Anomaly Detection:
Air Pressure System Failure in Scania Trucks

Chirav Dave, Dhrumil Shah, Nagarjun Chinnari,
Nisarg Trivedi, Rama Kumar Kana Sundara, Sushant Trivedi
[cdave1, dpshah, nchinnar, ntrived3, ramakuma, strived6] @asu.edu

ABSTRACT

Our reference paper models a prediction system that can detect failure of a specific component in the Air Pressure System in the heavy Scania trucks. The problem is a classification based problem with the goal to determine if there is a component failure or not. Our system is a hybrid of five different types of models. Accuracy for each of the model is calculated separately. The failure of a component is associated with a cost as well. So a higher cost is incurred for every wrong classification and vice-versa. Each model gives out a cost and then weight is associated with a model according to the quality of its classification. Finally, classification is done based on the weighted polling. Various types of data cleaning methods are used to deal with the missing values in the dataset like filling out with median values, selecting the best features based on some evaluating function and many more. Our results show that the random forest model is the best one and hence has the highest weight as it gives the minimum cost of approximately 10,000. Although further tweaking of the models can be done to improve the performance and accuracy.

1. INTRODUCTION

In this paper, we are trying to model a prediction system for detecting a component failure in the Air Pressure System (APS) in the heavy Scania trucks. The prediction will tell us if there is an imminent failure in the heavy trucks. The data were thus collected from the APS system that is used in these day-to-day trucks. The APS, in general, is a system which generates pressurized air to use in different component functions in the trucks such as braking, gears, suspension, etc. A positive class is given to component failure that belongs to the APS system and a negative class is given to component failure related to anything else. Other than just predicting the failure of the component, we are also trying to optimize the cost of a failure. A cost of 10 is given to a correct prediction i.e. predicting a failure of APS component and a cost of 500 is given to a false negative i.e. to failure that was not predicted by our model. Thus, penalty minimizing is also one

of our main goals. The problem can be said to be a classification problem. We have used 5 different models, Logistic regression, Support Vector Machine, Random forest, Gaussian model, Random model and K-Nearest Neighbour. All of these models use a different type of data cleaning techniques and feature selection. In the end, according to the cost predicted by each model, a weight is assigned to each one of them and a final decision is taken based on this weighted polling. Our experiments show that the best classifier is the Random Forest with the cost of around 10,000. Also, the accuracy of all the other models came out to be approximately 95%.

2. PROBLEM DESCRIPTION

A component failure in a truck can be as small as a simple horn not working to the failing of the brakes while the truck is making a descent. As we can see these problems can be as little as a bother or can prove to be fatal if not treated properly. Apart from the risk involved in a component failure, there is the price that comes with the repairing of that and as components as important as the APS go, they can prove to be really costly. Here we are trying to predict the failure of a component related to APS. Not just predicting, but predicting it correctly. A cost is associated with the prediction and we are trying to bring that cost down as well. A false negative i.e. predicting that a failed component is good, proves to be the most costly affair. The cost associated with a False negative (FN) is 500 and for the False positive (FP) is 10. Thus, while calculating the final cost we have an equation,

$$\text{Total Cost} = 500 * FN + 10 * FP \qquad \textbf{[Formula (1)]}$$

FN -> False Negative, FP-> False Positive

3. DATASET DESCRIPTION

The dataset consists of data collected from heavy Scania trucks in everyday usage. The system which we are considering is an Air Pressure system (APS) of these trucks. Every instance of the dataset is classified into negative and positive classes. The positive class denotes a failure in component related to the Air Pressure system and a negative class denotes a non-APS failure. The training dataset consists of 60,000 instances and 171 attributes and the missing values have been denoted by 'na'. The dataset is a highly imbalanced dataset with only 1000 positive class samples out of 60,000 samples. Thus, FN has greater cost as compared to FP and as such is undesirable in our predictions.

The test dataset contains 16,000 samples with 375 samples of positive class.

4. RELATED WORK

The IDA 2016 Industrial Challenge consisted of using machine learning in order to predict whether a specific component of the Air Pressure System of a vehicle faces imminent failure or not.

4.1 Classification Algorithms

This problem was considered as a classification problem and many well-known classification algorithms such as Logistic Regression, k-nearest neighbours, Support Vector Machine (SVM), Decision Trees and Random Forests were applied to it. The experimental results showed that the best classifier was cost-wise 92.56 % better than a straightforward solution where a random classification was performed.

It could be observed from the evaluated classifiers that the cost was highest using Random classifier. With the data set provided, faced two main challenges while dealing with classifying new instances as positive or negative. Firstly, the dataset provided was highly imbalanced. Secondly, the missing data rate in the dataset was very high. The performance of all the classification algorithms is presented in Table 1 and resorted to the implementations provided by the Scikit-learn library. With this library, class-specific weights in the loss function for SVM and LR classifiers was used, which allowed resolving the first issue. In the k-NN and RF classifiers, the likelihood of an instance belonging to each class was calculated. These two algorithms were not able to classify positive instances properly, which greatly increased the misclassification cost. That problem was then handled by increasing the threshold value.

CLASSIFIER	COST
Support Vector Machine	27109
Logistic Regression	21500
kNN	16781
Random Forest	12781
Random	171875

Table 1: Classifiers and their evaluated cost

4.2 Dealing with Missing Data.

In order to deal with the missing data, below two methods were used:

- Mean imputation
- Soft-Impute.

Soft-Impute was found to be an efficient algorithm for large-scale matrix completion.

5. OUR METHODOLOGY : VARIOUS MODEL EVALUATION

5.1 Random Forest:

Random forests or Random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

Since there were a lot of missing values in the dataset, we handled them by replacing with the median of their respective column/feature values. We used Chi-squared score to select the best features. Also, since the dataset was very imbalanced, we maintained an appropriate proportion of the positive class and negative class data. We randomly took samples of size 1000 from the negative class data and all the positive class data at every iteration. We implemented Random Forest with this constraint and performed classification with this model. We saw that accuracy was getting better with reduction in the number of features with the same criteria for selecting a column. We observed that with 110 features and taking 1000 negative samples and all positive samples, random forest worked best on the given data set under the given constraint.

Our random forest model classified only 6 as false negative and 769 as false positive and it gave a classification cost of 10690 (following formula 1).

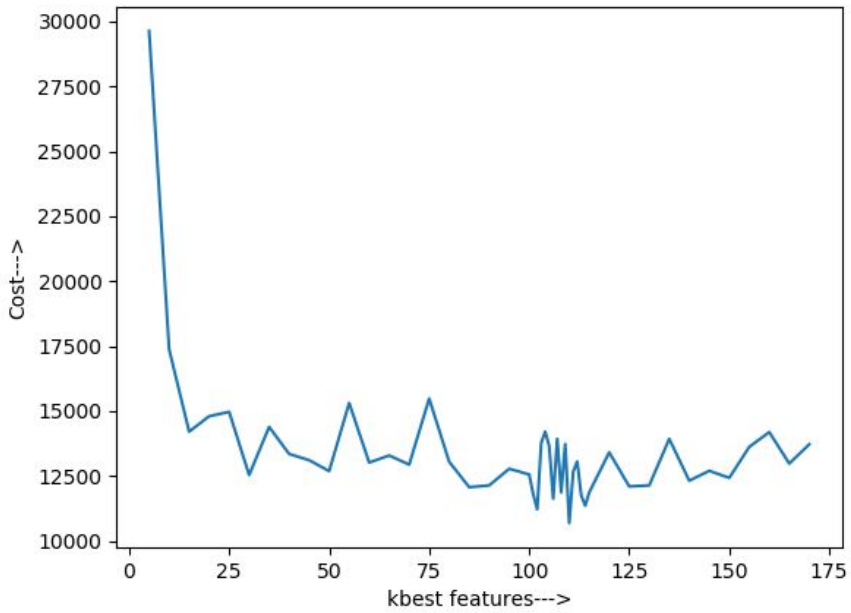

Figure 1: Cost vs K-best features

5.2 Gaussian Based Model:

In probability theory, the normal (or Gaussian) distribution is a very common continuous probability distribution. Normal distributions are important in statistics and are often used in the natural and social sciences to represent real-valued random variables whose distributions are not known. The normal distribution is useful because of the central limit theorem. In its most general form, under some conditions (which include finite variance), it states that averages of samples of observations of random variables independently drawn from independent distributions converge in distribution to the normal, that is, they become normally distributed when the number of observations is sufficiently large.

Similar to Random Forest, we handled the missing values by replacing them with the median of their respective column/feature values. Also, we maintained an appropriate proportion of the positive class and negative class data. We randomly took samples of size 800 from the negative class data and all the positive class data at every iteration. Next, we reduced the dimensionality of the selected data to top 35 features based on their Chi-squared scores. Finally, we created 35 Gaussian distribution models(one for every feature) and then took an offset value equal to 0.001 to determine anomalies. Our prediction then considered a threshold of 10 features to predict the anomaly. We saw that the model incurred a total cost of 14820 (following formula 1) with 9 as false negative and 1032 as false positive.

5.3 Support Vector Machine:

SVM is a discriminative classifier which is defined by a separating hyperplane. In the SVM algorithm, we are looking to maximize the margin between the data points and the hyperplane. We have used Synthetic Oversampling to manipulate the given data before feeding it to the classifier. We have used the sklearn library to reduce the features to a lower dimension of 85 by implementing PCA. Instead of using a generic SVM we tuned its parameters to get better results. Using GridSearchCV function of the sklearn library, we tuned the parameters ‘C’, ’gamma’ and ‘kernel’. SVM have different types of kernels: linear, polynomial, radial basis and sigmoidal functions. While using radial basis function or ‘rbf’, a third tuning parameter must be considered, gamma. If this parameter is large, the variance is low implying the support vector does not have widespread influence. Different kernels including ‘rbf’ and ‘linear’ were used for our model development.

The total misclassification cost using SVM is 16350 with 9 as false negative and 1185 as false positive.

5.4 Logistic Regression:

Logistic Regression is a binary probabilistic classifier that estimates the probability of a samples’ classes and classifies to the class that has a maximum probability estimate over a cutoff value (which is usually 0.5). Logistic Regression is a linear classifier and thus, it doesn’t perform well on non-linear dataset who’s classes cannot be separated by a straight line.

In our reference paper's implementation, the LR algorithm has been directly implemented while modifying the class-specific weights in the loss function to compensate for the imbalanced dataset. The weight of each class was set to be inversely proportional to the fraction of cases of the corresponding class. The findings of this paper were that LR has a misclassification cost of 61470 with FP: 2.36%, FN: 9.5%. In this section, we will elaborate on the techniques tested.

For handling missing values, we attempted to replace them with mean, median of the feature values and with zero values. Feature reduction was done using the Principal Component Analysis (PCA) technique and Recursive Feature Elimination (RFE). Furthermore, we also normalised the dataset. To deal with the imbalance of the classes in the dataset we attempted to use various sampling based technique such as Synthetic Minority Oversampling Technique (SMOTE) and Undersampling. We attempted to analyse different combinations of the above-mentioned techniques and have tabulated our result in Table 2.

SMOTE is an oversampling technique which swells the dataset size to 83,536 samples. This technique analyses the feature values for the imbalanced class and synthetically generates data samples with feature values similar to the pre-existing class data samples. While Recursive Feature Elimination (RFE) recursively models with the given dataset and eliminates features with lowest weights until we have brought down the feature count to a pre-decided count. This doesn't deal with the imbalance in the dataset.

We also attempted to vary the threshold cutoff for the Logistic Regression Classifier from 0.1 to 0.9 and observed favourable behaviour upon this change but the behaviour was highly dependent on how we dealt with the missing values (replace with 0, mean or median). An analysis is present in the Figure 2.

KEEPING ZERO	MEAN	MEDIAN
Cutoff Prob, FP, FN, Cost	Cutoff Prob, FP, FN, Cost	Cutoff Prob, FP, FN, Cost
0.1 FP: 5 FN: 975 Cost: 487550	0.1 FP: 27 FN: 559 Cost: 279770	0.1 FP: 0 FN: 1000 Cost: 500000
0.2 FP: 25 FN: 957 Cost: 478750	0.2 FP: 45 FN: 504 Cost: 252450	0.2 FP: 0 FN: 999 Cost: 499500
0.3 FP: 64 FN: 929 Cost: 465140	0.3 FP: 56 FN: 454 Cost: 227560	0.3 FP: 0 FN: 994 Cost: 497000
0.4 FP: 102 FN: 889 Cost: 445520	0.4 FP: 79 FN: 417 Cost: 209290	0.4 FP: 3 FN: 987 Cost: 493530
0.5 FP: 407 FN: 843 Cost: 425570	0.5 FP: 114 FN: 368 Cost: 185140	0.5 FP: 23 FN: 970 Cost: 485230
0.6 FP: 22226 FN: 781 Cost: 612760	0.6 FP: 156 FN: 336 Cost: 169560	0.6 FP: 21399 FN: 899 Cost: 663490
0.7 FP: 34746 FN: 706 Cost: 700460	0.7 FP: 215 FN: 303 Cost: 153650	0.7 FP: 37441 FN: 806 Cost: 777410
0.8 FP: 43225 FN: 599 Cost: 731750	0.8 FP: 311 FN: 249 Cost: 127610	0.8 FP: 43666 FN: 713 Cost: 793160
0.9 FP: 46703 FN: 456 Cost: 695030	0.9 FP: 531 FN: 180 Cost: 95310	0.9 FP: 47147 FN: 518 Cost: 730470

Figure 2: LR Cutoff Threshold Cost Analysis

METHOD	ACCURACY	CONFUSION MATRIX	COST
LR + Missing Mean Values replaced with Mean values	98.88 %	[15575, 50, 129, 246]	65000

LR + Dataset Normalized + Missing Values replaced with Mean values	98.6025 %	[15608, 17, 206, 169]	103170
LR + Dataset Normalized + Missing Values replaced with Mean values + PCA	97.581 % (90 % Info kept)	[15607, 18, 369, 6]	184680
	97.54 % (95 % Info kept)	[15598, 27, 367, 8]	183770
	97.26 % (98 % Info kept)	[15550, 75, 362, 13]	181750
LR+ Dataset Normalised + Missing values replaced with Mean + PCA + Undersampling	75.706 % (95 % Info kept)	[12071, 3554, 333, 42]	202040
LR+Missing Values replaced with Mean + Dataset Normalised + PCA + SMOTE	81.65 % (95 % Info kept)	[12865, 2760, 175, 200]	115100
LR+Dataset Normalised + Missing Values replaced with Mean + SMOTE	97.98 %	[15370, 25, 68, 307]	34250
LR+Dataset Normalised + Missing Values replaced with Mean + SMOTE+RFE	97.96 % (165 features)	[15362, 263, 64, 311]	34630
	97.99 % (100 features)	[15366, 259 62, 313]	33590
	97.84 % (80 features)	[15335, 290, 55, 320]	30400
LR+Dataset Normalised + Missing Values replaced with Mean+SMOTE+RFE	2.53 %	[44, 15581, 14, 361]	162810

Table 2: LR Classifier Methods and Evaluated Costs

Having analysed the above performances we found that we get the best cost with (LR+Dataset Normalised+Missing Values replaced with Mean+SMOTE+RFE) with 80 features and cost as 30,400.

5.5 K-Nearest Neighbor:

KNN is a supervised machine learning algorithm which estimates how likely a point belongs to a certain group based on what its nearest neighbours are. In a traditional KNN algorithm, a positive number K is taken as input and an unseen instance. This new instance is classified by finding the k- nearest neighbours (using distance as a metric) and then sees the most common class among them.

This algorithm gives good results if the dataset is balanced which unfortunately is not the case with our dataset. We have used a slight variation to the traditional KNN algorithm in which after finding the 150 nearest neighbours, the unseen instance is assigned to be positive if there are at least two positive class instances near it otherwise negative class is assigned.

Since there are a lot of outliers in the dataset, we calculated z-scores for each column/feature and removed the column if more than ten percent of it are outliers. After removing the outliers the missing values are dealt by imputing them with the mean of a column/feature. We see that this model incurs a total cost of 14670 with 8 as false negative and 1067 as false positive. Below is our model flow:

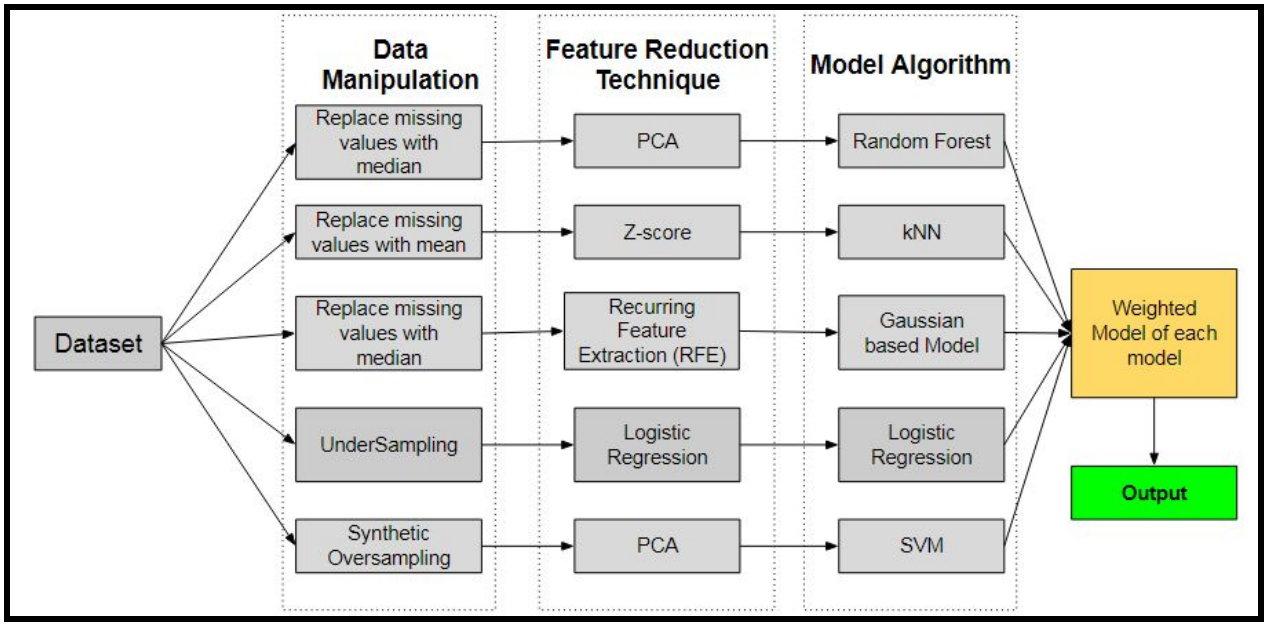

Figure 3: Our Model Flow

5.6 Weighted Polling:

Previously, every model was calculated independently and optimized to find the best possible result in terms of cost. For the final phase of the concept flow, all the models were merged together for doing prediction. As there are 5 models, the final prediction was done using a polling system. But simpler polling would have given equal weight to each model. Equal weight would mean that the efficiency in the final prediction is compromised due to models which are less efficient. There is a cost associated with each model which can be used to decide weights. Less cost would mean that the model is more efficient. So the weight of the model would be inversely proportional to the cost of the model. That way, random forest resulted with the highest weight and Logistic regression with the lowest weight. Weighted polling was performed as follows:

Every model predicted the output independently. As here the output is binary (0 or 1), the output is multiplied with the weight of that model. In the end, summation on each model prediction with their weight was performed. The summation was then divided with the total sum of weights. If the average was greater or equal to 0.5 than the output was predicted as 1 else predicted as 0.

6. RESULTS

By performing weighted polling, the final prediction had only 5 False Negative and 758 False Positives. Following is the confusion matrix for the final prediction:

	ACTUAL POSITIVE	**ACTUAL NEGATIVE**
PREDICTED POSITIVE	14867	758
PREDICTED NEGATIVE	5	370

Table 3: Confusion matrix for the weighted model

Here accuracy is approx 95.2% but the classification cost calculated with formula (1) is 10080. Below is the aggregated result of our system:

CLASSIFIER	**COST**
SVM	16350
Logistic Regression	30400
kNN	14670
Random Forest	10690

Gaussian Model	14820
Final	10080

Table 4: Classifiers and their costs

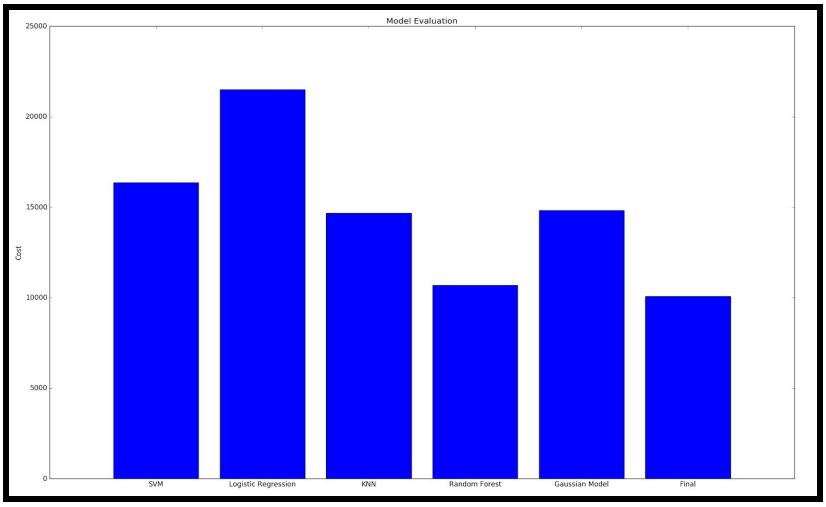

Figure 4: Model Cost

7. CONCLUSION

Given problem statement is not only about classification accuracy but also about optimizing classification accuracy using given constraints. Given dataset has many features with abundant missing values. We implemented various methods to deal with missing values like predicting missing values based on mean or median. As there are 171 features in total, feature reduction is necessary to implement to reduce computational complexity. We implemented Z score analysis, recursive feature elimination, chi square methods, Principal Component Analysis to reduce the dimension of the features. We found that feature selection techniques and missing data has relation among themselves. We used that relation and mapped procedures among each other to replace missing data to select features. Data given is skewed in manner so traditional algorithms are not useful for optimizing the cost for given constraints in problem statement. So, we implemented various algorithms by tweaking according to the data and given constraints. We again found that feature selection techniques and model implementation should also be mapped to get better accuracy. This leads to independently optimized classification cost for each model. As each model had some classification cost with them and final goal was to optimize the classification cost, weighted polling was used to get better accuracy i.e. lower the classification cost, higher the weight to that prediction. That gave us good accuracy as well a highly optimized cost in overall. In the given problem, top 3 lowest classification which has been achieved till date is: 1) 9920 2) 10900 3) 11480[5]. With our work, we achieved classification cost of 10800.

8. GROUP INFORMATION AND CONTRIBUTION

Name	ASU ID	Email Address	Contribution	Contribution(%)
Chirav Dave	1212637307	cdave1@asu.edu	Gaussian Based Model	16.67
Dhrumil Shah	1213096220	dpshah8@asu.edu	Random Forest	16.67
Nagarjun Chinnari	1213287788	nchinnar@asu.edu	KNN, Z-scores	16.67
Nisarg Trivedi	1213314867	ntrived3@asu.edu	Pre-processing	16.67
Rama Kumar Kana Sundara	1213347614	ramakuma@asu.edu	SVM & PCA	16.67
Sushant Trivedi	1213366971	strived6@asu.edu	Logistic Regression	16.67

9. REFERENCES

1. Costa C.F., Nascimento M.A. (2016) IDA 2016 Industrial Challenge: Using Machine Learning for Predicting Failures. In: BostrÃ¶m H., Knobbe A., Soares C., Papapetrou P. (eds) Advances in Intelligent Data Analysis XV. IDA 2016. Lecture Notes in Computer Science, vol 9897.Springer, Cham
2. Gondek C., Hafner D., Sampson O.R. (2016) Prediction of Failures in the Air Pressure System of Scania Trucks Using a Random Forest and Feature Engineering. In: BostrÃ¶m H., Knobbe A., Soares C., Papapetrou P. (eds) Advances in Intelligent Data Analysis XV. IDA 2016. Lecture Notes in Computer Science, vol 9897. Springer, Cham
3. Ozan E.C., Riabchenko E., Kiranyaz S., Gabbouj M. (2016) An Optimized k-NN Approach for Classification on Imbalanced Datasets with Missing Data. In: BostrÃ¶m H., Knobbe A., Soares C., Papapetrou P. (eds) Advances in Intelligent Data Analysis XV. IDA 2016. Lecture Notes in Computer Science, vol 9897. Springer, Cham
4. Towards an Understanding of the Misclassification Rates of Machine Learning-based Malware Detection Systems Nada Alruhaily, Behzad Bordbar and Tom Chothia School of Computer Science, University of Birmingham, Birmingham, U.K. {N.M.Alruhaily, bxb, T.P.Chothia}@cs.bham.ac.uk