Enhancing Transformer Oil Dissolved Gas Analysis Using Deep Learning and Ensemble Classification

A thesis submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfilment of the requirements of the degree of

Master of Science

by

Kale K.J. Ewasiuk

Department of Electrical and Computer Engineering
Price Faculty of Engineering
University of Manitoba
Winnipeg, MB, Canada

University of Manitoba

Examining Commitee

This thesis was examined on March 18, 2024, and was approved by the examining committee consisting of:

- **Dr. Behzad Kordi**, Electrical & Computer Engineering (advisor);
- **Dr. Nathan Jacob**, Electrical & Computer Engineering (co-advisor);
- **Dr. Shaahin Filizadeh**, Electrical & Computer Engineering; and
- **Dr. David Swatek**, Electrical & Computer Engineering.

Abstract

Power transformers are critical assets in the power system, and routine condition assessment of them is necessary to ensure reliable and cost-effective energy transmission. As a transformer operates, thermal and electrical stresses degrade its electrical insulation and remaining lifespan, and these degradation processes produce gas compounds that are dissolved in its insulating oil. Dissolved gas analysis (DGA) assesses a transformer's condition by measuring these gasses, and numerous interpretation techniques exist based on physical mechanisms and experimental data to determine the associated faults that caused the gassing. Artificial neural networks (ANN) are a deep learning topology, which is a type of artificial intelligence (AI). ANNs are proven to be useful tools for determining complex relationships in data — particularly for classification purposes — and lend themselves suitable for DGA interpretation. Due to the dependence of AI models on their provided datasets, uncertainty regarding their accuracies when predicting new assessments is expected. This thesis explores the use of ANNs to interpret DGA for improved certainty in fault classification, and postulates that the existing DGA interpretation methods encode relationships that are not easily learned by ANNs. This work proposes a methodology that combines select methods (e.g., Duval's Triangle 1, Duval's Pentagon 1, and IEC Ratios) as input features along with a structure that also learns novel gassing-fault relationships in an attempt to improve performance and trust with DGA interpretation. Three models with varied input features are optimized with respect to their hidden-layer dimensions and training parameters, and are evaluated using repeated random subsampling with K-fold validation. The results from this work demonstrate that using ANN's for DGA interpretation offers improvements over conventional methods — notably in low-temperature fault classification — and shows the best-performing model for the given dataset is achieved when industry methods along with gassing data are used as input features. Where the existing industry methods can conflict on assessments or provide unreliable results, the proposed models can offer a more compelling diagnosis of a transformer fault, and ultimately empower the asset's owner to optimize its lifecycle.

Acknowledgements

The author of this thesis would like to acknowledge:

- **Dr. Behzad Kordi**, for his support and patience throughout this endeavour;
- **Dr. Nathan Jacob**, for his valuable insight and contributions;
- **the examining committee**, for their time and attention;
- **Camlin Energy**, for providing the crucial data required for this work;
- **Natural Sciences and Engineering Research Council (NSERC) of Canada**, for partial financial support; and
- **my family**, for their unwavering love and encouragement.

Contents

List of Tables

List of Figures

Abbreviations and Symbols

Adam	Adaptive Moment Estimation
AI	artificial intelligence
ANN	artificial neural network
C_2H_2	acetylene
C_2H_4	ethylene
C_2H_6	ethane
CH_4	methane
CNN	convolutional neural network
CO	carbon monoxide
CO_2	carbon dioxide
D1	low energy discharge (sparking)
D2	high-energy discharge (arcing)
DGA	dissolved gas analysis
DL	deep learning
DP1	Duval's Pentagon 1
DP2	Duval's Pentagon 2
DRM	Doernenburg Ratios Method
DT1	Duval's Triangle 1
DT4	Duval's Triangle 4
DT5	Duval's Triangle 5
GAN	generative adversarial network
H_2	hydrogren
IECR	IEC Ratios Method
kNN	k-nearest neighbours
LSTM	long short-term memory
ML	machine learning
MLP	multilayer perceptron
N_2	nitrogen
NLTC	no-load tap changer
O_2	oxygen
OLTC	on-load tap changer
PD	partial discharge
ppm	parts per million
R1	Ratio 1: CH_4/H_2
R2	Ratio 2: C_2H_2/C_2H_4
R3	Ratio 3: C_2H_2/CH_4
R4	Ratio 4: C_2H_6/C_2H_2
R5	Ratio 5: C_2H_4/C_2H_6
RNN	recurrent neural network
SVM	support vector machine
T1	low-temperature overheating
T2	mid-temperature overheating
T3	high-temperature overheating

1

Introduction

This chapter provides context and motivation for this thesis. The objectives, outline, and machine learning framework for which the work was completed in is discussed.

1.1 Motivation

Transformers can be thought of as electrical "gear trains" in that they are used to adjust voltage (V) and current (I) inverse-proportionally; i.e., $V_1/V_2 = I_2/I_1$, which is analogous to how mechanical gear trains affect torque and angular velocity. Transformers allow a power system to transmit input power at a higher voltage and lesser current. The necessity of using high voltages is mainly economically driven: for the same power transfer, the current — and consequentially, volume of costly conductors for efficient transmission — is significantly reduced.

Power transformers are critical elements in a high voltage power system, and their unexpected failures can lead to power outages resulting in hardship and financial loss. A consequence of utilizing high voltages is that the subjected apparatuses are required to tolerate excessive electrical stress, and their insulation systems must be carefully designed accordingly. As transformers transmit relatively large amounts of power, thermal stress due to unavoidable copper and core losses, and physical stress from magnetic forces and vibrations are persistent.

Transformers are sometimes designed with little margin in electrical, thermal, and physical tolerances to achieve economical production [1]. Because the quality of the overall asset, including electrical insulation, degrades over the in-service life of the unit, insights towards if, when, and how severe problems are to occur in transformers are useful for asset management and planned maintenance. Nowadays, with increased economic pressures and supply chain limitations, leveraging data

using relatively inexpensive sensors and computational power presents a potentially low-cost and powerful tool for maximizing an asset's lifecycle.

Dissolved gas analysis (DGA) was introduced in the 1960s and still today remains one of the most valuable techniques for transformer condition assessment [2]. DGA is used to determine abnormal conditions in high voltage electrical apparatus by examining the chemical compounds dissolved in the electrical insulating oil. Thermal and electrical stresses contribute to the generation of gasses, and the quantities and ratios of chemicals can provide insight as to the location and severity of faults. DGA is considered a trusted tool; however, different interpretation methodologies can lead to conflicting results, non-diagnoses, often requiring analysis by a qualified expert.

The utilization of artificial intelligence (AI) and the inception of "machine learning" dates back to the 1950s; where every outcome in the game of checkers was programmed in a computer system [3]. Artificial neural networks (ANNs) were introduced in the 1970s, but received little interest until the 1990s. Since the 2010s, AI has been a hot topic in light of big data and cloud storage, and is now viewed as a tool available for technical fields outside of computer science.

Well-trusted methods like DGA interpretation for condition assessment has proven effective historically; however there is opportunity to improve an asset owner's confidence level. These immutable methods are based on known physical relationships and the culmination of empirical data derived from historical records. Employing machine learning to determine complex relationships between gassing quantities dissolved in transformer oil and incipient faults presents a suitable application for AI, given that DGA is fundamentally a classification problem. In addition to the well-known tools for DGA, it is conceivable to improve classification performance by combining these with an ANN trained on a dataset with gassing and the corresponding faults.

A gap in the current literature is the lack of using of trusted traditional techniques as feature inputs to neural networks. The hypothesis of this thesis that the existing DGA interpretation methods — which are developed upon known physical relationships — might not be easily learned by ANNs. Combining select methods

(e.g., Duval's Triangle 1, Duval's Pentagon 1, and IEC Ratios) as input features in a structure that also interprets new relationships based on the data it is trained on, may offer improved performance and trust from those who depend on DGA. The central work of this thesis is to investigate the combination of traditional methods in an ANN as the input features, and combining them in a structure that allows an ANN to determine gassing-fault relationships using relative concentrations of gassing and ratios of gasses as additional input features.

1.2 Thesis Objectives and Contributions

The objective of this thesis is to explore contributions to transformer condition assessment toolbox by enhancing DGA interpretation with deep learning. An ANN is proposed and optimized to improve DGA classification for mineral oil-based transformers beyond traditional methods. Input features including gas proportions, ratios of select gasses, and industry DGA interpretation predictions are combined in numerous ways and validated to determine the best performing group.

1.2.1 Publication

A publication titled: "Enhancing Transformer Oil Dissolved Gas Analysis Methods by Utilizing Artificial Neural Networks with Ensemble Classification" related to the work in this thesis was submitted for and presented in the CIGRE Canada 2023 conference in Vancouver, Canada [4].

1.3 Thesis Organization

This thesis is organized such that:

- Chapter 2 discusses fundamental concepts of transformers and DGA;
- Chapter 3 summarizes the applicable deep learning concepts;
- Chapter 4 provides literature review of machine learning concepts applied to DGA;
- Chapter 5 proposes an ANN-based methodology for DGA interpretation and analyses its performance; and

- Chapter 6 provides concluding remarks, including future work.

1.4 Framework

The work of this thesis outlined in Chapter 5 was completed with Python 3.10[1] and uses the PyTorch 1.13.0[2] library to construct the deep learning model. Additional libraries such as Pandas, NumPy, Scikit-Optimize, and Matplotlib were used for numerical analyses.

1. http://www.python.org
2. https://pytorch.org

2

Fundamentals of Transformer, Condition Assessment, and DGA

This chapter discusses fundamental transformer and DGA concepts. A brief introduction to transformer design and insulation systems is discussed. Fault gasses, transformer fault classifications, and interpretation techniques applicable to DGA are reviewed.

2.1 Transformer Design

Transformers are necessary, albeit costly apparatus. They are stationary devices that leverage electromagnetic induction to adjust voltage and current inverse-proportionally through the turns ratio between the primary and secondary side: that is, $V_p/V_s = I_s/I_p = N_p/N_n = N$, where N is the turns ratio formed by the winding; V_p and V_s are the primary and secondary voltages; and I_p and I_s are the primary and secondary currents. Transformers are typically available as single or three-phase devices, and can transfer relatively large quantities of power at high-voltages.

Thorough investigation and condition assessment throughout a transformer's life is necessary to ensure long term reliability and avoidance of forced outages. The basic design and operation mechanism of a transform is described in this section. Figure 2.1 depicts a cut-out view a transformer.

Windings of a transformer form a coil around the core and are energized to conduct the current. A primary winding is excited by a voltage which establishes magnetic flux that is coupled with a secondary winding. The changing flux present from an alternating voltage source on the primary side induces voltage and current in the secondary winding, which allows for galvanically isolated power transfer between the windings, which are typically made of copper or aluminium. Insulation is

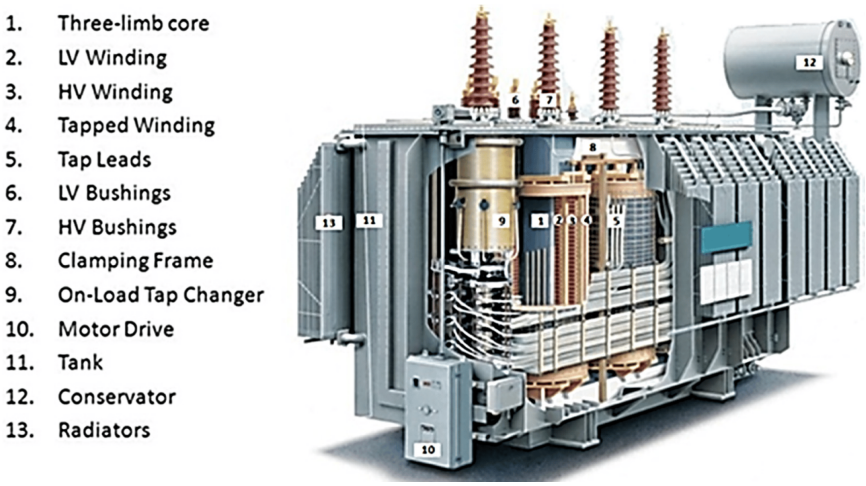


Figure 2.1 Cross-section of a transformer, adapted from [5] with permission. The windings and core, including the dielectric insulation system, form the active part which are necessary for power transfer to occur.

required to provide reliable electrical isolation between adjacent turns of the same windings, between high and low voltage windings, and between windings and the grounded core and tank or electrostatic shield if used. As windings conduct the current and are energized with a high voltage, they typically experience the greatest stress in a transformer and are considered attributable to most transformer faults [6].

Core of a transformer is a ferromagnetic loop used to contain the shared magnetic flux produced by the windings. Its principal function is to optimize the mutual inductance (magnetic coupling) between windings and reduce the leakage magnetic flux (leakage inductance). Laminated steel sheets are used to form the core and reduce heat-producing eddy currents. There exist core and shell types of transformers: core types utilize windings on separate legs, and shell types have the core surrounding the windings.

Bushings are the interface between the transformer internal windings and an external circuit. Bushings represent approximately 15 % of transformer failures [6].

Tank of a transformer houses the transformer components, protecting them from contamination and providing mechanical stability. The tank contains the transformer's oil and the conservator tank to allow for expansion due to heating.

Tap-Changers adjusts the turns ratio of the transformer for voltage control. There exists no-load tap changer that adjust the turns ratio in coarse steps but are designed for operation when the transformer is de-energized. On-load tap changers (OLTCs), on the other hand, are designed to operate while transformer is energized. Because current flows through OLTCs as they operate, arcing can occur and its byproducts may obfuscate DGA results.

2.2 Transformer Electrical Insulation System

The majority of transformers — especially high power transformers — utilize insulating oil and cellulose (paper and pressboard) for electrical insulation. Major insulation isolates the windings from ground potential, and minor insulation isolates the turns of the winding from each other. The solid insulation provides mechanical support, cooling ducts, and is a dielectric insulator.

The oil — typically hydrocarbon-based naphthenic, paraffinic, or aromatic mineral oil — provides dielectric strength, arc suppression, and heat absorption and transfer. The most common type of transformer oil is mineral oil, but synthetic oils such as silicone-based oils or ester-based oils are also used in certain applications [7]. Ester oil typically forms the same gases as mineral oils, but in different quantities [8, 9]. The work in this thesis is concerned with mineral oil.

2.3 Dissolved Gas Analysis (DGA)

DGA is applied to essentially all electrical apparatus with paper-oil based insulation systems, including current transformers and bushings. It employs gas chromatography to separate chemical compounds and quantify — typically in parts per million (ppm) — gasses of interest that arise from hydrocarbons formed from decomposing oil and cellulose due to thermal and electrical stress.

Generally, gas formation indicates stresses experienced by the transformer; however, normal temperature ageing will also produce gas over time [10]. A transformer is generally considered to be healthy and unconcerning if the gases of interest are

below certain values and is not increasing at a faster than normal rate [11]. DGA samples are drawn as part of routine transformer condition assessment, and for critical assets can be sampled daily and monitored "online". The following sections discuss the generation of gasses of interest and diagnostic techniques considered by DGA.

2.3.1 Gas Generation and Key-Gasses

Mineral oils are composed of various hydrocarbon molecules [11] (CH, CH_2, and CH_3 chemical groups). Chemical bonds of C-H and C-C can break as a result of electrical and thermal activity, producing radical or ionic H, CH_3, CH_2, CH, or C, for example. These recombine rapidly and produce chemicals such as H_2, CH_4, CH_4, and hydrocarbon polymers (X-wax) which are dissolved in the oil. Figure 2.2 provides a graphical depiction of the formation of gasses which are briefly discussed in the following paragraphs.

Acetylene (C_2H_2) is generally formed from temperatures above 700°C [2], and negligible quantities are created at lower temperatures. It is generated from arcing in oil or paper and can be associated with increased hydrogren and ethylene levels, and some CO and CO_2 accumulation as carbon particles form and the oil oxidizes. acetylene, generally speaking, is the most important gas for detecting severe faults and the presence of even trace amounts of a few ppm is a cause for concern. On-load tap changers are expected to produce acetylene due to transient arcs from breaking and making contacts between taps.

Hydrogen (H_2) production occurs under most faults, but high amounts are generally considered an indicator of discharge activity — predominately partial discharge (PD) — but is also correlated with sparking and arcing. Hydrogen also arises as a stray gas in oil from chemical reactions with steel [2].

Methane (CH_4) is generally associate with lower temperature faults, and is generated along with ethane and ethylene from the heating of oil or paper and lower energy discharges.

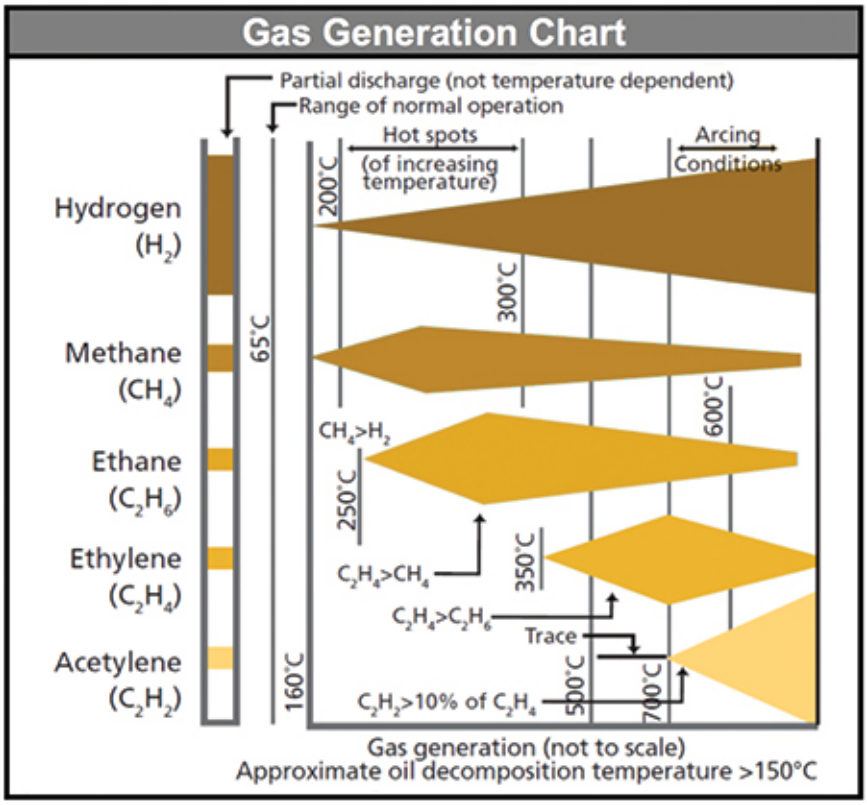


Figure 2.2 Gas generation chart, adapted from [12] with permission. Gas formation tends to begin at 150°C and above. H_2 production steadily increases with temperature, while CH_4, C_2H_6 and C_2H_4 increase to maximum production at particular temperatures, but reduce as temperature further increases. C_2H_2 begins formation under extremely high temperature conditions like arcing. Comparing C_2H_2, C_2H_4, C_2H_6, and CH_4, as they begin production at different temperatures can be used to determine fault temperatures. Under partial discharge conditions, a specific proportion of the gasses is typically produced.

Ethane (C_2H_6) indicates lower temperature faults with production beginning at around 250°C. Fault temperatures are then determined by comparing concentrations of dissolved acetylene, ethylene, ethane, and methane.

Ethylene (C_2H_4) Above 500°C, ethylene formation occurs at a greater rate than ethane and methane, but is still present at lower temperatures, and is therefore an indicator of higher temperature faults.

Carbon Dioxide (CO_2) and Carbon Monoxide (CO) indicate overheating of paper insulation and cellulose degradation. Generally, high values of carbon monoxide, on the order of 1000 ppm, and CO_2/CO ratios less than 3 may indicate faults occurring near the paper, causing possible carbonization. High carbon dioxide quantities, above 10 000 ppm and high CO_2/CO ratios above 10 can indicate mild overheating of paper or oil oxidation. carbon monoxide can accumulate on transformers with low breathing and yield CO_2/CO ratios less than 3.

Oxygen (O_2) is not a product of faults typically considered by DGA interpretations. Measuring oxygen is important as it facilitates oxidation of paper and oil and is coupled with moisture and heat.

Nitrogen (N_2) is not a fault gas but can be an indicator of air ingress into the tank of a sealed transformer, which may be associated with moisture and the issues discussed with oxygen.

2.3.2 Diagnoses

This section elaborates on the diagnoses typically determined by DGA. Although the following diagnostics and classifications are not inclusive of all faults that a transformer may experience, they are useful to direct an asset owner or transformer manufacturer to the root-cause of a problem or provide a level of urgency with which a transformer should be taken out of service and investigated.

Transformer faults assessed by DGA are broadly categorized into electrical and thermal faults. Thermal faults are generally caused by:

- insufficient cooling;

- excessive currents in adjacent metal components such as eddy currents caused by leakage flux;
- high dielectric losses leading to thermal runaway;
- heating of bushing winding connection lead; and
- overloading.

Electrical faults are generally caused by:

- insulation aging: over time, the insulation in electrical equipment can degrade, leading to the formation of voids or cracks that can increase partial discharge;
- electrical stress: overvoltages that exceed the insulation capability of the equipment cause discharge internal or external to the insulation that may cause puncture;
- contamination: dirt, dust, moisture, and other contaminants in the oil or surface of the insulation forming conductive paths;
- mechanical damage: physical damage to the insulation such as cracks, cuts, or punctures; and
- manufacturing defects and insulation deficiencies: defects in the manufacturing process can lead to voids, air pockets, or other defects in the insulation that can trigger partial discharge.

Stray Gassing refers to the natural decomposition of oil, cellulose, or unusual production of gassing under normal operating temperatures of 90 to 200°C, and is not attributable to fault conditions [13]. Refinement of oil can saturate it with hydrocarbons which are later released, or the oxidation of oil can cause gassing. Hydrogen is mostly generated at lower temperatures and methane and ethane at higher temperatures, but can be associated with stray gassing [10]. Stray gassing poses a challenge for DGA interpretation as it masks gassing caused by problematic phenomena and is commonly mistaken for PD due to hydrogren generation, and recent efforts are put towards discriminating it, including Duval's Pentagon 2 or HS-TS [14].

Other Gassing Corrosion, exposure to sunlight or other chemical reactions can produce gas. Coatings and steel reacting to water can produce hydrogren with oxy-

gen available from oil, and large amounts of hydrogren can be found in transformers that have yet to be energized.

Partial Discharge (PD) is a partially bridging electrical discharge in an insulation system. PD usually occurs in gaseous cavities due to humidity in paper, incomplete oil-impregnation, or cavitation. PD is useful as an indicator of insulation health, and when severe enough, erodes the insulation quality. An indicator for PD is the formation of X-wax which are polymerized fragments of liquid, and high amounts of hydrogen generation.

Thermal Faults Below 300°C (T1) are due to temporary overloading of the transformer, oil flow blockages, and stray flux inducing eddy current in non-active parts (beams, yokes). Paper, pressboard, and wood, contain weak carbon-oxygen bonds that are less thermally stable than hydrocarbon bonds in oil, and therefore decompose at lower temperatures starting at above 105°C, and considerably at temperatures above 300°C. T1 faults are indicated by paper turning brown in color.

Thermal Faults Between 300–700°C (T2) can be caused by poor electrical contacts between cable, defective welds on bushing draw-rods or gliding contacts, circulating current between laminations, or abraded insulation between parallel connections. T2 faults are indicated by the formation of carbon particles.

Thermal Faults 700°C and Above (T3) are caused by large circulating currents in tank and core. T3 faults are evidenced by large quantities of carbon particles in oil, metal coloration (occurs above 800°C), or metal fusion (occurs above 1000°C).

Discharge of Low Energy (Sparking) (D1) is categorized as sparking or arcing between poor connections or brazing, shielding electrodes, adjacent conductors of a winding, or high voltage and ground potential. D1 discharges can produce pinhole perforations in the paper and carbon surface tracking.

Discharge of High Energy (Arcing) D2 involve flashovers or severe tracking with power follow-through between conductors and are of high energy. They are evidenced by insulation and structural destruction, large quantities of carbon extensive metal fusion, and equipment trips.

Thermal Faults and Discharge is a diagnosis offered by some traditional methods and indicated a combination of thermal and electrical faults.

2.3.3 Industry Diagnostic Methods

This section discusses the traditional DGA methods. These methods are routinely used and considered reliable. The following gas ratios are defined [2]:

- Ratio 1: CH_4/H_2 ;
- Ratio 2: C_2H_2/C_2H_4 ;
- Ratio 3: C_2H_2/CH_4 ;
- Ratio 4: C_2H_6/C_2H_2 ; and
- Ratio 5: C_2H_4/C_2H_6 .

Duval's Triangle 1 (DT1) was developed from the IEC TC-10 database and [11] where numerous transformers were inspected. Methane (CH_4), ethylene (C_2H_4), and acetylene (C_2H_2) as a percentage over the sum of all three are used to produce one of seven faults — three discharge classes (including PD), three thermal, and a discharge and thermal. Graphically, DT1 can be interpreted as a ternary graph with each gas on a separate axis and delimiting zones to plot the specific fault. DT1 does not provide an "Ok" (no-fault) assessment.

Duval's Pentagon 1 (DP1) utilizes the five combustible gasses in a similar way that Duval's Triangle 1 does. The relative proportions are calculated, and various zones are defined for seven faults. Contrast to DT1, DP1 has a zone for stray gassing. A graphical depiction of DT1 and DP1 is shown in figure 2.3.

IEC Ratios Method (IECR) proposes the use of three ratios: CH_4/H_2, C_2H_2/C_2H_4, and C_2H_4/C_2H_6, and the implementation of a simple look-up table to determine the fault [11]. This method produces some overlap with D1 and D2 cases. The IEC Ratios method is simular to the Rogers Ratio Method [15], but uses different ratio ranges and fault diagnoses. A simplified scheme of interpretation is offered where:

- PD occurs when CH_4/H_2 is less than 0.2;
- Discharge (D1 or D2) occurs when C_2H_2/C_2H_4 is greater than 0.2; and
- Thermal (T1, T2, or T3) faults occurs when C_2H_2/C_2H_4 is less than 0.2.

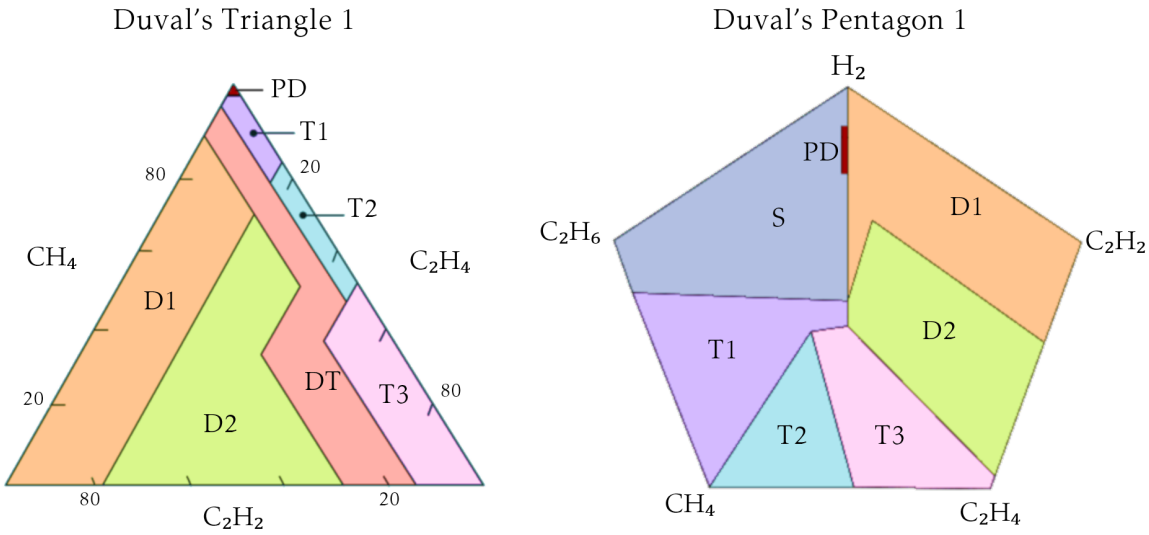


Figure 2.3 Graphical depiction of Duval's Triangle 1 and Duval's Pentagon 1.

Table 2.1 IEC ratios method [11].

Case	C_2H_2/C_2H_4	CH_4/H_2	C_2H_4/C_2H_6
PD	n/a	< 0.1	< 0.2
D1	> 1	$0.1 - 0.5$	> 1
D2	$0.6 - 2.5$	$0.1 - 1$	> 2
T1	n/a	> 1	< 1
T2	< 0.1	> 1	$1 - 4$
T3	< 0.2	> 1	> 4

n/a: not applicable

Key Gas uses the dominant gas to provide a cursory indication of a problem. CH_4 and C_2H_6 indicate low temperature thermal faults, C_2H_4 indicates high temperature heating, CO and CO_2 indicates thermal stress affecting oil and cellulose, H_2 indicates PD and D1, and C_2H_2 indicates D2.

Doernenburg Ratio Method (DRM) utilizes four ratios: CH_4/H_2, C_2H_2/CH_4, C_2H_4/C_2H_6, and C_2H_2/C_2H_4 [16]. Conditions of the four ratios can, and often lead to non-diagnostic results.

Additional Gas Ratios The CO_2/CO ratio is used to determine the severity of faults with paper but its accuracy is obfuscated as these gasses are produced in oil oxidization and normal cellulose aging. O_2/N_2 can be used to indicate heating and oil preservation in sealed transformers, and C_2H_2/H_2 often indicates OLTC contamination.

Duval's Triangles 4 and 5 and Pentagon 2 Duval's Triangles 4 and 5 are used as supplementary tools to assist in the further identification of low energy and medium to high energy faults, respectively. Duval's Pentagon 2 is used to discriminate between electrical faults and heating in oil (T3-H) compared with carbonization cause by heating (T3-C, T2-C, and T1-C).

2.3.4 Challenges with DGA

There are a number of challenges and opportunities for improvement with DGA, some of which lend themselves to be suitable for AI. Distinguishing normal operation gas quantities from abnormal is of importance. Normal gassing values are typically provided in literature, but these can be subjective and dependent on specificities of the transformer. This may be overcome by introducing additional information, such as transformer age, proclivities based on type and particular history of the transformer. Furthermore, load-break switches immersed in the same oil as the transformer or on-load tap changers will introduce gassing due to arcing which can mask gasses of interest. Noise introduced in the measurement due to imperfect sampling (contamination), measuring, and general uncertainty in measurements can lead to erroneous fault diagnoses and inappropriate actions being taken. Aside from a coarse assessment of thermal and discharge, multiple faults are typically

not assessed in conventional DGA methods. Solving these deficiencies by designing improved closed-loop systems like a look-up tables or graphs — as is the case with conventional DGA tools — presents an onerous challenge, but is still a topic of interest; for example, recent developments propose using a hexagon of the five hydrocarbons as well as carbon monoxide [17]. Applying AI however, is arguably the more contemporary and popular approach, and some of these efforts are described in Chapter 4. AI has the ability to self-learn input-output relationships from given datasets — in this case, gassing that corresponds to a particular fault — and is useful for classification problems like DGA interpretation. Additional relationships not covered by conventional DGA interpretation methods can be learned by an AI system in an attempt to improve interpretation performance. The following chapter presents the fundamental framework required to understand deep learning (DL) and why it can be used to enhance DGA.

3 Deep Learning Fundamentals

This chapter provides an overview of machine and deep learning concepts and techniques. The fundamental framework of artificial neural networks (ANN) is outlined with a focus on applicable features and advancements that were utilized for this thesis as described in Chapter 5.

3.1 Machine and Deep Learning

Machine learning is a type of algorithm that enables computer systems to make predictions with data without being explicitly programmed. Machine learning algorithms can be broadly categorized into three types: supervised learning, unsupervised learning, and reinforcement learning [18]. Unsupervised learning is focused towards associating and clustering unlabeled data sets and may include dimensionality reduction, principal component analysis, autoencoders, or determining association rules. Unsupervised learning can be useful for anomaly detection or recommendation engines, for example. Reinforcement learning uses trial and error to understand a system described by performing "actions" in a given system "state" and adjusting the model based on the "reward" (or penalty) arisen from a particular action. Reinforcement learning is useful for decision-making applications. Supervised learning — which is the type of machine learning in consideration for this thesis — typically deals with classification and regression problems. Supervised learning typically uses labeled data points to learn input-output relationships. Predictive analytics and object detection are common applications of supervised learning. Some challenges with supervised learning are that it requires some degree of expertise to structure the model, the performance is sensitive to the training dataset (and human error in building the dataset), and lacks the ability to cluster data on its own.

Deep learning is a subset of machine learning that generally involves the use of ANNs, and is employed in supervised, unsupervised, and reinforcement learning. Deep learning algorithms are characterized by their ability to learn multiple levels of representation in data in the form of "layers". Complex tasks such as image and speech recognition, natural language processing, or multivariate classification are proven to be well-suited for DL. There are numerous types of DL algorithms.

Artificial neural network (ANN) and multilayer perceptron (MLP) model layers of abstraction through matrix multiplications and non-linear activation functions, which loosely resembles the structure of interconnected neurons in the human brain. These types are often used for classification and regression problems [19].

Convolutional neural network (CNN) involves generating numerous "filters" through training that scan (i.e., are convolved with) input data. Convolutional neural networks (CNNs) are typically used for classification of data with spatial features: e.g., image or speech recognition [20].

Recurrent neural network (RNN) and long short-term memory (LSTM) is a type of neural network with memory that processes a sequence of inputs, typically used on time-series/temporal data for predicting next samples [21].

Generative adversarial network (GAN) is used for generative purposes. GANs use two neural networks combating each other in a zero-sum game [22] — i.e., the loss of one network is the gain of the other network — which results in one network trained to generate samples and the other to discriminate samples as real or generated. As training progresses, and the generator network improves its ability to produce samples which are difficult to distinguish from the training data.

3.2 Artificial Neural Networks (ANNs)

An ANN is fundamentally a sequence of matrix multiplications with intervening non-linear functions [19]. Input data is multiplied by said matrices — commonly referred to as "weights", and (optionally) offset by a "bias" array to produce an output. A non-linear activation function is applied to this output, which is then input to the next matrix. The input, final output, and intervening — known as

"hidden" — vectors results are referred to as "layers". Each layer of the network can be thought of as a feature mapping, where the new features are weighted sums of the input features to that matrix. The goal of training an ANN is to optimize the values of the weight and biases such that the error between the ANN's output and target output are minimized, but in a way such that the generalization — rather than memorization — of patterns in the data is learned. The goal of optimizing a neural network is determining the dimensions, layers, and training algorithm parameters. Several definitions pertinent to ANNs are discussed in the following sections.

3.2.1 Feedforward

Feedfoward, or forward pass, refers to the process of forwarding input data through the ANN to produce an output or prediction. It is the fundamental operation that occurs during both the training and prediction phases of the neural network. The term "feedforward" is used to distinguish this phase from the feedback or learning phase, wherein the network adjusts its parameters based on the error in its predictions (quantified by a loss or cost function).

A graphical depiction and mathematical formulation of an ANN operating in feed-forward if shown in figure 3.1. In this figure, the complete network is represented by an input vector x, hidden layers a, and an output vector y. The computation of a hidden layer's output is shown in the lower graphic, where the weights are depicted as the arrows between nodes, and are constituted by matrix $\mathbf{W}^{(0)}$ of size $m \times n$. The weights matrix multiplies an input column vector $a^{(0)}$ of size $n \times 1$ (a previous layer's output), and a bias column vector $b^{(0)}$ of size $m \times 1$ is added to the product to result in an output column vector $a^{(1)}$ (layer "1"). The activation function σ is applied element-wise to the $a^{(1)}$ vector. Rather than using a single column vector for an input, multiple data samples — or "batches" — are simultaneously computed by passing an input matrix of size $d \times n$ instead, where d is the batch size (in this graphic, $d = 1$).

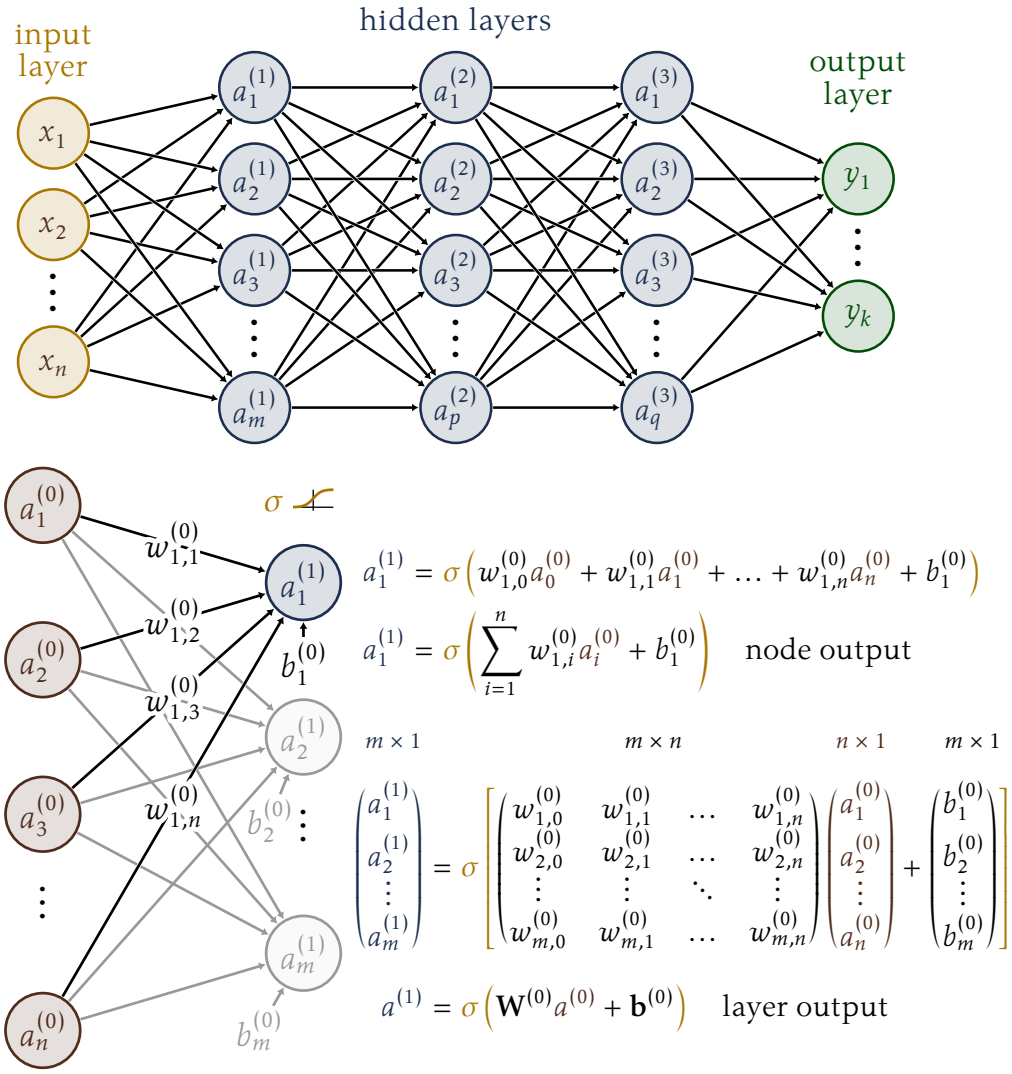


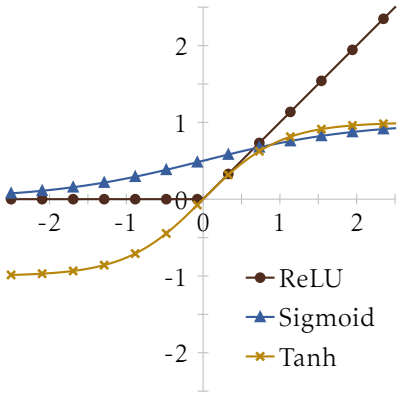
Figure 3.1 Graphical depiction of an ANN and mathematical formulation of a node and layer output during the feedforward process.

3.2.2 Activation Functions

Activation functions are functions applied to the output of each layer. An activation function generally has the following properties in consideration:

- introducing non-linearity;
- introducing a clamping/saturation effect on one or both ends of the input-output curve to improve numerical stability;
- have low impact on computational complexity;
- have a simple solution for the derivative of the function to improve gradient descent algorithm performance (see section 3.3.1); and
- retains the data distribution (i.e., the curve is mostly one-to-one).

The use of activation functions are what provides *depth* to an ANN, and the ability to imitate non-linear relationships [23]. Without activation functions, fully-connected layers would resolve into a single matrix multiplication. Select activation functions are demonstrated in figure 3.2, and several others are discussed in [24].



$$\text{ReLU}(x) = \max(0,x)$$
$$\text{Sigmoid}(x) = \sigma(x) = 1/(1+e^{-x})$$
$$\text{Tanh}(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} = 2\sigma(2x) - 1$$

Figure 3.2 Examples of commonly used activation functions.

3.3 ANN Training

Gradient descent, backpropagation, and automatic differentiation are interconnected concepts, but they serve different purposes in the context of training machine learning models, particularly in ANNs. Backpropagation (short for the backwards propagation of errors) is a supervised learning algorithm that forms part of the

training of ANNs, and is the method in which weights are adjusted. Automatic differentiation is the computational technique that is used to achieve this. In automatic differentiation, the input-output gradients at each node in the network are stored in the forward pass (feedforward). In the backward pass, the gradient — i.e., the partial derivative — of the loss function with respect to all input variables is computed. Gradient descent is the overarching algorithm that determines the adjustment of an ANN's weights.

3.3.1 Gradient Descent

Gradient descent is an algorithm that attempts to find the minima of a multivariate function by adjusting the input parameters of each iteration in proportion to the partial derivative of that function with respect to the input parameter of the previous iteration. For ANNs, the loss function is dependent on the weights, biases, and inputs. This algorithm is summarized as follows:

1. initialize weights and biases in the network;
2. feedforward training inputs to produce an output (prediction), computing the output of each node and the loss;
3. backpropagate the loss and compute the weight update, which requires the partial derivative of the loss with respect to each trainable parameter; and
4. repeat 2 – 4 until the loss is sufficiently low.

Stochastic gradient descent is the "basic" gradient descent algorithm, where all weights W are updated by adding a proportional (constant γ, called the learning rate) of the gradient of the loss function with respect to the weights of the prior iteration, $\nabla f_t(W_{t-1})$, as given by:

$$W_t = W_{t-1} - \gamma \nabla f_t(W_{t-1}). \tag{3.1}$$

Figure 3.3 provides a graphical depiction of the gradient descent algorithm and the importance of utilizing a suitable learning rate. A numerical example demonstrating the gradient descent algorithm and backpropagation is provided in appendix A. The ANN is ultimately a composition of several nested activation functions, multi-

plication with weights, and additions of biases. Using calculus, the partial derivative of the loss with respect to a particular weight can be determined using the chain rule: where the derivative depends on the multiplication of the other partial derivatives of weights appearing in later layers. Therefore, as an ANN becomes deeper, partial derivatives of each node are successively multiplied, which poses the risk of numerical instability. This problem is known as exploding and vanishing gradients. Regularization techniques such as weight decay, momentum, and proper initialization are used to adjust weights in such a fashion to prevent this [25].

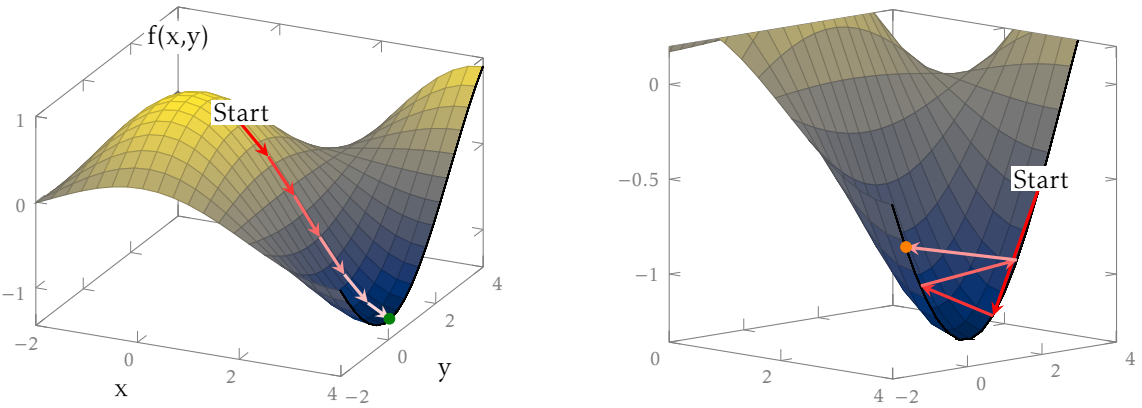


Figure 3.3 Graphical depiction of gradient descent, showing a convergent case (left) and divergent case (right). Each arrow represents an iteration in the algorithm. Each iteration, the ANN parameters are updated proportional to the partial derivative of the multi-variate loss function $f(x, y)$. Steeper parts of the surface are updated with larger step changes, shown by the longer lines. The figure to the right shows a case where the starting point and too large of learning rate result in a missed minima.

3.3.2 AdamW Optimizer

The "Adam" (Adaptive Moment Estimation) optimizer is a gradient descent algorithm that adapts the learning rate of each optimizable parameter by considering the moving averages of the "momentum" and "variance" of the past gradients. Momentum is applied by adding a small proportion of the previous gradient during the weight update, and expedites weight change if the gradient is in a consistent direction. The advantage of using momentum is that it can help avoid local minima by inceasing weight updates and reduce convergence time. RMSprop — a gradient descent variant — introduces a "variance" term that is adopted by Adam [26]. The variance term is applied to the denominator of the weight update term, and penal-

izes the update for parameters where the gradient's magnitude is greatly changing between iterations (i.e., a high a variance). The novelty of the Adam algorithm is that it incorporates moving averages of the momentum and variance, rather than other versions of gradient descent that use only the previous iteration's momentum/variance, or not at all. The AdamW variant of the algorithm differs from Adam slightly, in that it applies weight decay with the learning rate.

Table 3.1 shows the AdamW algorithm steps and mathematical formulation. In this figure, m_0, v_0 and $\hat{v}_{0_m}$ are initialized to 0, and ε represents a small number to prevent numerical instability. Time-steps are labelled as t, the learning rate is γ, the network weights are W, and $\nabla_W f(W)$ is the gradient of the loss function with respect to a particular weight. Step 2 depicts weight decay and is governed by constant λ (typically $0.01 - 0.1$). Weight decay is used to regularize the weight-update by adding a proportion of the sum-of-squares of weights to the loss function [27]. Mathematically, it is equivalent to decreasing the magnitude of the weight a small amount each time-step. β_1 and β_2 are the momentum and variance's moving-average coefficients, and are typically between $0.9 - 0.999$. Steps 3A and 3B apply the moving average to the momentums. The momentums are divided by $(1 - \beta^t)$, as shown in step 4, to increase value in the early stages of training (since moments are initialized at 0). Step 5 imparts another regularization technique by setting the second moment $\hat{v}$ to the maximum observed during training, which can prevent the weight update from accelerating too quickly. This variant is proposed in AMSGrad [28]. Step 6 shows the final weight update.

Table 3.1 AdamW algorithm.

1.	g_t	$\leftarrow$	$\nabla_W f_t(W_{t-1})$	compute gradients of loss f w.r.t. weights W
2.	g_t	$\leftarrow$	$g_t + \gamma\lambda W_{t-1}$	apply weight decay
3A.	m_t	$\leftarrow$	$\beta_1 m_{t_1} + (1 - \beta_1) g_t$	apply moving average to first
3B.	v_t	$\leftarrow$	$\beta_2 v_{t_1} + (1 - \beta_2) g_t^2$	and second moment
4A.	$\hat{m}_t$	$\leftarrow$	$m_t / (1 - \beta_1^t)$	regularize $\hat{m}_t$ and $\hat{v}_t$ during the
4B.	$\hat{v}_t$	$\leftarrow$	$v_t / (1 - \beta_2^t)$	first few iterations
5.	$\hat{v}_{t_m}$	$\leftarrow$	$\max(\hat{v}_{t_m}, \hat{v}_t)$	regularize $\hat{v}_t$ (AMSGrad)
6.	W_t	$\leftarrow$	$W_{t-1} - \gamma\hat{m}_t / (\sqrt{\hat{v}_{t_m}} + \varepsilon)$	update weights with modified learning rate

3.3.3 Loss Functions

The loss function measures the disparity between the network's predictions and the true target values. Common loss functions include mean squared error for regression, or cross-entropy for classification tasks [29, 30]. The cross-entropy loss function employed by PyTorch for multi-class classification was used for this thesis and is given by:

$$l = -\sum_{c=1}^{C} w_c y_c \log \text{Softmax}(x_c) \tag{3.2}$$

$$\text{where} \quad \text{Softmax}(x_c) = \frac{e^{x_c}}{e^{x_1} + \ldots + e^{x_C}}. \tag{3.3}$$

In equation 3.2, l is the loss, y is the target array, and x is the output array of the ANN, both of size C (number of classes). The target array is usually represented by a one-hot encoded array: a vector whose indices correspond to each possible class, and all elements are set to 0 except for the index that corresponds to the correct classification, that is set to 1. The softmax function clamps extreme values and normalizes classes such that the sum of the array is 1. The ideal loss is then zero (log 1) if the output of the ANN matches the target array. w_c is an optional weighting coefficient that can be applied to each class, which could be used to avoid bias in an unbalanced distribution of classes. In the practical training of ANN, batches of data are simultaneously fed forward and the gradient update is based on the sum of all losses of the batch [31].

4

Literature Review

This chapter provides a literature review of deep learning applied to DGA interpretation and transformer condition assessment. Considerable research has been conducted on the application of machine learning (ML) and DL to improve DGA interpretation, and numerous methodologies and architectures have been studied. DGA presents a suitable problem for DL applications as it is fundamentally a classification problem. In recent years, deep learning and artificial intelligence have shown great promise for improving DGA-based fault detection. However, some uncertainty in its suitability is naturally expected due to the "black-box"-like nature of DL architectures and its dependency on high-quality, abundant training data, and need for thorough validation.

4.1 Machine Learning Architectures

Several deep learning architectures have been applied to DGA interpretation, including support vector machine (SVM), k-nearest neighbours (kNN), ANN, CNN, and tree-based classifiers. "Shallow" machine learning architectures include SVM, kNN and decision trees [32, 33, 34]. Demirci et al. (2021) [35] provides a comparison between a these models with inputs features derived by taking the logarithm of gassing (in ppm), tested with 167 data points from the IEC TC-10 [36] database. The intention of using the logarithm of gassing is to normalize the data, with the assumption that the order-of-magnitude of gassing is arguably more indicative than the precise value, especially considering noise in DGA measurements. The highest accuracy achieved in this work was 91 % for the SVM using 34 test samples. Ghosh et al. (2021) [37] proposes numerous tree-based classifiers using 5 gas ratios as inputs, as well as combining them in a wisdom-of-the-crowd approach, and 91 % accuracy is reported. Benmahamed et al. (2017) [38] also applied SVM and kNN, but to augment the behaviour of Duval's Pentagon 1 (DP1). In this work,

the x-y centroid of DP1 gas was used as an input to SVM and kNN models. It was found that an accuracy of 80 % could be increased to 82 % with kNN, where the Euclidean distance, city block, and cosine distance metrics [39] were compared. A final reported accuracy of 88 % was achieved with the SVM.

ANNs are shown to accurately approximate complex relationships with sufficient data, and therefore, have gained popularity in recent years over shallow models. Farooque et al. (2015) [40] proposed a 15-neuron single hidden-layer ANN with five hydrocarbon gasses as input features, and compares the results to DP1. The accuracy of DP1 was 87 %, and the proposed ANN achieved 92 % accuracy for the same dataset. Training auto-encoders [41] tuned to each fault classification has shown to be useful when discriminating between fault and no-fault cases of transformers with and without OLTC [42]. These so-called auto-associative neural networks use inputs and outputs of the same dimension, but a hidden layer with a reduced size that acts to "compress" the data is added. For this work, each fault type used an autoencoder that operates in parallel with each other. A 100 % accuracy on the test data was achieved; however, the model was trained using synthetic data that was generated based on the test data. Concerns against synthetic data are discussed in section 4.3.

Barbosa et al. (2012) [43] uses ANN for a different purpose: rather than predicting a fault from gassing, predicting the gassing content from other oil properties is attempted. Oil acidity, breakdown voltage, power factor, water content, density, and interfacial tension are used as input features to train a model that predicts the gassing content of various hydrocarbons in transformer from which the oil was sampled. This work suggests that lower-costing chemical tests can be used as a proxy for DGA.

4.2 Insight on ANN Design and Optimization

Deep learning is a growing field, and there are numerous methods proposed for improving model performance. As expected, several of these techniques have appeared in literature concerning applications on DGA interpretation. Ou et al. (2019) [44] uses the Adam optimizer with a dynamic learning rate — i.e., a learning rate that

increases as model training progresses — and shows that dropout [45] serves as a useful regularization technique. In this work, gas concentrations in ppm were used, and different network dimensions were tested. "Fat and short" (hidden layers of sizes 50, 400, and 50) and a "deep" (5 hidden layers of 100) model were compared, and showing that the latter was more accurate. This work also compares networks that use the ReLU, Sigmoid, and the Softplus activation function, where ReLU produces convincingly the best accuracy of 85.6 %. Zakaria et al. (2012) [46] uses an ANN with three gas ratios as the input features, and proposes early stopping [47] during training as a method to avoid overfitting. Taha et al. (2021) [48] uses a CNN to combat noise in DGA measurements, with the intuition that the CNN learns filters that are capable of smoothening features. This work shows that CNN model's accuracy degrades less as injected noise increased as compared to the ANN model. Interestingly, this work also shows that conventional methods — although lower in accuracy when comparing the noise-free datasets — are relatively immune to noise, in that the accuracy of most conventional methods remain with 2 percent-points with noise injection up to 20 %.

Input data transformations and normalization are shown to be effective for model convergence and regularization [49]. Some numerical modifications to input data are explored; for example, using the arc-tangent function to normalizes gas ratios and avoid extreme values input to the model [50]. It is common practice to standardize input features according to:

$$\hat{x} = \frac{x - \mathrm{E}(x)}{\sqrt{\mathrm{Var}(x)}}, \tag{4.1}$$

where x is the input vector, $\mathrm{E}(x)$ is the expected — or mean value — of x, $\mathrm{Var}(x)$ is the variance of x, and $\hat{x}$ is the normalized input.

4.3 Datasets

The volume of publicly available datasets of DGA are still limited. Most research relies on proprietary datasets collected from power utilities [51, 52, 53] in combination with publicly available databases, including the IEC TC-10 database [36].

Some research efforts have used datasets specific to certain types of transformers such as wind turbine transformers [54]. Using bespoke datasets of specific transformer types highlights one of the advantages of using ML compared with traditional methods: custom models can be developed and conveniently deduce nuanced relationships between gassing and faults that are not accounted for with conventional methods.

Regardless, a small dataset size presents challenges for training complex models. To address this, some works have generated synthetic datasets using physical and statistical models of transformer aging, fault progression, or other oil properties [43]. Data augmentation techniques such as rotation, noise addition, and mean-shift clustering algorithms have also been used to expand real datasets [42, 55]. The use of synthetic data requires careful attention when training and validating a model, as it can be difficult to discriminate between an over-trained or the desired generalized model. The test data, if synthetically generated, will presumably be numerically close to the training data, and the novelty of using an "unseen" test set for performance evaluation is lost. It is generally recommended, therefore, that real data samples be used to test a model [56].

4.4 Ensemble Models Using Conventional Methods

Further enhancements of existing DGA methods using ML have also been considered. Subroto et al. (2017) [57] uses fuzzy logic and weighted sums of RRM, IECR, and GB/T 7252 interpretation methods, and an average accuracy of 90 % is achieved. Sutikno et al. (2021) [58] proposes a multi-method interpretation scheme with IECR, DRM, DT1, and DP1. The scheme involves using a weighted-sum table where stronger methods in predicting a particular faults have proportionally higher voting power on an prediction.

4.5 Gaps and Challenges

Generally, the greatest limitation in applying ML to DGA interpretation is the lack of training data, however, works using larger datasets have shown accuracies of 86 %, as shown in [59] with a 3043 point dataset. By design, transformer failure events are rare and forensic investigation results may be costly, private, or contain proprietary information. Imbalanced datasets, including lack of non-fault cases, can skew a model's prediction. Some works combined fault classifications, for example T1 with T2 [38, 40] or T2 with T3 [42] which is expected to improve model performance. Section 4.4 discussed works that use the trusted conventional methods, however an ANN approach to using these conventional methods as inputs is lacking. The following chapter endeavors to propose, optimize, and assess an ANN incorporating these features along with gassing data and ratios.

5 DGA Interpretation with ANN and Ensemble Classification

This thesis proposes a methodology to leverage the output of select DGA interpretation techniques as input features to a neural network, in addition to the commonly-used gassing quantities and ratios of gasses. The central presumption is that the existing methodologies encode gas-fault relationships that might not be easily learned by the existing DL architectures. The predictions of existing industry methods, however, often provide a different subset of classifications, may yield conflicting or no-diagnostic results, and an ANN can presumably address these problems. This chapter discusses the dataset used for this work, describes the ANN architecture, optimization, and validation processes, and provides an evaluation of the results.

5.1 Dataset

The development of a machine learning model is highly sensitive to its available data, and is described in this section. A dataset consisting of 543 samples was used to train and validate the model. The dataset contained 485 points with fault conditions including: partial discharge (PD), low-energy (spark) discharge (D1), high-energy (arcing) discharge (D2), low-temperature ($T < 300\,°C$) overheating (T1), mid-temperature ($300\,°C < T < 700\,°C$) overheating (T2), and high-temperature ($T > 700\,°C$) overheating (T3). An additional 58 no-fault cases (Ok) were included in the dataset. The data was obtained from a combination of the IEC TC-10 database and data from in-service transformers individually verified and inspected[1]. The distribution of classes in the dataset is shown in table 5.1. Limited data containing CO_2 and CO was obtained, and therefore methods and classifications dependent

1. Data provided and assessed by Camlin Energy.

on these gasses (carbonized paper, overheated insulation, and stray gassing) were omitted from this work. The chemicals used were limited to the five hydrocarbons typically used in DGA interpretation: H_2, CH_4, C_2H_4, C_2H_6, and C_2H_2. Cases marked as "stray gassing" were labelled as "Ok" as they are typically not cause for concern [10], and only one sample of stray gassing was obtained in the dataset which would pose a challenge for training and validating on this classification. A variety of transformers mostly without OLTC — but a few with OLTC — and some instrument transformers are included.

Table 5.1 Summary of dataset and distribution of fault classifications used in this thesis. The dataset is relatively well-balanced and considered large enough such that synthetic data was not deemed as required.

	Ok	PD	D1	D2	T1	T2	T3	Total
Counts:	58	52	88	125	55	60	105	543
Proportion:	11	10	16	23	10	11	19	100%

5.2 Model Validation

A generally preferred evaluation method of a machine learning models is to use a train-validation-test split [60]. In this scheme, a training set is used to adjust a model's weights during training and the validation set is used determine the optimal network performance — i.e., the performance of the network is checked at each training iteration (epoch) with the validation set. After training, the model is typically reverted back to the one that gave the optimal performance with the validation set, and then tested with an unseen test set to evaluate the model's performance. For example, 75 % of the data could be used train the model, 12.5 % could be used to validate the model during training and revert back to the optimal parameters, and a final, unseen 12.5 % is then used to test the model. K-fold cross-validation ensures that all data samples are used for testing by splitting the entire dataset into K different folds (e.g., K=4), where K-1 (e.g., 3/4) is used for training, and the rest for validation (e.g., 1/8) and testing (e.g., 1/8). The model is trained over K-independent instances, or "folds", where the training, validation, and test

sets are rotated, and the average performance across all K runs is reported as the final performance metric.

A training-validation split of 75/25 % was used for the model optimization described in section 5.4, and testing was not performed during the optimization. For the final model analysis described in section 5.5, five instances of repeated random subsampling validation was combined with K-fold validation using 4 folds with a train-validation-test split. That is, the dataset was shuffled 5 times. For each shuffle, it was then split into 4 folds with 407 samples for training, 68 for validation, and 68 for testing, rotated 4 times, for a total of 20 independent instances of model training and testing. Pseudocode outlining the training procedure is provided in appendix B. The dataset was shuffled and split in such a way that the distribution of fault classes was approximately consistent between the training, validation, and testing sets, as shown in table 5.1.

As a final performance metric, the F1-score was mostly used as it is generally considered a more indicative of performance in contrast with accuracy, which is typically reported on [61]. The F1-score is the harmonic-mean of precision (P) and recall (R), represented each as a percentage by:

$$\mathrm{F1} = \frac{2 \cdot \mathrm{P} \cdot \mathrm{R}}{\mathrm{P} + \mathrm{R}}, \tag{5.1}$$

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \cdot 100\,\%, \tag{5.2}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \cdot 100\,\%, \tag{5.3}$$

where TP is true positives, FP is false positives, and FN is false negatives. TP+FN is the total counts of a particular class, and TP+FP is the counts of a particular class that the model predicted. Recall can be thought of as class-specific accuracy, and precision can be thought of as the accuracy of positive predictions. The average F1-score, precision, and recall reported in the remainder of this thesis refer to the "macro" average, where the average between classifications is taken (treated as equally important, as opposed to weighted by class size), unless otherwise specified. The overall accuracy is equivalent to the "micro"-averaged recall, which is the total numer of correct predictions over the size of the dataset.

The cross entropy loss, although correlated with performance, is not necessarily one-to-one with the F1-score. Therefore, the best overall F1-score of the validation test set was tracked during training and used to determine the optimal model during training.

5.3 Architecture

The overall architecture of the proposed model is a multi-layer feed-forward ANN that incorporates three subnetworks for each "feature-set", including:

- a "PPM" subnetwork that interprets relative percentage of five hydrocarbons over the sum;
- a "Ratios" subnetwork that interprets five gas ratios; and
- an "Ensemble" subnetwork that interprets conventional DGA methods' predictions as inputs features.

An ANN was chosen for this work because of its proven ability to solve classification problems, noting that other types of neural networks such as CNNs are typically used for data with spatial features (e.g., image recognition), and RNNs are typically used for data with temporal features (e.g., time-series prediction or language models).

Figure 5.1 provides a graphical depiction of the proposed ANN. The raw gassing data of five hydrocarbons in ppm is transformed by the three subnetworks: PPM, Ratios, and Ensemble, then standardized using equation 4.1. The dimensions of each subnetworks are controlled by variables "d" and "h", where "d" is the number of hidden layer vectors in each subnetwork, and the "h" parameter is used to scale the size of the hidden layer vectors. Any combination of these networks can be concatenated at the "Con-Cat" by enabling the $\widehat{P}$, $\widehat{R}$, and/or $\widehat{E}$ "gates" to allow for testing different model variations. The combined network uses the concatenation of the enabled subnetworks as an input features, and passes them through additional hidden layers with dimensions controlled by "d_c" and "h_c". The activation functions for the combined part is "σ_c", and "σ" for the prior subnetworks. The prediction of

the model is then taken as the index of the maximum value of the output layer of the combined network.

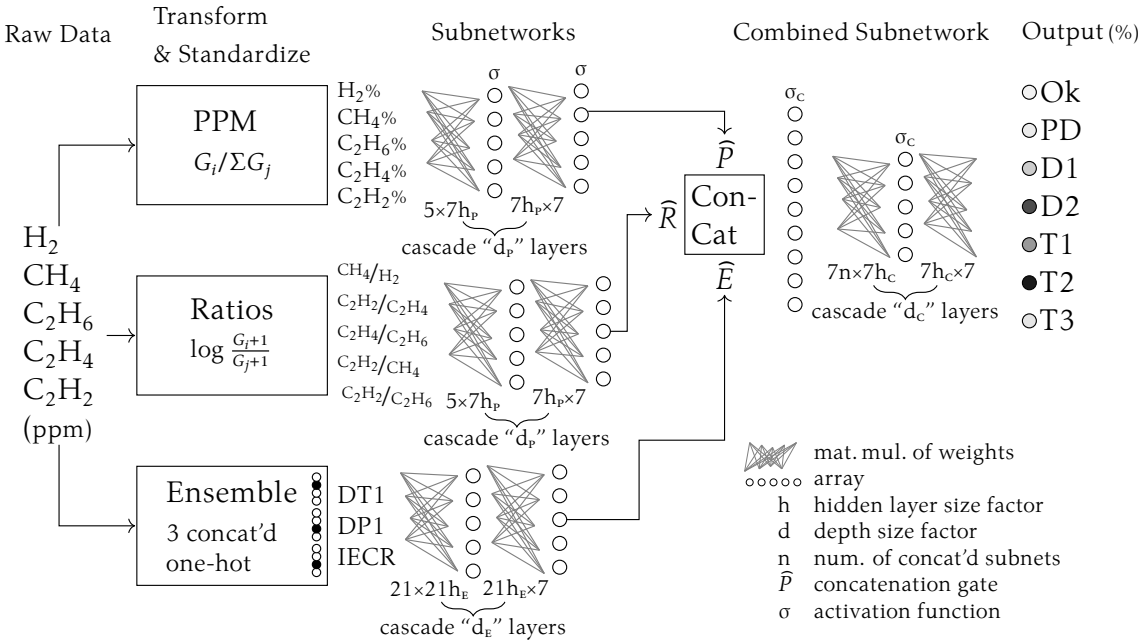


Figure 5.1 Schematic of proposed ANN.

5.3.1 PPM Subnetwork

The PPM subnetwork normalizes the input gases in terms of their relative concentrations with the sum of all gases as per:

$$G_i = \frac{G_{i_{\text{ppm}}}}{\sum_{j=1}^{5} G_{j_{\text{ppm}}}}, \qquad 5{\cdot}4$$

where G_i is the normalized quantity $(0-1)$ and $G_{j_{\text{ppm}}}$ is a gas quantity in ppm.

5.3.2 Ratios Subnetwork

The ratios of gasses chosen for the model are the combination of the ratios used by IECR and DRM [11]. The ratios are calculated by:

$$\log \frac{G_{i_{\text{ppm}}} + 1}{G_{j_{\text{ppm}}} + 1}, \tag{5.5}$$

where $G_{i_{\text{ppm}}}$ and $G_{j_{\text{ppm}}}$ are the gassing quantities in ppm of two gasses. Rather than taking the quotient of two gasses, equation 5.5 is used to avoid extreme values by adding 1 to each gas, and supress differences in orders of magnitude at the ANN input (e.g., if G_j is comparatively small). The logarithm was chosen as it maps the ratios in a one-to-one fashion as opposed to asymptotic functions like tanh or sigmoid (as shown in figure 3.2) or the arc-tangent ($\tan^{-1}$) function [50].

5.3.3 Ensemble Subnetwork

DT1, DP1, and IECR were used as input features for the Ensemble subnetwork of the ANN. Doernenburg Ratios was initially considered, but due to a possibility of no-diagnosis along with IECR, was not chosen for the Ensemble network as to avoid too many non-diagnoses. IECR and Roger's Ratio Method use the same ratios but different levels, and IECR was shown to produce slightly more accurate assessments and was therefore chosen.

Conventional DGA methods are typically used only when gassing is beyond a certain threshold. Following this, input gassing is determined as a fault or no-fault case first by comparison with typical gassing levels of non-faulted transformers. If all key-gase philosophy are below the particular thresholds outlined in table 5.3, the output result of all methods is Ok (non-fault), otherwise, the method's assessment is used. The outputs of each method are output as a one-hot encoded array, and all are concatenated as an input array to the Ensemble subnetwork and then standardized with equation 4.1 — with three methods used, an array of size 21 values with three elements set to 1, and the remaining set to 0 is used as the ANNs input. For the no-diagnosis cases which can be given by IECR, all elements of the one-hot array for that method are set to zero. Only three instances of combined thermal and discharge

fault classifications were obtained in the original dataset, and therefore omitted as classification for the proposed ANN. Only DT1 is capable of providing such a classification out of the selected methods, and its one-hot array output was set to 0 in these instances. DP1's diagnoses of stray gassing were set to "Ok". Table 5.2 compares the performance of the conventional DGA interpretation methods that are used by the ensemble subnetwork.

Table 5.2 Performance comparison of conventional DGA interpretation methods. The methods are shown to compliment eachother: DT1 tends to perform the best overall, but DP1 produces the best T1 classifications, which are the more difficult cases to capture. IECR did not provide a diagnostic for several instances (result did not match table 2.1 criteria) in the dataset, resulting in a low F1-score. However, with those cases removed — i.e., when IECR does provide a diagnosis — it offers a performance advantage over DT1 and DP1, shown by IECR*. IECR was shown to offers improvements in numerous classifications, and was therefore included as a feature set for the ensemble subnetwork.

	F1-Scores, %							F1-Score	Recall	Precision
Method	Ok	PD	D1	D2	T1	T2	T3	Avg., %	Avg., %	Avg., %
DT1	68	59	76	85	38	47	84	65	66	69
DP1	69	59	62	82	41	42	79	62	63	69
IECR	69	32	38	52	39	38	77	43	37	65
IECR*	73	69	78	91	44	49	83	70	70	74

Table 5.3 Ok-classification threshold for existing DGA methods. If all gasses are below their threshold values, the case is considered non-fault by industry methods. The quantities are based off of IEC 60599 [11] suggested 90-percentile gassing level increases over a 1-year period.

Gas:	H_2	CH_4	C_2H_6	C_2H_4	C_2H_2
Threshold, ppm:	100	120	65	60	1

5.3.4 ANN Models

For this thesis, specific configurations of the architecture incorporating a different combination of subnetworks are referred to as *models*. Three models are optimized, tested and compared in the remainder of this chapter:

- a "PPM+Ratios Model that uses the PPM and Ratios subnetworks, but not the ensemble;
- a "Ensemble Model" that uses the ensemble feature-set and subnetwork only; and
- a "Full Model that uses all feature-sets: ppm, ratios, and ensemble.

5.4 Model Optimization

A plethora of design and training options exist for any given deep learning model, including the proposed ANNs. For example, the number and size of hidden layers, activation functions selection, or the inclusion of regularization techniques such as drop out or batch training can greatly affect a model's training and performance outcome. Furthermore, loss-function parameters such as learning rate, weight-decay, and label smoothening are under consideration. Insight towards how to design a model without exhaustive experience in deep learning is elusive. This section details the optimization process of each model and training specifications.

To assist with designing the model and selecting suitable training parameters, a Bayesian optimization using Gaussian processes algorithm [62] was used. Bayesian optimization is a technique used for optimizing black-box functions that are computationally expensive to evaluate — in this case, training an ANN. This technique builds a surrogate Gaussian process model of the objective function (best F1-score for the validation set during training), which captures the uncertainty in the function's behavior as a function of the parameters. This model is updated iteratively as evaluations are made to identify the most promising points in the search space. Bayesian optimization was chosen to reduce computation time [63], noting the relatively large number of parameters. Other optimization approaches like grid search or random search [64] were considered but not selected due to the large parameter space.

The optimizer was configured with parameters described in table 5.4, and was conducted for Full Model, PPM+Ratios Model, and Ensemble Model described in section 5.3. The optimization was conducted for a training-validation split of 75/25% on the same data distribution for each model, repeated over 100 simulations

with 20 unique starting parameter sets. The achieved optimal parameters are shown in table 5.5 and discussed in the following sections. The height scale of each subnetwork is allowed to be a fraction as to test "autoencoder" variations of each subnetwork, where the number of hidden layers is smaller than the input size. The array size of the hidden layers are rounded to the nearest integer after multiplying the height scale. Regularization tools such as batch-size, dropout, and label smoothening were tested. The training hyperparameters optimized were learning rate and weight decay. The β parameters of AdamW were decidedly not optimized and set to their default values of 0.9 and 0.999. Because β is multiplied by itself during training, a value closer to 1 reduces drastic step-size change between iterations (see table 3.1).

Table 5.4 Model and training parameter optimization space.

	Description	Range
Model Parameters		
d_P	Depth of PPM and Ratios subnetworks	1 – 4
d_E	Depth of Ensemble subnetwork	1 – 4
d_C	Depth of Combined subnetwork	1 – 4
h_P	Height scale of PPM and Ratios subnetworks	0.25 – 6
h_E	Height scale of Ensemble subnetwork	0.25 – 6
h_C	Height scale of Combined subnetwork	0.25 – 6
σ	Activation function of subnetworks	ReLU, Tanh, Sigmoid
σ_C	Activation function of Combined subnetwork	ReLU, Tanh, Sigmoid
Training Parameters		
BS	Mini-batch size	0, 12, 24, 102, 204
DO	Dropout, %	0, 0.1, 0.2
γ	Learning rate	0.0005 – 0.02
RL	Ramping learning rate	yes, no
λ	Weight decay	0.0001 – 0.01
LS	Label smoothening	0, 0.1, 0.2

5.4.1 Network dimensions

The optimal network dimensions varied considerably between the PPM+Ratios Model and Ensemble Model. The PPM+Ratios Model followed a short (smaller d_P

Table 5.5 Optimal network and training parameters for PPM+Ratios Model, Ensemble Model, and Full Model. The Full Model resembles a combination of parameters of the PPM+Ratios Model and Ensemble Model.

Model Parameters			
	PPM+Ratios	Ensemble	Full
d_P	1	n/a	1
d_E	n/a	2	2
d_C	2	3	3
h_P	4	n/a	3
h_E	n/a	0.6	0.6
h_C	2	4	4
σ	Tanh	ReLU	ReLU
σ_C	ReLU	Sigmoid	Sigmoid
Training Parameters			
	PPM+Ratios	Ensemble	Combined
BS	24	102	24
DO	0.1	0.1	0.1
γ	0.01	0.004	0.008
RL	yes	yes	yes
λ	0.001	0.008	0.004
LS	0.1	0.1	0.1

and d_C) and tall (higher h_P scheme), and the Ensemble Model was comparatively narrower in the ensemble subnetwork (h_E) but deeper overall (d_E and d_C). The height and depth of the PPM and ratios subnetworks were chosen to be the same (d_P and h_P) under the assumption that the input feature size is the same and should be comparable. As the input features for the ensemble subnetwork are large and sparse (one-hot encoded arrays), it is reasonable to see that some form of data compression ($h_E < 1$) was used. The Full Model, as one could expect, adopts the dimensions from each subnetwork. The total number of trainable parameters of each model is outlined in table 5.6.

Table 5.6 Number of trainable parameters for each optimized model.

Model:	PPM+Ratios	Ensemble	Full
Number of parameters:	3073	3611	5466

5.4.2 Network Activation Functions

The activation functions of each subnetwork, excluding the combined subnetwork, were chosen to match, given that they are concatenated before being fed to the combined network. Out of interest, it was decided to test changing the activation function of the input from the concatenated hidden layer to the combination network, σ_C. Using multiple activation functions was hypothesized to have a best of both-worlds outcome [65]. Interestingly, there were some differences between the models. The PPM+Ratios Model first uses Tanh in the subnetworks then ReLU in the combined network, whereas the Ensemble Model and Full Model were optimal with ReLU in the subnetworks then Sigmoid in the combined network.

5.4.3 Batch size

Figure 5.2 shows a comparison of the cross-entropy loss and F1-score during training the Full Model network with a small batch (BS=24) and a large batch (BS=204). When larger batches or no batches are used, the loss of the training and validation set smoothly decrease after several epochs, but after some iterations (approximately 1000 in figure 5.2), the loss of the validation starts to increase. This indicates that the network tends to over-fit the training-set, i.e., remember the training data without generalizing it. The deviation between the training and validation set is less pronounced with the smaller batch size.

With smaller batches, the loss of the training set fluctuates because the parameters are updated with a different sub-set of training data each iteration — with no batches, the same data-set updates the weights each iteration and therefore a more smooth change is expected. The noisy fluctuation in loss of the training set coincides with fluctuating performance in the validation set, but a lower likelihood of overfitting and greater likelihood of reaching a more performant configuration. Intuition on the maximum allowable training iterations was found from analyzing these graphs. The maximum iterations was set to 2400, but terminated early if the best validation F1-score previously encountered did not improve after 600 additional epochs. The Full Model network and PPM+Ratios Model found optimal batch sizes of 24 (17 batches per epoch) where as the Ensemble Model was 102 (4 batches per epoch).

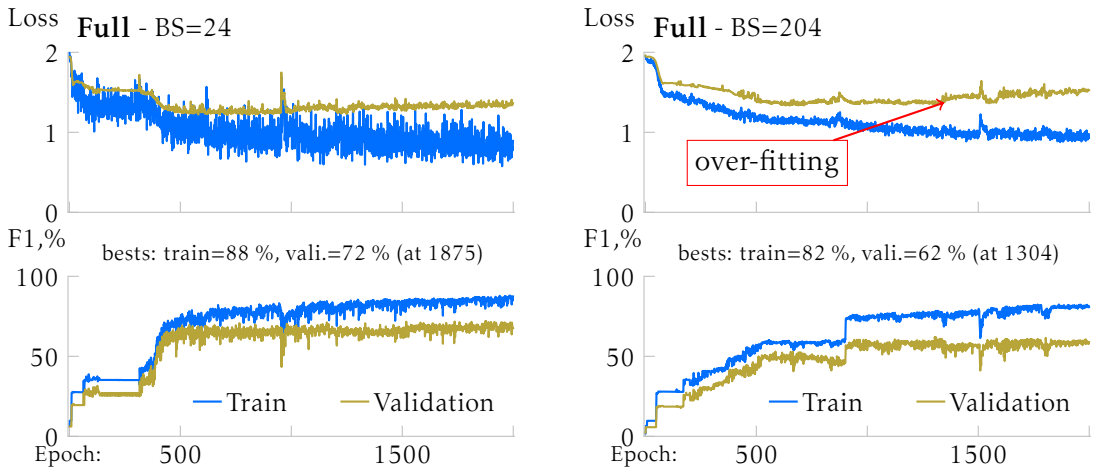


Figure 5.2 Comparison of loss (top graphs) and F1-score (bottom graphs) during training with a batch size of 24 (graphs on the left), and 204 (graphs on the right). This example shows the Full Model with optimal parameters with the batch size varied. The loss and F1-score of the training and validation were both tracked to monitor training progression.

5.4.4 Dropout

Dropout is thought to "strengthen" an ANN by dropping out (setting to 0) a portion of the weights during training only [66]. The dropout value refers to the proportion that are dropped out each training iteration (i.e., 0.1 means 10 % of the weights are ranomly set to zero). All models arrived at the same optimization point with a dropout of 0.1.

5.4.5 Learning Rate, Ramping, and Weight Decay

A ramping learning rate inspired by [44] was tested, where the learning rate was configured to double every 200 epochs, up to a maximum value of 0.01. It was found through trial and error that values above 0.05 severely degraded the model. Because the philosophy of early-stopping is adopted, the thought behind a ramping learning rate is that the best model is continuously tracked and chosen, whether the model is over-fit or not. Ramping the learning rate could hypothetically help the model parameters become "un-stuck" out of a local minima. Figure 5.2 shows some notable steps present as the training progressed: this is attributable to the

ramping learning rate. All of the models found the best performance using this technique. The initial learning rate, and weight decays varied between models, but were relatively comparable.

5.4.6 Label Smoothening

Label smoothening regularizes the training by introducing fuzziness — or smoothening — to the target labels, and is shown to reduce over-fitting [67]. All of the models arrived at the same optimization point with a label smoothening value of 0.1.

5.5 Analysis

The model and training parameters shown in table 5.5 were used for the PPM+Ratios Model, Ensemble Model, and Full Model. The validation process described in section 5.2 was followed to analyze each models performance, and the results are described in this section.

Confusion Matrices are used to graphically depict a multi-class classifiers performance, and are shown in figures 5.3 and 5.4. For all of the confusion matrices described in this section, the aggregate F1-score, accuracy, and precision is reported for each model by adding the results of all 20 model tests. The average F1-score, accuracy, and precision is shown in each of the plots titles, and the class-wise precisions, recalls, and F1-scores are reported along the axes. Numbers within each matrix were normalized and expressed as a percentage of the number of classifications of that particular fault. in the dataset (i.e., the sum of the rows, the actual classification, will equate to 100 %). A "perfect" classifier would yield a confusion matrix with 100 % along the bottom-left to top-right diagonal, and 0 % elsewhere.

5.5.1 Ensemble Model

A comparison of the Ensemble Model's results with conventional DGA interpretation methods is shown in figure 5.3. The overall performance of the Ensemble Model is improved over each method that it was trained on: DP1, DT1, and IECR. Because the Ensemble Model is shown to yield performance improvements compared with each

method individually — yet solely relied on information provided by these methods during training — it is inferred that an intricate interpretation of the combination of these methods is used by the model to better predict faults. This emphasizes that each method performs better at classifying particular faults, and the strengths of each can be leveraged for a better performance overall. Am improvement of 5 percentage points in F1-score, accuracy, and precision compared with DT1 — the best performing conventional method — is observed.

Notable classifications that were improved are T1 and PD, highlighted in figure 5.3. As hypothesized, the Ensemble Model appears to inherit the flaws and strengths of the methods it was trained with as features. The general distribution is comparable among all cases, but PD and T1 classes are notably improved and highlighted in the figure. IECR — when it does not give a no-diagnosis[1] (see table 5.2) — shows faults classification improvements over the remaining methods, and this appears to be adopted by the Ensemble Model where PD, T1, and T2 faults are better-detected. All methods tend to mis-diagnose several cases as "Ok" (high recall, but low precision), indicating that the no-fault gassing thresholds in table 5.3 could be further optimized.

5.5.2 Comparison of All Models

On average, each model outperformed all industry methods operating individually. A comparison of the F1-score for each classification, and average of F1-score, precision, and recall between the models and DT1 is provided in table 5.7. The recalls of each fault classification and "micro" recall, which is colloquially referred to as overall accuracy (i.e., total number of correct guesses in relation to the size of the data set) is also reported. Overall improvements relative to DT1 are shown in table 5.8, and confusion matrices for results of each model and DT1 are provided and discussed in figure 5.4.

The PPM+Ratios Model shows an overall comparable performance to the Ensemble Model with an average F1-score of 71 % but compromises performance in some

1. the IECR confusion matrix includes the ND category, and a single artificial data point was added to facilitate plotting.

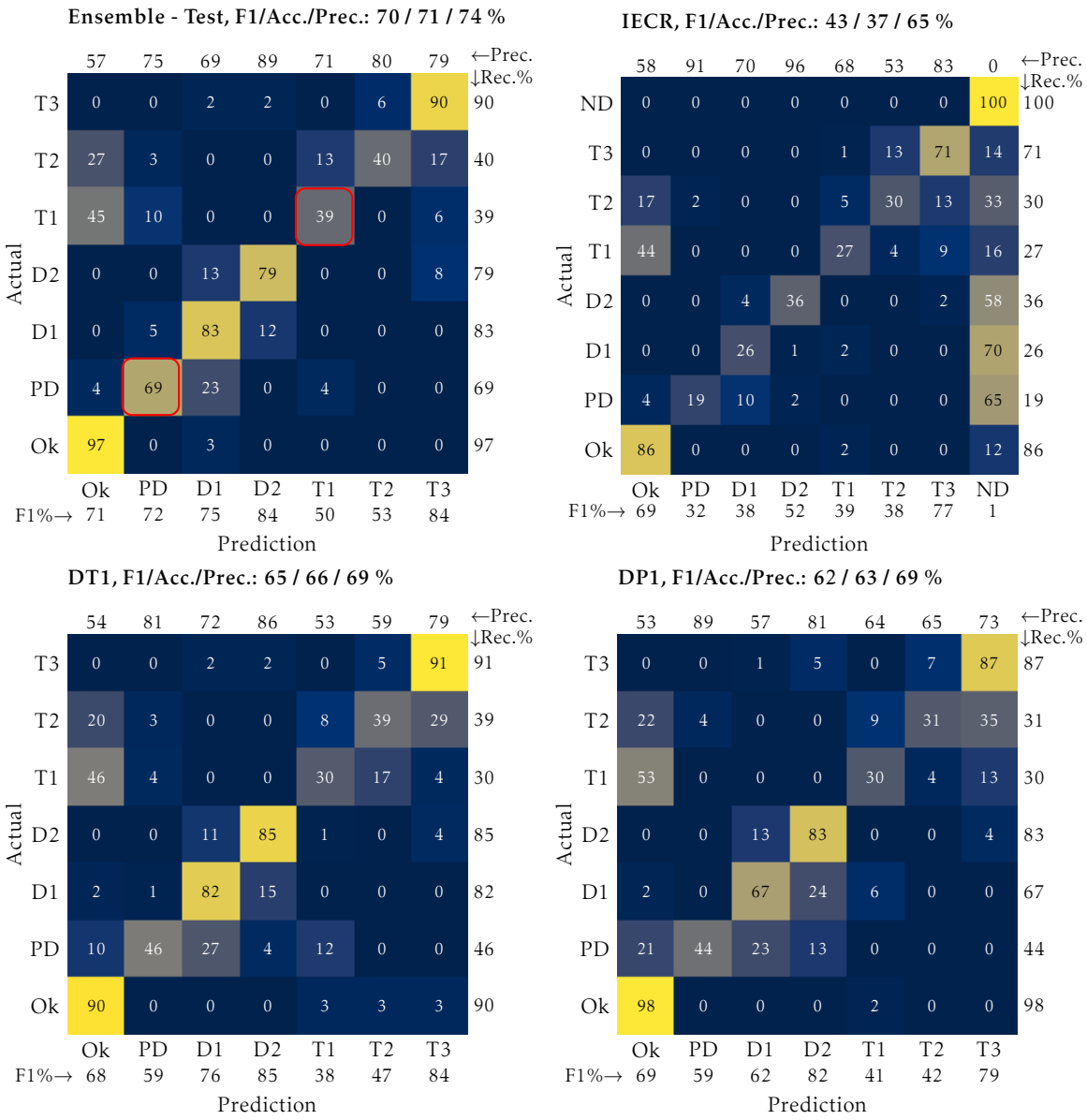


Figure 5.3 Confusion matrices for the Ensemble Model compared with conventional methods.

faults in favor of others. Combining all feature sets to produce the Full Model, however shows the greatest performance in terms of all metrics used: F1-score, recall, precision, and overall accuracy; and producing nearly the best performance in each fault classification. These results imply a lack of coverage with existing methods collectively, and that the PPM and Ratios subnetworks of the models were capable of learning novel DGA interpretation patterns from the gassing in ppm and the ratios of select gasses. In the Full Model, the combined network effectively selects hidden features from the output of each the PPM, Ratios, and Ensemble subnetworks for a more reliable assessment than using either subnetwork alone with an average F1-score of 73 %. The overall accuracy (micro-recall) of each model and DT1 is slightly higher than the corresponding macro-recall. This difference is expected due to the dataset's imbalance and each model producing stronger results for classifications which had more training samples.

Each model shows a modest performance improvement over the conventional interpretation methods, and demonstrates that they could be considered a viable option for DGA interpretation. The overall accuracy of the Full Model improves upon DT1 by 5 percent-points. Although this improvement could be considered incremental, given that a limited dataset of 543 points was used and seven unique classifications were required, these results are considered encouraging. Most prominently, a notable improvement in T1 classification is observed, where the Full Model and PPM+Ratios Model improve the F1-score compared with conventional interpretation methods by over 15 percent-points.

5.5.3 Spread of Performance Between Tests

A six-point statistical summary — i.e., minimum, lower-quartile, mean, median, upper-quartile, and maximum — of the test set performance among each 20 model tests is shown in table 5.9. A notably high spread of performance was produced by the Full Model (interquartile range (IQR)=6.0 %) and PPM+Ratios Model (IQR=12.3 %), which shows a potentially high-sensitivity to the training and testing data, and indicates that a larger set should be used. The Ensemble Model, albeit with a lower performance overall, had a greater consistency in its performance (IQR=3.8 %), highlighting the benefit of including it in an ANN designed for DGA

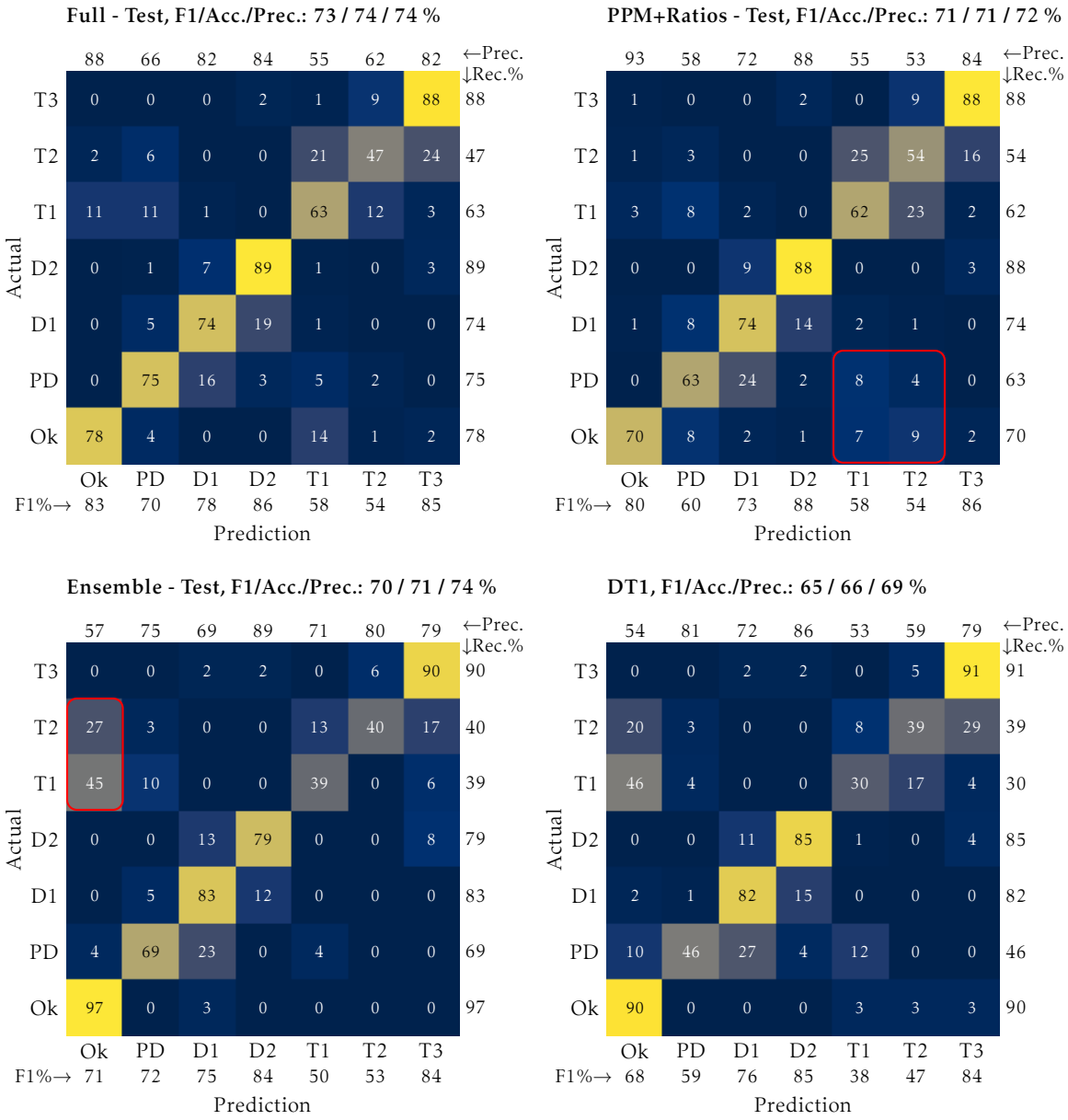


Figure 5.4 Confusion matrices of the proposed models. Within each model, clusters of confusion around adjacent faults appear: T1, T2 and T3 faults are mislabeled with each other; and PD, D1, and D2 are mislabeled. As highlighted in this figure, the Ensemble Model inherits the industry methods' problem of mislabeling T1 and T2 cases as Ok. The PPM+Ratios Model improves upon these missed T1 and T2 cases, but tends over-prescribes them (loss of precision). These problematic areas of confusion are rectified by the Full Model.

Table 5.7 Comparison of F1-score between classifications on the test set. The "macro" averages are reported in the upper section of the table as to treat each fault classification of equal importance and reject the imbalanced distribution of classes shown table 5.1. The overall accuracy is presented in the lower table and is equivalent to computing the recall with "micro"-averaging.

	F1-Scores, %							F1-Score	Recall	Precision
Method	Ok	PD	D1	D2	T1	T2	T3	Avg., %	Avg., %	Avg., %
Full	83	70	78	86	58	54	85	73	74	74
PPM+Ratios	80	60	73	88	58	54	86	71	71	72
Ensemble	71	72	75	84	50	53	84	70	71	74
DT1	68	59	76	85	38	47	84	65	66	69
	Recalls, %							Overall Accuracy, %		
Full	78	75	74	89	63	47	88	77		
PPM+Ratios	70	63	74	88	62	54	88	75		
Ensemble	97	69	83	79	39	40	90	74		
DT1	90	46	82	85	30	39	91	72		

Table 5.8 Relative improvements over DT1. All models improved in each performance metric compared with the best performing conventional DGA interpretation method.

	Improvement, %-points			Overall
Model	F1-score	Recall	Precision	Accuracy, %
Full	8	8	5	5
PPM+Ratios	6	5	3	3
Ensemble	5	5	5	2

interpretation, as it is more resilient to the differences in data that it is provided with. The Full Model produced the best performing minimum, mean, and maximum F1-scores, demonstrating reasonable performance even under an unfavorable dataset split.

5.5.4 Training, Validation, and Test Set Performances

A model evaluated with its training set is expected to show better performance than evaluating it with the validation or test set, and evaluating a model with the

Table 5.9 Performance sensitivity among tests.

	F1-scores, %					
Model	Min.	LQ	Mean	Med.	UQ	Max.
Full	66	69.2	73.2	73.4	75.2	80.0
PPM+Ratios	59	62.5	70.5	72.5	74.8	78.0
Ensemble	65	67.2	69.5	69.0	71.0	72.0

LQ=lower quartile, UQ=upper quartile.

validation set is expected to give slightly higher performance than the test set. Closeness in performance between all sets, however, indicates a well-generalized model. Table 5.10 provides a comparison of the models' performance when evaluated with the training, validation, and testing sets. The PPM+Ratios Model resulted in the greatest difference in performance between the training, validation and test sets, where the Ensemble Model's performance was closer among evaluation data, indicating a well-generalized model. The Full Model resembles a combination of the two. Overall, this shows reasonable performance across each of the models, but indicates that the dataset was likely too small for the PPM+Ratios Model and Full Model to produce strong generalizations between the gassing and faults.

Table 5.10 Performance comparison between models evaluated with the training, validation, and test datasets.

	Average F1-Score, %		
Model	Training	Validation	Testing
Full	88.1	79.6	73.2
PPM+Ratios	88.7	79.6	70.5
Ensemble	71.8	70.0	69.5

5.5.5 Summary

Overall, each model shows potential to be a useful tool for DGA interpretation and performance of each model improved over the best-performing conventional method, DT1. The Full Model produced an average F1-score of 73 %, improving upon DT1's F1-score of 65 %. The Full Model is shown to leverage strengths of both the Ensemble Model and PPM+Ratios Model models by using all feature

sets. The Ensemble Model, although lowest in performance among the proposed models, produced a relatively consistent performance among tests and datasets, highlighting that it is a well-generalized model and suitable with a smaller dataset. The PPM+Ratios Model was demonstrated to improve T1 classification, an area where conventional methods are shown to be weaker in. The highest performing conventional method in terms of T1 classification was DP1 with an F1-score of 41 %. Each network was shown to improve this area, especially when using utilizing the PPM and Ratios subnetworks: where the Full Model and PPM+Ratios Model improved the F1-score to 58 % (+17 %).

6

Conclusions and Future Work

This thesis provides a review of DGA interpretation, ANNs, and contemporary research that utilize these tools together. Based on this review and identified gaps in literature, ANN models are proposed using feature inputs of conventional DGA interpretation methods, relative gassing proportions, and select ratios of gasses, under the pretense that the strengths of the conventional methods can be combined with an ANNs ability to learn novel patterns. Models using combinations of these feature sets were optimized and evaluated.

This thesis demonstrates that the utilization of conventional DGA interpretation methods' assessments as input features to ANNs in combination with gassing and gas ratios can improve DGA classification. A proposed Ensemble Model trained on three conventional DGA methods outperformed each method, and compared with other proposed models, was resilient to variations in the training and testing data it was provided with. A proposed PPM+Ratios Model trained on only gassing and gas ratios is shown to significantly improve T1 classification. The T1 classification improvement is found in the proposed Full Model, which incorporates all feature sets. This research reinforces that ANNs should be considered a viable tool for DGA interpretation, especially when combined with existing industry methods.

6.1 Future Work

A few avenues for future exploration related to this thesis are:

- expanding the dataset and evaluating test performance among particular types and ratings of transformers (e.g., electric arc furnace transformers);
- investigate the incorporation of multi-fault classifications (e.g., thermal and discharge);
- conduct optimization on the "Ok" thresholds used by the conventional methods;

- perform additional analysis of the models' sensitivity to dimensions and training parameters;
- investigate pre-training each subnetwork of the models before combining to improve model training; and
- attempt synthetic data generation.

Anomaly Detection with Time-Series Data

An interesting and useful topic in DGA is anomaly detection. A deep learning model based on long short-term memory, for example, could be trained on time-series data for a given transformer. The time-series could contain daily gassing data and a statistical summary of transformer loading and temperature. A DL model could then be trained on this historical data in an attempt to predict what the next-day gassing evolution should look like based on the prior correlation of gas generation with loading and temperatures. If real gassing data obtained the next day occurs outside of a predicted band by the model, that event could be considered anomalous, and potentially be used to identify an incipient fault.

References

1. M. Heathcote. *J and P Transformer Book*. Elsevier Science & Technology Books, 2011. ISBN: 9780080551784.
2. *IEEE C57.104-2019 – IEEE Guide for the Interpretation of Gases Generated in Mineral Oil-Immersed Transformers*. DOI: 10.1109/ieeestd.2019.8890040.
3. J. Schmidhuber. *Annotated History of Modern AI and Deep Learning*. 2022. DOI: 10.48550/ARXIV.2212.11279.
4. K. Ewasiuk, N. Jacob, and B. Kordi. "Enhancing Transformer Oil Dissolved Gas Analysis Methods byUtilizing Artificial Neural Networks with Ensemble Classification (654)". *CIGRE Canada 2023 Conference in Vancouver, Canada* (2023).
5. J. V. Costa, D. F. F. d. Silva, and P. J. C. Branco. "Large-Power Transformers: Time Now for Addressing Their Monitoring and Failure Investigation Techniques". *Energies* 15.13 (June 2022), p. 4697. ISSN: 1996-1073. DOI: 10.3390/en15134697.
6. Y. Kim et al. "Classification of Fault and Failure Types Determined by Dissolved Gas Analysis for Transformers". *Journal of Electrical Engineering & Technology* 14.4 (June 2019), pp. 1665–1674. DOI: 10.1007/s42835-019-00175-0.
7. Y. Lu and S. J. Liu. "Performance Comparison and Selection of Transformer Fluid". *MATEC Web of Conferences* 63 (2016). Ed. by J. Kao and W.-P. Sung, p. 02009. ISSN: 2261-236X. DOI: 10.1051/matecconf/20166302009.
8. M. Meira et al. "Dissolved Gas Analysis Differences Between Natural Esters and Mineral Oils Used in Power Transformers: A Review". *IET Generation, Transmission & Distribution* 13.24 (2019), pp. 5441–5448. ISSN: 1751-8695. DOI: 10.1049/iet-gtd.2018.6318.
9. Working Group D1.32. "DGA in Non-Mineral Oils and Load Tap Changers and Improved DGA Diagnosis Criteria". *Cigre Technical Brochure 443* (2010).
10. T. Buchacz, J. Buchacz, and M. Duval. "Stray Gassing of Oil in HV Transformers". *IEEE Transactions on Dielectrics and Electrical Insulation* 28.5 (Oct. 2021), pp. 1729–1734. DOI: 10.1109/tdei.2021.009520.
11. *IEC 60599:2022 – Mineral Oil-Filled Electrical Equipment In Service – Guidance on the Interpretation of Dissolved and free Gases Analysis*. 2022.
12. J. Golarz. "Understanding Dissolved Gas Analysis (DGA) techniques and interpretations". *2016 IEEE/PES Transmission and Distribution Conference and Exposition (T&D)*. May 2016. DOI: 10.1109/tdc.2016.7519852.

13. M. Duval and T. Heizmann. "Identification of Stray Gassing of Inhibited and Uninhibited Mineral Oils in Transformers". *Energies* 13.15 (July 2020), p. 3886. ISSN: 1996-1073. DOI: 10.3390/en13153886.

14. S. Kim, H. Seo, and J. Jung. "Advanced dissolved gas analysis method with stray gassing diagnosis". *2016 International Conference on Condition Monitoring and Diagnosis (CMD)*. Sept. 2016. DOI: 10.1109/cmd.2016.7757877.

15. I. B. M. Taha, A. Hoballah, and S. S. M. Ghoneim. "Optimal ratio limits of rogers' four-ratios and IEC 60599 code methods using particle swarm optimization fuzzy-logic approach". *IEEE Transactions on Dielectrics and Electrical Insulation* 27.1 (Feb. 2020), pp. 222–230. ISSN: 1558-4135. DOI: 10.1109/tdei.2019.008395.

16. A. Wajid et al. "Comparative Performance Study of Dissolved Gas Analysis (DGA) Methods for Identification of Faults in Power Transformer". *International Journal of Energy Research* 2023 (Sept. 2023). Ed. by A. L. C. Minh, pp. 1–14. ISSN: 0363-907X. DOI: 10.1155/2023/9960743.

17. P. R. M. Giri, I. M. Y. Negara, and D. A. Asfani. "Recent Development in DGA Diagnosis Using Graphical Analysis Method". *2021 International Seminar on Intelligent Technology and Its Applications (ISITIA)*. July 2021. DOI: 10.1109/isitia52817.2021.9502223.

18. I. H. Sarker. "Machine Learning: Algorithms, Real-World Applications and Research Directions". *SN Computer Science* 2.3 (2021). ISSN: 2661-8907. DOI: 10.1007/s42979-021-00592-x.

19. D. Dai, W. Tan, and H. Zhan. *Understanding the Feedforward Artificial Neural Network Model From the Perspective of Network Flow*. 2017. DOI: 10.48550/ARXIV.1704.08068.

20. K. O'Shea and R. Nash. *An Introduction to Convolutional Neural Networks*. 2015. DOI: 10.48550/ARXIV.1511.08458.

21. A. Sherstinsky. "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network" (2018). DOI: 10.48550/ARXIV.1808.03314.

22. F. Farnia and A. Ozdaglar. *GANs May Have No Nash Equilibria*. 2020. DOI: 10.48550/ARXIV.2002.09124.

23. G. Cybenko. "Approximation by Superpositions of a Sigmoidal Function". *Mathematics of Control, Signals, and Systems* 2.4 (Dec. 1989), pp. 303–314. ISSN: 1435-568X. DOI: 10.1007/bf02551274.

24. S. R. Dubey, S. K. Singh, and B. B. Chaudhuri. "Activation functions in deep learning: A comprehensive survey and benchmark". *Neurocomputing* 503 (Sept. 2022), pp. 92–108. DOI: 10.1016/j.neucom.2022.06.111.

25. I. Nusrat and S.-B. Jang. "A Comparison of Regularization Techniques in Deep Neural Networks". *Symmetry* 10.11 (Nov. 2018), p. 648. ISSN: 2073-8994. DOI: 10.3390/sym10110648.
26. T. Kurbiel and S. Khaleghian. *Training of Deep Neural Networks based on Distance Measures using RMSProp*. 2017. DOI: 10.48550/ARXIV.1708.01911.
27. I. Loshchilov and F. Hutter. "Fixing Weight Decay Regularization in Adam" (2018).
28. T. T. Phuong and L. T. Phong. "On the Convergence Proof of AMSGrad and a New Version" (2019). DOI: 10.48550/ARXIV.1904.03590.
29. L. Ciampiconi et al. "A survey and taxonomy of loss functions in machine learning" (Jan. 2023). DOI: 10.48550/ARXIV.2301.05579. arXiv: 2301.05579 [cs.LG].
30. I. J. Good. "Rational Decisions". *Journal of the Royal Statistical Society. Series B (Methodological)* 14.1 (1952), pp. 107–114. ISSN: 00359246. URL: http://www.jstor.org/stable/2984087 (visited on 07/10/2023).
31. P. M. Radiuk. "Impact of Training Set Batch Size on the Performance of Convolutional Neural Networks for Diverse Datasets". *Information Technology and Management Science* 20.1 (Jan. 2017). DOI: 10.1515/itms-2017-0003.
32. M. Hearst et al. "Support vector machines". *IEEE Intelligent Systems and their Applications* 13.4 (July 1998), pp. 18–28. ISSN: 1094-7167. DOI: 10.1109/5254.708428.
33. Y. Izza, A. Ignatiev, and J. Marques-Silva. "On Tackling Explanation Redundancy in Decision Trees" (2022). DOI: 10.48550/ARXIV.2205.09971.
34. P. Cunningham and S. J. Delany. "k-Nearest Neighbour Classifiers - A Tutorial". *ACM Computing Surveys* 54.6 (July 2021), pp. 1–25. ISSN: 1557-7341. DOI: 10.1145/3459665.
35. M. Demirci, H. Gozde, and M. C. Taplamacioglu. "Comparative Dissolved Gas Analysis with Machine Learning and Traditional Methods". *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. IEEE, June 2021. DOI: 10.1109/hora52670.2021.9461371.
36. M. Duval and A. dePabla. "Interpretation of gas-in-oil analysis using new IEC publication 60599 and IEC TC 10 databases". *IEEE Electrical Insulation Magazine* 17.2 (Mar. 2001), pp. 31–41. DOI: 10.1109/57.917529.
37. S. Ghosh and S. Dutta. "Ensemble Machine Learning Methods for better Dynamic Assessment of Transformer Status". *Journal of The Institution of Engineers (India): Series B* 102.5 (May 2021), pp. 1113–1122. DOI: 10.1007/s40031-021-00599-1.
38. Y. Benmahamed, M. Teguar, and A. Boubakeur. "Application of SVM and KNN to Duval Pentagon 1 for transformer oil diagnosis". *IEEE Transactions on Dielectrics and Electrical Insulation* 24.6 (Dec. 2017), pp. 3443–3451. DOI: 10.1109/tdei.2017.006841.

39. V. B. S. Prasath et al. "Distance and Similarity Measures Effect on the Performance of K-Nearest Neighbor Classifier – A Review" (2017). DOI: 10.48550/ARXIV.1708.04321.
40. M. U. Farooque, S. A. Wani, and S. A. Khan. "Artificial neural network (ANN) based implementation of Duval pentagon". *2015 International Conference on Condition Assessment Techniques in Electrical Systems (CATCON)*. IEEE, Dec. 2015. DOI: 10.1109/catcon.2015.7449506.
41. D. Bank, N. Koenigstein, and R. Giryes. *Autoencoders*. 2020. DOI: 10.48550/ARXIV.2003.05991.
42. V. Miranda, A. R. G. Castro, and S. Lima. "Diagnosing Faults in Power Transformers With Autoassociative Neural Networks and Mean Shift". *IEEE Transactions on Power Delivery* 27.3 (July 2012), pp. 1350–1357. DOI: 10.1109/tpwrd.2012.2188143.
43. F. R. Barbosa et al. "Application of an Artificial Neural Network in the Use of Physicochemical Properties as a Low Cost Proxy of Power Transformers DGA Data". *IEEE Transactions on Dielectrics and Electrical Insulation* 19.1 (Feb. 2012), pp. 239–246. DOI: 10.1109/tdei.2012.6148524.
44. M. Ou et al. "A Dynamic Adam Based Deep Neural Network for Fault Diagnosis of Oil-Immersed Power Transformers". *Energies* 12.6 (Mar. 2019), p. 995. DOI: 10.3390/en12060995.
45. N. Srivastava et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". *J. Mach. Learn. Res.* 15.1 (Jan. 2014), pp. 1929–1958. ISSN: 1532-4435.
46. F. Zakaria, D. Johari, and I. Musirin. "Artificial neural network (ANN) application in dissolved gas analysis (DGA) methods for the detection of incipient faults in oil-filled power transformer". *2012 IEEE International Conference on Control System, Computing and Engineering*. IEEE, 2012. DOI: 10.1109/iccsce.2012.6487165.
47. Y. Bai et al. *Understanding and Improving Early Stopping for Learning with Noisy Labels*. 2021. DOI: 10.48550/ARXIV.2106.15853.
48. I. B. M. Taha, S. Ibrahim, and D.-E. A. Mansour. "Power Transformer Fault Diagnosis Based on DGA Using a Convolutional Neural Network With Noise in Measurements". *IEEE Access* 9 (2021), pp. 111162–111170. DOI: 10.1109/access.2021.3102415.
49. S. Ioffe and C. Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. 2015. DOI: 10.48550/ARXIV.1502.03167.
50. S. Li et al. "Interpretation of DGA for transformer fault diagnosis with complementary SaE-ELM and arctangent transform". *IEEE Transactions on Dielectrics and Electrical Insulation* 23.1 (Feb. 2016), pp. 586–595. DOI: 10.1109/tdei.2015.005410.

51. Y. Zhang et al. "Fault diagnosis of transformer using artificial intelligence: A review". *Frontiers in Energy Research* 10 (Sept. 2022). ISSN: 2296-598X. DOI: 10.3389/fenrg.2022.1006474.

52. M. Badawi et al. "A Novel DGA Oil Interpretation Approach Based on Combined Techniques". *2022 23rd International Middle East Power Systems Conference (MEPCON)*. IEEE, Dec. 2022. DOI: 10.1109/mepcon55441.2022.10021795.

53. P. Mirowski and Y. LeCun. "Statistical Machine Learning and Dissolved Gas Analysis: A Review". *IEEE Transactions on Power Delivery* 27.4 (Oct. 2012), pp. 1791–1799. ISSN: 1937-4208. DOI: 10.1109/tpwrd.2012.2197868.

54. P. Singh and T. Blackburn. "Dissolved Gas Analysis Results in Wind Turbine Transformers". *2018 Australasian Universities Power Engineering Conference (AUPEC)*. IEEE, Nov. 2018. DOI: 10.1109/aupec.2018.8758061.

55. A. R. E. Soto, S. L. Lima, and O. R. Saavedra. "Incipient Fault Diagnosis in Power Transformers by DGA using a Machine Learning ANN - Mean Shift Approach". *2019 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*. IEEE, Nov. 2019. DOI: 10.1109/ropec48299.2019.9057143.

56. S. Hao et al. "Synthetic Data in AI: Challenges, Applications, and Ethical Implications" (Jan. 2024). DOI: 10.48550/ARXIV.2401.01629. arXiv: 2401.01629 [cs.LG].

57. C. Subroto et al. "Artificial intelligence for DGA interpretation methods using weighting factor". *2017 1st International Conference on Electrical Materials and Power Equipment (ICEMPE)*. IEEE, May 2017. DOI: 10.1109/icempe.2017.7982151.

58. H. Sutikno, R. A. Prasojo, and Suwarno. "Integration of Duval Pentagon to the Multi-Method Interpretation to Improve the Accuracy of Dissolved Gas Analysis Technique". *2021 IEEE International Conference on the Properties and Applications of Dielectric Materials (ICPADM)*. IEEE, July 2021. DOI: 10.1109/icpadm49635.2021.9493929.

59. L. Zhang et al. "A Fault Diagnosis Method of Power Transformer Based on Cost Sensitive One-Dimensional Convolution Neural Network". *2020 5th Asia Conference on Power and Electrical Engineering (ACPEE)*. IEEE, June 2020. DOI: 10.1109/acpee48638.2020.9136223.

60. M. A. Lones. "How to avoid machine learning pitfalls: a guide for academic researchers" (2021). DOI: 10.48550/ARXIV.2108.02497. arXiv: 2108.02497 [cs.LG].

61. M. W. Spratling. "Comprehensive Assessment of the Performance of Deep Learning Classifiers Reveals a Surprising Lack of Robustness" (Aug. 2023). DOI: 10.48550/ARXIV.2308.04137. arXiv: 2308.04137 [cs.LG].

62. P. I. Frazier. *A Tutorial on Bayesian Optimization*. 2018. DOI: 10.48550/ARXIV.1807.02811.

63. R. Turner et al. *Bayesian Optimization is Superior to Random Search for Machine Learning Hyperparameter Tuning: Analysis of the Black-Box Optimization Challenge 2020*. 2021. DOI: 10.48550/ARXIV.2104.10201.

64. P. Liashchynskyi and P. Liashchynskyi. "Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS" (Dec. 2019). arXiv: 1912.06059 [cs.LG].

65. H. Hu et al. *Adaptively Customizing Activation Functions for Various Layers*. 2021. DOI: 10.48550/ARXIV.2112.09442.

66. Z. Liu et al. *Dropout Reduces Underfitting*. 2023. DOI: 10.48550/ARXIV.2303.01500.

67. R. Müller, S. Kornblith, and G. Hinton. *When Does Label Smoothing Help?* 2019. DOI: 10.48550/ARXIV.1906.02629.

Appendices

A

Gradient Descent Example

An example is provided to demonstrate the gradient descent algorithm. A simple neural network is used to predict partial discharge or low temperature faults by using hydrogen and methane samples: cases with higher amounts of hydrogen should be classified as PD, and higher amounts of methane should be classified as T1. The network consists of a 2×2 weights matrix with ReLU applied at the output, and a sum-of-square error loss function is used. The feedforward and loss computation of the network operates by:

$$y = \text{ReLU}(xW^T), \tag{A.1}$$

$$L(y, y_{\text{true}}) = \Sigma(y - y_{\text{true}})^2, \tag{A.2}$$

where y is the model's output, x is the input, W is the weights matrix, ReLU is defined in figure 3.2, and L is the loss. Σ denotes that the sum of squared errors across all samples and outputs is used. The model's loss is computed by a composition of functions:

$$L = S(A(y(W, x))), \tag{A.3}$$

where S is the sum-of-square error function, A is the activation function, and y is the product of the weights with input. Equation 3.1 shows that in the gradient descent algorithm, the weights are updated by subtracting the partial derivative of the loss function with respect to that weight. During the feedforward process, each node of the computational graph that models the network stores the partial derivative of the node's output with respect to its input. The partial derivative of the loss function with respect to the network's weights is then computed via the

chain rule:

$$\frac{\partial L}{\partial W} = \frac{\partial S}{\partial A}\frac{\partial A}{\partial y}\frac{\partial y}{\partial W} \tag{A.4}$$

$$\frac{\partial L}{\partial W} = 2(\mathrm{ReLU}(xW^T) - y_{\mathrm{true}})\frac{\partial A}{\partial y}\frac{\partial y}{\partial W} \tag{A.5}$$

$$\frac{\partial L}{\partial W} = 2(\mathrm{ReLU}(xW^T) - y_{\mathrm{true}})(\frac{\partial A}{\partial y} \odot \frac{\partial y}{\partial W}) \tag{A.6}$$

$$\frac{\partial L}{\partial W} = 2\big((x \odot [1 \text{ if } y > 0 \text{ else } 0])^T(\mathrm{ReLU}(xW^T) - y_{\mathrm{true}})\big)^T, \tag{A.7}$$

noting that the ReLU function's derivative is 0 if its input is negative, and 1 if it is positive. Python defines one-dimensional arrays as row vectors, and this convention is followed. With n samples, the x matrix is of size $n \times i$, the prediction is of size $n \times o$, and the weights are of size $i \times o$. An example set of training data is given:

$$x = \begin{bmatrix} \mathrm{H}_{2(1)} & \mathrm{CH}_{4(1)} \\ \mathrm{H}_{2(2)} & \mathrm{CH}_{4(2)} \end{bmatrix} = \begin{bmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \end{bmatrix} \tag{A.8}$$

$$y_{\mathrm{true}} = \begin{bmatrix} \mathrm{PD}_{(1)} & \mathrm{T1}_{(1)} \\ \mathrm{PD}_{(2)} & \mathrm{T1}_{(2)} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{A.9}$$

with starting weights and an initial output of:

$$W_n = \begin{bmatrix} 1.0 & -0.2 \\ 0.1 & 0.9 \end{bmatrix} \quad y_n = \begin{bmatrix} 0.76 & 0.26 \\ 0.28 & 0.58 \end{bmatrix}. \tag{A.10}$$

Using these values and equation A.7, the gradient and weight-update with a learning rate of 0.2 is:

$$\frac{\partial L}{\partial W} = \begin{bmatrix} -0.16 & 0.24 \\ 0.08 & -0.40 \end{bmatrix} \tag{A.11}$$

$$W_{n+1} = W_n - 0.2\frac{\partial L}{\partial W} = \begin{bmatrix} 1.031 & -0.248 \\ 0.084 & 0.98 \end{bmatrix}. \tag{A.12}$$

With the updated weights, the output of the network becomes:

$$y_{n+1} = \begin{bmatrix} 0.776 & 0.263 \\ 0.264 & 0.621 \end{bmatrix}, \quad \text{A.13}$$

which more closely resembles the target value, y_{true}, when compared with the model's initial output. After an additional 5 iterations, the output is:

$$y_{n+6} = \begin{bmatrix} 0.838 & 0.213 \\ 0.194 & 0.741 \end{bmatrix}. \quad \text{A.14}$$

B

Model Training Pseudocode

The following pseudocode listing outlines the training loop used for each model.

```
for Model in PPM_Ratios, Ensemble, Full:
  # instantiate optimizer and loss
  Optimizer = AdamW(Model.parameters(), *opti_args)
  Loss = CrossEntropy(*loss_args)
  # load data
  data_train, data_validate, data_test =
              load_dataset(all_data, fold_num)
  gas_validate, fault_validate =
              data_validate.input, data_validate.target
  for e in 0,1,2...max_epochs: # iterate epochs
    for gas_train, fault_train in iter_batch_data(data_train):
      # forward pass: outputs and gradients are stored
        # at each node within the model
      fault_pred = Model(gas_train) # make prediction
      loss = Loss(fault_pred, fault_targ) # calc loss
      Optimizer.zero_grad()
      Loss.backward() # back-propagate loss
      # Optimizer updates Model weights according to
        # its hyper-parameters, loss, and the
          # stored gradients and outputs of the Model's nodes
      Optimizer.step() # update weights
    # see if model improved after batches complete
    f1_validate = compute_F1(Model(gas_validate),
                                    fault_validate)
    if f1_validate > f1_best: # if model improves
      Model_Best = Model  # store model as best
      f1_best = f1_validate
      e_best = e
    if use_ramp_learning_rate:
      Optimizer.LR = min(2*Optimizer.LR, lr_max)
    if e > e_best + early_term:
      break # terminate if not improving after early_term
```

End of Thesis.