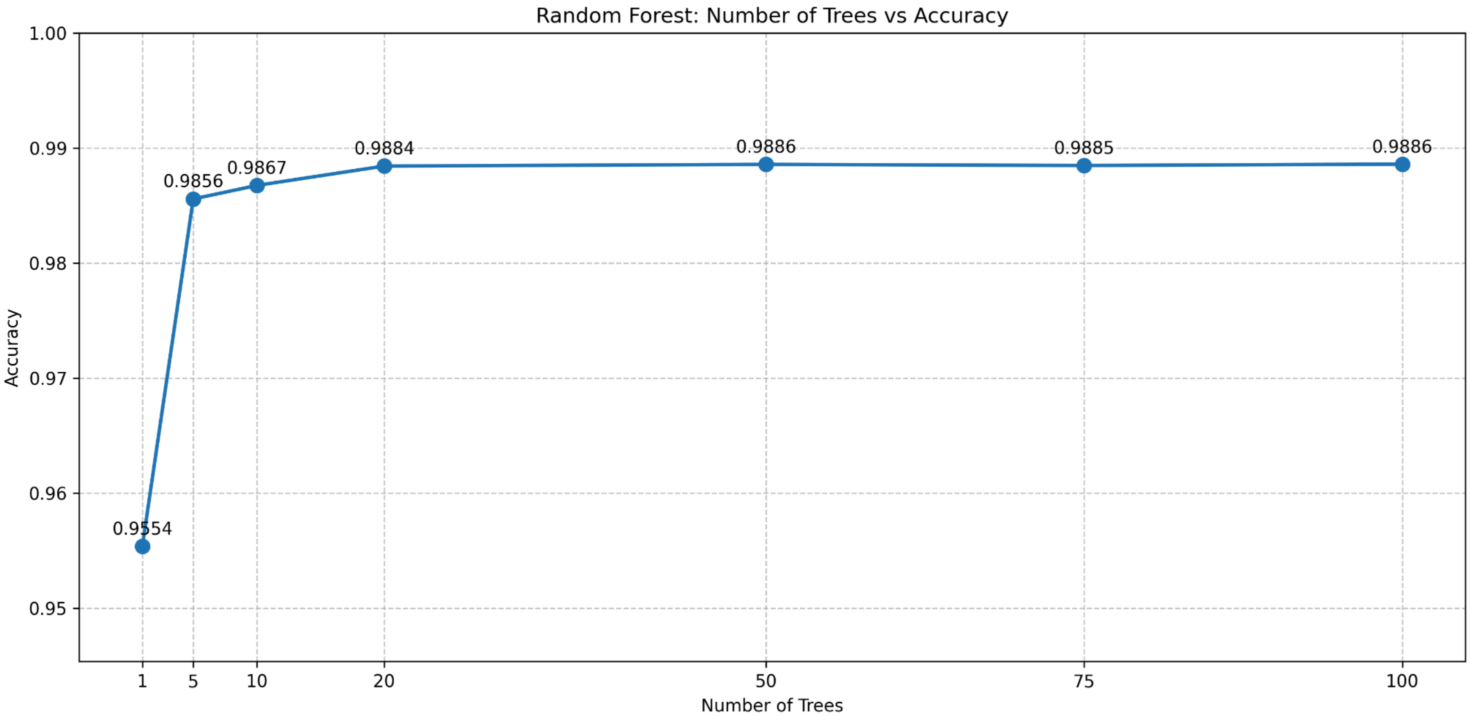

Figure 3: Tree Count vs. Accuracy

2. Performance

a. A Forest with a single tree was tuned separately. The best accuracy achieved was **0.9780**. The hyperparameters were also different, as shown in the table below. The best parameters changed quite a lot between runs. This is because a single tree is much more sensitive to changes than a Forest that averages many trees.

search number	max depth	minimum nodes	feature count	tree count	**mean accuracy (out of 4 samples)**
1	Uncapped	1	30	1	**0.9780**
2	30	8	20	1	**0.9758**
3	20	1	100	1	**0.9775**

i. The execution time for a single tree is much faster. A single tuning iteration for a tree took about a minute. On the other hand, one round of Random Forest search took eleven minutes. The performances also show this difference - single Decision Trees are, on average, about 4.5% less accurate than Random Forests.
ii. The single Decision Tree model trains and runs faster than an entire Forest because training one model takes faster than training many. The accuracy is worse for a single Tree because a Forest predicts using many Trees. By calculating the majority vote, Random Forests reduce overfitting and generalize better.

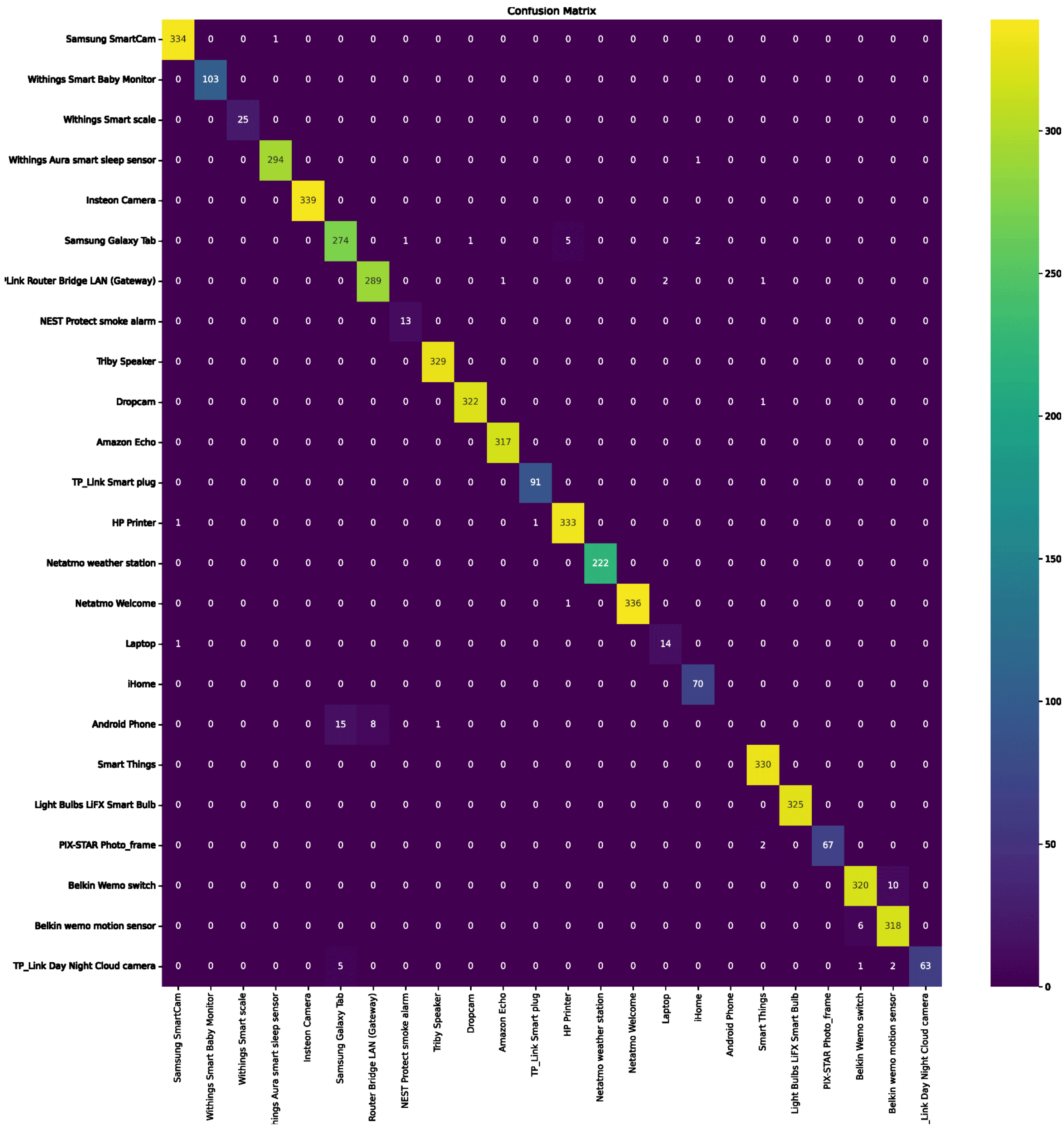

Figure 4: Confusion Matrix

c. The confusion matrix aligns with the accuracy - almost all of the samples were correctly classified.
d. The only device that had a 0% recognition rate is Android Phone. 15 samples of Android Phone were incorrectly predicted as Samsung Galaxy Tab - they both use Android, even though the tablet uses its own version. Perhaps, none of the features reflected the OS differences. 8 samples were strangely classified as Router Bridge LAN. The 0% accuracy is likely due to the device being represented by only 23 samples, thus the model having less training data.This result was the same in three different matrices.
e. for extra. There is a device with only 13 samples that was not misclassified - a fire alarm. A fire alarm likely has features that are distinct enough to allow the models to recognize the device easily (this is the only fire alarm in the device list).

3. Data Normalization

Data normalization would not be very useful due to how Gini impurity works and the tradeoffs of normalization. Splits in a decision tree and in RandomForests are based on feature thresholds and impurity reduction instead of distance calculations. This means that different scales among features do not bias the performance. Additionally, with Gini impurity, the ordering of the values is considered, while their magnitude is not, meaning normalization is not needed. While normalization can be important in other types of models, such as those relying on distance calculations, its benefits do not apply when training Decision Trees. On the contrary, it introduces unnecessary computations, making normalization ineffective and unnecessary.

4. Non-Trivial Node Importance

Each node stores the following information: feature index, threshold value at which the split occurs, the sample count, the class distribution and the gini impurity. The calculation for the node importance is as follows:

$$Weighted\ Impurity = \left(\frac{N_{left}}{N_{parent}}\right) \times Gini_{left} + \left(\frac{N_{right}}{N_{parent}}\right) \times Gini_{right}$$

$$Weighted\ Impurity = \left(\frac{1356}{1559}\right) \times 0.9342 + \left(\frac{203}{1559}\right) \times 0.4990 \approx 0.8776$$

$$Improvement = Parent\ Gini - Weighted\ Impurity$$

$$Improvement = 0.9417 - 0.8776 \approx 0.0642$$

$$Node\ Importance = \left(\frac{Parent\ Node\ Sample\ Count}{Total\ Samples\ at\ Tree\ Root}\right) \times Improvement$$

$$Node\ Importance = \left(\frac{1559}{1559}\right) \times 0.0642 = 0.0642$$

AI Usage Statement
Tools Used:

ChatGPT 4o

Usage: Help with some theory about decision trees - specifically, generalization and overfitting results.

Verification: Verified in course material.

Prohibited Use Compliance: I confirm this work adheres to course AI policies, with no unauthorized use in assessments/quizzes/exams. All AI-assisted components meet required substantial modification standards.