Assignment Report

Starsha Odelia
20311517
(Dated: November 5, 2023)

Three classifier models were developed using a provided dataset characterised by imbalanced class distributions and a notable amount of correlated features, and were employed to identify malicious activity and categorise the specific type of malicious behaviour. k-Nearest Neighbours and Decision Trees emerged as the best classifiers for the given task, with classification accuracies of 81% and 73%, respectively. The estimated prediction accuracy was determined based on the validation process, and was computed to be $(77.0 \pm 3.85)\%$.

I. METHODOLOGY

A. Data Preparation

This goal of this step is to ensure that the data used for model training and validation is of high quality. The provided training and testing datasets underwent preprocessing steps that included feature extraction, scaling, and label encoding.

The feature extraction step began by identifying and removing an irrelevant feature called `ackdat` as the feature `tcprtt` is described as the sum of the values `ackdat` and `synack`. Correlated features then need to be taken into account. Figure 4 shows the features that are highly correlated. In cases where feature pairs exhibit a correlation of 90% or higher, one of the features will be eliminated.

Scaling all features of the float and numeric data types was implemented to mitigate skewing, which has the potential to introduce bias into the model. Additionally, given the utilization of distance-based algorithms like k-nearest neighbours for classification, scaling ensures that no single feature unduly influences the outcome by preventing certain features from dominating others based on their scale.

Label encoding, a technique that transforms categorical variables into numerical values, was applied to all features of categorical data types to make them compatible with decision trees, which rely on numerical input. This method also conserves memory compared to one-hot encoding, which can be especially beneficial when dealing with large datasets.

To address the issue of class imbalance, oversampling and under-sampling techniques, specifically SMOTE (Synthetic Minority Over-sampling Technique) and Edited Nearest Neighbours, were applied. These techniques aim to balance the class distribution by either generating synthetic examples (SMOTE) or removing some instances (Edited Nearest Neighbours). As a consequence of these class balancing efforts, the dataset size expanded, creating a larger dataset compared to the original training set provided. This larger dataset with balanced class proportions helps improve the model's ability to learn from both classes and increase prediction accuracy.

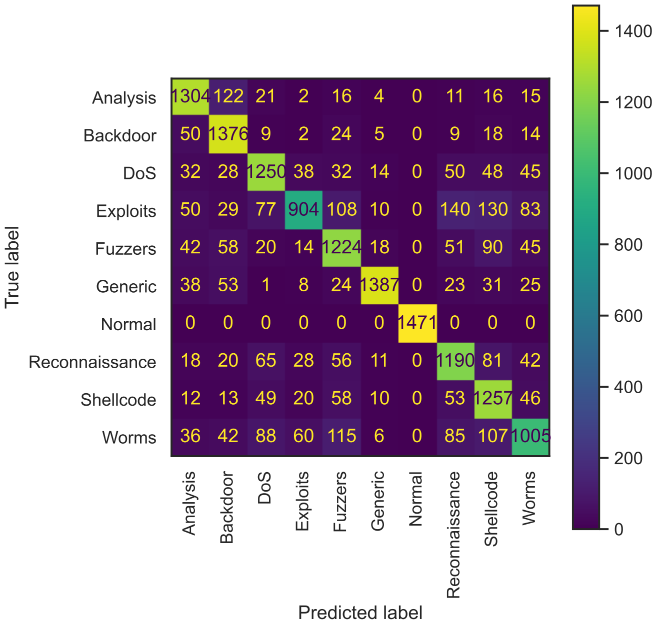
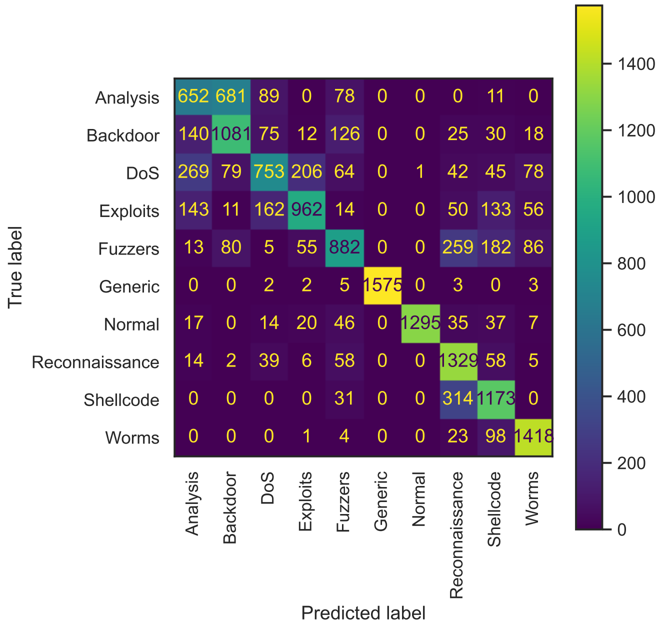


FIG. 1: Confusion matrix for k-NN

B. Data Classification

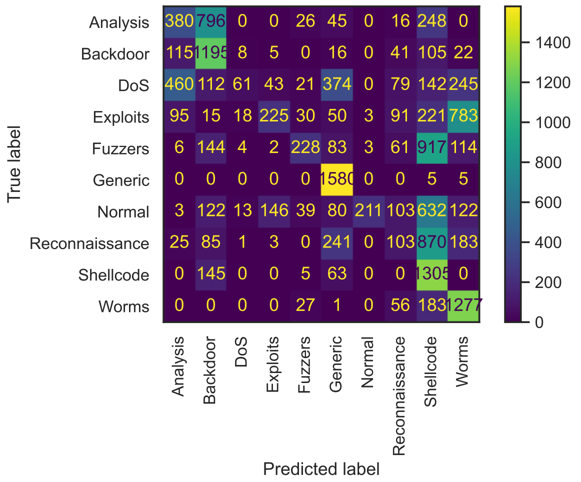
Three classifiers, namely k-NN (k-nearest neighbours), decision trees, and naive Bayes, were selected for this analysis. `GridSearchCV` was employed to optimize the hyperparameters and evaluate the performance of the k-NN and decision tree models. The evaluation process involved a ten-fold cross-validation approach, where the validation data in each fold constitutred 10% of the training dataset. This technique was chosen as it reduces the variance in the performance estimate and maximizes the utilization of the available data for training. Additionally, k-fold cross-validation prevents over-fitting by exposing the model to various subsets of the data during each iteration, ensuring a more reliable assessment of the model's performance. The second plot in Figure 5 visualises the k-fold cross-validation process.

In the case of k-NN and decision trees, the cross-validation process involved the search for optimal parameter values from a predefined set of parameters in a grid. This ensures that the chosen hyper-parameter



	Analysis	Backdoor	DoS	Exploits	Fuzzers	Generic	Normal	Reconnaissance	Shellcode	Worms
Analysis	652	681	89	0	78	0	0	0	11	0
Backdoor	140	1081	75	12	126	0	0	25	30	18
DoS	269	79	753	206	64	0	1	42	45	78
Exploits	143	11	162	962	14	0	0	50	133	56
Fuzzers	13	80	5	55	882	0	0	259	182	86
Generic	0	0	2	2	5	1575	0	3	0	3
Normal	17	0	14	20	46	0	1295	35	37	7
Reconnaissance	14	2	39	6	58	0	0	1329	58	5
Shellcode	0	0	0	0	31	0	0	314	1173	0
Worms	0	0	0	1	4	0	0	23	98	1418

FIG. 2: Confusion matrix for decision tree



	Analysis	Backdoor	DoS	Exploits	Fuzzers	Generic	Normal	Reconnaissance	Shellcode	Worms
Analysis	380	796	0	0	26	45	0	16	248	0
Backdoor	115	1195	8	5	0	16	0	41	105	22
DoS	460	112	61	43	21	374	0	79	142	245
Exploits	95	15	18	225	30	50	3	91	221	783
Fuzzers	6	144	4	2	228	83	3	61	917	114
Generic	0	0	0	0	0	1580	0	0	5	5
Normal	3	122	13	146	39	80	211	103	632	122
Reconnaissance	25	85	1	3	0	241	0	103	870	183
Shellcode	0	145	0	0	5	63	0	0	1305	0
Worms	0	0	0	0	27	1	0	56	183	1277

FIG. 3: Confusion matrix for naive Bayes

values yielded the most effective model configurations. Conversely, for the naive Bayes classifier, there are no adjustable parameters to fine-tune, and thus there was no need for an optimization process. Confusion matrices deliver a thorough breakdown of accurately and inaccurately classified samples, rendering it necessary for evaluating the classification accuracy of the models. Among the three classifier models, the k-NN and decision tree models demonstrated the highest validation scores. Figures 1, 2, and 3 show the confusion matrices that respectively correspond to k-NN, decision tree, and naive Bayes classifiers.

In the k-NN model, the optimal hyperparameters were determined to be using distance-based weights and settingi the number of nearest neighbours to 110. Under these settings, the classification method scored a weighted average precision and weighted average recall score of approximately 81% (Table II).

In the decision tree model, the utilization of the Gini index as the splitting criterion, along with a minimum requirement of 5000 samples to split an internal node, resulted in a weighted average precision and recall scores of approximately 72% (Table IV).

II. PREDICTION

By taking the mean of the accuracy scores, the estimated accuracy of the predictions is $(77.0 \pm 3.85)\%$. This result provides an indication of the model's expected performance when applied to new, unseen data.

TABLE I: Overview of the best performing classifiers.

	k-NN	Decision Tree	Mean
Accuracy	0.810	0.730	0.770
Precision	0.814	0.733	0.774
Recall	0.809	0.730	0.769
F1 Score	0.808	0.724	0.766

TABLE II: Classification report for k-NN.

	precision	recall	f1-score	support
Analysis	0.824273	0.863005	0.843194	1511.0
Backdoor	0.790350	0.913072	0.847291	1507.0
DoS	0.791139	0.813273	0.802053	1537.0
Exploits	0.840149	0.590464	0.693517	1531.0
Fuzzers	0.738684	0.783611	0.760485	1562.0
Generic	0.946758	0.872327	0.908020	1590.0
Normal	1.000000	1.000000	1.000000	1471.0
Reconnaissance	0.738213	0.787558	0.762088	1511.0
Shellcode	0.706974	0.828063	0.762743	1518.0
Worms	0.761364	0.650907	0.701816	1544.0
accuracy			0.81	
macro avg	0.813790	0.810228	0.808121	15282.0
weighted avg	0.813580	0.809318	0.807555	15282.0

TABLE III: Classification report for decision tree.

	precision	recall	f1-score	support
Analysis	0.522436	0.431502	0.472635	1511.0
Backdoor	0.558945	0.717319	0.628306	1507.0
DoS	0.661106	0.489915	0.562780	1537.0
Exploits	0.761076	0.628347	0.688372	1531.0
Fuzzers	0.674312	0.564661	0.614634	1562.0
Generic	1.000000	0.990566	0.995261	1590.0
Normal	0.999228	0.880354	0.936032	1471.0
Reconnaissance	0.638942	0.879550	0.740184	1511.0
Shellcode	0.663837	0.772727	0.714155	1518.0
Worms	0.848594	0.918394	0.882115	1544.0
accuracy			0.73	
macro avg	0.732848	0.727334	0.723447	15282.0
weighted avg	0.733515	0.727653	0.723977	15282.0

TABLE IV: Classification report for naive Bayes.

	precision	recall	f1-score	support
Analysis	0.350554	0.251489	0.292871	1511.0
Backdoor	0.457154	0.792966	0.579956	1507.0
DoS	0.580952	0.039688	0.074300	1537.0
Exploits	0.530660	0.146963	0.230179	1531.0
Fuzzers	0.606383	0.145967	0.235294	1562.0
Generic	0.623766	0.993711	0.766432	1590.0
Normal	0.972350	0.143440	0.250000	1471.0
Reconnaissance	0.187273	0.068167	0.099951	1511.0
Shellcode	0.281979	0.859684	0.424666	1518.0
Worms	0.464195	0.827073	0.594645	1544.0
accuracy			0.43	
macro avg	0.505527	0.426915	0.354830	15282.0
weighted avg	0.505235	0.429590	0.356683	15282.0

[1] Batista, Gustavo E. A. P. A. and Prati, Ronaldo C. and Monard, Maria Carolina, *Study of the Behavior of Several Methods for Balancing Machine Learning Training Data*, 6, Association for Computing Machinery, New York (2004).

Appendix A: Figures

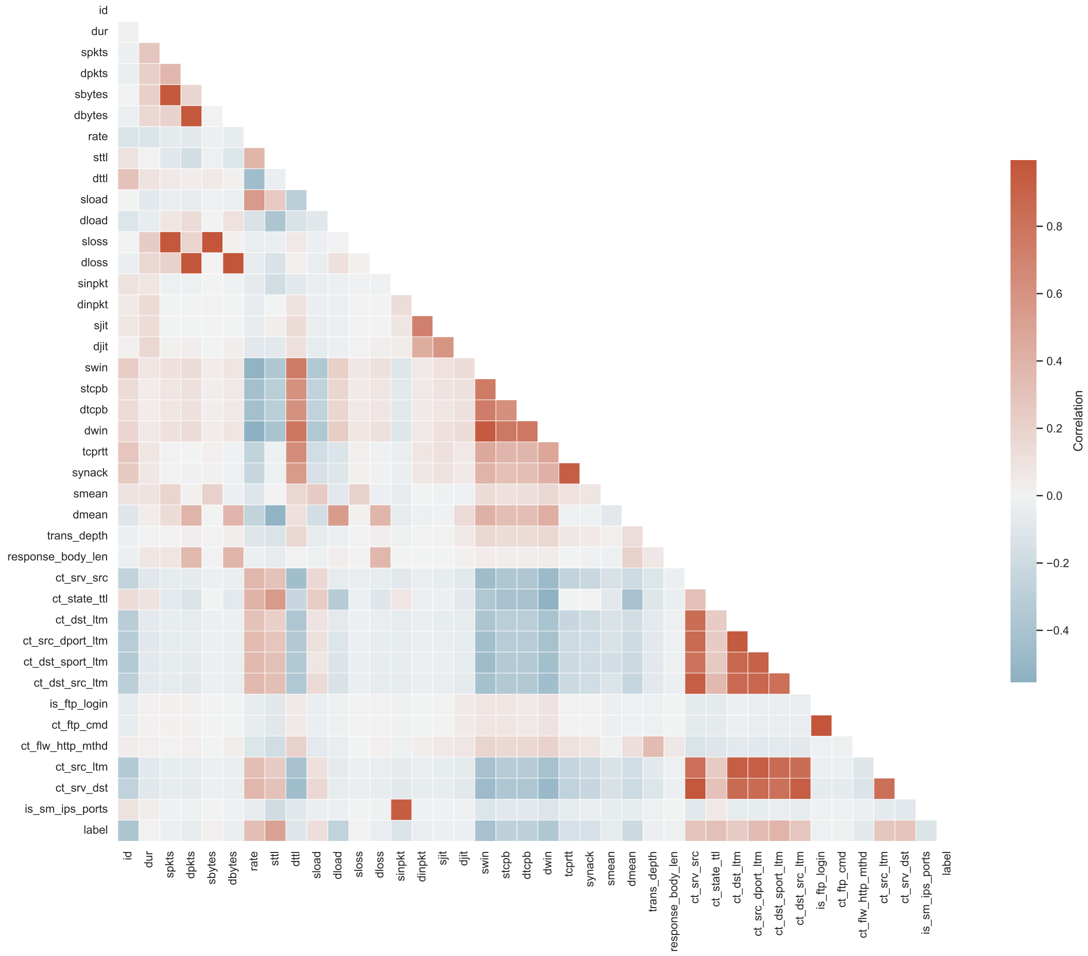


FIG. 4: Correlation of features in the dataset provided