UNIVERSITY OF MINES AND TECHNOLOGY (UMaT)
TARKWA

FACULTY OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

A PROJECT ENTITLED

PREDICTIVE MAINTENANCE OF POWER TRANSFORMER USING MACHINE LEARNING - A CASE STUDY

BY
AMANING RICHMOND OFORI

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING

PROJECT SUPERVISOR

…..……
DR JOSEPH C. ATTACHIE

TARKWA, GHANA
AUGUST, 2024

DECLARATION

I declare that this project work is my own work. It is being submitted for the degree of Bachelor of Science in Electrical and Electronic Engineering in the University of Mines and Technology (UMaT), Tarkwa. It has not been submitted for any degree or examination in any other University.

……

(Signature of Candidate)

……day of ……., 2024.

ABSTRACT

The importance of power transformers in electrical power systems cannot be overstated, as their failures can lead to considerable economic losses and disruptions. The typical malfunctions encountered by a power transformer comprise dielectric issues, thermal losses due to copper resistance, distortions in winding caused by mechanical faults, failure of bushings, malfunction of tap changers, core malfunction, tank malfunction, failure of the protection system, and failure of the cooling system. Traditional methods for transformer fault detection involve using the ratio of key gases present in the transformer oil when a fault occurs. These gases include Hydrogen (H_2), Methane (CH_4), Ethane (C_2H_6), Ethylene (C_2H_4), Ethyne (C_2H_2), Carbon Monoxide (CO) and Carbon Dioxide (CO_2). For accurate and early detection of faults, traditional methods require complex algorithms. This project focuses on the predictive maintenance of power transformers using machine learning techniques, aiming to identify and address potential faults pre-emptively. By analysing various fault types and leveraging machine learning tech like Decision Trees, Support Vector Machines (SVM), and K-Nearest Neighbour (KNN), the project develops models that predict transformer failures based on historical data. Dataspell software and Python libraries such as Numpy and Matplotlib were used to train the model. The testing results showed the efficiency of the SVM, KNN, and Decision Tree methods in detecting the faults experienced by the power transformer. The testing accuracy for SVM, KNN and Decision Tree models was 95.65%, 95.65% and 89.13%, respectively. It was observed that the SVM and KNN models performed better than the decision tree model.

DEDICATION

This work is dedicated to my parents, Mr Tawiah Maxwell Ebu and Mrs Cecilia Nyarkoa, and my sister, Ms Alberta Agyapomaa Tawiah, for their support and unconditional love throughout my life.

ACKNOWLEDGEMENTS

My ultimate gratitude goes to God Almighty for giving me the insight and strength in all the days I spent to acquire my first degree.

I am also grateful to my supervisor, Dr Joseph C. Attachie, for his maximum attention, time, and guidance while supervising this work.

Finally, I appreciate my parents and siblings for their support throughout my study.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Meaning
AC	Alternating Current
ANN	Artificial Neural Network
AP	Average Precision
AUC	Area Under Curve
DGA	Dissolved Gas Analysis
ECG	Electricity Company of Ghana
EMF	Electromotive Force
FN	False Negative
FP	False Positive
IDE	Integrated Development Environment
IoT	Internet of Things
KNN	K-Nearest Neighbour
PD	Partial Discharge
PDM	Predictive Maintenance
PR	Precision-Recall
ROC	Receiving Operating Curve
RUL	Remaining Useful Life
SVM	Support Vector Machine
TN	True Negative
TP	True Positive

LIST OF SYMBOLS

Carbon Dioxide	CO_2
Carbon Monoxide	CO
Ethane	C_2H_6
Ethylene	C_2H_4
Ethyne	C_2H_2
Methane	CH_4
Hydrogen	H_2
High Energy Discharge	D2
High-Temperature Thermal Fault	T3
Low Energy Discharge	D1
Low-Temperature Thermal Fault	T1
Medium-Temperature Thermal Fault	T2

INTERNATIONAL SYSTEM OF UNITS (SI UNITS)

Quantity	Units of Measurement	Symbol
Apparent Power	volts-ampere	VA
Electrical Voltage	volts / kilovolts	V / kV
Temperature	degree Celsius	ºC

CHAPTER 1

GENERAL INTRODUCTION

1.1 Research Background

The Electricity Company of Ghana (ECG) functions within different sectors, providing electricity to homes, industries and businesses. The power transformer is utilised in ECG as one of the essential pieces of equipment, guaranteeing the effective conveyance and dispersion of electrical power throughout the grid. Power transformers step up voltage for transmission after generation for economic transmission to reduce power losses. Power transformers are static machines that have very high efficiency. The power transformer plays a crucial role in the operation of a power grid, and its failure can significantly impact the grid's safe and stable operation and lead to significant economic losses (Duan and Wang, 2023). Faults may also happen in the transformer, like all other electrical devices, which cause the failures (Sudha *et al*., 2022). Failures that frequently happen in power transformers include dielectric faults, thermal losses due to copper resistance, winding distortion leading to mechanical faults, bushing failure, tap changer failure, core failure, tank failure, protection system failure, and cooling system failure.

The goal of predictive maintenance methods is to assess the operating condition of equipment and predict when it is prone to failure so that proactive maintenance can be scheduled (Amer *et al*., 2023). The application of machine learning for equipment prognostics has become more common since 2010, with significant progress in fault classification and limited progress in predicting Remaining Useful Life (RUL) (Howard, 2022). Machine learning applications provide benefits like predicting and preventing equipment failures, cutting maintenance expenses, minimising repair downtimes, improving plant safety and security, boosting production, and offering numerous other advantages (Amer *et al*., 2023).

1.2 Problem Definition

Power transformers are essential equipment for ECG’s power grid system. Transformer breakdown is one of the main reasons why a power system operation may be interrupted. Transformer maintenance is, therefore, highly essential but tedious. The abrupt or unforeseen malfunction of such machinery will not just impact power generation but also

result in additional expenses for maintenance and repairs, as well as potential casualties and societal repercussions in serious situations. Traditional maintenance approaches, such as reactive maintenance, are often reactive and time-consuming, posing challenges for ECG in ensuring a reliable power supply.

The reliable assessment of a power transformer's condition using machine learning algorithms is an important challenge that needs to be addressed promptly. Recent developments in information technologies and communication networks have made it possible for machines to collect and analyse large volumes of data and perform operational environmental tasks to detect and investigate machine failures (Amer *et al*., 2023). Predictive Maintenance (PDM) enables the enhancement of high-quality strategies for smart preventative maintenance using gathered data collections.

The goal of this project is to develop a model for predicting the health status of equipment, carry out the prediction of equipment health status, and proactively implement maintenance actions based on the predicted equipment status results to anticipate equipment "status repair" and investigate an innovative approach to equipment maintenance management.

1.3 Project Objectives

The objectives of this project are to:

i. Investigate machine learning algorithms that can be suitable for predictive maintenance application; and
ii. Build a model based on machine learning techniques for predictive maintenance of a power transformer.

1.4 Methods Used

The methods employed include:

i. Review of relevant literature;
ii. Model development and implementation; and
iii. Comparing the results of the developed model with existing models using standard metrics.

1.5 Facilities Used

The facilities employed are:

i. The University Library;
ii. Internet Facilities; and
iii. Laptop computer with the latest version of Python programming language installed and DataSpell, a data science integrated development environment (IDE).

1.6 Work Organisation

This project is organised into five chapters. Chapter 1 gives a general introduction that deals with the background to the research, the problem under study, project objectives, methods used, facilities used, and work organisation. Chapter 2 examines significant literature in the subject area, considering definitions, examples, and clarifications of the theoretical framework related to the study. This chapter forms the basis of the whole project, equipping the reader with the essential information needed to comprehend the approach and the chapters that follow. Chapter 3 discusses the methods, including implementing the proposed machine learning techniques. Chapter 4 presents the results and their discussions. Chapter 5 gives the conclusions and recommendations.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Power transformer failures can lead to significant economic losses and disruptions. To mitigate these risks, predictive maintenance strategies have gained prominence. Machine learning has become a powerful tool for analysing complex data patterns associated with transformer health. By leveraging this technology, utilities can accurately predict potential failures, optimise maintenance schedules and prevent costly breakdowns. This proactive approach aligns with the industry's growing emphasis on reliability and efficiency.

This chapter discusses power transformers, reviews power transformer faults, reviews software for machine learning, reviews machine learning techniques, and reviews related works.

2.2 Power Transformer

A power transformer efficiently transfers electrical power between circuits without altering the frequency. Its operation is based on electromagnetic induction. The purpose of power transformers is to increase or decrease the voltage of an alternating current (AC). They are considered static devices because they have no moving parts.

A power transformer is a type of transformer that operates within a voltage range of 33 to 400 kV and has a rating higher than 200 MVA. Voltage ratings of power transformers on the market range from 400 kV to 33 kV. The other transformers are distribution (230 V - 11 kV) and instrument transformers. Figure 2.1 (Anon., 2024a) shows a Transformer.

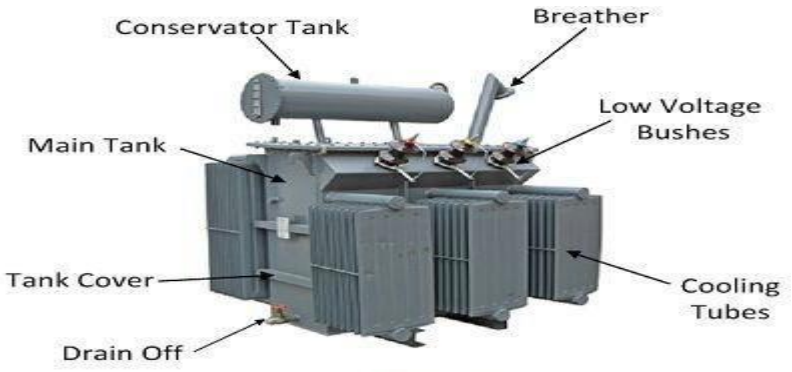

Figure 2.1 A Transformer

2.2.1 Principle of Operation

Faraday's principle of electromagnetic induction is the fundamental concept behind how a transformer operates. This fundamental law of electromagnetism explains the working principle of all transformers and inductors. Faraday's law asserts that an electromotive force (emf) will be generated across a closed loop when it is brought close to a changing magnetic field (Anon., 2024b).

When an alternating current passes through a coil or primary winding, it generates a changing magnetic flux around it. This magnetic flux, which is produced by the primary winding, travels through a ferromagnetic core to reach a secondary winding. As a result of electromagnetic induction, the magnetic flux induces an electromagnetic field in the secondary winding. This induced electromotive force (emf) then promotes the current flow in the secondary winding. Figure 2.2 shows the working principle of a transformer (Anon., 2024b).

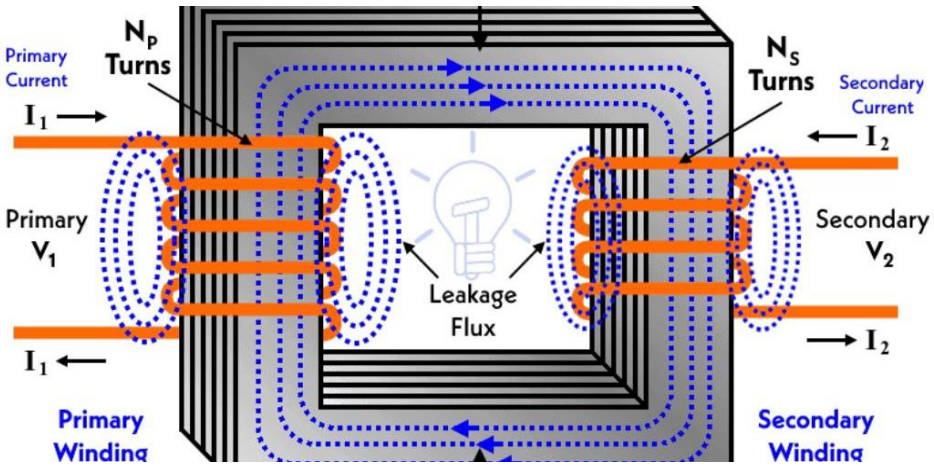

Figure 2.2 Transformer Working Principle

2.2.2 Types of Transformers

Different factors can be used to categorise power transformers, including their construction, application, and function. Some of the standard classifications of power transformers and various types are turns ratio, phases, core material and core and winding construction and arrangement

Turns ratio

Step-up and step-down transformer: These transformers increase or decrease the voltage level of an AC supply. In a step-up transformer, there are more turns in the secondary

winding than in the primary winding. Conversely, in a step-down transformer, there are fewer turns in the secondary winding compared to the primary winding (Anon., 2024c).

Isolation transformer: This type transfers electrical power between two circuits without changing the frequency and providing galvanic isolation. A transformer for isolation has an equal number of turns in its primary and secondary windings (Anon., 2024d).

Phases

Single-phase and three-phase transformer: A single-phase transformer has one primary winding and secondary winding, which produces a single alternating voltage in the form of a sine wave, while a three-phase transformer features three pairs of primary and secondary windings, interconnected in either a star or delta configuration (Anon., 2024e).

Autotransformer: This type of transformer has only one winding. The primary and secondary coils are linked electrically and wound around a common core. They find wide applications in induction motors, railway systems, audio, and lighting (Anon., 2024f).

Core material

Air-core transformers: Air-core transformers do not have a physical core, and their primary and secondary windings are wrapped in a solid insulating material. They are utilised to carry radio currents (Anon., 2019).

Ferrite core transformers: Transformers with ferrite cores have cores composed of ferrite, a magnetic ceramic material containing iron oxide. Manganese zinc ferrite and nickel-zinc ferrite are ferrites commonly used in transformers (Anon., 2019).

Iron core transformers: Iron core transformers contain a magnetic core of laminated iron sheets. This type of transformer is the most widely used in its category. They demonstrate powerful magnetic properties, resulting in a high flux linkage (Anon., 2019).

Toroidal core transformers: Transformers with toroidal cores feature a core made from iron or ferrite and have a torus or doughnut shape. Compared to traditional shell and core transformers, they provide increased design flexibility, efficiency, and space-saving characteristics (Anon., 2019).

Core and winding construction and arrangement

Berry-type transformers: Berry-type transformers feature a core configuration resembling the spokes of a wheel. They have distributed magnetic paths with more than two distinct magnetic pathways (Anon., 2020).

Core-type transformers: These transformers have primary and secondary windings that encircle the core. The core is constructed by joining two L-shaped steel strips and stacking them. The positioning of the strips is designed to eliminate uninterrupted joints and avoid high reluctance at the joints. The limbs and the yoke carry the complete flux (Anon., 2020). *Shell-type transformers:* The core in shell-type transformers surrounds the primary and secondary windings. It is constructed by layering E-shaped and I-shaped steel strips to create the central and side limbs. The central limb carries the complete magnetic flux, while the side limbs each have half (Anon., 2020).

2.3 Power Transformer Faults

The formation of gas molecules of different densities occurs when power transformers fail. These gas molecules are formed at various temperatures and with differing formation energies. The type and amount of gases produced during the fault vary, making them useful for fault detection. Dissolved Gas Analysis (DGA) is an effective method for detecting gases present in transformer insulation liquid. Utilising the chromatograph method, the DGA measures the number of gases in the insulating liquid for fault detection. The DGA method for transformer fault analysis utilises gases such as Hydrogen (H_2), Carbon Monoxide (CO) and Carbon Dioxide (CO_2), Methane (CH_4), Ethane (C_2H_6), Ethylene (C_2H_4) and Ethyne (C_2H_2).

Partial discharges, thermal overheating, and arcing are the primary power transformer faults that can be consistently identified during a visual inspection (Nanfak *et al*., 2021). These faults can further be classified into six types listed in Table 2.1.

Table 2.1 Fault Classification According to IEC 60599 and IEEE C57.104 Standard

Acronym	Fault Type
PD	Partial Discharge
D1	Low Energy Discharge
D2	High Energy Discharge
T1	Low-Temperature Thermal Fault T<300 °C
T2	Medium-Temperature Thermal Fault 300 °C <T<700 °C
T3	High-Temperature Thermal Fault T>700 °C

(Source: Nanfak *et al*., 2021)

Partial discharge

Partial Discharge (PD) is an electrical discharge that partially connects the insulation between conductors. The discharge can occur when the electrical stress exceeds the breakdown strength of a specific portion of the insulation system. The occurrence of PDs results in the deterioration of the transformer insulation system (Meitei *et al*., 2021).

Low energy discharge

Low-energy discharge (D1) refers to partial discharges that release relatively little energy compared to other discharges. Despite their lower energy, these discharges can still cause significant damage to transformer insulation over time.

High energy discharge

High-energy discharges (D2) in transformers are partial discharges that release significant energy. If not promptly addressed, these discharges can cause immediate and severe damage to the insulation system and lead to catastrophic failure.

Low-temperature thermal fault

Thermal faults (T1) in transformers refer to conditions where the temperature of the insulation material and transformer oil rises abnormally but to relatively lower temperatures compared to more severe faults. T1 thermal faults typically involve temperatures up to 300 °C and are often associated with overheating due to poor cooling, overloading, or localised heating effects.

Medium-temperature thermal fault

Faults related to temperature (T2) in transformers usually range from 300 °C to 700 °C and are considered moderate to high temperatures. These faults indicate more severe overheating conditions than T1 thermal faults and can cause significant degradation of the transformer's insulation and oil.

High-temperature thermal fault

Thermal faults (T3) in transformers refer to temperatures exceeding 700 °C. These severe faults indicate extremely high overheating, which leads to substantial degradation of the insulation system and transformer oil and a high risk of transformer failure.

2.4 DataSpell Software for Machine Learning

DataSpell, a robust IDE developed by JetBrains, is designed to improve productivity and streamline workflows for data science and machine learning tasks. It is particularly effective for predictive functions like classification, providing a comprehensive suite of tools and features modified to the needs of data scientists and machine learning practitioners.

One of DataSpell's key features is its support for Jupyter Notebooks. Users can write and run code, visualise data, and document workflow all within one interface in this interactive environment. Jupyter Notebooks are invaluable for exploratory data analysis and iterative model development, making testing different hypotheses and approaches for classification tasks more accessible.

DataSpell provides strong support for Python, which is the primary language for machine learning. This support includes advanced functionalities such as smart code suggestions, highlighting of syntax, and tools for debugging. These features help data scientists write cleaner, more efficient code and quickly identify and resolve errors. The IDE also supports various Python libraries and frameworks commonly used in machine learning, such as Scikit-learn, TensorFlow, and PyTorch, facilitating seamless integration into classification workflows.

Version control integration is another significant advantage of DataSpell. Users can effortlessly monitor changes, cooperate with team members, and handle various versions of their projects thanks to the integrated Git and other version control system support. This is particularly useful in machine learning projects, where iterative improvements and experimentation are expected.

Data visualisation is crucial for understanding data distributions, feature importance, and model performance. DataSpell provides built-in support for popular visualisation libraries like Matplotlib, Seaborn, and Plotly, allowing users to create detailed and informative visualisations. These visualisations help interpret the results of classification models and communicate findings to stakeholders.

DataSpell offers a comprehensive environment for developing, testing, and deploying machine learning models, particularly for classification tasks. The combined Jupyter Notebooks, strong Python support, incorporation of version control, and capabilities for data

visualisation make it an essential tool for data scientists and machine learning experts who want to boost their efficiency and accomplish more precise predictive results (Anon., 2024g).

2.5 Review of Classification Models

Classification is a supervised machine learning method where the model attempts to predict the correct label for a given input dataset. With classification, the model is trained using the training data and then assessed on test data before being utilised to predict new, unseen data. The initial phase is "training," where the model is established, and the classification rules are determined through the dataset containing known labels. The subsequent step is "testing," which involves evaluating the model's performance using other datasets with known labels. If the criteria are satisfied, the model is accepted. If not, the initial phase is repeated (Anon., 2022).

Lazy Learners and Eager Learners are two types of classification algorithms. Eager learners are machine learning algorithms that construct a model using the training dataset and then make predictions for future datasets. An example of an Eager Learning algorithm is the Decision Tree. Learners that are lazy or instance-based do not generate a model right away from the training data. Instead, they store the training data and locate the closest neighbor from the complete training data every time a prediction is required, which may cause them to be slow during prediction. An example of the lazy learner algorithm is K-Nearest Neighbour (KNN). Some classification methods used for this research include Decision trees, SVM and KNN.

2.5.1 Decision Tree

A decision tree serves as a strong supervised learning method used for classification and regression tasks. It organises data into a model resembling a tree, with internal nodes indicating attribute tests, branches representing test outcomes, and leaf nodes indicating class labels or continuous values. Decision trees are valued for their straightforwardness, interpretability, and capacity to manage both numerical and categorical data. They excel in feature selection, managing non-linear relationships, and handling datasets with missing values. Decision trees are extensively utilised in various domains because they can handle complex decision-making processes and provide precise, interpretable models (Doe and Smith, 2023).

2.5.2 Support Vector Machine

Support Vector Machine (SVM) is a supervised learning method used for classifying and predicting tasks. It operates by identifying the best hyperplane to separate classes, maximising the distance between the nearest points of different classes. SVMs are efficient in high-dimensional spaces and are less susceptible to overfitting due to their emphasis on maximising the margin. They also employ the kernel trick to handle non-linear classification. However, SVMs can be demanding in terms of computational resources and require careful parameter tuning. Generally, SVMs provide superior generalisation but can be less explainable and more resource-intensive compared to other algorithms such as decision trees and logistic regression (Johnson and Wang, 2024).

2.5.3 K-Nearest Neighbour

The KNN algorithm is a straightforward, non-parametric technique utilised for both classification and regression. It categorises data points by taking a vote from their nearest neighbours. KNN's strength lies in its simplicity and effectiveness when dealing with small datasets, while its weakness is its high computational intensity when handling large datasets. In comparison to algorithms such as decision trees and SVMs, KNN is straightforward to implement but may be slower and less precise when dealing with high-dimensional data (Lee and Brown, 2023).

2.6 Traditional Methods of Fault Detection in Transformers

Traditional fault detection methods based on DGA data are known as gas ratio methods. These techniques make use of key gas ratios to diagnose faults. The traditional techniques consist of Doernenburg’s ratio method, Roger’s ratio method, IEC ratio method, and the Three Ratio Technique.

2.7 Review of Related Works on Transformer Fault Detection

Venkataswamy *et al*. (2020) proposed a reliability-centred maintenance approach for distribution transformers using Internet of Things (IoT) and metaheuristic techniques. The approach is based on the metaheuristic optimisation techniques integrated with IoT technologies to enhance reliability-centred maintenance strategies for distribution transformers. The approach significantly enhanced the reliability-centred maintenance of

distribution transformers, reducing downtime and maintenance expenses. However, implementation requires a robust IoT infrastructure, which may be complex and expensive.

Sarro *et al*. (2020) proposed a technique that utilises machine learning to improve human expert effort estimates by learning from mistakes. The study utilises machine learning algorithms to enhance the precision of human expert effort predictions by drawing insights from past data and prior estimation discrepancies. The machine learning-enhanced estimates provided more accurate predictions of human expert effort, reducing the error margin in project planning. However, the drawback of machine learning-enhanced estimates is that the accuracy of the prediction relies on the excellence and comprehensiveness of historical data.

Aqueveque *et al*. (2021) presented a method for utilising wireless accelerometer sensor modules to conduct data-driven condition monitoring of mining mobile machinery in non-stationary operations. The authors suggested using wireless accelerometer sensors to collect data for monitoring the condition of mining machinery during non-stationary operations in this paper. Effective monitoring of mining machinery under non-stationary conditions was achieved, improving operational efficiency and reducing unexpected failures. However, the study focuses on monitoring in non-stationary environments, which poses challenges due to varying operational conditions and sensor reliability.

Laayati *et al*. (2021) suggested a novel concept for a self-diagnostic system that is integrated into an intelligent energy management system for monitoring oil-immersed power transformers. This paper proposes a smart energy management system with integrated self-diagnostic capabilities for oil-immersed power transformers to monitor and diagnose faults. The developed system successfully integrated monitoring and self-diagnostic capabilities, providing real-time fault detection and diagnosis. The drawback of this approach is that combining the system with existing infrastructure can be complex and require significant modifications.

Pileggi *et al*. (2021) proposed a method for implementing machine learning to predictively maintain gas turbines. The study examines the real-world implementation of machine learning models for predicting maintenance needs in gas turbines, covering data preparation and deploying models. The study provided valuable insights into the challenges and best practices for implementing machine learning in predictive maintenance and improving

operational outcomes. However, operationalising machine learning models in real-world settings involves data quality, model maintenance, and scalability challenges.

Vallim Filho *et al*. (2022) reviewed condition-based maintenance of power transformers using predictive analytics. The authors in this research suggested a framework for predictive maintenance using machine learning models, which is based on equipment load cycles, in a practical case scenario. The framework accurately predicted maintenance needs based on equipment load cycles, validated by real-world case studies. However, while the framework accurately predicted maintenance needs, its effectiveness depends on the availability and accuracy of detailed equipment load cycle data.

Yu *et al*. (2022) proposed a SVM approach utilising information granulation for detecting anomalies in primary transformers at nuclear power facilities. This paper uses an information-granulated SVM approach to detect anomalies in main transformers at nuclear power plants. The proposed SVM approach accurately detected anomalies in transformer operations, enhancing the safety and reliability of nuclear power plants. The drawback of this approach is that the SVM approach's performance is highly impacted by the quality and granularity of the information provided.

Raghuraman and Darvishi (2022) proposed a method for identifying various types of transformer faults from DGA data by employing machine learning methods. In this fault detection research, machine learning techniques are applied to DGA data to detect and classify fault types in power transformers. The machine learning models accurately identified fault types from DGA data, improving maintenance planning and transformer reliability. The drawback of this approach is that the precision of identifying fault types are contingent on the quality of the DGA data, which can vary significantly.

Kastelan *et al*. (2022) suggested an innovative idea of switchgear digitalisation. In this paper, the authors examine the current state and future directions of switchgear digitalisation, focusing on integrating sensor technology and digital tools. The review highlighted significant advancements and future opportunities in switchgear digitalisation, promoting better integration of digital tools. However, the high cost of implementing digitalisation technologies may be a barrier to widespread adoption.

Bai *et al.* (2023) proposed transformers as statisticians, a provable in-context learning method with in-context algorithm selection. The proposed method is an in-context learning

framework for transformers that can select appropriate algorithms based on contextual data, enhancing predictive accuracy. The proposed in-context learning framework enhanced the accuracy and efficiency of maintenance operations by improving the selection of algorithms for predictive maintenance. However, selecting the appropriate algorithms in context requires advanced understanding and can be computationally intensive.

2.8 Summary of Review of Related Works on Transformer Fault Detection

A significant amount of research has been conducted regarding the use of machine learning for predicting maintenance needs for power transformers including that of Yu *et al*. (2022), Raghuraman and Darvishi (2022), and Vallim Filho *et al*. (2021). However, the existing studies were limited due to the availability and quality of the dataset, the type of classifier used and accuracy issues that needed to be addressed to achieve an improved model with better prediction results. Classifiers like Decision Trees, SVM, and KNN can rectify this gap by predicting transformer faults based on the dissolved gases present in the transformer oil.

CHAPTER 3

METHODS USED

3.1 Introduction

This chapter describes the methodologies employed in the predictive maintenance of power transformers using machine learning techniques. This chapter presents a systematic approach to developing and implementing a machine learning model designed explicitly for detecting anomalies and faults in power transformers early. Figure 3.1 shows the procedure for data processing, developing, training the model and testing its ability to classify faults accurately.

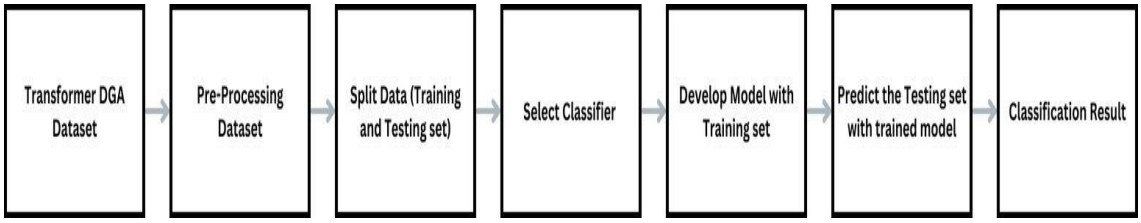

Figure 3.1 A Flow Chart of the Methodology

3.2 Data Acquisition and Pre-Processing

The dataset used in this model is obtained from existing works of literature. Most of the data were obtained from the IEC TC10 dataset. Figure 3.2 shows the distribution of the fault classes in the dataset.

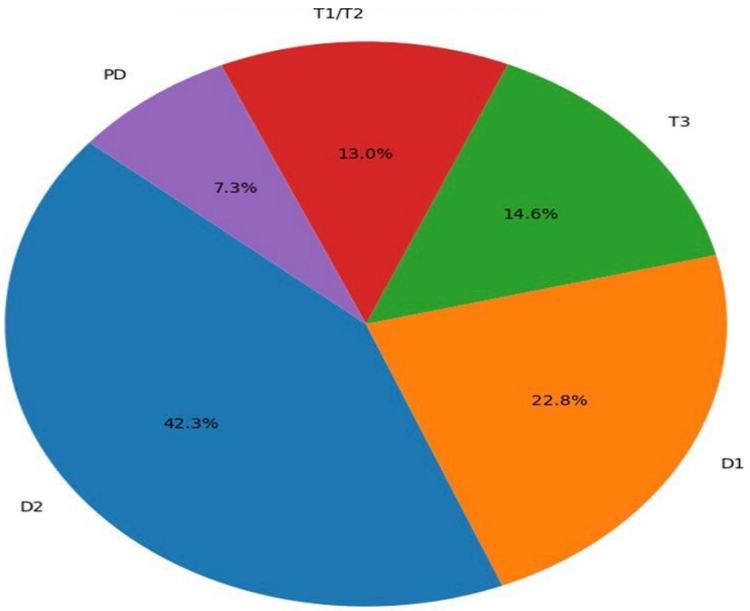

Figure 3.2 A Graphical Representation of the Distribution of Faults in the Dataset

The data consists of 123 DGA gas concentration samples containing five key gases and their labelled faults. They are Hydrogen (H_2), Methane (CH_4), Ethane (C_2H_6), Ethylene (C_2H_4) and Ethyne (C_2H_2).

The data pre-processing techniques used in this project include:

i. Feature Selection: It encompasses choosing the critical aspects of the data for model building. The goal of feature selection is to enhance the predictive model's performance and lower the computational expenses of modelling;
ii. Feature Engineering: This is the handling of datasets to improve model training. This includes handling missing data, variable encoding, feature scaling, feature creation and handling outliers;
iii. Data/Class Balancing: The minority class is resampled to ensure a balanced dataset that matches the size of the majority class; and
iv. Data Splitting: The dataset is divided into training and testing sets.

3.3 Model Selection and Development

The decision tree, SVM, and KNN classifier were selected and used to develop a model for classifying transformer faults. The decision tree, SVM, and KNN classifiers were chosen as classifiers for the model of interest because of their simplicity and interpretability. The classifiers were trained on a dataset containing 123 sets of transformer oil samples with their labelled faults. The dataset underwent division into training and testing sets in order to guarantee that the model gleaned from diverse examples and can effectively generalise to new, unseen data. The dataset is first pre-processed, and a feature selection approach was adopted to improve the model's efficiency.

3.4 Model Evaluation

Assessing a trained model's performance using various metrics and techniques is widely recognised as a critical step in the Machine Learning workflow, known as model evaluation. Model assessment shows how well the model can learn and correctly classify faults as needed. In the model evaluation phase, the test set data is utilised to objectively assess the model's performance. The dataset is pre-processed and transformed to meet the specific requirements of the model in use. The predictions generated on the test dataset demonstrate the model's capacity to understand underlying patterns and offer insight into its decision-making process.

3.5 Classification Result

A confusion matrix is utilised to demonstrate the classification ability of classification algorithms. It consists of True positives, true negatives, false positives, and false negatives. True positives (TP) indicate the number of correctly identified class instances, while true negatives (TN) represent accurately categorised patterns that do not belong to the class. Conversely, false positives (FP) are samples erroneously assigned to the class, and false negatives (FN) are samples not identified as class instances. Precision, Recall, and F1 scores are computed using these values. Precision reflects the proportion at which the data class tag matches the classifier tag, while Recall denotes the classifier's performance for each class tag. The F1 Score, the harmonic mean of precision and recall, is employed in unbalanced classification problems. Classification measures are provided in Equations (3.1), (3.2), and (3.3) accordingly.

$$Precision = \frac{TP}{TP+FP} \tag{3.1}$$

$$Recall = \frac{TP}{TP+FN} \tag{3.2}$$

$$F1\ Score = \frac{2*precision*recall}{precision+recall} \tag{3.3}$$

3.6 Performance Metrics

Accuracy is a performance measure for evaluating a model's ability to make accurate predictions. It is computed by comparing the number of correct predictions to the total number of samples in the dataset. A high level of accuracy is favourable as it indicates the model's effectiveness in classifying data into appropriate categories. In cases where one class is much more prevalent than the other in a dataset, simply using accuracy to evaluate a model may not accurately assess its performance. Therefore, additional metrics such as precision should also be considered. Equation (3.4) is used for calculating accuracy as follows:

$$Accuracy = \frac{Number\ of\ Correctly\ Predicted\ Samples}{Total\ Number\ of\ Samples} \tag{3.4}$$

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

This chapter presents the results after the models have been executed. It compares the models built using the three classifiers by evaluating their testing and training accuracy.

4.2 Computational Results for Decision Tree Classifier

The model was evaluated using two performance metrics: accuracy in training and testing. These metrics provided insights into the model's effectiveness, showing how well it worked with the dataset. Training accuracy is a complete pass through the entire dataset. This signifies the ability of the model to classify correctly and effectively learn from the training data. Table 4.1 shows the decision tree classifier's classification result and confusion matrices. The matrix diagonal contains 41 correctly predicted faults, and there are 5 faults outside the diagonal.

Table 4.1 Classification Result and Confusion Matrice for Decision Tree Classifier

DECISION TREE								
Actual Class / **Predicted Class**	**PD**	**D1**	**D2**	**T1/T2**	**T3**	**Classification Measures**		
						Precision	**Recall**	**F1 Score**
PD	**23**	0	0	0	0	0.96	1	0.98
D1	0	**5**	1	0	0	0.62	0.83	0.71
D2	0	2	**8**	0	0	0.89	0.80	0.84
T1/T2	0	1	0	**1**	0	1	0.33	0.5
T3	1	0	0	0	**4**	1	1	1

4.3 Computational Results for SVM Model

The SVM model was evaluated in testing and training using accuracy as the primary performance metric. It was noticed that the SVM model performed well over the dataset. Table 4.2 shows the confusion matrices and classification results for the SVM model. In the matrix diagonal, 44 faults were correctly predicted, and there were 2 faults outside the diagonal.

Table 4.2 Classification Result and Confusion Matrice for SVM Classifier

SVM								
Actual Class / **Predicted Class**	**PD**	**D1**	**D2**	**T1/T2**	**T3**	**Classification Measures**		
						Precision	**Recall**	**F1 Score**
PD	**23**	0	0	0	0	0.96	1	0.98
D1	0	**6**	0	0	0	0.86	1	0.92
D2	0	0	**10**	0	0	1	1	1
T1/T2	1	1	0	**1**	0	1	0.33	0.5
T3	0	0	0	0	**4**	1	1	1

4.4 Computational Results for KNN Classifier

The performance of the KNN classifier was assessed using accuracy. The model performed equally in terms of testing accuracy with the SVM model. Table 4.3 shows the KNN model's confusion matrices and classification results. The matrix diagonal contains 44 accurately predicted faults, while there are 2 faults outside the diagonal.

Table 4.3 Classification Result and Confusion Matrice for KNN Classifier

KNN								
Actual Class / **Predicted Class**	**PD**	**D1**	**D2**	**T1/T2**	**T3**	**Classification Measures**		
						Precision	**Recall**	**F1 Score**
PD	**23**	0	0	0	0	0.92	1	0.96

Table 4.3 Classification Result and Confusion Matrice for KNN Classifier Cont'd

D1	0	**6**	0	0	0	1	1	1
D2	0	0	**10**	0	0	1	1	1
T1/T2	2	0	0	**1**	0	1	0.33	0.5
T3	0	0	0	0	**4**	1	1	1

4.5 Comparison of Accuracies of Classifiers

Table 4.5 shows the classification accuracies of machine learning techniques employed. SVM and KNN achieve the highest classification accuracy, 95.65%, and the decision tree achieves the lowest, 89.13%. When comparing the outcomes, SVM and KNN emerge as the most appropriate diagnostic techniques for the IEC TC10 dataset.

Table 4.4 Comparison of Classification Accuracies

SN	Classification Method	Actual Prediction/Test Samples	Testing Accuracy Score (%)	Training Accuracy Score (%)
1.	Decision Tree	41/46	89.13	100
2.	SVM	44/46	95.65	92.86
3.	KNN	44/46	95.65	91.21

4.6 Comparison of Receiver Operating Characteristics Curves

The Receiver Operating Characteristic (ROC) curves and their corresponding Area Under Curve (AUC) are shown in Figure 4.1. The ROC and AUC curve values provide insights into each classifier's performance in distinguishing between classes.

The Decision Tree classifier's ROC curve is generally lower than the other classifiers, indicating less discriminative power. The curve might seem less even and may not extend as near to the upper-left corner of the graph. A lower AUC value suggests that the Decision Tree may struggle more with separating the classes, especially compared to more complex models.

The SVM classifier's ROC curve is higher and smoother, indicating better performance in distinguishing between the different classes. The curve is expected closer to the top-left corner, reflecting the SVM's strength in finding an excellent separating hyperplane. A higher

AUC value for SVM signifies that it has a better ability to correctly classify instances across various threshold values compared to the Decision Tree.

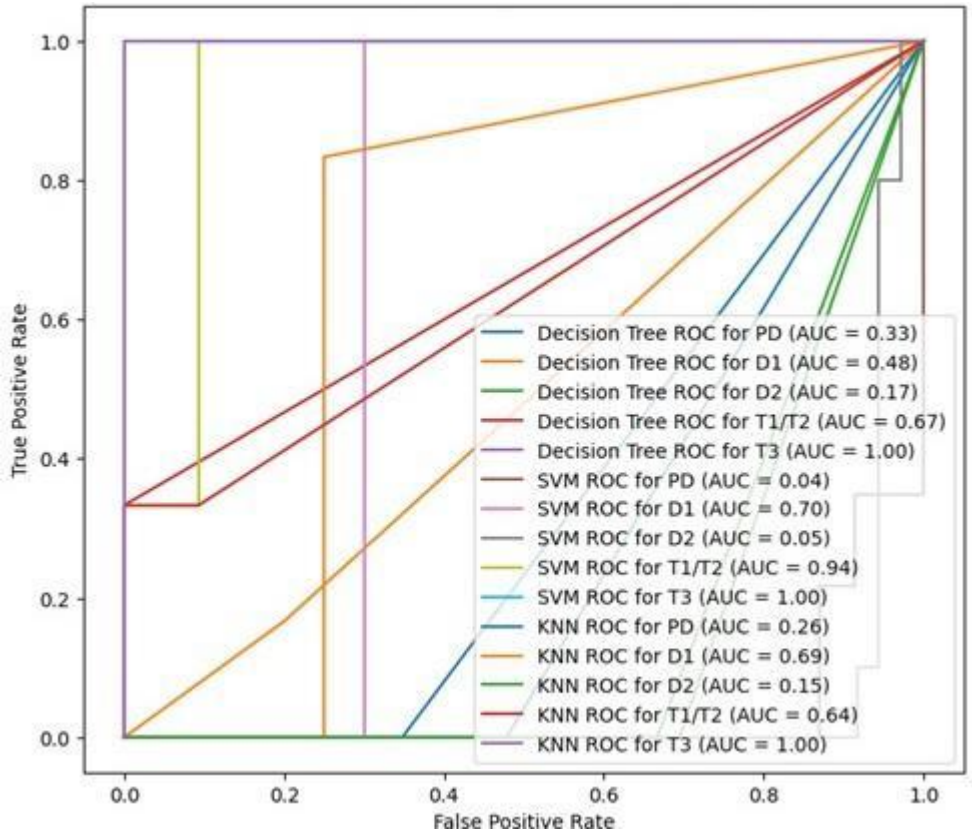

Figure 4.1 ROC Curve

The ROC curve for the KNN classifier show performance that falls between the Decision Tree and SVM. The variability it displays can change based on the selection of `k` and the distance measure employed. The AUC for KNN might be better than the Decision Tree but potentially lower than the SVM. The performance of KNN can be sensitive to local data characteristics and parameter settings.

Based on this evaluation, it can be concluded that the SVM classifier is the most suitable choice for a predictive model to diagnose and classify transformer faults.

4.7 Comparison of Precision-Recall Curves

Figure 4.2 shows the Precision-Recall (PR) curves and their corresponding Average Precision (AP) scores. The PR curves and AP scores provide insights into each classifier's performance, particularly regarding precision and recall.

The PR curve for the Decision Tree classifier show lower precision and recall than the other classifiers, especially for specific faults. This indicates that while the Decision Tree may correctly classify some instances, it might have a higher false positive or false negative rate. A lower AP score reflects that the Decision Tree may have difficulty maintaining high precision and recall across different thresholds, especially for imbalanced classes.

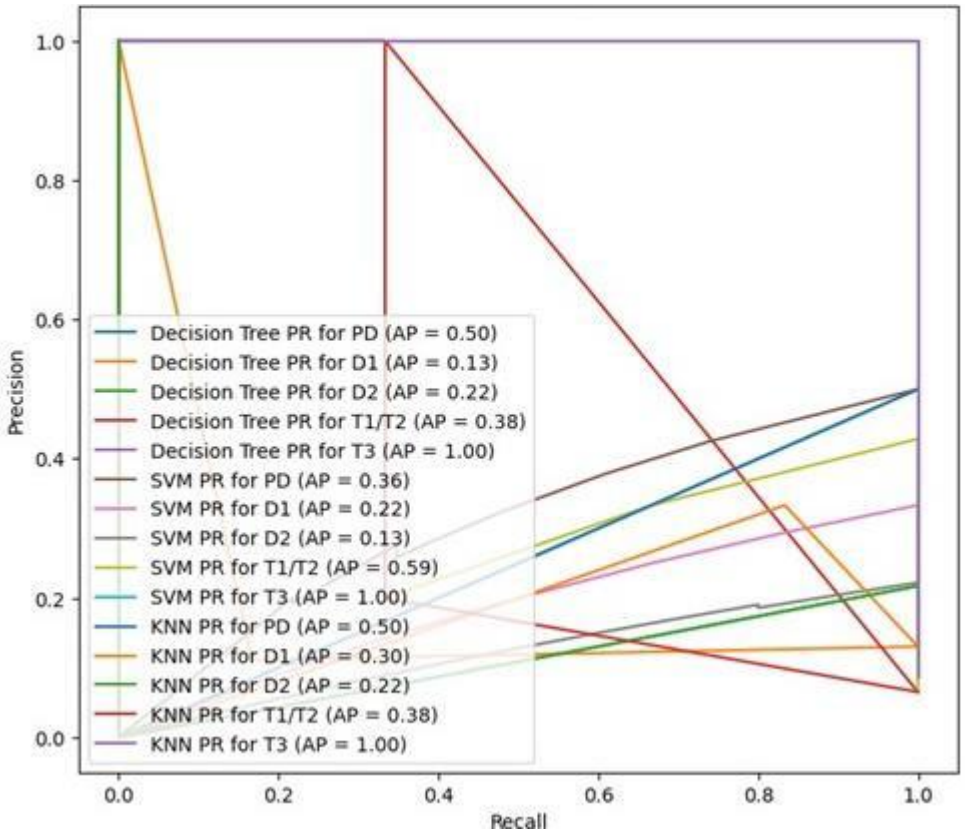

Figure 4.2 PR Curves

The SVM classifier's PR curve is closer to the top-right corner, indicating better overall precision and recall. SVM's performance in handling imbalanced data often results in a more balanced precision-recall trade-off. A higher AP score for SVM signifies strong performance in precision and recall, making it effective in scenarios with imbalanced classes.

The PR curve for KNN show competitive performance but not as strong as SVM. KNN's performance vary depending on the selected `k`, impacting the balance between precision and recall. The AP score for KNN might be better than the Decision Tree but can be lower than the SVM, indicating that while KNN can achieve good precision and recall, its performance is sensitive to parameter choices.

Based on this assessment, it can be stated that SVM classifier is best for a predictive model that can be used in diagnosing and classifying transformer faults.

4.8 Summary of Findings

Based on the results of the developed models, it can be stated that the models can work accurately and effectively and have the ability to:

i. Correctly classify and distinguish various datasets into appropriate classes due to the high training and testing accuracies.
ii. Capture and learn the hidden and complicated features of the input dataset, enabling the model to distinguish correctly between the six classes of transformer faults using their key gases.
iii. Apply knowledge from training to make predictions on unseen or hidden data used for testing.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

Based on the findings of this research, the following conclusions can be made:

i. The study's key findings on the predictive maintenance of power transformers using machine learning methods, including decision tree, SVM, and KNN, yielded impressive results of 89.13%, 95.65%, and 95.65%, respectively.

ii. The results obtained highlights the robust ability of the machine learning models to classify power transformer faults.

5.2 Recommendations

It is recommended that:

i. Power companies are strongly encouraged to adopt more machine learning approaches, such as the ones developed in this project, instead of relying solely on traditional maintenance methods for power transformers. This shift can significantly improve maintenance efficiency and transformer reliability.

ii. Power stations readily make available data on dissolved gases in transformer oil. This will facilitate future advanced studies and improve the accuracy and effectiveness of predictive maintenance for power transformers.

5.3 Future Works

Future research should consider:

i. Exploring techniques such as Artificial Neural Networks (ANN) to focus on more hidden patterns in the dataset that contribute most to prediction and aid the model's decision-making process.

ii. The use of ensemble learning, a technique that uses multiple models for prediction to boost accuracy and robustness.

REFERENCES

Amer, S., Mohamed, H. K. and Mansour, M. B. M. (2023), "Predictive Maintenance by Machine Learning Methods", *2023 IEEE Eleventh International Conference on Intelligent Computing and Information Systems (ICICIS)*, pp. 58 – 66.

Anon. (2019), "Different Types of Transformers and Their Applications" *https://circuitdigest.com/tutorial/different-types-of-transformers-and-their-applications.* Accessed: June 10, 2024.

Anon. (2020), "Types of Transformers Based on Classification - with Explanation" *https://www.electricaldeck.com/2020/12/types-of-transformers-based-on-classification.html.* Accessed: June 10, 2024.

Anon. (2022), "Classification in Machine Learning: An Introduction" *https://www.datacamp.com/blog/classification-machine-learning*. Accessed: July 01, 2024.

Anon. (2024a), "What is a Transformer" *https://circuitglobe.com/what-is-a-transformer.html*. Accessed: June 10, 2024.

Anon. (2024b), "Power Transformer" *https://www.iqsdirectory.com/articles/electric-transformer/power-transformers.* Accessed: June 10, 2024.

Anon. (2024c), "Difference between Step-up and Step-down Transformer" *https://circuitglobe.com/difference-between-step-up-and-step-down-transformer.html*. Accessed: June 10, 2024.

Anon. (2024d), "Isolation Transformer" *https://www.electrical4u.com/isolation-transformer.* Accessed: June 10, 2024.

Anon. (2024e), "Difference between Single-Phase and Three-Phase Transformer" *https://www.tutorialspoint.com/difference-between-single-phase-and-three-phase-transformer.* Accessed: June 10, 2024.

Anon. (2024f), "Autotransformer – Its Types, Operation, Advantages and Applications" *https://www.electricaltechnology.org/2019/07/autotransformer.html.* Accessed: June 10, 2024.

Anon. (2024g), "DataSpell- JetBrains Tool for Data Analysts" https://www.jetbrains.com/dataspell. Accessed: June 10, 2024.

Aqueveque, P., Radrigan, L., Pastene, F., Morales, A. S. and Guerra, E. (2021), "Data-Driven Condition Monitoring of Mining Mobile Machinery in Non-Stationary Operations Using Wireless Accelerometer Sensor Modules", *IEEE Access,* Vol. 9, pp. 17365 – 17381.

Bai, Y., Chen, F., Wang, H., Xiong, C. and Mei, S. (2023), "Transformers as Statisticians: Provable In-Context Learning with In-Context Algorithm Selection", *Advances in Neural Information Processing Systems,* pp. 1 – 87.

Doe, J. and Smith, A. (2023), "Decision Tree Classifier: A Detailed Survey", *IEEE Transactions on Knowledge and Data Engineering,* Vol. 29, No. 3, pp. 123 – 133.

Duan, Z. and Wang, X. (2023), "Application of Artificial Intelligence Technology in Power Equipment Condition Prediction and Maintenance", *2023 International Conference on Power, Electrical Engineering, Electronics and Control (PEEEC),* Athens, Greece, September 25th-27th, 2023, pp. 86 – 90.

Howard, W. P. (2022), "Machine Learning for Electric Machine Prognostics and Remaining Useful Life with Basic Motor Data", *2022 Electrical Insulation Conference (EIC),* Tennessee, USA, June 19th-23rd, 2022, pp. 245 – 248.

Johnson, M. and Wang, L. (2024), "Support Vector Machines: Optimisation and Applications", *IEEE Transactions on Neural Networks,* Vol 31, No. 2, pp. 220 – 230.

Kastelan, N., Vujović, I., Krčum, M. and Assani, N. (2022), "Switchgear Digitalization—Research Path, Status, and Future Work", *Sensors,* Vol. 22, No. 20, pp. 1 – 15.

Laayati, O., Bouzi, M. and Chebak, A. (2021), "Design of an Oil Immersed Power Transformer Monitoring and Self Diagnostic System Integrated in Smart Energy Management System", *2021 3rd Global Power, Energy and Communication Conference (GPECOM),* Antalya, Turkey, October 5th-8th, 2021, pp. 240 – 245.

Lee, R. and Brown, P. (2023), “A Comprehensive Survey on K-Nearest Neighbors”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 6, pp. 987 – 996.

Meitei, S. N., Borah, K. and Chatterjee, S. (2021), “Partial Discharge Detection in an Oil-Filled Power Transformer Using Fiber Bragg Grating Sensors: A Review”, *IEEE Sensors Journal*, Vol. 21, pp. 10304 – 10316.

Nanfak, A., Eke, S., Kom, C. H., Mouangue, R. and Fofana, I. (2021), “Interpreting Dissolved Gases in Transformer Oil: A New Method Based on the Analysis of Labelled Fault Data”, *IET Generation, Transmission & Distribution*, Vol. 15, pp. 3032 – 3047.

Pileggi, P., Lazovik, E., Snijders, R., Axelsson, L. U., Drost, S., Martinelli, G., De Grauw, M. and Graff, J. (2021), “A Lesson on Operationalizing Machine Learning for Predictive Maintenance of Gas Turbines”, *ASME Turbo Expo 2021: Turbomachinery Technical Conference and Exposition*, London, England, June 22nd-26th, 2021, Vol. 84966, pp. 1 – 12.

Raghuraman, R. and Darvishi, A. (2022), “Detecting Transformer Fault Types from Dissolved Gas Analysis Data Using Machine Learning Techniques”, *2022 IEEE 15th Dallas Circuit and System Conference (DCAS)*, Texas, USA, June 17th-19th, 2022, pp. 1 – 5.

Sarro, F., Moussa, R., Petrozziello, A. and Harman, M. (2020), “Learning from Mistakes: Machine Learning Enhanced Human Expert Effort Estimates”, *IEEE Transactions on Software Engineering*, Vol. 48, No. 6, pp. 1868 – 1882.

Sudha, B., Praveen, L. S. and Vadde, A. (2022), “Classification of Faults in Distribution Transformer Using Machine Learning”, *Materials Today: Proceedings*, Vol. 58, pp. 616 – 622.

Vallim Filho, A. R. A., Moraes, D. F., De Aguiar Vallim, M. V. B., Da Silva, L. S. and Da Silva, L. A. (2022), “A Machine Learning Modelling Framework for Predictive Maintenance Based on Equipment Load Cycle: An Application in a Real World Case”, *Energies*, No. 10, pp. 1 – 47.

Venkataswamy, R., Rao, K. U. and Meena, P. (2020), “Internet of Things Based Metaheuristic Reliability Centred Maintenance of Distribution Transformers”, *IOP Conference Series: Earth and Environmental Science*, Vol. 463, No. 1, pp. 1 – 8.

Yu, W., Yu, R. and Li, C. (2022), “An Information Granulated Based SVM Approach for Anomaly Detection of Main Transformers in Nuclear Power Plants”, *Science and Technology of Nuclear Installations*, Vol. 1, pp. 1 – 11.

APPENDICES

APPENDIX A

DATASET USED

SN	H_2	CH_4	C_2H_6	C_2H_4	C_2H_2	Equip.	Actual Fault
1.	13	3	1	3	6	U	D2
2.	10000	6730	345	7330	10400	U	D2
3.	13500	6110	212	4510	4040	U	D2
4.	34	21	4	49	56	U	D2
5.	137	67	7	53	104	U	D2
6.	310	230	54	610	760	U	D2
7.	420	250	41	530	800	U	D2
8.	620	325	38	181	244	U	D2
9.	800	160	23	260	600	U	D2
10.	1570	735	87	1330	1740	U	D2
11.	2850	1115	138	1987	3675	U	D2
12.	7020	1850	0.001	2960	4410	U	D2
13.	32930	2397	157	0.001	0.001	U	PD
14.	1450	940	211	322	61	U	T1/T2
15.	3675	6392	2500	7691	5	U	T1/T2
16.	100	200	110	670	11	U	T3
17.	150	22	9	60	11	U	T3
18.	860	1670	30	2050	40	U	T3
19.	1860	4980	0.001	10700	1600	U	T3
20.	8	0.001	0.001	43	101	S	D1
21.	35	6	3	26	482	S	D1
22.	6870	1028	79	900	5500	S	D1
23.	10092	5399	530	6500	37565	S	D1
24.	210	43	12	102	187	S	D2
25.	1084	188	8	166	769	S	D2
26.	1100	1600	221	2010	26	S	T3
27.	650	81	170	51	270	S	D1
28.	1900	530	35	383	434	R	D2

29.	2800	2800	234	3500	3600	R	D2
30.	5100	1430	0.001	1140	1010	R	D2
31.	5900	1500	68	1200	2300	R	D2
32.	8200	3790	250	4620	5830	R	D2
33.	480	1075	298	1132	0.001	R	T1/T2
34.	2031	149	20	3	0.001	R	T1/T2
35.	1	8	8	100	6	R	T3
36.	12705	23498	6047	34257	5188	R	T3
37.	300	700	280	1700	36	R	T3
38.	1550	2740	816	5450	184	R	T3
39.	3910	4290	626	6040	1230	R	T3
40.	4	1	2	7	52	U	D1
41.	543	120	41	411	1880	U	D1
42.	1900	285	31	957	7730	U	D1
43.	1270	3450	520	1390	8	P	T1/T2
44.	3420	7870	1500	6990	33	P	T1/T2
45.	6	2990	29990	26076	67	P	T3
46.	107	143	34	222	2	P	T3
47.	400	940	210	820	24	P	T3
48.	290	966	299	1810	57	P	T3
49.	290	1260	231	820	8	P	T3
50.	2500	10500	4790	13500	6	P	T3
51.	6709	10500	1400	17700	750	P	T3
52.	8800	64064	72128	95650	0.001	P	T3
53.	60	10	4	4	4	R	D1
54.	385	60	8	53	159	R	D1
55.	1790	580	321	336	619	R	D1
56.	120	25	1	8	40	R	D1
57.	2177	1049	207	440	705	R	D1
58.	4230	690	5	196	1180	R	D1
59.	6454	2513	121	2159	6432	R	D1
60.	7600	1230	318	836	1560	R	D1

61.	90	28	8	31	32	R	D2
62.	4419	3564	668	2861	2025	R	D2
63.	5000	1200	83	1000	1100	R	D2
64.	99	170	20	200	190	R	D2
65.	110	62	90	140	250	R	D2
66.	120	31	0.001	66	94	R	D2
67.	220	77	22	170	240	R	D2
68.	245	120	18	131	167	R	D2
69.	305	85	25	197	130	R	D2
70.	530	345	85	266	250	R	D2
71.	535	160	16	305	680	R	D2
72.	810	580	111	570	490	R	D2
73.	1900	530	35	383	434	R	D2
74.	1330	10	20	66	182	P	D1
75.	75	15	7	14	26	P	D2
76.	1820	405	35	365	634	P	D2
77.	60	5	2	21	21	P	D2
78.	2770	660	54	712	763	P	D2
79.	260	215	35	334	277	P	D2
80.	440	89	19	304	757	P	D2
81.	545	130	16	153	239	P	D2
82.	755	229	32	404	460	P	D2
83.	1170	255	18	312	325	P	D2
84.	1500	395	28	395	323	P	D2
85.	1570	1110	175	1780	1830	P	D2
86.	7150	1440	97	1210	1760	P	D2
87.	20000	13000	1850	29000	57000	P	D2
88.	3090	5020	323	3800	2540	P	D2
89.	3700	1690	128	2810	3270	P	D2
90.	12	18	4	4	0.001	P	T1/T2
91.	14	44	124	7	1	P	T1/T2
92.	48	610	29	10	0.001	P	T1/T2

93.	66	60	2	7	0.001	P	T1/T2
94.	57	24	2	27	30	B	D1
95.	1000	500	1	400	500	B	D1
96.	8266	1061	22	0.001	0.001	B	PD
97.	0.001	18900	410	540	330	B	T1/T2
98.	40000	400	70	600	6	B	T1/T2
99.	210	22	6	6	7	C	D1
100.	150	130	9	55	30	C	D2
101.	7940	2000	355	3120	5390	C	D2
102.	33046	619	58	2	0.001	I	PD
103.	92600	10200	0.001	0.001	0.001	I	PD
104.	9340	995	60	6	7	I	PD
105.	26788	18342	2111	27	0.001	I	PD
106.	36036	4704	554	5	10	I	PD
107.	37800	1740	249	8	8	I	PD
108.	40280	1069	1060	1	1	I	PD
109.	360	610	259	260	9	I	T1/T2
110.	960	4000	1290	1560	6	I	T1/T2
111.	1	27	49	4	1	I	T1/T2
112.	24700	61000	26300	42100	1560	I	T1/T2
113.	305	100	33	161	541	P	D1
114.	595	80	9	89	244	P	D1
115.	78	20	11	13	28	P	D1
116.	95	10	0.001	11	39	P	D1
117.	645	86	13	110	317	P	D1
118.	1230	163	27	233	692	P	D1
119.	1330	10	20	66	182	P	D1
120.	75	15	7	14	26	P	D1
121.	1820	405	35	365	634	P	D2
122.	60	5	2	21	21	P	D2
123.	2770	660	54	712	763	P	D2

APPENDIX B

CODES FOR DEVELOPING MODELS

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.metrics import roc_curve, auc, precision_recall_curve, average_precision_score
from sklearn.utils import resample
from collections import Counter
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import label_binarize

# Load the CSV file
file_path = 'Gas_Data2.csv'  # Ensure this path is correct on your local machine
gas_data = pd.read_csv(file_path)

# Drop the 'Equip' column
gas_data = gas_data.drop(columns=['Equip.'])

# Separate the features and the target variable
X = gas_data.drop(columns=['Actual Fault'])
y = gas_data['Actual Fault']

# Balance the dataset by resampling
df = pd.concat([X, y], axis=1)
df_minority = df[df['Actual Fault'] == df['Actual Fault'].value_counts().idxmin()]
df_majority = df[df['Actual Fault'] != df['Actual Fault'].value_counts().idxmin()]
```

```
df_minority_upsampled = resample(df_minority,
                                 replace=True,     # sample with replacement
                                 n_samples=len(df_majority),    # to match majority class
                                 random_state=123) # reproducible results

df_balanced = pd.concat([df_majority, df_minority_upsampled])

# Separate the features and the target variable again
X_balanced = df_balanced.drop(columns=['Actual Fault'])
y_balanced = df_balanced['Actual Fault']

# Split the data into training and testing sets with 80% training and 20% testing
X_train, X_test, y_train, y_test = train_test_split(X_balanced, y_balanced, test_size=0.2,
random_state=42, stratify=y_balanced)

# Identify categorical and numerical columns
categorical_cols = X_train.select_dtypes(include=['object']).columns
numerical_cols = X_train.select_dtypes(exclude=['object']).columns

# Preprocessing pipelines for both numeric and categorical data
numeric_transformer = Pipeline(steps=[
    ('scaler', StandardScaler())
])

categorical_transformer = Pipeline(steps=[
    ('onehot', OneHotEncoder(handle_unknown='ignore'))
])

preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, numerical_cols),
        ('cat', categorical_transformer, categorical_cols)
    ])
```

```
# Create a function to fit and predict using different classifiers
def fit_and_evaluate_model(model, X_train, X_test, y_train, y_test):
    pipeline = Pipeline(steps=[('preprocessor', preprocessor),
                               ('classifier', model)])
    pipeline.fit(X_train, y_train)

    # Training accuracy
    y_train_pred = pipeline.predict(X_train)
    train_accuracy = accuracy_score(y_train, y_train_pred)

    # Testing accuracy
    y_test_pred = pipeline.predict(X_test)
    test_accuracy = accuracy_score(y_test, y_test_pred)

    # Get prediction probabilities for ROC and AP
    y_test_prob = pipeline.predict_proba(X_test)

    return y_test_pred, y_test_prob, train_accuracy, test_accuracy

# Initialize classifiers
classifiers = {
    'Decision Tree': DecisionTreeClassifier(random_state=42),
    'SVM': SVC(kernel='linear', probability=True, random_state=42),  # probability=True
for predict_proba
    'KNN': KNeighborsClassifier()
}

# Fit and evaluate each classifier
results = {}
for name, clf in classifiers.items():
    y_test_pred, y_test_prob, train_accuracy, test_accuracy = fit_and_evaluate_model(clf,
X_train, X_test, y_train, y_test)
    results[name] = {
```

```
        'predictions': y_test_pred,
        'probabilities': y_test_prob,
        'train_accuracy': train_accuracy,
        'test_accuracy': test_accuracy,
        'classification_report': classification_report(y_test, y_test_pred, output_dict=True),
        'confusion_matrix': confusion_matrix(y_test, y_test_pred, labels=['PD', 'D1', 'D2',
'T1/T2', 'T3'])
    }
    print(f'Accuracy ({name}):')
    print(f'  Training Accuracy: {train_accuracy * 100:.2f}%')
    print(f'  Testing Accuracy: {test_accuracy * 100:.2f}%')

# Print the number of each fault classified by each model
for name, result in results.items():
    fault_counts = Counter(result['predictions'])
    print(f'\nFault Counts for {name}:')
    for fault, count in fault_counts.items():
        print(f'Fault: {fault}, Count: {count}')

# Calculate precision, recall, and F1-score
for name, result in results.items():
    print(f'\nClassification Report for {name}:')
    report = result['classification_report']
    for fault, metrics in report.items():
        if fault not in ['accuracy', 'macro avg', 'weighted avg']:
            print(f"Fault: {fault}")
            print(f"  Precision: {metrics['precision']:.2f}")
            print(f"  Recall: {metrics['recall']:.2f}")
            print(f"  F1-score: {metrics['f1-score']:.2f}")

# Display confusion matrices with fault types in the specified order
fault_order = ['PD', 'D1', 'D2', 'T1/T2', 'T3']
for name, result in results.items():
    cm = result['confusion_matrix']
```

```
    plt.figure(figsize=(10, 7))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=fault_order, yticklabels=fault_order)
    plt.title(f'Confusion Matrix for {name}')
    plt.xlabel('Predicted Fault')
    plt.ylabel('Actual Fault')
    plt.show()

# ROC and PR curves for each classifier
plt.figure(figsize=(14, 12))
for name, result in results.items():
    y_test_bin = label_binarize(y_test, classes=['PD', 'D1', 'D2', 'T1/T2', 'T3'])
    for i, fault in enumerate(['PD', 'D1', 'D2', 'T1/T2', 'T3']):
        fpr, tpr, _ = roc_curve(y_test_bin[:, i], result['probabilities'][:, i])
        roc_auc = auc(fpr, tpr)

        plt.subplot(2, 2, 1)
        plt.plot(fpr, tpr, label=f'{name} ROC for {fault} (AUC = {roc_auc:.2f})')
        plt.title('Receiver Operating Characteristic (ROC)')
        plt.xlabel('False Positive Rate')
        plt.ylabel('True Positive Rate')
        plt.legend(loc='lower right')

        precision, recall, _ = precision_recall_curve(y_test_bin[:, i], result['probabilities'][:, i])
        ap_score = average_precision_score(y_test_bin[:, i], result['probabilities'][:, i])

        plt.subplot(2, 2, 2)
        plt.plot(recall, precision, label=f'{name} PR for {fault} (AP = {ap_score:.2f})')
        plt.title('Precision-Recall (PR)')
        plt.xlabel('Recall')
        plt.ylabel('Precision')
        plt.legend(loc='lower left')

plt.tight_layout()
```

```
plt.show()

# Create a DataFrame to display the results
results_df = pd.DataFrame({
    'Test Data Index': X_test.index,
    'Predicted Fault (Decision Tree)': results['Decision Tree']['predictions'],
    'Predicted Fault (SVM)': results['SVM']['predictions'],
    'Predicted Fault (KNN)': results['KNN']['predictions'],
    'Actual Fault': y_test.values
})

# Print the DataFrame
print(results_df.head(10))  # Print the first 10 rows to check
print(f'Test Set Size: {len(X_test)}')
```