NIPES

Journal of Science and Technology Research

Journal homepage: www.nipesjournals.org.ng

An Empirical Analysis of Machine Learning Models for the Prediction of Incipient Fault in Power Transformer

Efosa Charles Igodan[1], Rose Iyonawan Izevbizua[2]

[1,2] Department of Computer Science, Faculty of Physical Sciences, University of Benin, Benin City, Nigeria.

Article Info

Keywords: *dissolved gas analysis (DGA), classification algorithms, power transformer, ensemble methods, machine learning algorithms*

Received 23 Feb. 2024
Revised 27 March 2024
Accepted 31 March 2024
Available online 8 April 2024

https://doi.org/10.5281/zenodo.10939081

Abstract

The urgent need to monitor oil-filled power transformers' health on daily bases is due to the incipient faults that lead to economic loss. However, the most used traditional technique which is dissolved gas analysis (DGA) for incipient fault detection is characterized by their inability to categorize the state of the faults. This is because the DGA datasets can be imbalanced, insufficient and overlapping; imposing limitation in obtaining accurate diagnosis. This study investigated an ensemble of classifiers used to build fault detection diagnostic model for power transformers using DGA. The proposed methods include using data transformation techniques, machine learning algorithms: Support Vector Machine, Logistic Regression, Multilayer Perceptron, and their ensembles: voting, stacking, boosting, bagging, and random forest classifiers. The prediction model was applied on 298 data samples with seven independent attributes. The research results showed that the AdaBoost Radom Forest ensemble model with an accuracy of 100% performed better than other methods for the prediction of incipient faults in power transformers. The findings, therefore, suggest that the performance of the use of ensemble of classifiers could be influenced by the type and size of the datasets, and models' parameters.

1.0 Introduction

An electrical transformer is an inert apparatus that facilitates the transmission of electrical energy between circuits using electromagnetic induction. It is an immensely important and expensive component in electricity generation, transmission, and distribution [1]. As power transformers operate, they experience significant mechanical, electrical, and thermal strain, which inevitably result in faults. Power transformer failures not only halt the continuous flow of energy but also seriously jeopardize the stability and security of the entire power system. Moreover, such failures can lead to substantial economic and societal losses [2]. For detecting faults in power transformers, dissolved gas analysis (DGA) is widely used as non-intrusive technique [1]. Heat generated by the transformer during its operation causes insulation materials to undergo degradation, leading to the emission of detectable quantities of certain gases. Analysing gas distribution provides important insights in knowing the faults types that occurs within the power transformer. The gases that are most commonly examined in DGA include methane (CH_4), ethane (C_2H_6), ethylene (C_2H_4), hydrogen (H_2), acetylene (C_2H_2), carbon monoxide (CO), and carbon dioxide (CO_2), which are measured and expressed in units of parts per million (ppm) [3]. Popular traditional methods for interpreting DGA data include

the Doernenburg method, Rogers' method, key gas method, Duval Triangle method, and IEC ratios [4]. These methods rely heavily on the technical knowledge of human experts to diagnose faults in power transformers, but can as well lead to miss-identification of the severity or type of faults [1]. This problem has led to the introduction of numerous methods utilizing intelligent methods to reliably and efficiently analyse DGA data and predict power transformer faults more accurately. Some popular intelligent methods include Artificial Neural Networks (ANN), support vector machine (SVM), fuzzy logic, adaptive neuro-fuzzy inference system (ANFIS), and other hybrid methods [5]. Numerous studies have investigated the efficiency of these intelligent techniques. It is important to consider previous research in order to identify opportunities for improvement. A summary of studies on DGA interpretation using machine learning and intelligent approaches are highlighted in Table 1.

Table 1 Related literature summary

Author	Models	Contribution for Knowledge	Limitations
[1]	KosaNET (ensemble method based on decision trees)	Exhibits an improved ability in classifying multinomial data with a classification accuracy of 99.98%.	Not used on regression problem
[6]	IEC and ROGER ratio methods combined with Artificial Neural Networks (ANN)	Increase in efficiency from 20% to 70% for the IEC ratio method and 40% to 70% for the ROGER ratio method	Still prone to producing misleading results
[4]	Parzen window estimation method	Higher performance than the traditional methods at 95% accuracy.	Small sample used
[7]	Back-propagation (BP), radial basis function (RBF) NN, and adaptive ANFIS	Obtained 98.85% accuracy	No validation ANFIS is slow and occupies more memory space
[8]	Multi-layer perceptron (MLP), Doernenburg ratio and Rogers ratio	Detects faults more accurately than contemporary ratio methods	No parameter tuning
[3]	Common Vector Approach (CVA)	Produces better fault diagnosis performance than all methods that were compared.	Lacks generalization ability
[9]	Particle Swarm Optimization with Support Vector Machine (PSO SVM)	Superior accuracy to standard SVM and GA-SVM, with 85.71% accuracy against 57.14% for SVM and 60.71% for GA-SVM	Lacks generalization ability
[10]	Probabilistic Neural Network (PNN) optimized by a modified differential evolution whale optimization algorithm (MDE-WOA)	Improves the convergence rate of PNN network, enabling it to quickly escape local optima and increasing efficiency.	Performance is impacted by the parameter setting
[5]	A Fuzzy Inference System (FIS), Artificial Neural Network (ANN), and Adaptive ANFIS	The ANFIS, FIS and ANN method were shown to have superior performance with 97.5%, 95%, 92.5% accuracy respectively while the Rogers ratio method had a 60%	Lacks generalization ability

[2]	Genetic algorithms, support vector machines, arctangent transformation (AT) and logarithm transformation filter methods	Outperforms all other methods in most power transformer fault categories	Lacks generalization ability
[11]	A new approach DGA Technique has been developed based on the gas concentrations.	Obtained higher agreement accuracy than traditional DGA techniques Obtained an overall accuracy of 84.71%	Poor performance compared to ML-based models.
[12]	Fuzzy logic system	Proposed system maintains an accuracy rate of 99% in identifying faults in transformers.	No learning Not adaptable
[13]	Roger's ratio and IEC ratio combined, Fuzzy Inference system (FIS)	FIS system was shown to improve the efficiency of diagnosing power transformer faults	No learning and model not adaptable
[14]	MLP Model and SVM	SVM obtained 81.4% accuracy while MLP obtained an accuracy of 76 %.	Lacks generalization
[15]	ANN and Fuzzy systems	Duval Pentagon method and Fuzzy Inference system was proven to have the best performance of all methods considered	ANN lacks explainability and interpretability. Fuzzy lacks learning and adaptability
[16]	Extreme Learning Machine (ELM) based technique	Proposed method performed better than existing ones.	Lacks generalization ability

It is evident from the literature that the old DGA techniques' accuracy limit remains a significant problem when diagnosing transformer defects caused by electrical and thermal stressors [11]. Due to their inability to precisely evaluate every faults, which typically arises when many faults occur in a transformer especially as the concentration of gas approaches the threshold, all conventional approaches have limits [5]. Furthermore, it is observed to the best of the authors' knowledge, ensemble machine learning methods – a method of combining different ML algorithms – is yet to be applied in addressing faults in power transformer so as to solve the problem associated with diversity with respect to datasets and base classifiers [19, 28, 29]. Motivated by these limitations, three well-known supervised machine learning techniques – MLP, SVM and LR, and Random Forest (RF), and three different ensemble learning methods – Bagging, Boosting, Stacking and Voting methods were investigated to diagnose the condition power transformer towards improving the accuracy and reliability of fault diagnosis and building confidence in results obtained.

2.0. Materials and Method

This section outlines the general methodology of our suggested approach as well as the primary research component. The suggested methodology's system architecture is shown in Figure 1.

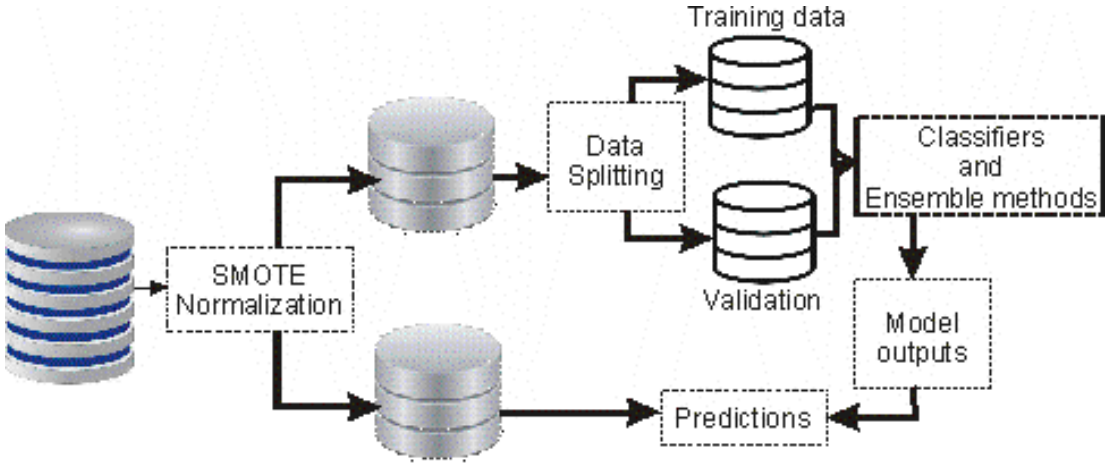

Figure 1 System architecture of the proposed methodology

2.1 Data

The primary concern for information accuracy is the data source [20]. The data used is the IEC-TC 10 dataset from [17], which consists of 166 instances and 8 features. Among these features, 7 are independent variables denoting the concentration of dissolved gases; H_2, CH_4, C_2H_4, C_2H_2, C_2H_6, CO_2, and CO in ppm (parts per million), while the 8^{th} attribute represents the observed fault. From the dataset, five fault categories were identified, and an extra label was assigned to represent the normal operational state. The fault types represented in Table 2 are as follows: partial discharge (PD), discharge of low energy (D1), high energy discharge (D2), thermal fault below 300°C and Thermal Fault above 300°C but below 700°C (T1&T2) and thermal Fault above 700°C (T3) [17,18].

Table 2 *Fault categories of power transformer faults*

Fault Type	Number Of Instances
PD	9
D1	26
D2	48
T1&T2	16
T3	18
Normal (no fault)	50

2.2 Processing the data

Data preprocessing is an essential stage in data preparation that assures data are well-formatted and high-quality prior to applying machine learning algorithms and building models. This is the most labor-intensive stage of the data mining life-cycle and needs to be done with extreme caution because bad data might result in bad models and poor performance. It involves various procedures, such as data cleansing, balancing, normalization, data wrangling etc., so as to make the data suitable for further analysis and modelling [19,20].

2.3 Missing Values

In this study, the missing values in the dataset were filled up using the median imputation technique. The choice of median imputation was made due to the relatively low number of missing values (less than 10%) and the implementation method is quick and straightforward.

The two steps are to: first, impute the median value of the respective feature, considering instances belonging to the same class; and second, repeat this process for all instances with missing values in the feature [21].

2.4 Data Imbalance

Data imbalance occurs when the distribution of classes in the dataset is skewed and the minority classes have significantly fewer examples than the majority class leading to overfitting and poor generalization. When dealing with data imbalance, the two techniques employed are under sampling and oversampling. Under sampling is involved with reducing the majority class which is not recommended as there is a chance of loss of information. Oversampling on the other hand involves increasing the minority classes so they are equal with the majority classes [22]. To address data imbalance in the dataset, the synthetic minority oversampling technique (SMOTE) was applied in our work. The following steps was used to implement SMOTE and using Equation 1.

1. Select a minority instance from the feature space.
2. Locate the K-nearest neighbours and obtain the distance between the neighbours using the Euclidean distance (Equation 2).
3. Next, we determine the vector that connects the chosen neighbor and the current data point.
4. To sum up the additional samples, we multiply the vector by a random integer between 0 and 1 and add it to the existing data point.

$$x_{syn} = x_i + (x_{knn} - x_i) * t \quad (1)$$

$$d(x, y) = \sqrt{(\sum_{i=1}^{N}(x_i - y_i)^2)} \quad (2)$$

After applying SMOTE on our dataset the number of instances increased from 166 to 300, with 50 samples for each fault type and depicted in Table 3.

2.5 Normalisation

The dataset was normalized to prevents bias, improves algorithm convergence and speed and stabilize variance. By bringing all features to a common scale, normalization enhances model performance, interpretability, and the reliability of statistical analyses. Min-Max normalization technique was applied as represented in Equation 3.

$$X' = \frac{x - min(x)}{max(x) - min(x)} \quad (3)$$

where X' is the normalized value of the original data point x, $min(x)$ is the lowest value of the whole dataset, $max(x)$ is the highest value of the whole dataset .

2.6 Data Splitting

The dataset is split with 60% allocated to train the model, 20% for validation and 20% to test the model. The distribution of the dataset is depicted in Table 3.

Table 3 Label count for training, validation and testing set

Fault Type	Training Set	Validation	Testing Set
Total (300)	180	60	60

3.0. Modeling

The models are formed in this stage. As shown in Figure 2, this process is an iterative step leading up to modeling and model evaluation. The fundamental idea is to repeatedly build different models in an effort to find the optimal model that meets the data requirements for performance criteria. Some selected supervised classification algorithms were used to build the models with their default parameter settings adopted [19]. The aim behind the application of these classifiers was to increase diversity and confidence of the results obtained [33]. The following subsections discusses the classification algorithms as:

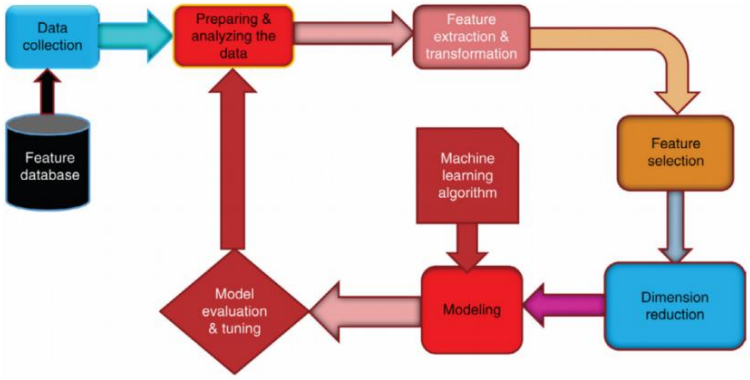

Figure 2 A typical machine learning framework

3.1 Multilayer Perceptron

A Multilayer perceptron (MLP) is an artificial neural network characterized by its fully connected feed-forward architecture, comprising a minimum of three layers: an input layer, an output layer, and at least one hidden layer [19]. In our Multilayer perceptron model implementation as represented in Equation 4, we utilized an input layer of 7 nodes, mapping to each of our input features, and use an output layer of 6 nodes for each of our fault types and a single hidden layer. The weight outputs are computed as follows [6]:

$$y_j = f\left(\sum_{i=1}^{n} X_i W_{ij} + \theta_j\right) \quad (4)$$

where X_i are the network inputs; W_{ij} translate the weight-connection between the input neuron i and the neighbouring hidden neuron j; θ_j is the bias of the jth hidden neuron, y_j is the output of the network, and $f(\,)$ is the transfer function or also called activation function.

3.2 Support Vector Machines

A support vector machine depicted in Equation 5, creates a hyperplane or multiple hyperplane within a high-dimensional or potentially limitless space, which can then be used for tasks such as classification or regression tasks [23]. The "one-vs-rest" method of multi-class SVM classification was used in this work [24].

$$(w * x) + b = 0 \quad (5)$$

Where w is the weight vector, b is the bias, and x is the feature vector.

3.3 Logistic Regression

Logistic regression is a statistical modeling technique that determines the probability of a specific result based on one or more independent variables. The logistic function, often depicted as an S-shaped curve (Sigmoid curve), generates an output value ranging between 0 and 1 which can effectively represent a probability [25]. This is implemented using the model:

$$y = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}} \quad (6)$$

Where y is the output value ranging from 0 to 1, x is the input variable, and β_0, β_1 are the coefficients

3.4 Random Forest

With ensembles of trees, one can make significant improvements in classification and regression accuracy, as each tree in the ensemble is developed according to a random parameter. By aggregating throughout the ensemble, final predictions are obtained. These processes are known as "random forests" because the fundamental components of the ensemble are tree-structured predictors and because each of these trees is built with the addition of randomness [27, 33, 34]. An RF is defined formally as a classifier that consists of a set of decision tree classifiers:

$$\{h_k(x, T_k)\}, k = 1,2,\ldots,L \quad (7)$$

Where h_k is the decision tree classifier, T_k is the independent identically distributed random sample, and each tree casts a unit vote for the most popular class at input x [19,26,27]. To build an RF classifier we:

1. decide on the number of trees to build, a larger number usually leads to better performance up to a point, a range between 64 and 128 trees has been suggested in literature for an optimal equilibrium among AUC performance, processing speed, and memory utilization,
2. we take the bootstrap sample of our training set data (i.e. repeatedly taking random samples from our datasets, with replacement, k times), this enables each decision tree to be trained on a different dataset creating variation,
3. each decision tree evaluates only a random subset of the available features during each division, introducing randomness that prevents any individual feature from overpowering the decision-building procedure, and;
4. finally, every tree makes a prediction about the class of a given data point; the anticipated classes are then combined via majority vote.

4.0 Ensemble (Combination) Learning Method

In machine learning, the ensemble technique aggregates the predictions of several different independent models to improve their overall performance and robustness, even though it brings an increased algorithmic cost and model complexity. There is no single answer to what the best

model is or what will give you the best results so the need for ensemble [19]. In this study, bagging, stacking, boosting, and voting are adopted using LR as the meta learner.

4.2 Bagging

The bagging ensemble method trains a basic classifier on random portions of the original dataset. The predictions of these individual classifiers are combined either through voting or averaging to produce a final prediction [23]. This is implemented for our training set T using the model:

$$T = \{(y_n, x_n), n = 1 \ldots N\} \quad (8)$$

where x is the instance and y is the target label, and a classifier ϕ of the form $\phi(x, T)$ that predicts the output label from the input data, we build a bagging ensemble by;

1. taking repeated samples $\{T^{(b)}\}$ from T, to create the model;

$$\phi_B(x) = av_B \phi(x, T^{(b)}) \quad (9)$$

2. the bootstrap samples are then derived from replicated dataset, each containing N instances, drawn randomly and with replacement from the training dataset,
3. we train multiple instances of a base model on these subsets independently from each other, and;
4. finally, we combine the predictions of these models to determine our final prediction.

4.3 Boosting

Boosting is an adaptive technique in machine learning that involves fitting a series of weak learners sequentially by assigning more importance to the data points that were poorly handled by the previous models [28]. There are two meta-algorithms used for fitting and aggregating weak learners for boosting; AdaBoost and gradient boosting. This work adopted the AdaBoost. We implemented a multi-class AdaBoost ensemble utilizing the Stage-wise Additive Modelling using a Multi-class Exponential loss function (SAMME) technique. The core concept in the SAMME algorithm involves calculating the weighted sum of the weak classifiers' predictions and focuses on minimizing an exponential loss function, which measures the difference between predicted and actual classes, by adjusting weights and selecting the best weak classifiers [29]. It is implemented based on the following mathematical model;

$$F_t(x) = F_{t-1}(x) + \alpha_t h_t(x) \quad (10)$$

where $F_t(x)$is the combined prediction up to iteration *t*, $F_{t-1}(x)$ is the combined prediction up to iteration *t-1*, $\alpha_t$is the weight assigned to the *t-th* weak classifier, $h_t(x)$ is the prediction of the *t-th* weak classifier for input *x*. The following steps implement the algorithm as:

1. initialize the dataset and allocate equal weights to all data points,
2. then feed the dataset as input to the model and pinpoint the data points, that have been classified incorrectly,
3. increase/boost the weights of the misclassified data points while reducing the weights of those that have been classified correctly, and normalize the weights of all data points.
4. This cycle is repeated until the desired outcomes are attained.

4.4 Stacking

Stacking involves the training of multiple diverse weak learners and then combining their predictions by using a meta-model in order to leverage the diverse perspectives of these weak models and build a more robust and accurate final prediction [30]. We implement stacking ensemble in the following steps:

1. Firstly, multiple weak learners are trained using different algorithms, hyper-parameters, or subsets of the data.
2. Each base model m_i generates predictions for each input data point x_j, resulting in an ensemble of predictions:

$$\{P_{ij} = m_i(x_j)\}\ i = 1 \ldots N, j = 1 \ldots k \qquad (11)$$

   where k is the number of data-points.
3. After training the weak learners, they are used to make predictions on the same dataset they were trained on. Each weak learner generates its set of predictions for the target variable based on its understanding of the data.
4. The meta-model is then trained on these predictions to learn how to weigh and combine them effectively:

$$P_{ensemble} = M(Z) \qquad (12)$$

4.5 Soft Voting

The most widely used voting method is majority voting. In this approach, we determine the final output by the following steps:

1. each classifier casts a vote for a specific class label,
2. we aggregate the votes cast and determine if any class receives more than half of the votes,
3. if such a class exists it is determined as the final output, else;
4. if not more than half of the votes are cast for any of the target labels, a rejection option is provided, and the combined classifier refrains from making a prediction.

The mathematical model for the above process is as follows;
This research study adopted the majority voting as in Equation 13 and 14 respectively.

$$\text{class(x)} = \underset{\text{c}_\text{i} \in \text{dom(y)}}{\arg\max} \sum_\text{k} \text{g}(\text{y}_\text{k}(\text{x}), \text{c}_\text{i}) \qquad (13)$$

where $y_k(x)$is the classification of the k^{th} classifier, and $g(y_k(x))$ is an indicator function defined as:

$$\text{g}(\text{y}_\text{k}(\text{x})) = \begin{cases} 1\ \text{y} = \text{c} \\ 0\ \text{y} \neq \text{c} \end{cases} \qquad (14)$$

5. Performance Evaluation Methods

The confusion matrix (CM) was used to summarize the performance of classifier models. This matrix offers crucial information on the generalization properties of the model as well as its capacity to forecast specific classes [1,19,31]. Other performance metrics that are derived from a confusion matrix are [19]:

Table 4. Confusion Matrix

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	(TP)	(FN)
	Class=No	(FP)	(TN)

(a) Accuracy: determines the model's accuracy, i.e. the ratio of accurate classification to all instances, as shown by Equation 15.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \quad (15)$$

(b) Precision: The ratio of correctly categorized positive cases to all positively classified positive instances, as represented by Equation 15, indicates the model's precision.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (16)$$

(c) Recall: calculates the sensitivity using the ratio of instances classified correctly as positive to all the positive instances, represented by the Equation 17.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (17)$$

(d) F-Measure analysis: one way is to calculate the harmonic mean of precision and recall, which in the information retrieval literature is known as the F-measure [32]. This statistical study uses the weighted harmonic mean of the recall and precision to determine the test's accuracy. The more realistic measure provided by the F-score can be achieved by using both recall and precision in the test performance [1,19,32].

$$\text{F} - \text{measure} = \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (18)$$

(e) Receiver operating characteristic (ROC) curve: The area under the coverage curve gives the absolute number of correctly ranked pairs; in an ROC plot the area under the ROC curve (AUC) is the ranking accuracy [19,32].

6. Results and Discussion

The results from utilizing machine algorithms and ensemble model for classification of power transformer faults are discussed in this section. All experiments were conducted on an 8th-generation Core i5 machine running a Linux operating system. We deployed Jupyter Notebook with the Python programming language to implement our machine learning models. These models' performance was evaluated employing the F1 Score, accuracy, precision, ROCAUC and recall. Table 5 presents the test results of the implemented models. AdaBoost achieved a 100% accuracy, precision, recall and f1-scores as the highest compared to others. The stacking ensemble and bagging (MLP) obtained 95% accuracies, while voting and MLP obtained 93.33% accuracies respectively. The least preforming models are the SVM and random forest with an 85% and 83.33% respectively which are depicted in Figure 2.

Table 5 Evaluation comparison of the models implemented

Metrics	SVM	MLP	RF	Bag (SVM)	Bag (MLP)	Ada (RF)	Stack	Voting
accuracy	85.00	93.33	83.33	83.33	95.00	**100.0**	95.00	93.33
precision	87.21	94.05	83.94	84.30	95.51	**100.0**	95.15	94.20
recall	85.00	93.33	83.33	83.33	95.00	**100.0**	95.00	93.33
f1_score	84.50	93.31	82.78	82.98	95.06	**100.0**	95.00	93.37

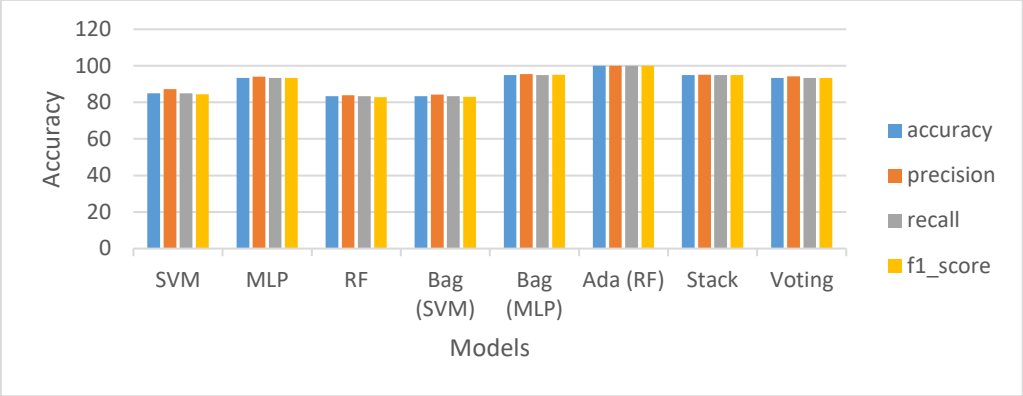

Figure 2 Performance metrics of all models

The confusion matrix of each of the models are shown in Figures 3 to 8. In Figure 3, Multilayer perceptron and voting misclassified 4 samples respectively, while SVM shown in Figure 4 misclassified 9 samples. In Figure 5, the bagging (MLP) and stacking has 3 misclassified samples, while AdaBoost has no misclassified samples in its prediction which shows AdaBoost model outperformed others as regards classification in power transformer faults.

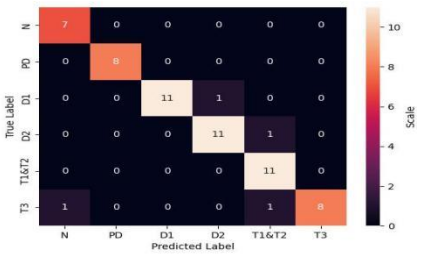

Figure 3 CM for MLP classifier

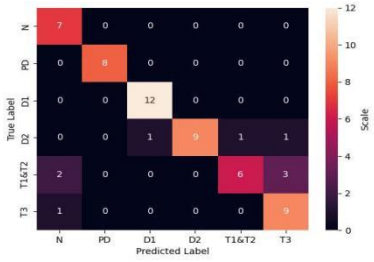

Figure 4 CM for SVM classifier

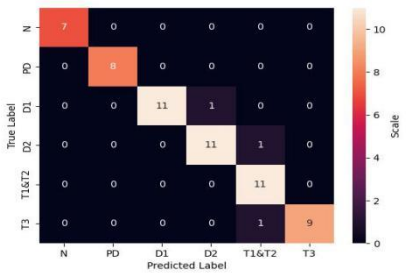

Figure 5 CM for Bagging MLP classifier

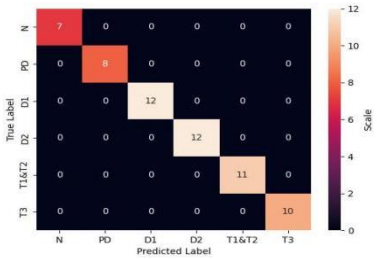

Figure 6 CM for AdaBoost Ensemble

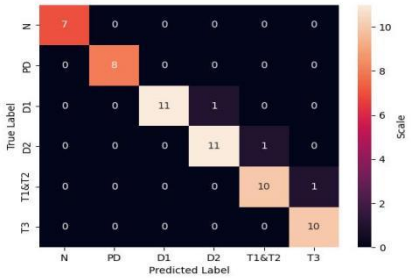

Figure 7 CM for Stacking classifiers

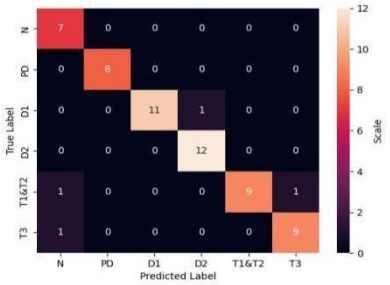

Figure 8 CM for Voting Ensemble

The area under the ROC provides values between (0, 1). When discrimination operates significantly better, its value is 1, and a faulty classification leads values to be near to 0.5 [35]. Figures 10 to 17 depicts the receiver operating characteristic area under (ROCAUC) curve showing the absolute number of correctly ranked pairs, i.e., the ranking accuracies of each model. From the ROCAUC scores from all classifiers shown in Table 6, we obtained good performance in the prediction models evaluated. The One vs Rest technique was used to binarize the models producing the results for each class. Most of the ensembles are shown to have a high sensitivity or true positive rate of above 0.9 across most classes and with an average AUC-ROC score between 96% to 99% respectively. This shows that the models are not randomly guessing the predicted outcome and produce optimal results in fault diagnosis.

Table 6 AUC-ROC scores (%)

ROC	SVM	MLP	RF	BAG SVM	BAG MLP	ADA RF	STACKING	VOTING
N	100	100	100	100	100	100	100	100
PD	100	100	100	100	100	100	100	100
D1	99.31	100	94.62	99.31	100	100	100	100
D2	94.62	99.65	90.28	94.44	99.65	98.44	99.65	99.65
T1&T2	94.06	99.63	96.29	94.43	99.07	100	99.81	99.26
T3	98.20	100	99.20	98.20	100	100	100	99.40
AVERAGE	97.70	99.88	96.73	97.73	99.79	99.74	**99.91**	99.72

The application of oversampling, normalization and cross-validation techniques resulted in the performance boosting of both models as depicted in Figure 9. The best performing models are the stacking, MLP, voting, Bagged MLP and ADA RF models respectively. Whilst the SVM, Bagged SVM and RF performed little above average. This may not be unconnected to the use of the default parameters in the study. However, it is important to note that the ensemble methods performed better than the base classifiers. In addition, the ensemble of MLP, RF, Stacking, and Voting improved in their classification due to the diversity introduced in to the model, making the proposed model robust and scalable and by implication increases model predictive confidence.

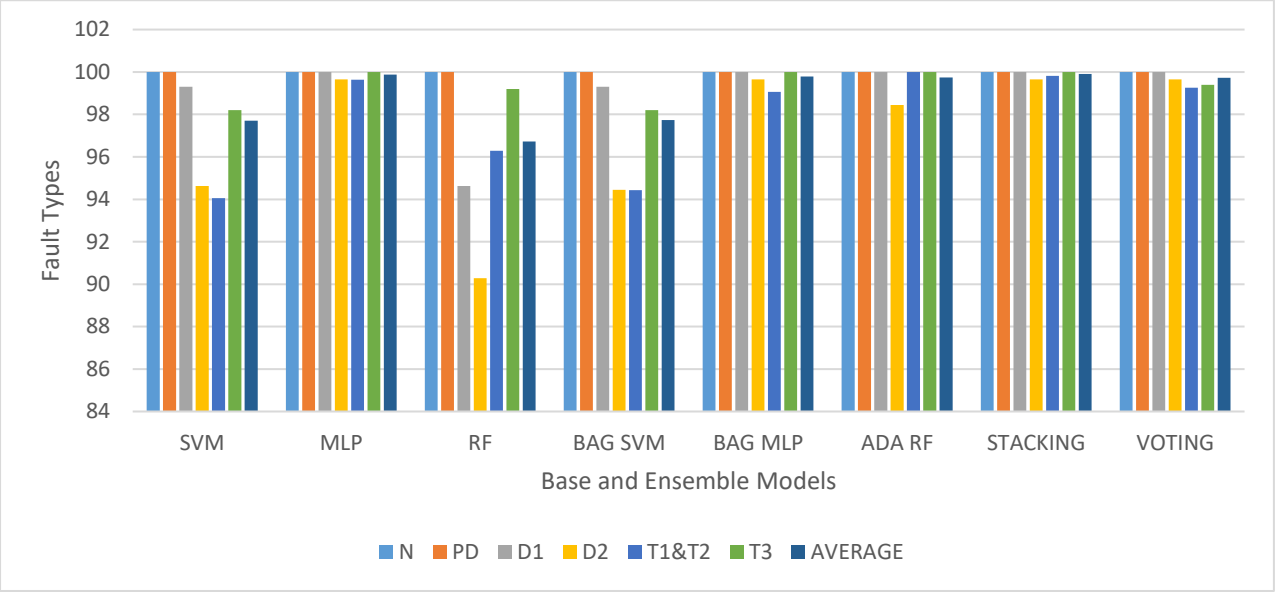

Figure 9. Empirical Analysis of Models' Performance

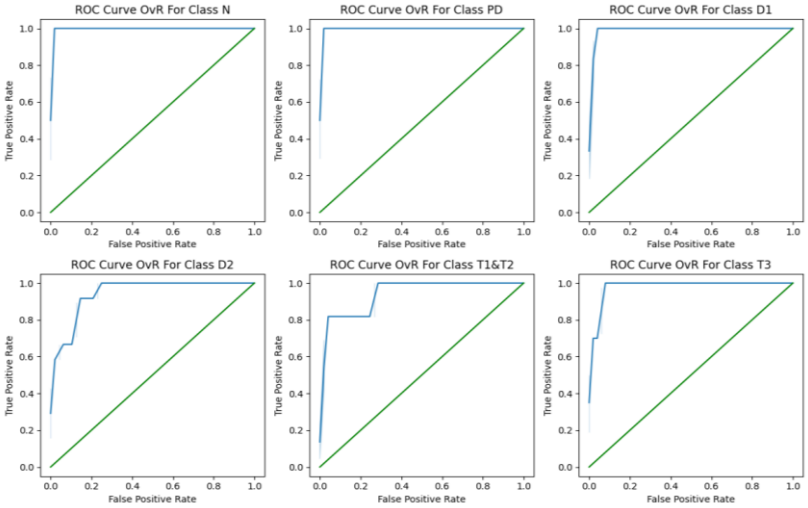

Figure 10 SVM ROC curve

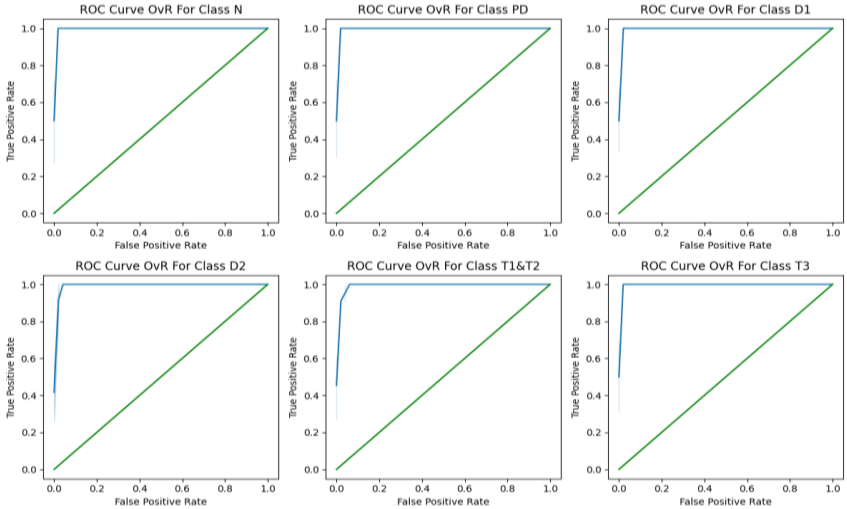

Figure 11 MLP ROC curve

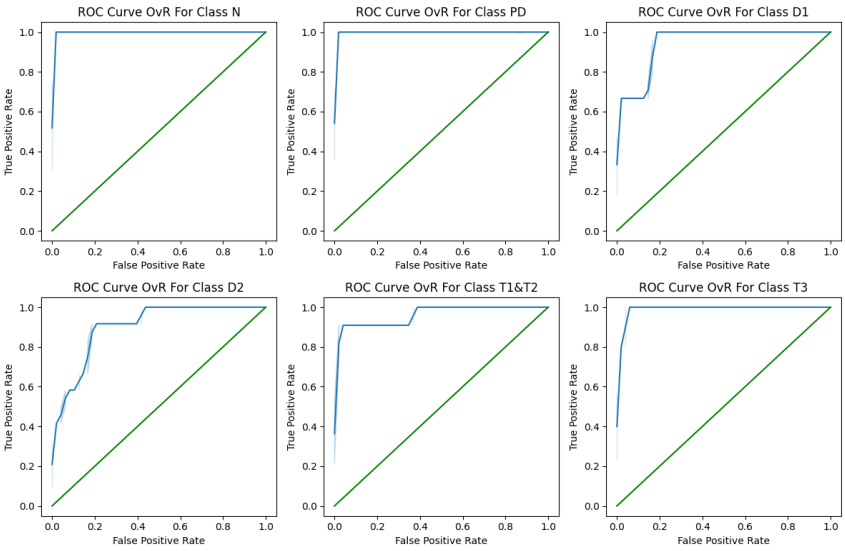

Figure 12 Random Forest ROC curve

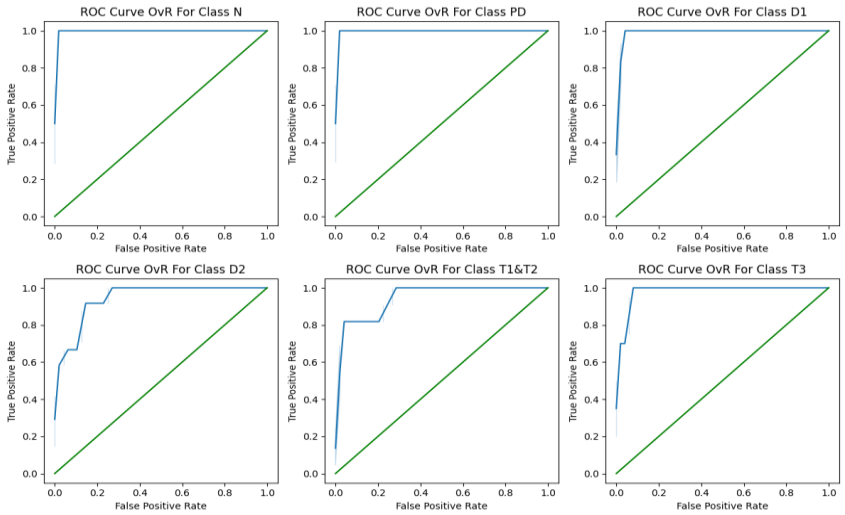

Figure 13 Bagging (SVM) ROC curve

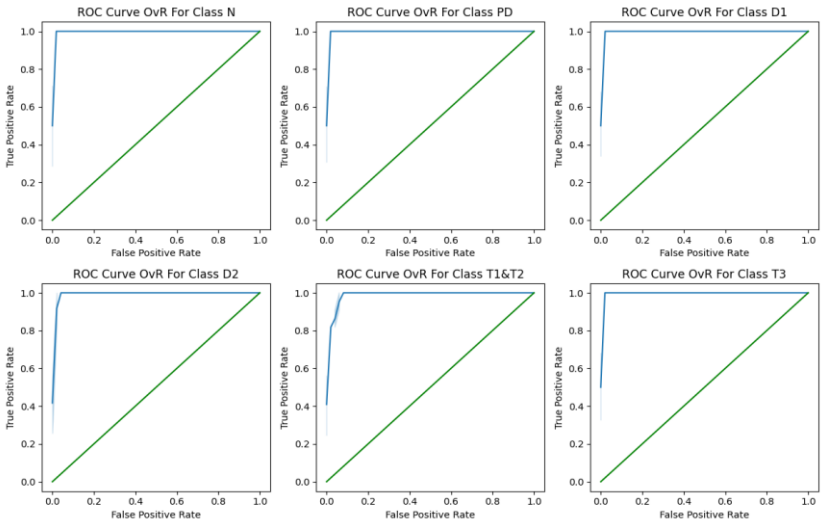

Figure 14 Bagging (MLP) ROC curve

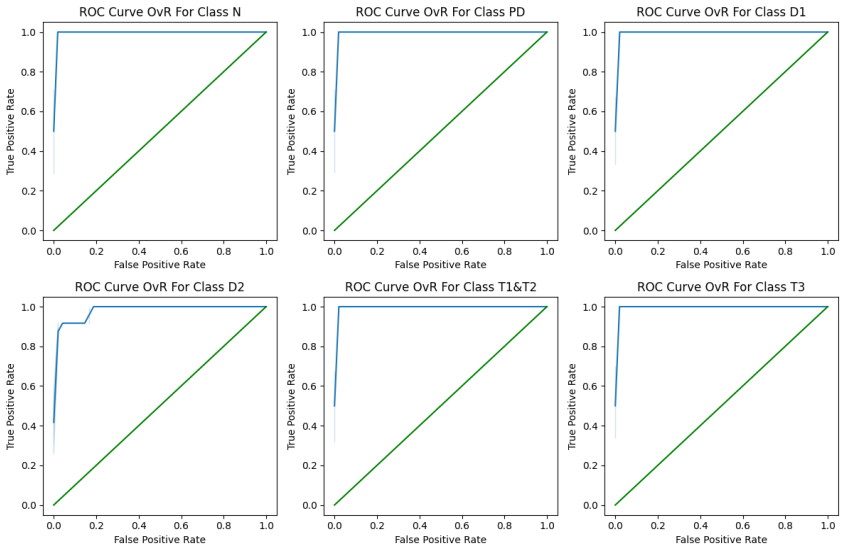

Figure 15 AdaBoost (Random Forest) ROC curve

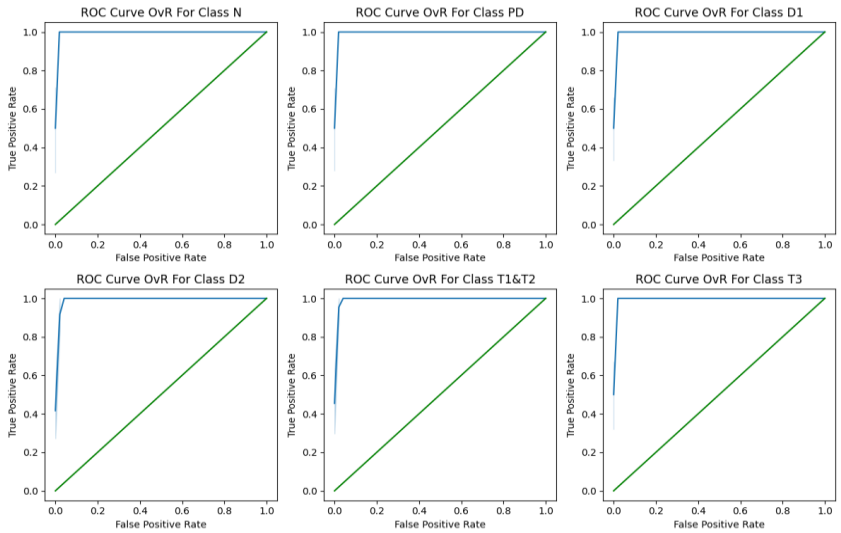

Figure 16 Stacking ROC curve

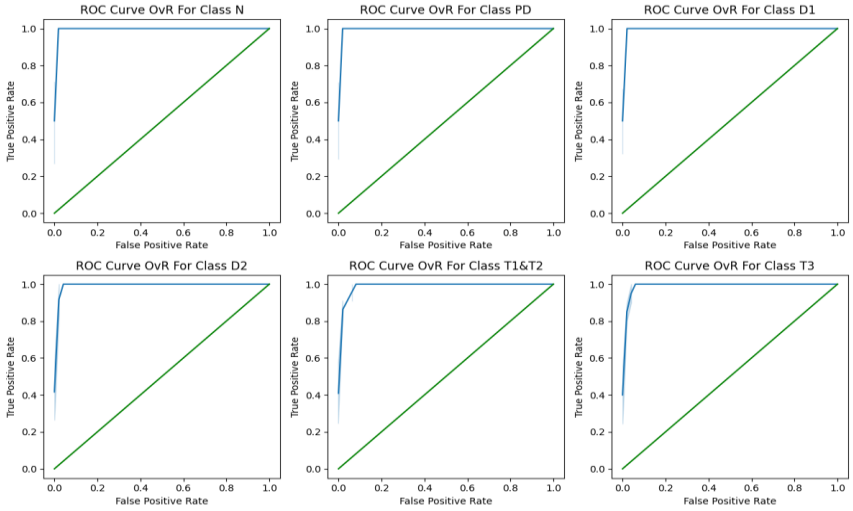

Figure 17 Voting ensemble ROC curve

7. Conclusion

This research demonstrates the advancements realized through the utilization of ensemble techniques in dissolved gas analysis for the classification of faults in power transformer. The DGA dataset used was first preprocessed using oversampling techniques, then normalized and finally cross validated respectively. We then created and evaluated ensemble methods including bagging, AdaBoost, stacking, Random forest and simple voting schemes as well as SVM and MLP base models to discover the most performing classification model. The findings unequivocally established that the AdaBoost ensemble implemented with a random forest as its base learner surpasses all other intelligent approaches, achieving a classification accuracy of 100%. For future research purpose, we plan to employ more robust real-world datasets, also introduce feature engineering techniques in determining principal features and feature relevance to use in modeling. Finally, the base classifier's parameters will also be tuned and carefully selected for optimal performance with regards to improve accuracy, but reduced time and space complexities. Furthermore, by pairing transformers with online-connected field-test instruments and then connecting the final model from this investigation to a broader online monitoring system, the current work can be extended.

References

[1] Odongo, G., Musabe, R., Hanyurwimfura, D.: A Multinomial DGA Classifier for Incipient Fault Detection in Oil-Impregnated Power Transformers. Algorithms," 14(4), 128, (2021). https://doi.org/10.3390/a14040128

[2] Kari, T., Gao, W., Zhao, D., Abiderexiti, K., Mo, W., Wang, Y., Luan, L.: Hybrid feature selection approach for power transformer fault diagnosis based on support vector machine and genetic algorithm. IET Generation, Transmission & Distribution, 12(21), 5672-5680, (2018). https://doi.org/10.1049/iet-gtd.2018.5482

[3] Kirkbas, A., Demircali, A., Koroglu, S., Kizilkaya, A.: Fault diagnosis of oil-immersed power transformers using common vector approach. Electric Power Systems Research, 184, (2020). https://doi.org/10.1016/j.epsr.2020.106346

[4] Islam, M.M. Lee, G., Hettiwatte, S.N.: Application of Parzen Window estimation for incipient fault diagnosis in power transformers," High Voltage, 3(4), 303-309, (2018). https://doi.org/10.1049/hve.2018.5061

[5] Rokani, V., Kaminaris, S.D.: Power transformers fault diagnosis using AI techniques. AIP Conference Proceedings, 2307(1), 020056, 2020. https://doi.org/10.1063/5.0032820

[6] Bouchaoui, L., Hemsas, K.E., Mellah, H.: Power transformer faults diagnosis using undestructive methods (Roger and IEC) and artificial neural network for dissolved gas analysis applied on the functional transformer in the Algerian north-eastern: a comparative study. Electrical Engineering & Electromechanics, (4), 3–11, 2021. https://doi.org/10.20998/2074-272X.2021.4.01

[7] Wagh, N., Deshpande, D.M.: Investigations on Incipient Fault Diagnosis of Power Transformer Using Neural Networks and Adaptive Neurofuzzy Inference System. Applied Computational Intelligence and Soft Computing, 2014(845815), 1-9, 2014. https://doi.org/10.1155/2014/845815

[8] Muthi, A., Sumarto, S., Saputra, W.S.: Power Transformer Interruption Analysis Based on Dissolved Gas Analysis (DGA) using Artificial Neural Network. IOP Conference Series: Materials Science and Engineering, 384(012073), 1-5, 2018. doi.org/10.1088/1757-899X/384/1/012073

[9] Ma, H., Zhang, W., Wu, R., Yang, C.: A Power Transformers Fault Diagnosis Model Based on Three DGA Ratios and PSO Optimization SVM. IOP Conference Series: Materials Science and Engineering, 339(012001), 1-6, (2018). https://doi.org/10.1088/1757-899X/339/1/012001

[10] Zhang, W., Yang, X., Deng, Y., Li, A.: An Inspired Machine-Learning Algorithm with a Hybrid Whale Optimization for Power Transformer PHM. Energies, 13(12), 3143, (2020). https://doi.org/10.3390/en13123143

[11] Ghoneim, S.S., Taha, I.B.: A new approach of DGA interpretation technique for transformer fault diagnosis. International Journal of Electrical Power & Energy Systems, 81, 265-274, (2016). https://doi.org/10.1016/j.ijepes.2016.02.018
[12] Aburaghiega, E., Farrag, M.F. Hepburn, D., Haggag, A.: Enhancement of Power Transformer State of Health Diagnostics Based on Fuzzy Logic System of DGA. 2018 Twentieth International Middle East Power Systems Conference (MEPCON), 400-405, (2018). https://doi.org/10.1109/MEPCON.2018.8635154
[13] Apte, S., Somalwar, R., Wajirabadkar, A.: Incipient Fault Diagnosis of Transformer by DGA Using Fuzzy Logic. 2018 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES), 1-5, (2018). https://doi.org/10.1109/PEDES.2018.8707928
[14] Saravanan, D., Hasan, A., Singh, A., Mansoor, H., Shaw, R.N.: Fault Prediction of Transformer Using Machine Learning and DGA. 2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON), 1-5, (2020). https://doi.org/10.1109/GUCON48875.2020.9231086
[15] Faiz, J., Soleimani, M.: Assessment of Computational Intelligence and Conventional Dissolved Gas Analysis Methods for Transformer Fault Diagnosis," IEEE Transactions on Dielectrics and Electrical Insulation, 25(5), 1798-1806, (2018). https://doi.org/10.1109/TDEI.2018.007191
[16] Venkatesan, R., Er, M.J.: Multi-Label Classification Method Based on Extreme Learning Machines. 2014 13th International Conference on Control, Automation, Robotics & Vision Marina Bay Sands, Singapore, 10-12th December 2014 (ICARCV 2014), pp. 619-634, (2014).
[17] Duval, M., dePablo, A.: Interpretation of Gas-In-Oil Analysis Using New IEC Publication 60599 and IEC TC 10 Databases. Dielectrics and Electrical Insulation Society, 17(2), 31-41, (2001). doi.org/0883-7554/01/$10.00©2001
[18] Dhini, A., Faqih, A., Kusumoputro, B., Surjandari, I., Kusiak, A.: Data-Driven Fault Diagnosis of Power transformers using Dissolved Gas Analysis (DGA). International Journal of Technology. 11(2), 388-399, (2020). http://ijtech.eng.ui.as.id
[19] Subasi, A.: Practical Machine Learning for Data Analysis Using Python. Elsevier Science. United Kingdom, (2020).
[20] Jassim, M.A., Abdulwahid, S.N.: Data Mining preparation: Process, Techniques and Major Issues in Data Analysis. IOP Conf. Series: Materials Science and Engineering 1090 012053, 2021. doi:10.1088/1757-899X/1090/1/012053.
[21] Acuna, E., Rodriguez, C.: The Treatment of Missing Values and Its Effect on Classifier Accuracy. In: Classification, Clustering, and Data Mining Applications. Springer, Berlin, Heidelberg, 639-647, (2004). https://doi.org/10.1007/978-3-642-17103-1_60
[22] Chawla, N.V. Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic Minority Over-Sampling Technique. Journal of Artificial Intelligence Research. 16, 321-357, (2002). https://doi.org/10.1613/jair.953
[23] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12(85), 2825-2830, (2011). https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.htm
[24] Vapnik, C.V..: Support-vector networks. Machine Learning," 20, 3, 273, (1995). doi:10.1007/BF00994018.
[25] Nield, T.: Essential Math for Data Science: Take Control of Your Data with Fundamental Linear Algebra, Probability, and Statistics. O'Reilly, (2022).
[26] Grus, J.: Data Science from Scratch: First Principles with Python. O'Reilly Media, (2019).
[27] Oshiro, T.M., Perez, P.S., Baranauskas, J.A.: How Many Trees in a Random Forest? Lecture Notes in Computer Science, 7376, 155-168, (2022). http://dx.doi.org/10.1007/978-3-642-31537-4_13
[28] Zhou, Z.-H.: Ensemble Methods: Foundations and Algorithms. CRC Press, (2012).
[29] Zhu, J., Rosset, S., Zou, H., Hastie, T.: Multi-class AdaBoost. Statistics and its Interface. 2(3), (2006). http://dx.doi.org/10.4310/SII.2009.v2.n3.a8

[30] Wolpert, D.H.: Stacked generalization. Neural Networks. 5(2), 241-259, (1992). https://doi.org/10.1016/S0893-6080(05)80023-1
[31] Cichosz, P.: Data mining algorithms: Explained using R. West Sussex, UK: John Wiley & Sons, 2014.
[32] Flach, P.: Machine learning: The art and science of algorithms that make sense of data. Cambridge, UK: Cambridge University Press, (2012).
[33] Seijo-Pardo, S., Porto-Diaz, I., Bolon-Canedo, V., Alonso-Betanzos, A.: Ensemble Feature Selection: Homogeneous and Heterogeneous Approaches. Knowledge-Based System. 118:124-139 (2017).
[34] Biau, G.: Analysis of a random forests model. J. Mach. Learn. Res. 13, 1063–1095 (2012).
[35] Senoussaoui, M.E., Brahami, M., Fofana, I.: Combining and comparing various machine-learning algorithms to improve dissolved gas analysis interpretation. IET Gener. Transm. Distrib. (2018), 12, 3673–3679