In [37]:
```
#Created by Rami ALmehdawi

# Import the necessary Libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import sklearn as sk
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix, classification_report, roc_
from imblearn.over_sampling import RandomOverSampler, SMOTE
```

In [38]:
```
# Read in the Data
MW_Data = pd.read_csv("Malware-staDyn-data.csv")
```

In [39]:
```
# Drops all Missing values in Label and selects the Feature Column
x = MW_Data.drop('label', axis = 1)
y = MW_Data["label"]

# Split into Training and Test Sets
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.1, random_state=42)
```

In [40]:
```
# Balances the training data
smote = SMOTE(random_state=42)
x_train_resampled, y_train_resampled = smote.fit_resample(x_train, y_train)
```

In [41]:
```
# Creats & Trains the Random Forest with 5 trees
rf = RandomForestClassifier(n_estimators=5, random_state=42, class_weight= {1:2, 0:1})
rf.fit(x_train_resampled, y_train_resampled)
```

Out[41]:
```
▾                    RandomForestClassifier

RandomForestClassifier(class_weight={0: 1, 1: 2}, n_estimators=5,
                       random_state=42)
```

In [42]:
```
y_pred = rf.predict(x_test)

y_pred_proba = rf.predict_proba(x_test)[:, 1]
```

```
AUC-ROC: 0.9995408973595393
```

In [68]:
```
# Prints out Relevant Metrics
print("Accuracy: ", accuracy_score(y_test, y_pred))

print("AUC-ROC:", roc_auc_score(y_test, y_pred_proba))

print("F1 Score: ", f1_score(y_test, y_pred))

print("Precision:", precision_score(y_test, y_pred))

print("Recall: ", recall_score(y_test, y_pred))
```

```
Accuracy:  0.9984
AUC-ROC: 1.0
F1 Score:  0.9990950226244344
Precision: 0.9981916817359855
Recall:  1.0
```

**The Kernel crashed while executing code in the the current cell or a previous cell. Please review the code in the cell(s) to identify a possible cause of the failure. Click <a href='https://aka.ms/vscodeJupyterKernelCrash'>here</a> for more info. View Jupyter <a href='command:jupyter.viewOutput'>log</a> for further details.**

In [45]:
```
# Creats & Trains the Random Forest with 5 trees
rf = RandomForestClassifier(n_estimators=10, random_state=42, class_weight= {1:2, 0:1})
rf.fit(x_train_resampled, y_train_resampled)
```

Out[45]:
```
▾                    RandomForestClassifier

RandomForestClassifier(class_weight={0: 1, 1: 2}, n_estimators=10,
                       random_state=42)
```

In [46]:
```
y_pred = rf.predict(x_test)

y_pred_proba = rf.predict_proba(x_test)[:, 1]
```

In [47]:
```
# Prints out Relevant Metrics
print("Accuracy: ", accuracy_score(y_test, y_pred))

print("AUC-ROC:", roc_auc_score(y_test, y_pred_proba))

print("F1 Score: ", f1_score(y_test, y_pred))

print("Precision:", precision_score(y_test, y_pred))

print("Recall: ", recall_score(y_test, y_pred))
```

```
Accuracy:  0.9968
AUC-ROC: 0.999875918205281
F1 Score:  0.9981884057971014
Precision: 0.9981884057971014
Recall:  0.9981884057971014
```

In [48]:
```
# Creats & Trains the Random Forest with 5 trees
rf = RandomForestClassifier(n_estimators=20, random_state=42, class_weight= {1:2, 0:1})
rf.fit(x_train_resampled, y_train_resampled)
```

Out[48]:
```
▾                      RandomForestClassifier

RandomForestClassifier(class_weight={0: 1, 1: 2}, n_estimators=20,
                       random_state=42)
```

In [49]:
```
y_pred = rf.predict(x_test)

y_pred_proba = rf.predict_proba(x_test)[:, 1]
```

In [50]:
```
# Prints out Relevant Metrics
print("Accuracy: ", accuracy_score(y_test, y_pred))

print("AUC-ROC:", roc_auc_score(y_test, y_pred_proba))

print("F1 Score: ", f1_score(y_test, y_pred))

print("Precision:", precision_score(y_test, y_pred))

print("Recall: ", recall_score(y_test, y_pred))
```

```
Accuracy:  0.9984
AUC-ROC: 0.9999751836410562
F1 Score:  0.9990950226244344
Precision: 0.9981916817359855
Recall:  1.0
```

In [51]:
```
# Creats & Trains the Random Forest with 5 trees
rf = RandomForestClassifier(n_estimators=10, max_features = 5, random_state=42, class_weight= {1:2, 0:1})
rf.fit(x_train_resampled, y_train_resampled)
```

Out[51]:
```
▾                      RandomForestClassifier

RandomForestClassifier(class_weight={0: 1, 1: 2}, max_features=5,
                       n_estimators=10, random_state=42)
```

In [52]:
```
y_pred = rf.predict(x_test)

y_pred_proba = rf.predict_proba(x_test)[:, 1]
```

In [53]:
```
# Prints out Relevant Metrics
print("Accuracy: ", accuracy_score(y_test, y_pred))

print("AUC-ROC:", roc_auc_score(y_test, y_pred_proba))

print("F1 Score: ", f1_score(y_test, y_pred))

print("Precision:", precision_score(y_test, y_pred))

print("Recall: ", recall_score(y_test, y_pred))
```

```
Accuracy:  0.992
AUC-ROC: 0.9996773873337303
F1 Score:  0.9954504094631482
Precision: 1.0
Recall:  0.9909420289855072
```

In [54]:
```
# Creats & Trains the Random Forest with 5 trees
rf = RandomForestClassifier(n_estimators=10, max_features = 10, random_state=42, class_weight= {1:2, 0:1})
rf.fit(x_train_resampled, y_train_resampled)
```

Out[54]:
```
▾                    RandomForestClassifier

RandomForestClassifier(class_weight={0: 1, 1: 2}, max_features=10,
                       n_estimators=10, random_state=42)
```

In [55]:
```
y_pred = rf.predict(x_test)

y_pred_proba = rf.predict_proba(x_test)[:, 1]
```

In [56]:
```
# Prints out Relevant Metrics
print("Accuracy: ", accuracy_score(y_test, y_pred))

print("AUC-ROC:", roc_auc_score(y_test, y_pred_proba))

print("F1 Score: ", f1_score(y_test, y_pred))

print("Precision:", precision_score(y_test, y_pred))

print("Recall: ", recall_score(y_test, y_pred))
```

```
Accuracy:  0.9888
AUC-ROC: 0.9997022036926742
F1 Score:  0.9936305732484076
Precision: 0.9981718464351006
Recall:  0.9891304347826086
```

In [57]:
```
# Creats & Trains the Random Forest with 5 trees
rf = RandomForestClassifier(n_estimators=10, max_features = 20, random_state=42, class_weight= {1:2, 0:1})
rf.fit(x_train_resampled, y_train_resampled)
```

Out[57]:
```
▾                        RandomForestClassifier

RandomForestClassifier(class_weight={0: 1, 1: 2}, max_features=20,
                       n_estimators=10, random_state=42)
```

In [58]:
```
y_pred = rf.predict(x_test)

y_pred_proba = rf.predict_proba(x_test)[:, 1]
```

In [59]:
```
# Prints out Relevant Metrics
print("Accuracy: ", accuracy_score(y_test, y_pred))

print("AUC-ROC:", roc_auc_score(y_test, y_pred_proba))

print("F1 Score: ", f1_score(y_test, y_pred))

print("Precision:", precision_score(y_test, y_pred))

print("Recall: ", recall_score(y_test, y_pred))
```

```
Accuracy:  0.992
AUC-ROC: 0.9996773873337303
F1 Score:  0.9954586739327882
Precision: 0.9981785063752276
Recall:  0.9927536231884058
```

In [60]:
```
auc = roc_auc_score(y_test, y_pred_proba)
```

In [61]:
```
# Creats & Trains the Random Forest with 5 trees
rf = RandomForestClassifier(n_estimators=10, max_features = 40, random_state=42, class_weight= {1:2, 0:1})
rf.fit(x_train_resampled, y_train_resampled)
```

Out[61]:
```
RandomForestClassifier(class_weight={0: 1, 1: 2}, max_features=40,
                       n_estimators=10, random_state=42)
```

In [62]:
```
y_pred = rf.predict(x_test)

y_pred_proba = rf.predict_proba(x_test)[:, 1]
```

In [63]:
```
# Prints out Relevant Metrics
print("Accuracy: ", accuracy_score(y_test, y_pred))

print("AUC-ROC:", roc_auc_score(y_test, y_pred_proba))

print("F1 Score: ", f1_score(y_test, y_pred))

print("Precision:", precision_score(y_test, y_pred))

print("Recall: ", recall_score(y_test, y_pred))
```

```
Accuracy:  0.9968
AUC-ROC: 0.9999131427436967
F1 Score:  0.9981884057971014
Precision: 0.9981884057971014
Recall:  0.9981884057971014
```

In [64]:
```
auc = roc_auc_score(y_test, y_pred_proba)
```

In [65]:
```
# Creats & Trains the Random Forest with 5 trees
rf = RandomForestClassifier(n_estimators=10, max_features = 100, random_state=42, class_weight= {1:2, 0:1})
rf.fit(x_train_resampled, y_train_resampled)
```

Out[65]:
```
RandomForestClassifier(class_weight={0: 1, 1: 2}, max_features=100,
                       n_estimators=10, random_state=42)
```

In [66]:
```
y_pred = rf.predict(x_test)

y_pred_proba = rf.predict_proba(x_test)[:, 1]
```

In [67]:
```
# Prints out Relevant Metrics
print("Accuracy: ", accuracy_score(y_test, y_pred))
```

```
print("AUC-ROC:", roc_auc_score(y_test, y_pred_proba))

print("F1 Score: ", f1_score(y_test, y_pred))

print("Precision:", precision_score(y_test, y_pred))

print("Recall: ", recall_score(y_test, y_pred))
```

```
Accuracy:  0.9984
AUC-ROC: 1.0
F1 Score:  0.9990950226244344
Precision: 0.9981916817359855
Recall:  1.0
```