Machine Learning in Fuel Consumption II: An Anomaly Detection Approach in Power Generation Plants

J. Mulongo[a,b], M. Garuti[a,e], M. Atemkeng[f], T. Ansah-Narh[c,d], Rockefeller[g], G.M. Nguegnang[a,h,*]

[a] *Mathematical Sciences (AIMS-Cameroon)*
[b] *Maseno University*
[c] *Centre for Radio Astronomy & Astrophysics, Ghana Space Science and Technology Institute, Ghana Atomic Energy Commission, P. O. Box LG 80, Legon - Accra, Ghana*
[d] *Space & Atmospheric Research Group, Ghana Space Science and Technology Institute, Ghana Atomic Energy Commission, P. O. Box LG 80, Legon - Accra, Ghana*
[e] *University of Padova*
[f] *Rhodes Centre for Radio Astronomy Techniques & Technologies(RATT)*
[g] *Mathematical Sciences (AIMS-Senegal)*
[h] *University of Yaounde 1*

Abstract

High fuel consumption is experienced in telecom based stations. In this paper, we show the use of supervised machine learning classification based techniques in detecting anomalies associated with the fuel consumed dataset from the base station using the generator as a source of power. We made use of classification based techniques to train the dataset, these include support vector machines (SVM), k-nearest neighbor (KNN), logistic regression (LR), and multilayer perceptron (MLP). The aim of the study is to detect anomalies in the recorded data by learning the pattern of the fuel consumed dataset and compare the performance of the four classification techniques on the unseen dataset. We evaluate models performance using K-fold cross validation and training test split techniques. Performance metrics such as area under Receiver Operating Characteristic curve (ROC), the area under the precision-recall curve, accuracy, F1-measure, and so on were used to evaluate the fitted models. Comparative study of the classifier is done for model selection, and the result of this study shows that MLP has the best performance in the evaluation measured used with a score of 0.96 in both K-fold cross validation and train test split techniques. In the area under the ROC and precision-recall curve, the model attained a score of 1.00 and 0.98 respectively.

Keywords: Anomalies, Multilayer Perceptron, K-Nearest Neighbor, Support Vector Machines, Logistic Regression, Machine Learning, Classification

1. Introduction

The expansion of mobile services of the telecom industries has resulted in the installation of cell towers to diverse parts of the world. In developing countries like Cameroon, the supply of the power grid is irregular and the network companies find it difficult to manage their base stations particularly, their data centers and other IT equipment. Due to this, the management of the base stations uses diesel generators, solar panels, batteries and another secondary source of power as a backup plan to keep running their systems. However, these other power sources have also generated other problem, specifically, fuel pilferage from those who have access to fuel distribution in the various base stations and also, from other people in the vicinity and high fuel consumption as a result of the generator malfunctioning.

With an objective to provide base station management service such as refueling and site maintenance, TeleInfra[1] company in Cameroon faces the problem of electricity shortage hence the company had a generator as an alternative source of power. The company works with an objective is to provide management services such as site maintenance and refueling of generators in the base stations. Like any other power dependence company in Africa, the company faces the problem of electricity shortages and hence, chooses to have a generator as a main back up plan. The main goal of this paper is to use machine learning (ML) techniques to learn the pattern of the fuel consumed by generator power plants in the TeleInfra[2] company in order to observe some outliers produced by the system. The dataset used in this paper is obtained from base stations under supervision of TeleInfra company. It contains attributes of fuel consumed by the generators and other informations such as maintenance details.

Anomaly in a data is considered as an observation which do not conform to the expected standard in the data. Meanwhile,

*Corresponding author
Email addresses: `jecinta.mulongo@aims-cameroon.org` (J. Mulongo), `marco@aims-cameroon.org` (M. Garuti), `m.atemkeng@gmail.com` (M. Atemkeng), `t.narh@gaecgh.org` (T. Ansah-Narh), `rockefeller@aims-senegal.org` (Rockefeller), `maxime.nguegnang@aims-cameroon.org` (G.M. Nguegnang)

[1] `http://www.art.cm/en/node/3111`
[2] `http://www.art.cm/en/node/3111`

recent works in anomaly detection such as fraud detection analysis (Zanin et al., 2017; Chouiekh and Haj, 2018), thermal power plant (Banjanovic-Mehmedovic et al., 2017), medical image analysis (Taboada-Crispi et al., 2009), etc., have shown how well ML techniques can be used to address these challenges. From the various anomalies detection research, the algorithms used can outperform each other based on the type of data and the underlying assumptions employed. From the literature, most of the competitive ML techniques used in classification task like in our case, consist of SVM, KNN, LR, and MLP. In this study, we do a comparative analysis of the performance of these classifiers to determine the best classifier that can accurately detect the anomalies in the fuel consumption dataset. SVM aims in generating an optimal hyperplane that maximizes the margin between two class, that is, in the case linearly separable binary classification whereas KNN stores the training data and predict the test instance based on distance measure and the majority votes from the training sample. LR uses probabilities to predict the chance of a sample to belong in a certain class. MLP learns the complexity of data and optimizes the weights to minimize classification error. A more detailed explanation of these classifiers is given is Section 3.

The rest of this paper is organized as follows: Section 2 describe the dataset, feature selection and descriptive findings from the data. Section 3 we describe machine learning methods used in the study. Section 4 we conclude the results and finding from the study discussed with comparative analysis performance of the methods used.

2. EXPLORATORY DATA ANALYSIS

2.1. Dataset

The dataset is collected from the base stations powered mainly by the generator. It has 32 features both numerical and categorical data with a sample size of 5905 inputs of the fuel consumed within September 2017 to September 2018. The variables used to fit the models were rescaled to ensure all the features have the same treatment. The dataset contains details of fuel consumed by generator and maintenance report made on the generator. This information recorded includes the working hours of the generator, the rate of consumption, fuel consumed, the quantity of fuel added in the generator and so on.

The important features used to fit the models were selected using the Gradient Boosting Classifier (GBC) to reduce the number of features that do not impact the model performance. Compared to other feature selection technique such as random forest, GBC has been preferred to fit feature importance due to its ability to minimize bias and overfitting. This has been achieved in gradient boosting at error correction which is done based at every stage hence optimizing the objective function. From Figure 1, the running time of the generator, which is the number of hours the generator is working has a great influence on the output and as from Figure 5 how long the generator is working per day has a positive correlation with fuel consumed. Daily fuel consumed and consumption_HIS variables gives the fuel consumed per day and fuel consumed per period before the next refueling is done respectively. Other features such as generator capacity, maximum fuel the generator can take in one day, the rate of consumption and so on, have less information about the classification output class. The variables considered to fit the model included variables with at least 20% relative importance. Figure 1 shows feature importance fitted using Gradient Boosting classifier.

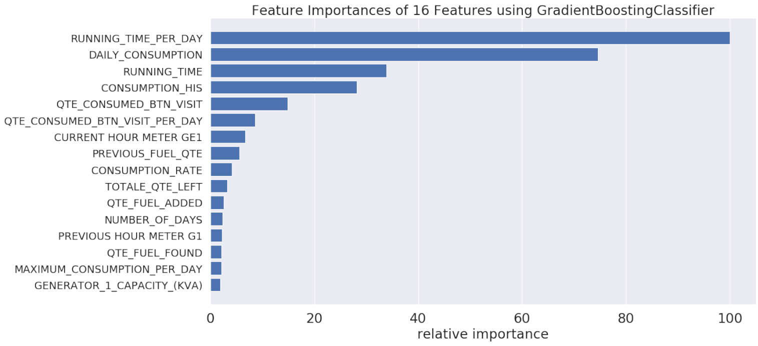


Figure 1: Feature importance ranking fitted using Gradient Boosting classifier

From a total sample of 5905, 35.11% of the sample were labeled as anomaly class and the rest as a normal class. The classification classes labeling was done based on the anomalies observed in the dataset. Anomalies in the data observed include outliers, wrong entry on the number of hours the generator was working in a day, consumption per day exceeding maximum consumption the generator can take in a day and generator fuel reducing when the generator was not working. All these observations was considered to label the classification class as either normal consumption or anomaly.

Class imbalance is a problem that highly affects classification base task. This is where the number of one class is significantly large compared to the other class. The effects of class imbalance are seen in the performance measure such as accuracy whose computation depends on both classes i.e., normal and anomaly class Awoyemi et al. (2017). A corrective measure of class imbalance have been proposed, that is, the use of sampling method Brownlee (2016). Sampling method can be classified as either undersampling or oversampling. Oversampling methods involves duplicating minority class to add more inputs so as to have equal sample size as to that class with the majority sample. On the other side, undersampling involves reducing the majority of data to have the same sample size as a minority.

2.2. Descriptive Analysis

Gaussian distribution of fuel consumed in Figure 2 shows the probability frequency distribution of the random variables. The fuel consumed by generator has asymmetry normal distribution with mean 261.80 and standard deviation of 175.06. Large variance is as a result of dispersion and the skewness of the fuel consumed data. Classifiers assumes the underlying distribution of the dataset hence the distribution has a great impact on the performance of classifiers Japkowicz and Shah (2011). Figure 2 shows the kernel density estimation and normal probability plot for the fuel consumed dataset.

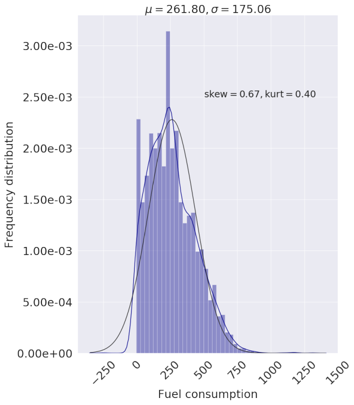
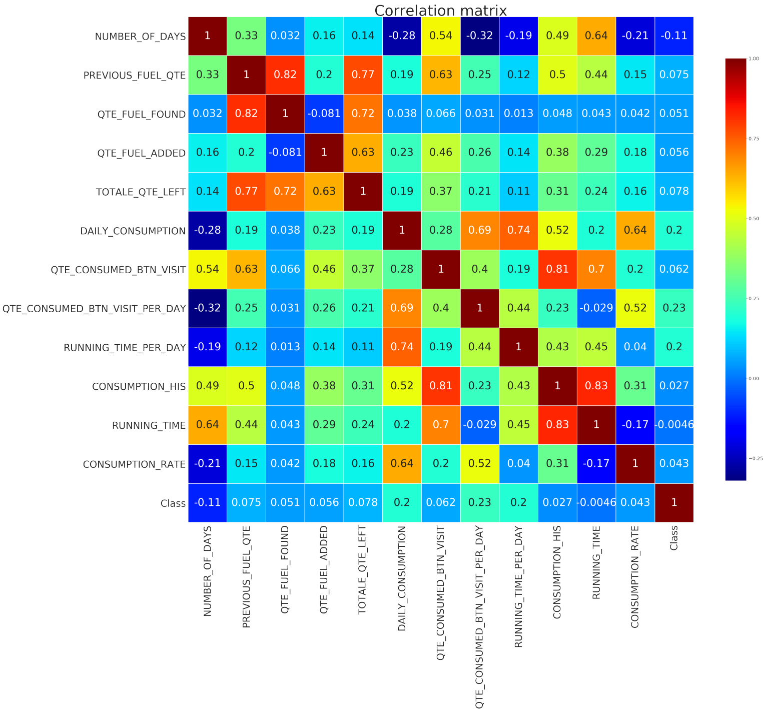


Figure 2: Average distribution of the fuel consumed

From the work done by G.M. Nguegnang (2018) on prediction of fuel consumption on the same dataset with all anomalies put into consideration, the mean of the fuel consumed within the period is 289.04 litters with a standard deviation of 160.09. The mean improvement is a result of the zero consumption and fuel consumption with anomalies not taken into consideration. The fuel consumed is positively skewed with a skew of 0.40. From Figure 2, the Gaussian distribution plot show that the dataset has outliers, as a result of this, the normal distribution curve deviating from the mean. From Figure 2, zero consumption of fuel consumed entries in the dataset causes the normal curve to start above the normal line. Due to this, the zero consumption samples were compared with the difference between what was left in the generator previously visit and what was found in the generator to make sure that consumption was actually zero fuel. As a result of the outliers, the normal distribution curve of the fuel consumed is shifted to the left, this contributes to the skew distribution of the fuel consumed.

Figure 3 displays a correlation matrix between numerical variables in the dataset. Correlation is a normalized covariance with its values ranges from -1 to 1. The matrix measures a linear relationship between variables with -1 indicates two variables have a strong negative relationship, that is, as one variable increases the other one decreases and 1 indicates strong positive relation, that is, an increase in one variable results to increase in the other one. The diagonal values indicate the correlation of a variable with itself. From Figure 3, fuel consumed have strong positive relation of 0.83 with the number of hours the generator. This is explained as fuel consumed by the generator increases depending on how long the generator is working. The quantity of fuel added and the fuel consumed has a positive correlation coefficient of 0.14. The rate at which the generator consumes fuel has positive correlation of 0.31 with the consumption. The classification class which is the output variable has also positive correction of 0.027 with fuel consumed. A shortcoming of the correlation matrix is that, variables with non-linear relationship the correlation matrix might not detect.



	NUMBER_OF_DAYS	PREVIOUS_FUEL_QTE	QTE_FUEL_FOUND	QTE_FUEL_ADDED	TOTALE_QTE_LEFT	DAILY_CONSUMPTION	QTE_CONSUMED_BTN_VISIT	QTE_CONSUMED_BTN_VISIT_PER_DAY	RUNNING_TIME_PER_DAY	CONSUMPTION_HIS	RUNNING_TIME	CONSUMPTION_RATE	Class
NUMBER_OF_DAYS	1	0.33	0.032	0.16	0.14	-0.28	0.54	-0.32	-0.19	0.49	0.64	-0.21	-0.11
PREVIOUS_FUEL_QTE	0.33	1	0.82	0.2	0.77	0.19	0.63	0.25	0.12	0.5	0.44	0.15	0.075
QTE_FUEL_FOUND	0.032	0.82	1	-0.081	0.72	0.038	0.066	0.031	0.013	0.048	0.043	0.042	0.051
QTE_FUEL_ADDED	0.16	0.2	-0.081	1	0.63	0.23	0.46	0.26	0.14	0.38	0.29	0.18	0.056
TOTALE_QTE_LEFT	0.14	0.77	0.72	0.63	1	0.19	0.37	0.21	0.11	0.31	0.24	0.16	0.078
DAILY_CONSUMPTION	-0.28	0.19	0.038	0.23	0.19	1	0.28	0.69	0.74	0.52	0.2	0.64	0.2
QTE_CONSUMED_BTN_VISIT	0.54	0.63	0.066	0.46	0.37	0.28	1	0.4	0.19	0.81	0.7	0.2	0.062
QTE_CONSUMED_BTN_VISIT_PER_DAY	-0.32	0.25	0.031	0.26	0.21	0.69	0.4	1	0.44	0.23	-0.029	0.52	0.23
RUNNING_TIME_PER_DAY	-0.19	0.12	0.013	0.14	0.11	0.74	0.19	0.44	1	0.43	0.45	0.04	0.2
CONSUMPTION_HIS	0.49	0.5	0.048	0.38	0.31	0.52	0.81	0.23	0.43	1	0.83	0.31	0.027
RUNNING_TIME	0.64	0.44	0.043	0.29	0.24	0.2	0.7	-0.029	0.45	0.83	1	-0.17	-0.0046
CONSUMPTION_RATE	-0.21	0.15	0.042	0.18	0.16	0.64	0.2	0.52	0.04	0.31	-0.17	1	0.043
Class	-0.11	0.075	0.051	0.056	0.078	0.2	0.062	0.23	0.2	0.027	-0.0046	0.043	1

Figure 3: Correlation Matrix of features

The anomalies detected in the data include hours the generator was working in one day, consumption exceeds what fuel generator can consume in one day and the case where generators consume fuel and running time is zero and outliers. The anomalies discovered in the dataset were used to generates a classification class of the samples as either normal or anomaly consumption.

Figure 4 shows the distribution of fuel consumed.

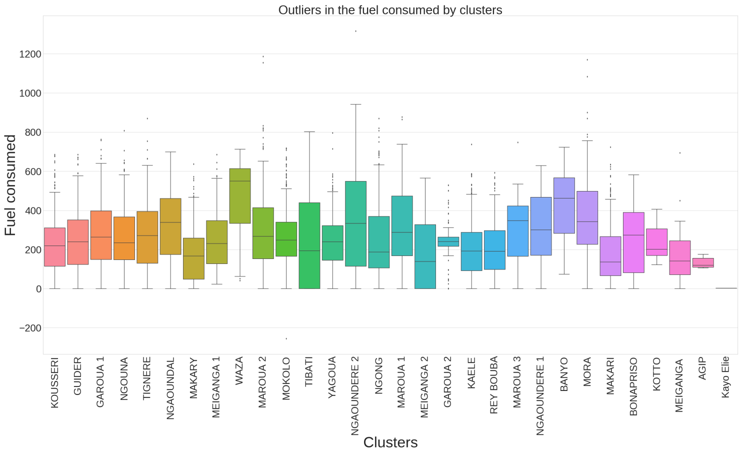


Figure 4: Fuel consumed per cluster.

A cluster is a group of sites where generators are located. Each site has a generator and therefore the fuel consumed by a cluster is the total fuel consumed by different generators in various sites. Clusters shows variation of the fuel consumed with median of the fuel consumed in most clusters ranges between 200 and 400. To label a sample in the dataset as an anomaly, the maximum consumption of each generator was considered.

The graph below shows a plot of the number hours generator was working in one day. From the graph, recorded data indicates anomalies of working hours was more than twenty hours in one day.

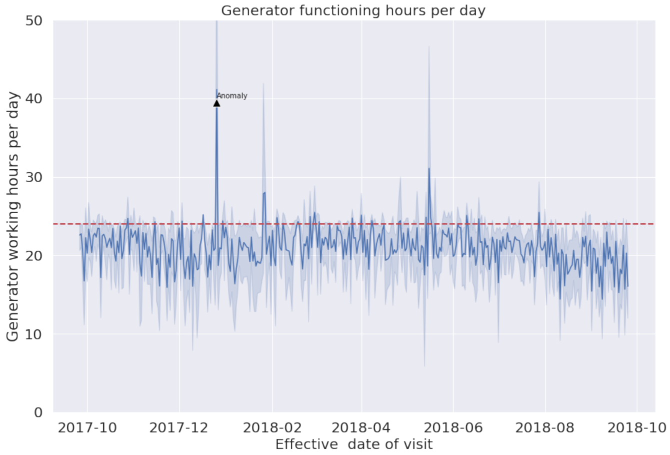


Figure 5: Graph of number of hours the generator function in one day.

From Figure 5, the running hours of most of the generator per day fluctuates around 18 to 25hours in a day. An observation with more than 24hours per day was considered as an anomaly.

3. METHODOLOGY

3.1. Support Vector Machines (SVM)

Support Vector Classifier is a classification algorithm used to separate two or more class by finding a hyperplane which maximizes the margin between classes. In the case of two separable class, a decision boundary separates two class with the widest margin. When the two class are linearly separable, the ideal hyperplane is the one with the widest margin between the boundary of two classes Friedman et al. (2001). Given a paired of learning examples of m number of features and corresponding class;

$$\{x_i, d_i\}, i = 1 \cdots N. \tag{1}$$

where x_i are input examples and d_i the classification class. The classifier find a function that correctly maps each input variable x_i to its corresponding class $d_i \in \{-1, 1\}$.

$$f(x) = \mathbf{W}^T X + b = 0. \tag{2}$$

$\mathbf{W}$ is the weight vector and b the bias. Equation (2) is a decision boundary determining the class of the input variable. The support vectors, that is, the samples on the boundary of the margin determines the decision boundaries. If the function of Equation (2) is greater than zero then the input variable belongs to class 1 otherwise it belongs to class 0. To get an ideal hyperplane between the two classes is as same as minimizing the norm of the weight vector Friedman et al. (2001). For two-class classification, the input variable is either on the positive or negative side of the decision variable, such that;

$$d_i(W^T x_i + b) \geq 1, \forall(x_i, d_i) \quad , d_i \in \{-1, 1\}. \tag{3}$$

The margin can be maximized by minimizing the weight vector $\mathbf{W}$. In case of non-separable, penalizing term is introduced to allow misclassification. The slack variable introduced in the case of non-separability measures how far the data deviate from the correct class Kelleher et al. (2015) .

$$d_i(w^T x_i + b) \geq 1 - \xi_i, \quad i = 1 \cdots N, \quad 0 \leq \xi_i \leq 1 \tag{4}$$

When a sample is correctly classified then the slack variable corresponding to that input value is equal to zero. For non-linear classes, the kernel functions transform the two class. A non-linear kernel function transforms the inputs to a more separable space and defining a hyperplane that clearly separates the classes. Commonly used non-linear function includes the hyperbolic tangent, polynomial and radial basic kernel Aggarwal (2014).

3.2. Multilayer Perceptron (MLP)

MLP is a supervised machine learning neural network inspired with how the human brain works. The classifier consists of input layer, hidden layers, activation function and output layer. The input layer receives the variables $X = \{x_0, x_1, \cdots x_n\}$ and assigned to weight vector $w = \{w_0, w_1, \cdots w_n\}$. MLP is a forward feed, that is, weighted input variables move from the input layer to the inner hidden layer Aggarwal (2014). The hidden layers enhance the model capabilities by allowing the network to learn complex problem and give result in the output layer. A nonlinear activation function applied to the weighted linear summation of the input variables to extract a relationship between the output and input variable.

$$z = \sum_{i=1}^{m} w_{ji} x_i, \tag{5}$$

$$y = \phi(z). \tag{6}$$

w_{ji} is the synaptic weight connecting neurons between the layers and ϕ is the activation function. The weights vector is unknown, therefore, weights in the input layer are randomly initialized based on the feature importance of the input variables. Hidden layers and activation function allow the model to learn nonlinear function, as result, low weight value at the input layer allows the model to start as linear and due to increasing hidden layers, the model turns nonlinear with increase weights.

Commonly used nonlinear function is the sigmoid function. The weights adjustment is with respect to the error, that is, computed at each neuron to make sure error minimization. As a result of these connections, each node is penalized as every node contributes to the global error computed at the output layer. The aim is to minimize the error, therefore the error correction with respect to the weights Haykin and Network (2004) .

$$\Delta w_{ji}^t = -\eta \frac{\partial \varepsilon(n)}{\partial w_{ji}}. \tag{7}$$

Where η is the learning rate and $\varepsilon(n)$ is the error term. Learning rate regulates the change of the adjusted weight in the direction of weight that makes sure the error is minimized, as a result of weight adjustment from Equation (7) the new weight in the corresponding layer became;

$$w_{kj}^{t+1} = w_{ji}^{t} - \eta \frac{\partial \varepsilon(n)}{\partial w_{ji}}. \tag{8}$$

At an output node, the total error is the difference between true value and the observed value. Since all the hidden layers connection contributes to the final error generated at the output node, then the error contributed from each node is also derived by taking the partial derivative of the error with respect to the weight connecting the node.

3.3. K- Nearest Neighbor (KNN)

KNN is a lazy learning algorithm that depends on the knowledge gained from the training data to predict the test data. The algorithm does not make any assumptions about the data and therefore the prediction depends on the k neighboring term. The new instance in the test dataset is compared with the training samples, checks the nearest neighbors and predict based on the majority class. Given a new data point, x_j the classifier checks k neighbors on the training samples and return the output based on the majority class in the k neighbor.

A small value of k can result in high accuracy but also pose a threat of overfitting the model. The large value of k causes underfitting of the model. To predict the class of test data point, Euclidean or Manhattan distance is applied in the case of the continuous variable.The commonly used distance is the Euclidean as it can be used for both nominal and numerical variable and distance is easy to interpret. Euclidean distance is measured by taking the difference between features in the data point.

$$D_i = ||x_j - x_0|| \tag{9}$$

To predict a new data point, the classifier compares each training set with new observation hence classifier suffer high computation cost as a result of comparing each point in the train set with test data.

3.4. Logistic Regression (LR)

Logistic regression is a probabilistic statistical model. Given an input variable the model predicts both the class and probabilities of a sample to belong to that class. Suppose given an input variable X such that , $X = (x_0, x_1, \cdots x_n)$, non-linear sigmoid function is applied to give the relationship between input and output variables $Y_i, \forall i = 1, \cdots, P$. Where P represent number of class.

In this case, we will consider a binary classification such that, Y $\in \{0, 1\}$ with a probability of being in either class 1 or 0 being;

$$p(Y|X) = \frac{1}{1 + e^{-z}}, \quad z = w_0 + \sum_{i=1}^{n} W^T X \tag{10}$$

The aim of logistic regression is to optimize the weights parameters **W** such that the classification error is minimized. Parameters are estimated by either gradient ascent or stochastic gradient ascent method Bonaccorso (2017). The log-likelihood and Newton methods are commonly applied to find optimal parameters Qi and Sun (1993). In both gradient ascent or stochastic gradient ascent method, parameters are adjusted until the model has a minimum error between the observed value and the predicted. Stochastic gradient ascent updates its weight at every single point depending on the direction of the weight.

The gradient ascent differs from the stochastic gradient ascent method in the sense that weights are adjusted at every level of the input variable and the gradient ascent considers input dataset in batches at each step, that is, to optimize the parameters using gradient ascent the problem is solved by taking the partial derivate of log-likelihood function with respect to parameters.

3.5. Performance evaluation

The dataset was split with 80% for training set and 20% test set. The testing sample validates the performance of the model in the confusion matrix and the generalized score of the model is obtained. The classifiers are trained on the same dataset and the performance is evaluated using the test dataset. The confusion matrix is a matrix that shows the classifier ability to predict the test data. Below is a representation of Confusion matrix used to decide the different performance of classifiers.

	Positive	Negative
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

Table 1: Confusion Matrix

From Table 1, the following information about the classifier's performances can be obtained. TP is when the model predicts positive when it's actually positive and on the other hand, FP is when the model predicts a negative class as positive. TN occurs when the model predicts a negative class when the class is actually negative and FN is when its positive class. The general performance of the model, that is, the ratio of correct prediction and the total sample gives the accuracy of the model. It is a measure that is affected by skew nature of the class distribution as it's computation involves output from both classes. From the confusion matrix, the True positive rate (TPR) is obtained by diving the true positive predicted by the total number of positive in the sample. The False positive rate (FPR) derived from the division of the false positive and the total number of negative in the sample. Recall or sensitivity is the ratio of true positive and all positive in the sample. Its show how sensitive is the classifier to detect the normal class from all label of the normal class. Precision compares the true positive in the confusion matrix and the total number the classifier predicted as positive. Specificity is also known as the true negative rate of the classifier, it is the ratio of true negative and negatives sample. F-measure the Harmonic mean of recall and precision. Equation (11) to (17) gives the formulas of the computation generated from the confusion matrix

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{11}$$

$$Recall(Sensitivity) = \frac{TP}{TP + FN} \tag{12}$$

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

$$F - measure = 2\frac{Precision * Recall}{Precision + Recall} \tag{14}$$

$$Specificity(TNR) = \frac{TN}{TN + FP} \tag{15}$$

$$TPR = \frac{TP}{TP + FN} \tag{16}$$

$$FPR = \frac{FP}{FP + TN} \tag{17}$$

For ease of comparison, models from all techniques in our study were developed using the same derived attributes.

Cross validation is a technique used to check how the model will perform in general. K-fold cross-validation with 10 fold was also tested. The data are split into K folds, that is, 10 folds, nine folds are used to train the data and the tenth fold is used for testing, this process is repeated until all the folds are trained and tested. The empirical risk of a classifier is obtained by taking the average of all error in the test fold.

Due to the imbalanced nature of the normal and anomaly class, ROC curves were used to measure how classifier is doing in each class. As we saw earlier, the data is skewed and therefore class distribution might affect classifier performance. The area under the curve (AUC) visualize the classifier behavior on how often the model will classify positive class correctly and when the actual classification is negative how often the model predicts positive. We want to maximize the true positive rate and minimize the false positive rate. The curve has TPR on the vertical axis and FPR on the horizontal axis. The plot range from $0 - 1$ with position (0,1) indicates a perfect classification model.

ROC curve has better visualization of models performance and comparative analysis of classifiers Japkowicz and Shah (2011). The precision-recall curve is more appropriate to study the behavior of the classifier. Precision and recall of a classifier is given by Equation (13) and (12). The precision-recall curve gives a clear relation of a true positive classified sampled and false positive classified sampled. The relationship between precision-recall and ROC curve shows that there exists a one-to-one relationship between the two curves Davis and Goadrich (2006). If the curve of classifier dominates the area under the curve in the precision-recall curve then curve must also dominate the ROC curve.

4. RESULTS & ANALYSIS

In this study, classification techniques were fitted with real-life data with 80% as a training sample and 20% for testing. To conduct a comparison of the fitted models, the models were fitted with the same features. We compared the performance of the classification methods used on the test data term of accuracy, graphical evaluation techniques such as ROC and precision-recall curve and K-fold cross validation techniques were used to evaluate the performance of the classification techniques. Anomalies detection abilities of the algorithms were also studied, that is, to identify which algorithm classifies the two classes correctly with minimal classification error. SVM, KNN, and MLP shows an impressive result in identifying the anomalies in the data. Below is a confusion matrix results of the classifiers on test data of 1181 sample points.

Confusion matrix				
	SVM	MLP	KNN	LR
TR	752	773	696	741
FP	30	9	86	41
TN	369	362	310	95
FN	30	37	89	304

Table 2: Classifier confusion matrix

The summary of classifiers performance on the test dataset are shown in Table 3.

Classification performance					
Classifier	Accuracy	F1-Measure	Recall	Precision	Specificity
LR	0.708	0.811	0.709	0.943	0.699
SVM	0.949	0.962	0.962	0.962	0.925
KNN	0.851	0.888	0.887	0.890	0.783
MLP	0.961	0.971	0.954	0.988	0.976

Table 3: Classifier evaluation performance

SVM fitted with radial basis function (RBF) had a general score of 94.9% on the test data. Although, the SVM had higher accuracy compared to KNN and LR on the test it is outperformed by MLP. From results obtained in the 10-fold cross validation as shown in Figure 6, SVM had an average score of 95.3%. The ROC and precision-recall curve of SVM as from Figure 7 shows that SVM has an AUC of 0.96 and 0.98% in precision-recall curve and ROC curve respectively. Figure 6 shows the accuracy distribution of each classifier in the 10-fold cross validation.

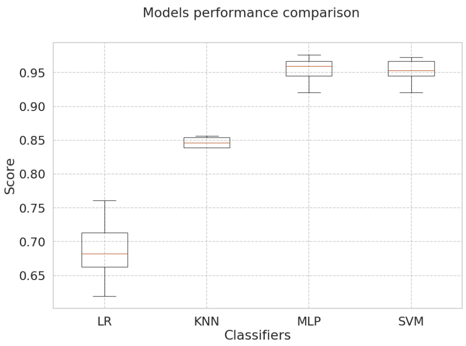


Figure 6: 10-Fold cross validation score for LR, KNN, MLP, and SVM

MLP fitted with relu activation shows better performance compared to other classifiers. The input variable used to fit the model were all standardized to ensure the feature are equally treated in the regularization. From the 10-fold cross validation, the classifier had an average score of 95.5%. As shown from Figure 7, the classier had an AUC of 1.00 and 0.99 for the ROC and precision-recall curves respectively. The precision-recall curve of MLP dominates with a score closed to perfect performances as compared to other classifiers.

KNN fitted with k equals to three had a general score of 85.1% on the prediction of the test data. For the case of anomalies detection, KNN performed better as compared to LR. From the 10-fold cross validation score, the KNN has a score of 84.3%, which outperformed LR. From Figure 7, KNN had an AUC of 0.89 and 0.96 in ROC and precision-recall curve respectively.

LR fitted with parameter C equal to 10. The classifier had a general performance of 70.8% on the test data and 69% K-fold cross validation score with ten fol. From Table 2, the classifier identified normal class with a precision of 94.3% which is slightly better as compared to KNN, and recall of LR indicates 70.9%. From Figure 7, AUC of ROC curve in LR is close to the random guess with an AUC of 69% in the ROC curve. In this case, the classifier will randomly classifier a sample, hence low predictive performance. LR had the lowest predictive power in identifying the anomaly class.

Across all validation tests, the LR was underperforming. In the study, accuracy techniques were used to determine the performance. Comparing the performance of the models, MLP has best accuracy score of 96.1% on test data. SVM, KNN, and LR had an accuracy score of 94.9%, 85.1% and 70.8% respectively. From figure 7, ROC and precision-recall curve of MLP is leading SVM.

Figure 7 shows the precision-recall and ROC curves representation of the classifiers used in this study. Precision-recall curve evaluates the number of positives that the model predicted. The curve at the right corner of the precision-recall curve is considered to have more true positive predicted by the model as compared to false negative. Precision-recall curve focus on how much positive have been predicted correctly given the total positive predicted. The LR in the precision-recall curve have a drastic change, this is explained by the change of positive and negative sample in Table 2. The graphs below show the graphical evaluation measure of the classifiers using precision-recall and ROC curves.

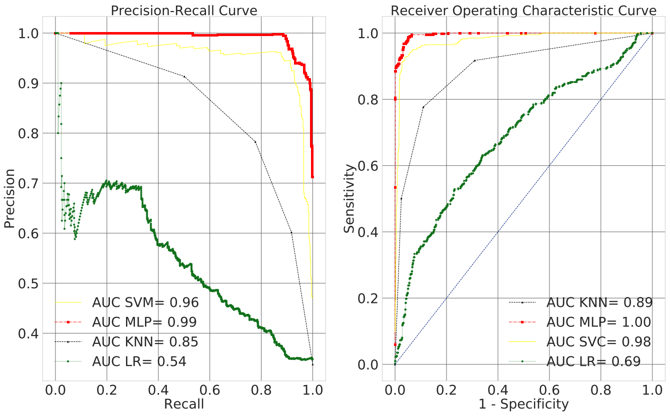


Figure 7: *Precision-Recall curves and ROC curves of classifiers*

4.1. Bias-Variance analysis

Using cross validation with ten split, we observe the relationship between training score and cross validation score of different classifiers. Variation of cross validation curve is as a result of high variance and training score curve variation is as a result of high bias. Bias-variance in the model affects the performance of the model. High variance is an indication that the model learns every noise in the data. High bias indicates that the model has less information about the dataset. Regularization of bias and variance help the model to attain better predictive performance. From cross validation with ten fold, Figure 8 shows the cross validation and the training scores on the fitted dataset.

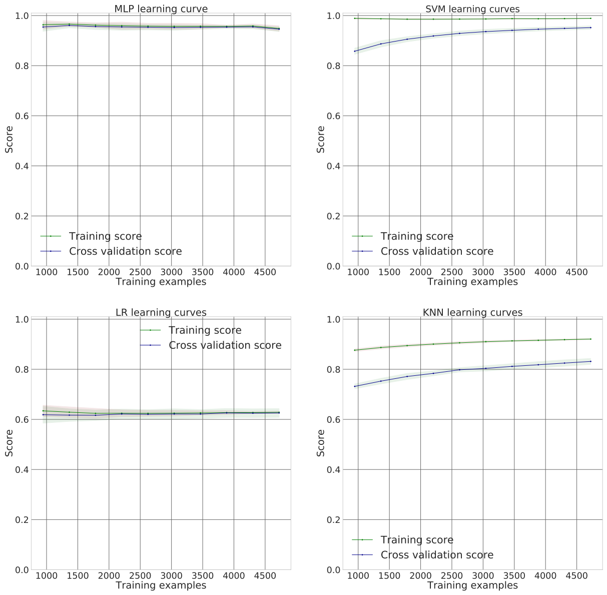


Figure 8: Training and cross validation score curve of MLP, KNN, SVM, and LR

In all the models the cross validation curve tends to converge with increasing training sample although SVM and KNN the convergences is slow with increasing in the sample. The training and cross validation curve of MLP from Figure 8 convergence faster and therefore increase in training data does not improve its performance. In LR, training score and cross validation curves converge completely with an increase in the data examples.

5. CONCLUSION

In this paper, we aimed at evaluating classification algorithm by training the model to learn and identify the anomalies in the fuel consumed dataset from a telecom base station. Four classification techniques were evaluated namely, SVM, LR, MLP, and KNN. MLP had best performance generally with an accuracy score on the test data of 96.1%. Although SVM outperformed other classifiers such as KNN and LR, using K- fold cross validation technique MLP performed best with score of 96.1%. From the confusion matrix, MLP had a best predicting power of the anomalies. LR performed better in precision score as compared to KNN, the classifier had the lowest performance as compared to all the classifiers . LR predicts more of the normal class as compared to anomaly class. The dataset was imbalance and as a result, evaluation methods which depend on both class for computation provides wrong impression due to skew nature of the class. ROC and Precision-recall curve used to visualize and evaluates classifiers. From Figure 7, MLP dominates two graph with higher performance on AUC in both curves. For anomalies detection in the dataset, MLP had a highest performance. Anomalies is our class of interest and since the ROC curve take into consideration of the negative class, ROC curve became the most relevant graphical visualization measure for not only the study of classifier behavior but also for model selection through comparative evaluation of different classifiers. We therefore conclude MLP show an overall better performance compared to other classification techniques in the performance measures, that is, the classifier best fit the training examples compared to other classifiers in terms of anomaly detection and in evaluation performance such as K-fold cross validation, train-test split, precision-recall and ROC curves.

6. SUMMARY

We fitted SVM, MLP, LR and KNN on the dataset of fuel consumed by generator with two class, anomaly and normal class. Comparative analysis of classifiers performance using K-fold cross validation, train test spit, graphical representation techniques such as ROC and precision-recall curve were used to study the behavior of the classifier. Key indicating factors consider to evaluate the classifiers includes how well the classifier identify anomalies that exist in the test data, accuracy and AUC in ROC graph. From the results of this study, MLP performed well in all metrics evaluated. The distribution of the two class were not equal, therefore the classification accuracy was less considered in the model selection.

7. Acknowledgement

The author would like to thank TeleInfra Company and Group One Holding Company for providing the dataset for this study. We wish to acknowledge Mr. OKALI DIMA Patrick from Group one holding for ensuring the data was available. Grateful for SciKit [3] learn library as this work has been made possible from the use of their library packages.

8. References

Aggarwal, C. C., 2014. Data classification: algorithms and applications. CRC Press.

Awoyemi, J. O., Adetunmbi, A. O., Oluwadare, S. A., 2017. Credit card fraud detection using machine learning techniques: A comparative analysis. In: Computing Networking and Informatics (ICCNI), 2017 International Conference on. IEEE, pp. 1–9.

Banjanovic-Mehmedovic, L., Hajdarevic, A., Kantardzic, M., Mehmedovic, F., Dzananovic, I., 2017. Neural network-based data-driven modelling of anomaly detection in thermal power plant. Automatika 58 (1), 69–79.

Bonaccorso, G., 2017. Machine Learning Algorithms. Packt Publishing Ltd.

Brownlee, J., 2016. Master Machine Learning Algorithms: discover how they work and implement them from scratch. Jason Brownlee.

Chouiekh, A., Haj, E. H. I. E., 2018. Convnets for fraud detection analysis. Procedia Computer Science 127, 133–138.

Davis, J., Goadrich, M., 2006. The relationship between precision-recall and roc curves. In: Proceedings of the 23rd international conference on Machine learning. ACM, pp. 233–240.

Friedman, J., Hastie, T., Tibshirani, R., 2001. The elements of statistical learning. Vol. 1. Springer series in statistics New York, NY, USA:.

G.M. Nguegnang, M. Garuti, M. A. T. A.-N. J. M., 2018. Machine learning in fuel consumption:a prediction approach in power generation plants. AIMS.

Haykin, S., Network, N., 2004. A comprehensive foundation. Neural networks 2 (2004), 41.

Japkowicz, N., Shah, M., 2011. Evaluating learning algorithms: a classification perspective. Cambridge University Press.

Kelleher, J. D., Mac Namee, B., D'Arcy, A., 2015. Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies. MIT Press.

Qi, L., Sun, J., 1993. A nonsmooth version of newton's method. Mathematical programming 58 (1-3), 353–367.

Taboada-Crispi, A., Sahli, H., Hernandez-Pacheco, D., Falcon-Ruiz, A., 2009. Anomaly detection in medical image analysis. In: Handbook of Research on Advanced Techniques in Diagnostic Imaging and Biomedical Applications. IGI Global, pp. 426–446.

Zanin, M., Romance, M., Moral, S., Criado, R., May 2017. Credit card fraud detection through parenclitic network analysis. ArXiv e-prints.

[3] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.