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

J.Mulongo[a,*], T. Ansah-Narh[b,c,**], Dr M. Atemkeng[d] (Co-ordinator), Rockefeller[e] and M. Garuti[d,**]

[a]*Mathematical Sciences (AIMS-Cameroon)*

[b]*Centre for Radio Astronomy & Astrophysics, Ghana Space Science and Technology Institute, Ghana Atomic Energy Commission, P. O. Box LG 80, Legon - Accra, Ghana*

[c]*Space & Atmospheric Research Group, Ghana Space Science and Technology Institute, Ghana Atomic Energy Commission, P. O. Box LG 80, Legon - Accra, Ghana*

[d]*Rhodes Centre for Radio Astronomy Techniques & Technologies(RATT)*

[e]*Mathematical Sciences (AIMS-Senegal)*

ARTICLE INFO

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

ABSTRACT

High fuel consumption is experienced in telecommunication based stations. As telecommunication base stations require electricity to operate, high fuel consumptions remains a challenge in base stations management as the working capital increases every year. To cope with this problem, we aim 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. In this paper, we show the use of supervised machine learning classification based techniques in detecting anomalies associated with the fuel consumed dataset from TeleInfra[1] base station using the generator as a source of power. We made use of machine learning techniques to train the dataset, these include support vector machines (SVM), k-nearest neighbor (KNN), logistic regression (LR), and multilayer perceptron (MLP). We evaluate models performance using K-fold cross validation and training test split techniques and classification performance metrics 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. In the area under the receiver operating characteristic and precision-recall curve, the model attained a score of 1.00 and 0.98 respectively. Our results also show that the working hours of the generator explains much the classification class.

1. Introduction

The expansion of mobile services of the telecommunication 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 equipments. 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) algorithms 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, recent works in anomaly detection such as fraud detection analysis [15, 6], thermal power plant [3], medical image analysis [14], 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,

*Corresponding author
**Principal corresponding author
***Corresponding author
Principal corresponding author
jecinta.mulongo@aims-cameroon.org (J.Mulongo); t.narh@gaecgh.org (T. Ansah-Narh); m.atemkeng@gmail.com (M. Atemkeng); rockefeller@aims-senegal.org (Rockefeller); marco@aims-cameroon.org (M. Garuti)
ORCID(s): 0000-0001-7511-2910 (J.Mulongo)

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

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

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 classes, 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 in Section 5.

The rest of this paper is organized as follows: Section 2 we describe the dataset used in the study, data preprocessing conducted and exploratory data analysis. In Section 5 we provide a brief description of machine learning algorithms and evaluation metrics used to perform model selection. Section 11 gives the result of the evaluation matrices capabilities in predicting whether an input is an anomaly or normal consumption.

2. EXPLORATORY DATA ANALYSIS

3. Dataset

The dataset is collected by cell site management company is Cameroon; TeleInfra[3]. The dataset contained different power type used by telecommunication cell site and we mainly focus on base stations powered by the generator. We limited to this kind of dataset as our major goal is to identify the cause of high fuel consumption by performing anomaly detection. 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. Variables description is given by Table 3 in Section A. The missing observation which accounts to 1.75% of the full dataset were dropped to reduce the assumption made on the data by replacing. 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, e.t.c.

From the 16 variables associated with fuel consumption, we extracted the variables that give more information about the output class. 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. We preferred GBC to fit our dataset 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.

Figure 1 shows that the variable *RUNNING_TIME_PER_DAY* of the generator is the key important feature which has a great influence on the output.

[3] http://www.art.cm/en/node/3111

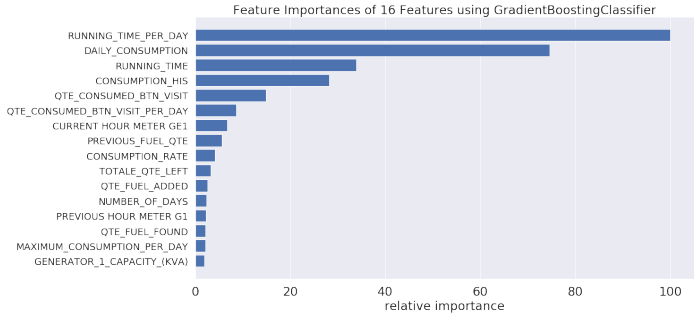

Figure 1: Feature importance ranking fitted using Gradient Boosting classifier

We also understand from Figure 5 the generator is working per day has a positive correlation with fuel consumed which explains why an increase in the running hours of the generator increases fuel consumption. *DAILY_CONSUMPTION* and *CONSUMPTION_HIS* variables give the fuel consumed per day and fuel consumed per period before the next refueling is done respectively. Other features such as *MAXIMUM_CONSUMPTION_PER_DAY*, the *CONSUMPTION_RATE* , *QTE_FUEL_FOUND* e.t.c, 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.

From a total sample of 5905, 35.11% of the sample were manually 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 were 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 [2]. A corrective measure of class imbalance has been proposed, that is, the use of sampling method [5]. Sampling method can be classified as either undersampling or oversampling. In the case of the oversampling method, classification class with small sample size is increased by generating a duplicate of the existing sample. This is done to ensure class attains the same sample size as the majority class. Undersampling takes the reverse of oversampling where the classification class which represent the majority is reduced in number to have both classification classes to have an equal sample size.

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, under-

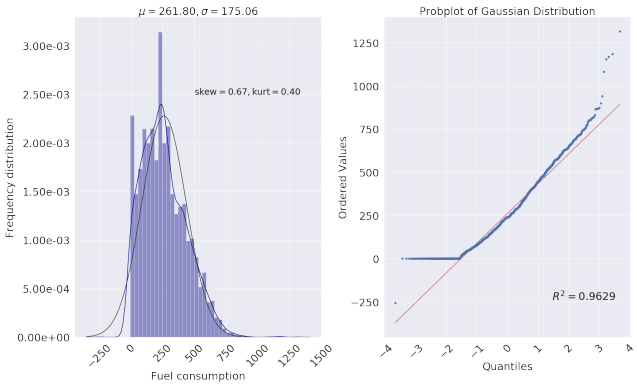

Figure 2: Average distribution of the fuel consumed

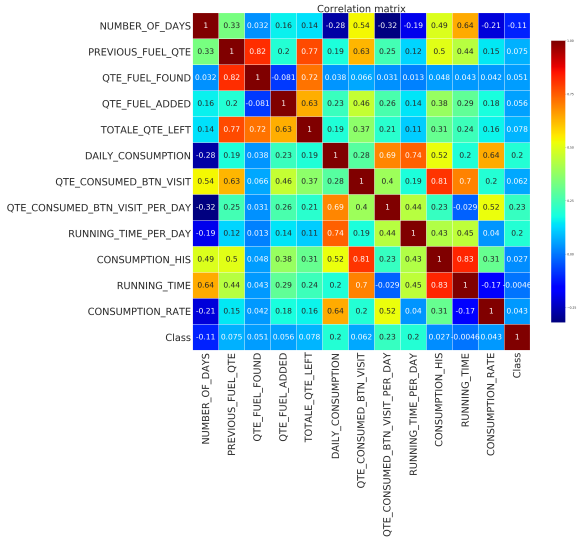

Figure 3: Correlation Matrix of features

sampling involves reducing the majority of data to have the same sample size as a minority.

4. Descriptive Analysis

Figure 2 shows the probability frequency distribution of the fuel consumed. 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 assume the underlying distribution of the dataset hence the distribution has a great impact on the performance of classifiers [11]. Figure 2 shows the kernel density estimation and normal probability plot for the fuel consumed dataset.

Anomalies in the data provide a wrong estimate of the fuel consumed. Considering all the anomalies we observed in the data, the normal consumption observation was used to make the prediction of the estimated fuel consumed [9]. From the normal consumed dataset, we observed that the mean and the standard deviations were 289.04(L) and 160.09 respectively. Figure 2 shows the Gaussian distribution plot which indicates that the dataset has outliers, as a result of this, the normal distribution curve deviating from the mean. Zero consumption of fuel consumed entries in the dataset causes the normal curve to start above the normal line. This explains the positive skewness of the fuel consumed with a skew of 0.40. Due to the high number of the zero consumption in the data as shown in Figure 2, observations with this characteristics were considered as an anomaly in the case where fuel reduced in the generator tank despite having zero consumption.

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 indicating that 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. It can be seen from From Figure 3 that the fuel consumed have strong positive relation of 0.83 with the number of hours the generator is working. The strong positive correction explained the dependence of the fuel consumed on the number of hours the generator is working and the rate of consumption of the generator. The two variables determine the quantity consumed. 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 a 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. The correlation matrix provides the linear relationship existing between variables but it does not explain in the case where the variables have a non-linear relation.

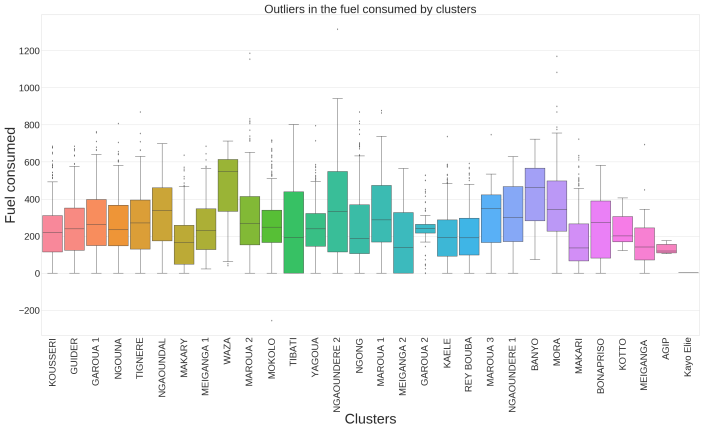

Figure 4: Fuel consumed per cluster.

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 shows the fuel consumed by clusters. A cluster is a group

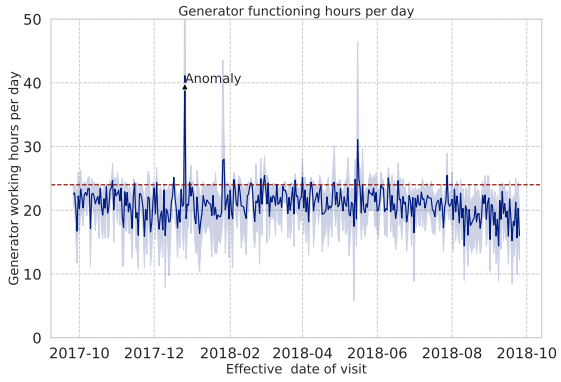

Figure 5: Graph of number of hours the generator function in one day.

of sites where generators are located and 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 show the variation of the fuel consumed with the median of the fuel consumed in most clusters ranges between 200 and 400. To label a sample in the dataset as an anomaly, in this case, the maximum consumption of each generator was considered and the samples which show the consumptions exceeding the maximum consumption was labeled as an anomaly.

The graph depicted in Figure 5 shows a plot of the number hours a generator was working in one day and we observed that the running hours of most of the generator per day fluctuates around 18 to 25 hours in a day. From the graph, the trend of working hours per day of the generator indicates an abnormal trend with input with more than 24 hours labeled as anomaly

5. METHODOLOGY

6. Support Vector Machines

Support vector classifier is a supervised machine learning technique used to separate two or more classes by finding a hyperplane which maximizes the margin between them. In the case of linearly separable classes, an ideal hyperplane called decision boundary is defined to separates two classes with the widest margin that ensures the distance between the boundary of the two or more class is maximized [8]. Given a paired of input examples $x \in R$ and corresponding class d_i such that $d_i \in \{1, -1\}$,. The classifier find a function that correctly maps each input variable x_i to its corresponding class d_i. The decision boundary is defined as;

$$f(X) = \mathbf{W}^T X + b. \tag{1}$$

Where $\mathbf{W}$ is the weight vector and b the bias. Equation (1) 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 $f(X)$ in Equation (1) is greater than zero then the input variable belongs to class 1 otherwise it belongs to class 0. To obtain an ideal hyperplane between the two classes is the same as minimizing the norm of the weight vector [8]. For two-class classification, the input variable is either on the positive or negative side of the decision variable, such that;

$$d_i(\mathbf{W}^T x_i + b) \geq 1, \forall(x_i, d_i) \quad , d_i \in \{-1, 1\}. \tag{2}$$

The margin can be maximized by minimizing the weight vector $\mathbf{W}$. In the 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 [12].

$$d_i(\mathbf{W}^T x_i + b) \geq 1 - \xi_i, \quad i = 1 \cdots N, \quad 0 \leq \xi_i \leq 1 \tag{3}$$

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 input example to a more separable space. 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 [1].

7. Multilayer Perceptron

MLP is a supervised machine learning neural network inspired by the human brain [10]. The classifier consists of input layer, hidden layers, activation function and output layer. The input layer receives the vector $X = [x_0, x_1, \cdots x_n]$ and assigned 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 [1]. 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.

$$z = \sum_{i=1}^{m} w_{ji} x_i, \tag{4}$$

$$y = \phi(z). \tag{5}$$

Where w_{ji} is the synaptic weight connecting neurons between the layers and ϕ is the activation function which transforms the weighted sum of input. 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 non-linear 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 [10]. The weight update is done using Gradient descent method given as;

$$\Delta w_{ji}^{t} = -\eta \frac{\partial \varepsilon(n)}{\partial w_{ji}}. \tag{6}$$

Where η is the learning rate and $\varepsilon(n)$ is the error term. At the output node, the network error is computed. Noting that weights in the hidden layers are updated based on the error computed at the output node. Learning rate regulates the change of the adjusted weight in the direction of weight. A small value of η controls the step size towards convergence whereas high value will cause divergence. To ensure the error is minimized, weights are adjusted such that the new weight became;

$$w_{kj}^{t+1} = w_{ji}^{t} - \eta \frac{\partial \varepsilon(n)}{\partial w_{ji}}. \tag{7}$$

The weight updates are done either in a batch, that is, using batch gradient descent or is the stochastic gradient descent method.

8. K- Nearest Neighbor

We also made use of KNN classifier to perform anomaly detection. 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 base its prediction according to the k neighboring terms. The k neighbors parameter that determined the number of neighbors in the training dataset to consider. 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 i.e;

$$D = ||x_j - x_0||. \tag{8}$$

Since KNN depends on the number of neighbors k during its testing phase and the algorithm uses distance measure to determine test class, the algorithm suffers from high computation cost. The choice of the value k influences the performance of the model. A small value of k can result in high accuracy but can results in overfitting the model whereas for a large value of k, although the effect of noise is reduced, KNN is prompt to have lose boundary hence underfitting the model.

9. Logistic Regression

Logistic regression makes use of conditional probability to map the input variable to a corresponding classification class. Given an input variable, the model predicts the probability of an input variable to belong to classification class. Suppose given an input variable X such that , $X = [x_0, x_1, \cdots x_n]$, and the corresponding class $Y_i, \forall i = 1, \cdots, P$. Where P represents the number of class, LR uses non-linear activation, that is, the sigmoid function to give the relation between the input variables and the output class. In this case of binary classification such that, $Y \in \{0, 1\}$ with a probability of being in either class 1 or 0 is given by;

$$p(Y|X) = \frac{1}{1 + e^{-z}}, \quad z = w_0 + \sum_{i=1}^{n} W^T X \tag{9}$$

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 [4]. The log-likelihood and Newton methods are commonly applied to find optimal parameters [13]. 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.

10. Performance evaluation

To evaluate the predictive capabilities performance of the fitted models, we abort methods such as train-test split and K-fold cross validation methods. From prediction of the test data, the model prediction was evaluated using confusion matrix. Table 1 shows the representation of a binary classification in a confusion matrix.

	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.

True positive (TP): correct prediction of positive class
True negative (TN): predictions of negative class when the class is actually negative
False negative (FN): wrong prediction of positive class as negative.
False positive (FP): model predicts negative class predicted as positive.
Recall or sensitivity gives the classifier capability to correctly classify the positive class which is given as a ratio of TP and positive in the sample in the dataset. Other measures such as precision compare the true positive in the confusion matrix and the total number of positively predicted by the classifier. 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 (10) to (14) gives the formulas of the computation generated from the confusion matrix

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}. \quad (10)$$

$$Recall(Sensitivity) = \frac{TP}{TP + FN}. \quad (11)$$

$$Precision = \frac{TP}{TP + FP}. \quad (12)$$

$$F - measure = 2\frac{Precision * Recall}{Precision + Recall}. \quad (13)$$

$$Specificity(TNR) = \frac{TN}{TN + FP}. \quad (14)$$

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. We performed K-fold cross-validation with 10 fold. 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 accuracy of the classifier is obtained by taking the average of all accuracy obtained in the test fold.
Due to the imbalanced nature of the normal and anomaly class in our dataset, 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. The TPR and the FPR of the classifier is given by;

$$TPR = \frac{TP}{TP + FN}. \quad (15)$$

$$FPR = \frac{FP}{FP + TN}. \quad (16)$$

ROC curve has better visualization of models performance and comparative analysis of classifiers [11]. The precision-recall curve is more appropriate to study the behavior of the classifier. Precision and recall of a classifier is given by Equation (12) and (11). 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 [7]. If the curve of classifier dominates the area under the curve in the precision-recall curve then curve must also dominate the ROC curve.

11. RESULTS & ANALYSIS

In this section, we are discussing the ability of the model to correctly predict the test dataset. 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, we compared the performance of the classifiers on the test data term of accuracy, graphical evaluation techniques such as ROC and precision-recall curve and K-fold cross validation technique. Anomalies detection abilities of the algorithms were also studied, that is, to identify which algorithm correctly predicts the two classes with minimal classification error. SVM, KNN, and MLP show an impressive result in identifying the anomalies in the data. Table 2 shows the 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 evaluation metrics used in this study are shown in Table 11.

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

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. MLP fitted with relu activation shows better performance compared to other classifiers. The input variable used to

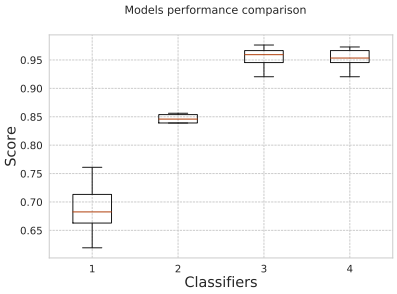

Figure 6: 10-Fold cross validation score for LR, KNN, MLP, and SVM

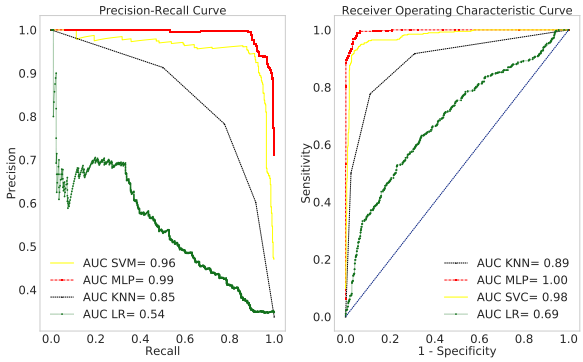

Figure 7: *Precision-Recall curves and ROC curves of classifiers*

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 fold. 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 performance in identifying the anomaly class.

Across all validation tests, the LR is under performing 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 compared to other classifier.

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 precision-recall curve of LR have a drastic change, this is explained by the change of positive and negative samples predicted as shown in Table 2. The graphs below show the graphical evaluation measure of the classifiers using precision-recall and ROC curves.

12. 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.

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.

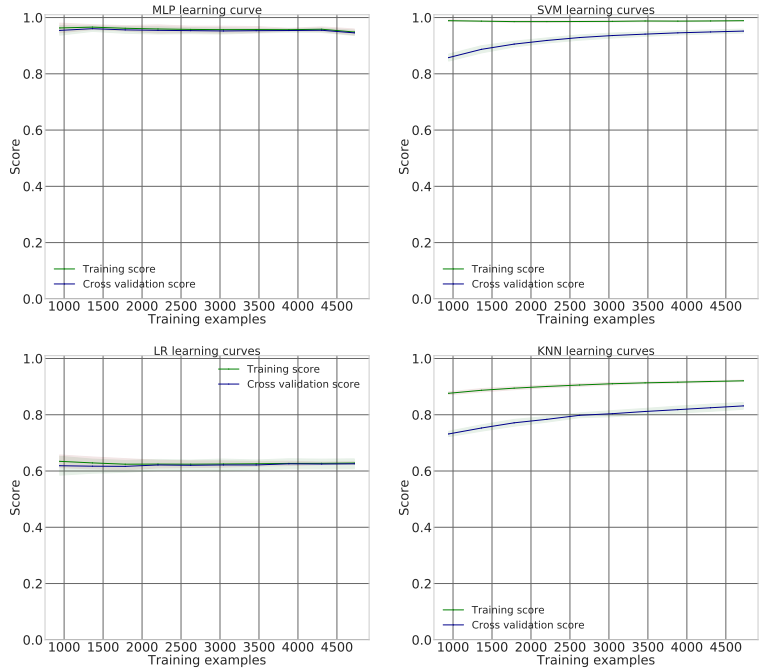

Figure 8: Training and cross validation score curve of MLP, KNN, SVM, and LR

13. CONCLUSION

In this paper, we aimed at evaluating the classification algorithm by training the model to learn and identify the anomalies in the fuel consumed dataset from a telecommunication base station. Four classification techniques were evaluated namely, SVM, LR, MLP, and KNN. MLP had the 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 a score of 96.1%. From the confusion matrix, MLP had the best predicting power of the anomalies. LR performed better in the 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 graphs with higher performance on AUC in both curves. For anomalies detection in the dataset, MLP had the highest performance. The anomaly is our class of interest and since the ROC curve takes 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 a 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.

14. SUMMARY

We fitted SVM, MLP, LR, and KNN on the dataset of fuel consumed by the 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 was not equal, therefore the classification accuracy was less considered in the model selection.

15. 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 [4] learn library as this work has been made possible from the use of their library packages.

16. References

References

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

[2] 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.

[3] 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.

[4] Bonaccorso, G., 2017. Machine Learning Algorithms. Packt Publishing Ltd.

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

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

[7] 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.

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

[9] 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.

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

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

[12] 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.

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

[14] 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.

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

A. My Appendix

Table 3 give the key variables associated with fuel consumption.

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

Feature description	
RUNNING_TIME	The total number of hours the generator worked before the next refueling is done.
RUNNING_TIME_PER_DAY	The number of hours the generator is working in one day.
NUMBER_OF_DAYS	The number of days before the next refueling process of the generator.
CONSUMPTION_HIS	The total fuel consumed between a specific period of time before the next refueling is done.This feature is determined by NUMBER_OF_DAYS, CONSUMPTION_RATE and RUNNING_TIME.
DAILY_CONSUMPTION	The quantity of fuel the generator consumes in a day base on its rate of consumption per hour and working hours in a day.
QTE_FUEL_FOUND	The quantity of fuel found inside the generator tank before refueling is done.
QTE_FUEL_ADDED	The quantity of fuel added in the generator during refueling process.
TOTAL_QTE_LEFT	Quantity left in the generator after refueling. This is the summation of what was found in the generator QTE_FUEL_FOUND and the quantity added QTE_FUEL_ADDED.
QTE_CONSUMED_BTN_VISIT	This is the difference between the total quantity which was left in the generator (TOTAL_QTE_LEFT) and the quantity found (QTE_FUEL_FOUND) during the next refueling.
QTE_CONSUMED_BTN_VISIT_PER_DAY	This variable is extracted from division of QTE_CONSUMED_BTN_VISIT and NUMBER_OF_DAYS to obtain the actual consumption of the generator.
CURRENT HOUR METER GE1	The hour meter reading of the generator.
PREVIOUS HOUR METER G1	The previous meter reading of the generator.
MAXIMUM_CONSUMPTION_PER_DAY	The maximum fuel the generator can consume in a day based on its rate of consumption.
CONSUMPTION_RATE	The number of liters the generator consumes per hour.

Table 3
Variable description.