HKIE THE HONG KONG INSTITUTION OF ENGINEERS 香港工程師學會

Machine learning model for dissolved gas analysis: methodological review with a case in Hong Kong

Philip TH Chan and Terry HW Chan
MTR Corporation Limited, Hong Kong, People's Republic of China

ABSTRACT

Dissolved gas analysis is a valuable diagnostic tool used to monitor transformer health by analysing the gases dissolved in insulation oil. However, its practical application is hindered by the absence of a universal standard, leading to varied interpretations and implementations across different contexts. Scholars have turned to machine learning to advance DGA anomaly detection, but the existing literature prioritises model development over methodological rigour; issues such as dataset imbalance, appropriate evaluation metrics, and testing and validation procedures are often overlooked. This study addresses the existing gaps by critically reviewing the methodological steps involved in machine learning modelling with DGA datasets. The proposed design considerations are justified, and the analysis of the IEC TC 10 dataset and a dataset from a Hong Kong company is enhanced. Using Python's scikit-learn, over 18 combinations of datasets, data preprocessing techniques, and model architectures were assessed, and the random forest model ($F1$ = 0.88) and the support vector classifier with an RBF kernel ($F1$ = 0.87) were identified as the top-performing models, after applying Yeo-Johnson transformation. The models' source code is available on the author's GitHub repository.

KEYWORDS Electrical Engineering; machine learning; dissolved gas analysis (DGA); anomaly detection; transformer
CONTACT Philip TH Chan philip.chan.th@gmail.com
Received 4 March 2024

1. Introduction

Insulation oil is essential in electrical transformers, serving to insulate and cool the transformer's core and windings, thus preventing electrical breakdown and overheating. Chemically, insulation oil consists of various hydrocarbons and additives designed to improve performance and enhance the resistance to oxidation, moisture, and other factors (Kaplan et al., 2010). When a transformer experiences faults such as overheating, electrical arcing, or partial discharge, the resulting electrical or thermal stress causes the insulation oil to decompose, generating dissolved gases such as methane, ethylene, and hydrogen. Chemical theory explains that different energy levels associated with transformer faults lead to the formation of specific gases-hydrogen and methane are linked to low-energy faults, while higher energy levels can produce ethylene and acetylene. By analysing the composition of these gases through dissolved gas analysis (DGA), it is possible to identify the underlying issues in the transformer. For example, IEC 60599 outlines seven key gases and relates them to various fault types, where D1 means "discharges of low energy", T1 means "thermal fault <300°C", etc. The primary focus will be on the key gases which are H_2, CH_4, C_2H_2, C_2H_4, C_2H_6, CO, and CO_2.

1.1. Traditional DGA approaches

Certain gas patterns are widely recognised as indicative of specific types of faults in a transformer. Traditional approaches, such as the Rogers Ratio method (Rogers, 1978), the Key Gas method (Bagheri et al., 2022), and the Duval Triangle method (Duval, 1989), rely on fixed thresholds and empirical rules to interpret the gas concentrations. The Rogers Ratio method diagnoses faults by comparing gas ratios, the Key Gas method links specific gases to particular faults, and the Duval Triangle uses a graphical representation of gas concentrations to identify the fault types.

Due to the complex nature of gas generation, the presence of one or multiple gases does not always indicate specific faults. Assessing the health of a transformer using traditional methods is often challenging, particularly because these methods can be inflexible and may not adapt well to varying transformer conditions; it is also uncertain whether the insights derived from the large transformers in these studies can apply to those with smaller ratings (Mirowski and LeCun, 2012). Relative gas ratios can also be ineffective, as a slight deviation from the fault zone can lead to false diagnoses (Duval and Dukarm, 2005). In practice, when operators and maintainers interpret the laboratory reports on gas concentration, they typically treat the thresholds as a starting point and must consider additional factors, such as maintenance logs and other qualitative data. These interpretations are highly dependent on the maintainer's knowledge, and in the experience of the authors, they can often derive conflicting conclusions from the same DGA values.

1.2. Machine learning as an alternative

When rule-based methods struggle with anomaly detection due to the absence of a universal standard, machine learning (ML) presents a promising alternative. ML techniques, once trained on extensive datasets, offer several advantages over traditional methods, including the automation of analysis and the ability to generate insights without requiring deep domain expertise. For DGA, ML models can detect patterns indicative of transformer faults and automate analysis without needing an exhaustive understanding of the chemical properties of each gas. This automation extends to cases where data custodians can adapt a trained model to their existing and evolving datasets, tailor the model to their specific transformer fleet, and align it with their organisational strategies.

This study frames a supervised binary classification problem, aiming to classify DGA test results as "healthy" or "faulty" using a labelled DGA dataset. Section 2 critically reviews the methodologies, addresses the limitations found in prior studies, and introduces an incremental sampling technique to tackle dataset imbalance. To enhance model interpretation, a dataset was sourced from a Hong Kong company, and model performance was compared. Section 3 details the data preprocessing and modelling processes, using the scikit-learn Python library as the primary tool. Python's flexibility and scalability make it the de facto standard choice for exploratory ML, with scikit-learn proving efficient for experimentation and model prototyping. The source code for the models is published on the author's GitHub repository. Interested readers are encouraged to conduct their own experiments and validations.

2. Literature review

This study identifies numerous papers within the related field, and efforts have been concentrated on comparing the methodologies, including data preprocessing techniques, dataset sources, and evaluation methods.

Some preliminary findings include that most studies (Fang et al., 2018; Rao et al., 2023; Zhang et al., 2020) use the IEC TC 10 database, while others rely on private datasets (Ekojono et al., 2022; Thango, 2022). Many training datasets are either moderately or severely imbalanced, yet this issue is often not explicitly addressed. Additionally, as far as the research covers, only Mirowski and LeCun (2012) have made their source code publicly available, and only the results from this study could be verified and reproduced.

2.1. Data preprocessing

DGA values exhibit notable right-skewness and heteroscedasticity, which can hinder the performance of many statistical models, including those used in this study (Antonini et al., 2019; Guo et al., 2019). To address the errors arising from data variability, common strategies include normalisation (Zhang et al., 1996), standardisation, and log-transformation (Mirowski and LeCun, 2012). However, these studies did not compare the effectiveness of different transformations on DGA values, and some did not consider preprocessing at all (Shang et al., 2019; Thango, 2022). As an example, Table 1 illustrates the effects of applying data transformation before (with data leakage) and after (without leakage) the train-test split.

2.2. Cross-Validation and Evaluation Metrics

In conventional ML practice, a dataset is divided into a training set and a testing set, with a subset of the training data sometimes used as a validation set for hyperparameter tuning and model evaluation (Kubat, 2017). While the said terminology is widely accepted, inconsistencies arose when the terms were used interchangeably (Ekojono et al., 2022), when the same data were used for both training and testing (Thango, 2022), or when cross-validation was improperly executed (Shang et al., 2019). Some studies (Equbal et al., 2018; Illias and Zhao Liang, 2018) did not perform cross-validation and lacked clarity regarding the hyperparameter tuning procedures. An excessively high training accuracy, e.g., over 90% after only a few iterations when training neural networks (Shang et al., 2019), typically hints at overfitting or improper training.

2.3. Balancing the dataset

In imbalanced datasets, minority classes often exert less influence during model training, which could potentially yield unreliable diagnoses as models may converge to local optima (Li et al., 2022). For DGA-related studies, the imbalance ratio can range from 5:1 (Yuan et al., 2019) to as severe as 10:1 for minority classes (Shang et al., 2019). The TC 10 dataset shows a relatively moderate imbalance ratio of 2.2:1.

When the data for minority classes are sparse, imbalances can be addressed through oversampling or undersampling. Oversampling the dataset, however, is not imperative, neither does it always guarantee performance improvement; whether the endeavour is favourable is dependent on the data size, data complexity, model type

Table 1. Comparing the performance with and without data leakage.

Architecture	Dataset	Preprocessing	Accuracy	ROC AUC	Precision	Recall	PR AUC	F1 Score
SVC (rbf)	[5]	Data leakage	0.829787	0.846154	0.724138	1	0.724138	0.84
SVC (rbf)	[5]	No data leakage	0.808511	0.826923	0.7	1	0.7	0.823529

etc; in practice, it is often a trial-and-error process. Studies have shown mixed results, with some finding significant improvements in accuracy and AUC from dataset balancing (Guo et al., 2019; Mirowski and LeCun, 2012), while others report similar benefits from hyperparameter tuning alone (Laurikkala, 2001; Zheng and Jin, 2020).

2.4. Evaluation metrics

In the reviewed literature, accuracy is the most frequently employed evaluation metric (Fang et al., 2018; Rao et al., 2023; Shang et al., 2019). Other studies (Ekojono et al., 2022; Fang et al., 2018) use additional metrics such as AUC, precision, recall, and F1 score for hyperparameter tuning, but rely solely on accuracy for model assessment, despite acknowledging the dataset's inherent imbalance. For general performance assessment, Japkowicz (2013) and Jeni et al. (2013) provide extensive lists of recommended metrics for addressing imbalanced datasets.

While numerous studies apply ML to DGA datasets for transformer fault diagnosis, few provide a comprehensive methodological framework beyond model development, and only Mirowski and LeCun (2012) effectively address challenges specific to DGA, such as dataset imbalance and long-tail data distributions. Overall, this field lacks a comprehensive review of issues like data preprocessing, cross-validation, and model evaluation, which impedes the reproducibility and comparability across studies employing ML techniques for DGA.

3. Methodology and implementation

Supervised machine learning presents challenges primarily in classification or regression tasks. The former categorises outputs into discrete classes and the latter predicts continuous quantities. The ML models aim to classify transformer DGA outcomes into two categories: faulty and healthy (non-faulty). Consider a dataset $\mathcal{D}$ comprising pairs (x_i, y_i) where $x_i \in R^7$ corresponds to the concentration of seven dissolved gases in an oil sample and $y_i \in \{0,1\}$ denotes the binary label indicating the transformer's condition: 0 for healthy (negative) and 1 for faulty (positive). The objective is to develop a predictive model $f: R^7 \rightarrow \{0,1\}$ that accurately determines whether a given oil sample originates from a healthy or faulty transformer.

This study primarily uses the scikit-learn Python library for modelling. Scikit-learn is a widely popular library for classical ML models for quick prototyping and finetuning, as attested by over 900 conference papers in IEEE Xplore® that use it for big data analysis and practical data science problems. For example, initializing an SVC with a RBF kernel requires only a single line of code:

```
clf = SVC(kernel='rbf')
```

Training the model is equally straightforward:

```
clf.fit(X, y)
```

This ease of operation, however, should be balanced with a competent understanding of the library's model architectures, design considerations, and the interplay of various methods and parameters. The models from Mirowski and LeCun (2012), which were originally implemented in MATLAB, were also recreated. This validates the choice of using Python libraries as it was possible to achieve comparable results with significantly less effort.

3.1. IEC TC 10 database

In ML, obtaining more data is akin to the quest for the "holy grail", as a larger dataset typically allows a model to identify finer distinctions. This study has undertaken an extensive search for relevant datasets; however, the IEC TC 10 database (as cited by Duval and DePabla [2001]) emerged as providing the only publicly available DGA data from a reputable organisation, alongside a few other papers that list their data within their publications (Sarma and Kalyani, 2004; Zhang et al., 1996). The scarcity of data is compounded by the fact that most of them are neither readily accessible nor machine-readable (often presented in PDF tables), thus complicating the creation of a continuous integration/continuous deployment pipeline. Since many studies in the field rely on the IEC TC 10 database as a foundational resource, to promote comparability, it was also incorporated into the training data.

The IEC TC 10 database gives five tables of DGA data with a total of 117 sets of positive results, extracted from various reports and journals. These are categorised by the fault codes representing the types of failures and is hereinafter referred to as the "TC 10 dataset".

This study follows the convention that the DGA results obtained after the transformer experienced a fault are labelled as 'positive' or 'faulty' data. In other words, when the model predicts a positive result, it should indicate that the input suggests a fault in the transformer from which the oil sample was taken. As with typical anomaly detection problems, obtaining positive data is challenging because (1) most transformers in operation have a regular (preventive) maintenance regime, operate within their recommended lifespan and rating, and so, empirically, are less prone to faults; and (2) such data might be sensitive to a power company or corporation, and they usually have little incentive to release their data publicly.

Data from the contrary class (labelled as "negative" or "healthy" data) are also essential for model training, but this classification is even more elusive. Although many transformers may never experience faults, the notion of "healthy" is nebulous as transformer health exists on a spectrum rather than a binary state, and there

is no universally accepted definition. On top of standard values (such as the "typical values" described in IEC 60599), practitioners often consider operational factors like service year, loading, transformer type, rating, and oil type. To complicate the matter more, the insulation oil in transformers begins to degrade as soon as the transformer is energised, albeit slowly. Factors such as heat, moisture, and contaminants, and actions like oil filtering and topping up can cause huge changes in the gas concentrations, both making it difficult to obtain representative data. As the author's experience attests, it is possible for DGA results from a transformer to fall outside the typical thresholds, yet still allow the transformer to function for years without apparent faults.

Annex 2 of Duval and DePabla (2001) presents four tables of typical DGA values from transformers in service, comprising a total of 50 negative results. The paper does not provide extensive clarification on the labels "typical" or "normal". Nonetheless, these values are generally interpreted as the antitheses of those in Annex 1. For comparability, Annex 2 is incorporated into the negative dataset in this study.

3.2. Balancing the dataset

As discussed in Section 2.3, although data balancing is not strictly necessary for the current use cases, several balancing experiments are conducted to explore their potential impact.

3.2.1. Synthetic minority oversampling technique

Many studies benefit from large corporate databases that allow for undersampling, but the TC 10 dataset (with 117 positive results and 50 negative results) is too small for such an approach. The Synthetic Minority Oversampling Technique (SMOTE) addresses this limitation by generating synthetic minority samples through interpolation between neighbouring instances. The underlying assumption is that if a cluster of data points represents a particular class, then new points interpolated within this cluster should also belong to that class (Chawla et al., 2002). The technique is incentivised in the case of DGA due to (1) the initially limited and dispersed nature of DGA data points; and (2) observed improvements in metrics like F1, suggesting that the introduced noise is manageable. SMOTE enriches the minority class without simply duplicating the existing instances, avoids introducing too many noises into the training data, and improves the model's ability to generalise to unseen data.

For each chosen instance of samples from the minority class, SMOTE identifies its k-nearest neighbours within the feature space. A new synthetic instance x_{new} is generated by $x_{new} = x + \lambda \bullet (x_{nn} - x)$, where x_{nn} is the randomly selected nearest neighbour, and λ is a random number. The algorithm continues until the desired balance of the minority and majority classes is achieved. The parameters can be adjusted depending on the dataset's characteristics and the desired level of oversampling. For example, a lower k gives synthetic instances that are closer to each other, thus reducing the variability, whereas a higher k introduces more diversity. The balanced dataset sampled by SMOTE will be referred to as the "SMOTE dataset".

3.2.2. Hong Kong dataset

An alternative balancing strategy involves integrating DGA data from a Hong Kong company, which manages a network of several hundred transformers subjected to regular maintenance and oil sample tests in laboratories. To balance the TC 10 dataset, 76 sets of negative data were collected from this source, originating from oil samples extracted from transformers at least three years prior to significant maintenance events, e.g., oil filtering or replacement, or transformer refurbishment. This negative dataset operates under the assumption that if the models classify a sample as 'negative', the corresponding transformer would be able to remain in service for at least three years without requiring significant maintenance. This combined dataset is denoted as the "Hong Kong dataset" and is available on the author's GitHub repository.

3.3. Feature scaling

The distribution of DGA results exhibits notable right-skewness and heteroscedasticity, with values ranging from negligible amounts to several tens of thousands. Training models directly on such data may introduce biases, as larger values can overshadow the subtler variations in smaller values. To mitigate this skewness, the Yeo-Johnson transformation was used (Yeo, 2000), denoted as $\phi(x;\lambda)$. This transformation stabilises variance and makes the data more Gaussian-like:

$$\phi(x;\lambda) = \begin{cases} \frac{[(x+1)^{\lambda}-1]}{\lambda} & \text{if } \lambda \neq 0, x \geq 0 \\ \ln(x+1) & \text{if } \lambda = 0, x \geq 0 \\ -\frac{[(-x+1)^{(2-\lambda)}-1]}{(2-\lambda)} & \text{if } \lambda \neq 2, x < 0 \\ -\ln(-x+1) & \text{if } \lambda = 2, x < 0 \end{cases} . \quad (1)$$

This transformation offers flexibility through the parameter λ to determine the degree of the transformation (for positive x) and to ensure continuity and symmetry (for negative x), allowing for precise adjustments.

The data were subsequently standardised using the z-score transformation (herein referred to as "z-transform"). In some models, adherence to a normal distribution is necessary; for instance, a Support Vector Classifier aims to maximise the margin between different classes in the feature space. When features are not normalised, the algorithm may disproportionately weight features with larger scales. Normalisation also helps in distributing data

Figure 1. The distributions of CO, CO_2 and C_2H_4 after transformations are closer to normal. The Yeo-Johnson transformation noticeably outperforms the log-transformation, particularly for CO_2 and C_2H_4.

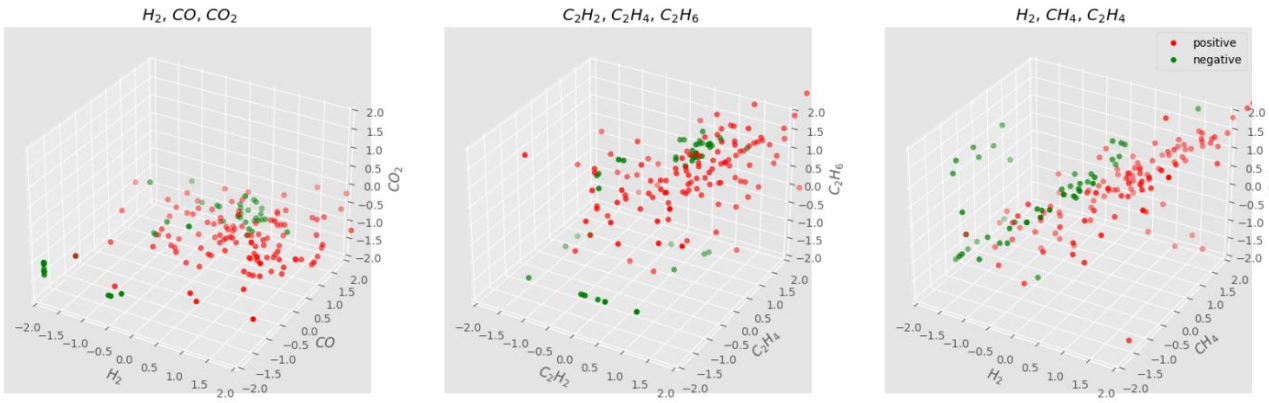

Figure 2. The TC 10 Dataset (Duval and DePabla, 2001), represented by 7-dimensional vectors, is visualised in three 3D plots. For clarity and sparsity, the dataset undergoes Yeo-Johnson transformation before presentation.

points more evenly across the feature space (see Figure 1) such that patterns can be identified and decision boundaries defined better.

Figure 2 visualises the TC 10 dataset. Certain features, prima facie, distinguish between positive and negative data points clearly (e.g., H_2 in the right-most plot), while other features do not offer such clear differentiation.

3.4. Train-test split

The dataset $\mathcal{D}$ is randomly partitioned into training and testing subsets, denoted as $\mathcal{D} = D_{\text{train}} \cup D_{\text{test}}$ with 80% of the data allocated to $\mathcal{D}_{\text{train}}$ and the remaining to $\mathcal{D}_{\text{test}}$. The latter serves as a novel dataset to evaluate the model's performance metrics by comparing the predicted values $\hat{y}$ against the actual values y using a predefined metric or loss function.

A train-test split is always necessary for preventing overfitting, i.e., a model might memorise the noise in the training data such that it negatively impacts its performance on unseen data. Furthermore, applying a train-test split before any data transformations is crucial to prevent data leakage. Consider a case where the mean and standard deviation of the entire dataset are used for standardisation before the train-test split. This results in a leakage of information, where the training data are falsely enhanced with knowledge of the test set's distribution. As can be seen in Table 1, the effect of data leakage is significant.

3.5. Model architectures

This paper implements four classical classifier models: the k-NN classifier, Support Vector Classifier, Random Forest, and a supervised neural network with a single hidden layer. These models were selected based on their widespread use and proven efficacy in various studies. Figure 3 provides an intuitive understanding of the models' mechanisms and illustrates how they classify new sets of DGA values by determining their placement within the decision boundaries. This visualisation simplifies the actual models using only two features, C_2H_2 and C_2H_4, to produce 2D plane graphs.

3.5.1. k-NN classifier

The k-nearest neighbours (kNN) algorithm classifies a new instance x by exploiting the proximity of data points in $\mathcal{D}$. Although the Euclidean distance is a common choice for measuring proximity, scikit-learn uses the BallTree data structure (specified with the keyword algorithm = 'ball_tree') for efficient nearest neighbour searches. BallTree organises data points into a binary tree structure that partitions the feature space into hierarchical regions, allowing for the faster and more efficient querying of nearest neighbours. The algorithm's performance is highly contingent on the choice of k: a smaller k increases the sensitivity to noise, while a larger k smooths the decision

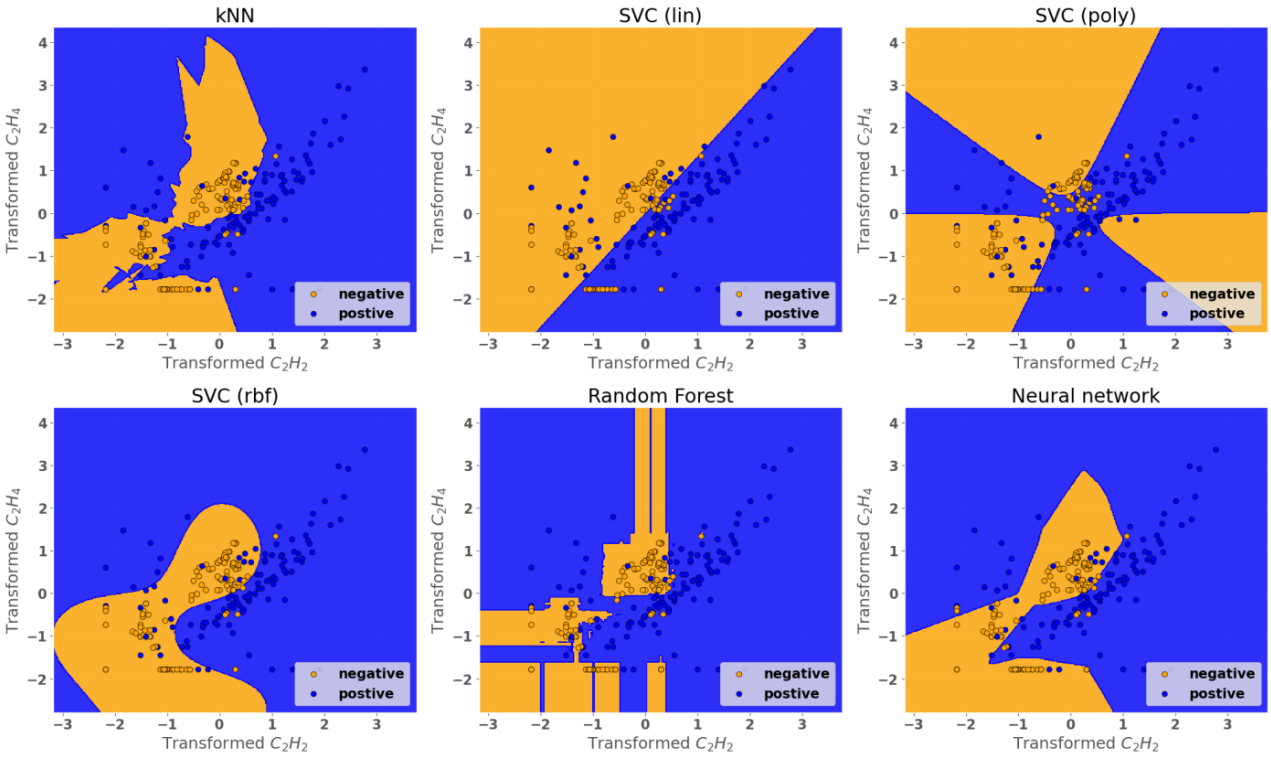

Figure 3. A demonstration of the decision boundary trained on transformed C_2H_2 and C_2H_4. SVC (lin) refers to a linear kernel, and SVC (poly) a third-degree polynomial kernel.

boundary. The optimal k is generally determined through cross-validation.

3.5.2. Support vector classifier

For a binary classification problem, a Support Vector Machine or Classifier (SVC) seeks to find w and b such that the hyperplane defined by $w^T x + b = 0$ maximises the distance between the closest data points of each class. This can be formulated as:

$$\min_{\mathrm{w},b} \frac{1}{2} \| \mathrm{w} \|^2 , \tag{2}$$

subject to:

$$y_i(\mathrm{w}^T \mathrm{x} + b) \geq 1 \quad \text{for } i = 1,2,\ldots,n , \tag{3}$$

where w is the weight vector, b is the bias term, and $(\mathrm{x}_i, \mathrm{y}_i)$ are the training samples with labels $y_i \in \{-1, +1\}$ for n samples (Bishop, 2006).

For datasets not separable by linear boundaries, SVCs use the kernel trick, which enables the model to operate in a high-dimensional Hilbert space where the inner products $\phi(\mathrm{x}_i) \cdot \phi(\mathrm{x}_j)$ are computed implicitly, and the classes may become linearly separable, before projection back to the original dimension. This method is very efficient because directly performing these transformations, such as second-degree polynomial mapping, would have a complexity of $O(n^2)$ for n features, making it computationally expensive.

Common kernels include the second-degree polynomial kernel:

$$K(\mathrm{x}_i, \mathrm{x}_j) = (\mathrm{x}_i \cdot \mathrm{x}_j + c)^2 , \tag{4}$$

which models the quadratic interactions between features, and the Radial Basis Function (RBF) kernel:

$$K(\mathrm{x}_i, \mathrm{x}_j) = \exp(-\gamma \| \mathrm{x}_i - \mathrm{x}_j \|^2) , \tag{5}$$

which maps data into an effectively infinite-dimensional space, allowing it to capture complex, non-linear relationships with γ controlling the kernel's impact on the decision boundary (Chang et al., 2010). A lower γ value yields a smoother boundary, potentially enhancing its generalisability, which may improve generalisability but risks underfitting.

3.5.3. Random forest

The Random Forest model is an ensemble learning technique that combines multiple decision trees to improve the prediction accuracy and reduce overfitting. Mathematically, the Random Forest model gives the average of the predictions made by N individual decision trees:

$$f_{\mathrm{RF}}(\mathrm{x}) = \frac{1}{N} \sum_{n=1}^{N} f_{\mathrm{DT}_n}(\mathrm{x}) , \tag{6}$$

where N is the number of trees in the forest, $\mathrm{x} \in \mathbb{R}^7$ is the input feature vector, and $f_{\mathrm{DT}_n}(\mathrm{x})$ is the prediction of the n-th decision tree. Each decision tree f_{DT_n} is trained using a bootstrap sample of the training data and a random subset of features at each split. The decision trees recursively partition the feature space based on feature values that best separate the classes.

3.5.4. Supervised neural network

A neural network is formed by sequentially interconnecting tensors via weighted connections; a tensor receives input x, undergoes a linear transformation $Wx + b$, with weights W and biases b, and the result is processed by an activation function, such as the Rectified Linear Unit (ReLU) $f(x) = \max(0, x)$, to introduce non-linearity into the model.

Weights and biases are iteratively adjusted using an optimization algorithm, such as the Stochastic Gradient Descent (SGD):

$$\mathrm{W}_{t+1} = \mathrm{W}_t - \eta \cdot \nabla L(\mathrm{W}_t) , \tag{7}$$

where W_t represents the weights at iteration t, η is the learning rate, and $\nabla_{\mathrm{W}} L(\mathrm{W}_t)$ is the gradient of the loss function L with respect to the weights at iteration t. The iterative process aims to minimise the loss function and optimises the model's performance on the training data.

Section 4 reports the model results of a one-layer neural network with ReLU activation, trained with SGD. With the simplicity and limited size of the dataset, the experiments conclude that more complex architectures are superfluous and do not significantly enhance performance.

4. Results

In evaluating the model's performance, the F1 score is prioritised, which is the harmonic mean of precision and recall, as the main metric due to its balanced consideration of these two measures. It is particularly helpful for this case where the costs of false positives and false negatives are equally significant. Although accuracy is a straightforward and easily interpretable measure, relying solely on it can be problematic since it may be skewed by the dominance of the majority class (Laurikkala, 2001). The F1 score, by addressing class imbalance, provides a more equitable assessment. For comparison, these metrics are also presented in Table 2.

In transformer health analysis, complicated metrics may present challenges for engineers with a limited data science background. Epigrammatically, a model of 90% accuracy can be paraphrased as "I can trust the model's predictions 90% of the time", whereas a 90% F1 score

means "When this model tells me something is true, I can trust it 90% of the time to be both accurate and comprehensive".

Visually, the receiver operating characteristic (ROC) curve depicts the performance of a classification model across various thresholds by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR). While it is a popular representation for overall model performance, ROC curves can be overly optimistic in cases of imbalanced datasets where the minority class is significantly underrepresented. As an alternative, the precision-recall (PR) curve focuses on the trade-off between precision and recall and provides a more informative evaluation.

Table 2. Formulas for the metrics for evaluating the model's performances. Note that Area Under the Curve (AUC) measures are estimated using numerical methods.

Metric	Formula
F1 score	$\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$
Precision	$\frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$
Recall	$\frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$
Accuracy	$\frac{\text{True Positives} + \text{True Negatives}}{\text{Total Samples}}$
AP	$\sum_{n}(\text{Precision atn}) \times (\text{change in Recall atn})$

Table 3. A summary of the model evaluation metrics of different ML models.

Architecture	Dataset	Preprocessing	Accuracy	ROC AUC	Precision	Recall	PR AUC	F1
kNN	TC 10	Y–J, z-transform	0.852071	0.920239	0.934579	0.847458	0.948939	0.888889
SVC (lin)	TC 10	Y–J, z-transform	0.840237	0.899385	0.888889	0.881356	0.952866	0.885106
SVC (poly)	TC 10	Y–J, z-transform	0.816568	0.905367	0.832061	0.923729	0.960458	0.875502
SVC (rbf)	TC 10	Y–J, z-transform	0.899408	0.951894	0.93913	0.915254	0.979234	0.927039
Random Forest	TC 10	Y–J, z-transform	0.91716	0.963443	0.933333	0.949153	0.983286	0.941176
Neural Network	TC 10	Y–J, z-transform	0.786982	0.872798	0.853448	0.838983	0.943727	0.846154
kNN	SMOTE	Y–J, z-transform	0.894068	0.955508	0.989474	0.79661	0.954226	0.882629
SVC (lin)	SMOTE	Y–J, z-transform	0.881356	0.932239	0.932692	0.822034	0.952049	0.873874
SVC (poly)	SMOTE	Y–J, z-transform	0.79661	0.900208	0.79661	0.79661	0.928254	0.79661
SVC (rbf)	SMOTE	No transform	0.860727	0.861775	0.986667	0.732246	0.926965	0.827197
SVC (rbf)	SMOTE	Log-transform	0.915426	0.914493	0.946338	0.889493	0.945309	0.911418
SVC (rbf)	SMOTE	Y–J transformation	0.961968	0.961594	0.983304	0.940217	0.976566	0.960715
Random Forest	SMOTE	Y–J, z-transform	0.961864	0.990305	0.973913	0.949153	0.991316	0.961373
Neural Network	SMOTE	Y–J, z-transform	0.851695	0.911556	0.927835	0.762712	0.93166	0.837209
kNN	Hong Kong	Y–J, z-transform	0.985915	0.99978	0.991228	0.957627	0.999276	0.974138
SVC (lin)	Hong Kong	Y–J, z-transform	0.99061	0.99923	0.991379	0.974576	0.997782	0.982906
SVC (poly)	Hong Kong	Y–J, z-transform	0.988263	0.99967	0.991304	0.966102	0.999103	0.978541
SVC (rbf)	Hong Kong	Y–J, z-transform	0.992958	0.999477	0.983193	0.991525	0.998546	0.987342
Random Forest	Hong Kong	Y–J, z-transform	0.995305	0.999518	0.983333	1	0.998681	0.991597
Neural Network	Hong Kong	Y–J, z-transform	0.983568	0.998679	0.982609	0.957627	0.996801	0.969957

Cross-validation was used to assess and benchmark the performance of the models by partitioning the $\mathcal{D}$ into five subsets, $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$, $\mathcal{D}_4$, $\mathcal{D}_5$, iteratively using four for training and one for validation, such that every data point is used for both training and validation. The average performance indicator, $\overline{P} = \frac{1}{5}\sum_{i=1}^{5} P_i$, is calculated by averaging the performance metrics P_i obtained from each of the five folds, and this averaged metric is less susceptible to the idiosyncrasies of any subset of the data.

4.1. Data imbalance

With appropriate data transformations, the performance of models trained on imbalanced datasets has a marginal difference to those trained on balanced datasets. Experiments conducted on both balanced and imbalanced datasets (Table 3) suggest no significant difference in the models' performance. For instance, using Yeo-Johnson transformation with an RBF SVC, the F1 scores for models trained on the imbalanced TC 10 dataset and the SMOTE dataset are consistently high, suggesting that the addition of synthetic data had minimal impact. Incorporating real data from a Hong Kong company remains valuable, as it offers insights into practical, real-world scenarios.

The PR curve and ROC curve in Figure 4 visualise the models' robustness across different datasets.

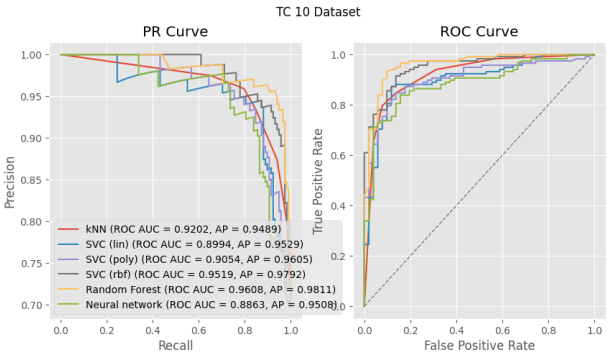

Figure 4. The PR curve and ROC curve of an RBF SVC, trained on TC 10 datasets. Notice how the curves converge to over 0.9 precision (left) and over 0.9 TPR (right).

4.2. Data leakage

Data leakage occurs when data transformations are applied before the train-test split. When transformations are applied only after the dataset is split, the metrics show a more gradual decline in precision with decreasing recall, indicating that the model is generalising and learning from the features of the data rather than remembering and regurgitating the answers.

4.3. Yeo-Johnson transformation

Yeo-Johnson transformation eliminates skewness more effectively than log-transformation. Using the SMOTE dataset, Table 3 reports the substantially higher metrics for Yeo-Johnson transformation, much preferable to other pre-processing techniques.

4.4. Model performance summary

Several ML models were trained using the TC 10 dataset, SMOTE dataset, and the Hong Kong dataset. Prior to the train-test split, the data underwent Yeo-Johnson transformation and z-transform. The performance of these models is illustrated through the PR curve and ROC curve in Figure 5, with the detailed metrics provided in Table 3. Among the trained models, the Random Forest and SVC with RBF kernel exhibited the best performances.

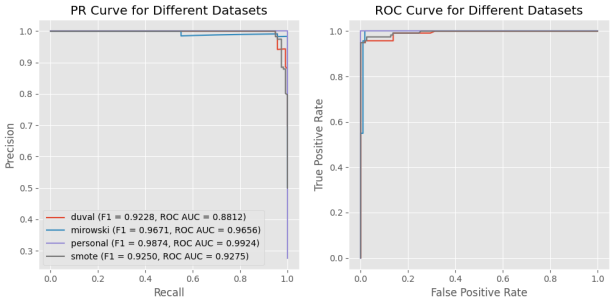

Figure 5. PR curve and ROC curve of ML models trained on different datasets.

5. Conclusion

This comprehensive analysis of ML models for DGA has set a methodological benchmark, particularly through a case study in Hong Kong. The analysis identified the random forest and RBF SVC as the most effective models, achieving *F1* scores of over 0.8 after Yeo-Johnson transformation were applied to the datasets. This finding emphasises that, with appropriate data preprocessing, the model's performance can improve hugely without the need for complex model architectures or extensive dataset sampling.

This study uses the *F1* score as the primary metric due to its balanced evaluation of precision and recall, supplemented by accuracy, average precision (AP), and ROC AUC. Cross-validation with a 5-fold approach enhanced the model evaluation reliability by providing averaged performance metrics for a generalised comparison. Models trained on both balanced and imbalanced datasets exhibited negligible performance variation, and preprocessing before the train-test split reduced the risk of data leakage. The Yeo-Johnson transformation is found to outperform log-transformation in addressing skewness, and

both are effective methods, as opposed to no preprocessing at all.

Several potential analyses and experimentations in the future are suggested. Firstly, exploring alternative data balancing techniques on larger datasets, such as undersampling or cost-sensitive learning, could provide a more comprehensive understanding of class imbalance. Additionally, evaluating the performance of other ML models, such as gradient boosting or deep learning architectures, may offer alternative approaches to classification tasks. Extending the models to diverse geographical contexts could further assess the generalisability of the findings. Pursuing these directions could refine the understanding of model training strategies and enhance the effectiveness of DGA assessments.

Acknowledgements

We would like to express our gratitude to the anonymous reviewers for their valuable comments. We also extend our thanks to the company that provided the dataset for the analysis.

Notes on contributors

Mr Philip TH Chan received his bachelor's and master's degrees in Engineering from The University of Hong Kong, specialising in transformer maintenance, smart technology, and AI applications in various domains. As a Support Engineer in MTR Corporation Limited, he monitors the health of MTR's entire fleet of transformers and oversees the analytics from various IoT devices on transformers, switchgear, and cables, enabling smart and preventive maintenance in power assets.

Mr Terry HW Chan received his bachelor's degree in Electrical Engineering from The Polytechnic University of Hong Kong. He has been working in the railway industry in the discipline of power distribution, including high voltage switchgear maintenance, periodic testing on protection schemes, and power system fault investigation. He is currently a Support Engineer in MTR Corporation Limited.

References

[1] Antonini P, Borsato E, Carugno G, Corso FD, Facco A and Fanin C (2019). Studies for the use of a dielectric liquid as insulator in a wireless high voltage generator. In: 2019 IEEE 20th international conference on dielectric liquids (ICDL). Roma, Italy: IEEE, pp. 1-4.

[2] Bagheri A, Allahbakhshi M, Arefi MM, Najafi N, and Javadi MS (2022). A new approach for top-oil thermal modelling of power transformers using unscented kalman filter considering IEEE C57. 91 standard. *IET Electric Power Applications*, 16(5), pp. 536-547.

[3] Bishop C (2006). *Pattern recognition and machine learning*. New York: Springer.

[4] Chang YW, Hsieh CJ, Chang KW, Ringgaard M and Lin CJ (2010). Training and testing low-degree polynomial data mappings via linear SVM. *Journal of Machine Learning Research*, 11(4).

[5] Chawla NV, Bowyer KW, Hall LO, and Kegelmeyer WP (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, pp. 321-357.

[6] Duval M (1989). Dissolved gas analysis: It can save your transformer. *IEEE Electrical Insulation Magazine*, 5(6), pp. 22-27.

[7] Duval M and DePabla A (2001). Interpretation of gas-in-oil analysis using new IEC publication 60599 and IEC TC 10 databases. *IEEE Electrical Insulation Magazine*, 17(2), pp. 31-41.

[8] Duval M and Dukarm J (2005). Improving the reliability of transformer gas-in-oil diagnosis. *IEEE Electrical Insulation Magazine*, 21(4), pp. 21-27.

[9] Ekojono, Prasojo RA, Apriyani ME and Rahmanto AN (2022). Investigation on machine learning algorithms to support transformer dissolved gas analysis fault identification. *Electrical Engineering*, 104(5), pp. 3037-3047.

[10] Equbal MD, Khan SA and Islam T (2018). Transformer incipient fault diagnosis on the basis of energy-weighted DGA usingan artificial neural network. *Turkish Journal of Electrical Engineering and Computer Sciences*, 26(1), pp. 77-88.

[11] Fang J, Zheng H, Liu J, Zhao J, Zhang Y and Wang K (2018). A transformer fault diagnosis model using an optimal hybrid dissolved gas analysis features subset with improved social group optimization-support vector machine classifier. *Energies*, 11(8), pp. 1922.

[12] Guo C, Dong M and Wu Z (2019). *Fault diagnosis of power transformers based on comprehensive machine learning of dissolved gas analysis*. In: IEEE 20th International Conference on Dielectric Liquids (ICDL). Roma, Italy: IEEE.

[13] Illias HA and Zhao Liang W (2018). Identification of transformer fault based on dissolved gas analysis using hybrid support vector machine-modified evolutionary particle swarm optimisation. *PLoS One*, 13(1) pp. e0191366.

[14] Japkowicz N (2013). Assessment metrics for imbalanced learning. *Imbalanced learning: Foundations, algorithms, and applications*, pp. 187-206.

[15] Jeni LA, Cohn JF, and De La Torre F (2013). *Facing imbalanced data-recommendations for the use of performance metrics*. In: 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction. Geneva, Switzerland: IEEE, pp. 245-251.

[16] Kaplan IR, Rasco J, and Lu ST (2010). Chemical characterization of transformer mineral-insulating oils. *Environmental Forensics*, 11(1–2), pp. 117-145,

[17] Kubat M (2017). *An introduction to machine learning*. Springer.

[18] Laurikkala J (2001). *Improving identification of difficult small classes by balancing class distribution*. In: Artificial intelligence in medicine: 8th conference on artificial intelligence in medicine in europe, AIME 2001 cascais, portugal, proceedings 8. Springer, pp. 63-66.

[19] Li Y, Xu Y, Li X, Li R, Lin J and Zhang G (2022). Addressing imbalance of sample datasets in dissolved gas analysis by data augmentation: Generative adversarial networks. *IET Generation, Transmission & Distribution*, 16(22), pp. 4494-4504.

[20] Mirowski P and LeCun Y (2012). Statistical machine learning and dissolved gas analysis: A review. *IEEE Transactions on Power Delivery*, 27(4), pp. 1791-1799.

[21] Rao S, Yang S and Zou G (2023). A methodology for transformer fault diagnosis based on the feature extraction from DGA data. *International Journal of Applied Electromagnetics and Mechanics*, Preprint, pp. S1-S8.

[22] Rogers R (1978). IEEE and IEC codes to interpret incipient faults in transformers, using gas in oil analysis. *IEEE transactions on electrical insulation*, no. 5, pp. 349-354.

[23] Sarma D S and Kalyani G (2004). ANN approach for condition monitoring of power transformers using DGA. *2004 IEEE region 10 conference TENCON*, IEEE, pp. 444-447.

[24] Shang H, Xu J, Zheng Z, Qi B, and Zhang L (2019). A novel fault diagnosis method for power transformer based on dissolved gas analysis using hypersphere multiclass support vector machine and improved d–s evidence theory. *Energies*, 12(20), pp. 4017.

[25] Thango BA (2022). Dissolved gas analysis and application of artificial intelligence technique for fault diagnosis in power transformers: A south african case study. *Energies*, 15(23), pp. 9030.

[26] Yeo IK and Johnson RA (2000). A new family of power transformations to improve normality or symmetry. *Biometrika*, 87(4), pp. 954-959.

[27] Yuan F, Guo J, Xiao Z, Zeng B, Zhu W and Huang S (2019). A transformer fault diagnosis model based on chemical reaction optimization and twin support vector machine. *Energies*, 12(5), pp. 960.

[28] Zhang Y, Ding X, Liu Y and Griffin P (1996). An artificial neural network approach to transformer fault diagnosis. *IEEE transactions on power delivery*, 11(4), pp. 1836-1841.

[29] Zhang Y, Wang Y, Fan X, Zhang W, Zhuo R, Hao J and Shi Z (2020). An integrated model for transformer fault diagnosis to improve sample classification near decision boundary of support vector machine. *Energies*, 13(24) pp. 6678.

[30] Zheng W and Jin M (2020). The effects of class imbalance and training data size on classifier learning: An empirical study. *SN Computer Science*, 1, pp. 1-13.