POWER TRANSFORMER OIL DIAGNOSIS BY USING DISSOLVED GAS-IN-OIL MEASURING RESULTS

Pius Victor Chombo

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Electrical Engineering

Suranaree University of Technology

Academic Year 2016

การประเมินอายุของหม้อแปลงแรงดันกำลัง โดยการตรวจสอบ แก๊สที่ละลายในน้ำมัน และความคงทนต่อการเกิดเบรกดาวน์ ของน้ำมันหม้อแปลง

นายพิอุส วิคเตอร์ ชอมโบ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2559

POWER TRANSFORMER OIL DIAGNOSIS BY USING DISSOLVED GAS-IN-OIL ANALYSIS MEASURING RESULTS

Suranaree University of Technology has approved this thesis submitted in partial fulfillment of the requirements for a Master's Degree.

Thesis Examining Committee

(Assoc. Prof. Dr. Kitti Attakitmongcol)

Chairperson

(Assoc. Prof. Dr. Thanatchai Kulworawanichpong)

Member (Thesis Advisor)

(Asst. Prof. Dr. Boonruang Marungsri)

Member

(Asst. Prof. Dr. Anant Oonsivilai)

Member

(Asst. Prof. Dr. Padej Pao-laor)

Member

(Prof. Dr. Sukit Limpijumnong)

Vice Rector for Academic Affairs and Innovation

(Assoc. Prof. Flt. Lt. Dr. Kontorn Chamniprasart)

Dean of Institute of Engineering

พิอุส วิคเตอร์ ช็อมโบ : การวินิจฉัยน้ำมันหม้อแปลงกำลังโดยใช้ผลการวัดแก๊สละลายในน้ำมัน (POWER TRANSFORMER OIL DIAGNOSIS BY USING DISSOLVED GAS-IN-OIL ANALYSIS MEASURING RESULTS) อาจารย์ที่ปรึกษา : รองศาสตราจารย์ ดร.ธนัดชัย กุลวรวานิชพงษ์, 226 หน้า

ในวิทยานิพนธ์เล่มนี้ โปรแกรมพื้นฐานของคอมพิวเตอร์ได้พัฒนาเพื่อให้ง่ายต่อการวินิจฉัยความสมบูรณ์ของหม้อแปลงโดยผ่านการหาจากข้อบกพร่องที่เกิดขึ้น ในซอฟต์แวร์ DGA ใช้ 5 วิธีการในการตรวจสอบข้อมูลที่เป็นปัญหาต่อหม้อแปลง ประกอบด้วย คีย์แก๊ส (Key gas method) อัตราส่วนดอร์เนนบูร์ก (Dornenburg ratio Method) อัตราส่วนโรเจอร์ (Roger's ratio method) อัตราส่วน ไออีซี (IEC ratio method) และสามเหลี่ยมดูวาล (Duval triangle method) ซอฟต์แวร์ DGA มีการติดตั้งเครื่องมือทำนายแนวโน้มและเครื่องมือเปรียบเทียบของสถานะของหม้อแปลงนั้น ๆ การเขียนหน้าต่างใช้งานของซอฟต์แวร์ DGA จะใช้ภาษาไพทอนซึ่งมีคำสั่งการทำงานที่หลากหลาย และมีการจัดการเรื่องระบบความจำที่ดี ในส่วนของฐานข้อมูลใช้ภาษา SQL ที่สามารถจัดการระบบฐานข้อมูลที่รวดเร็ว แม่นยำ ถูกต้อง และพื้นที่จัดเก็บเพียงพอ ในส่วนของการเชื่อมโยงสำหรับฐานข้อมูลและซอฟต์แวร์ DGA ใช้โปรแกรมที่เรียกว่า MySQL-python connector โปรแกรม DGA นี้ได้นำไปทดลองตรวจสอบค่าแก๊สละลายในน้ำมันจำนวน 50 ตัวอย่าง ที่ได้มาจากนักวิจัยคนอื่น ๆ พบว่าเทคนิคสามเหลี่ยมดูวาลมีความสามารถสูงสุด (90%) และมีความสม่ำเสมอสูงสุด (90%) เมื่อเทียบกับเทคนิคการวินิจฉัยแบบอื่น ในการเปรียบเทียบวิธีการต่าง ๆ นี้กับซอฟต์ระบบของนักวิจัยคนอื่น ๆ พบว่าเทคนิคสามเหลี่ยมดูวาลมีความแม่นยำมากที่สุด (90%) ไออีซี (89%) คีย์แก๊ส (75%) อัตราส่วนโรเจอร์ (68%) และอัตราส่วนดอร์เนนบูร์ก (44%) ตามลำดับ ดังนั้นโปรแกรมการวินิจฉัยแก๊สในน้ำมันที่พัฒนาขึ้นนี้มีความน่าเชื่อถือและสามารถนำไปใช้งานได้

น้ำมันหม้อแปลงที่นำมาทดสอบได้มาจากหม้อแปลงกำลังของระบบจำหน่ายของมหาวิทยาลัยเทคโนโลยีสุรนารีจำนวน 28 ตัวอย่าง แก๊สในน้ำมันถูกวินิจฉัยตามมาตราฐาน IEEEC57.104-2008 และมาตราฐาน IEC60599 ในการวินิจฉัยทั้ง 28 ตัวอย่าง วิธีการคีย์แก๊สให้ค่าการทำนาย 43.47 % ซึ่งมีความร้อนเกินในเซลลูโลส ขณะที่เทคนิคสามเหลี่ยมดูวาลให้ค่าการทำนาย 53.57 % ซึ่งมีความร้อนเกิดขึ้นมากที่สุดที่ T3 การใช้ฐานข้อมูลจัดเก็บข้อมูลของ DGA นั้นเป็นประโยชน์อย่างมากเมื่อต้องการเปรียบเทียบค่าแก๊สและแนวโน้มการเกิดแก๊ส หม้อแปลงที่เกิดข้อบกพร่องที่ T3 จะต้องถูกบำรุงรักษาอยู่เป็นประจำเนื่องจากปล่อยไว้นาน ความเสียหายที่ T3 จะ

เพิ่มพูลขึ้น การบำรุงรักษาที่ล่าช้าจะเป็นผลให้ฉนวนเกิดการเสื่อมสลายได้ไวขึ้นและเป็นเหตุให้อายุของหม้อแปลงไฟฟ้ากำลังสั้นลง

สาขาวิชา วิศวกรรมไฟฟ้า

ปีการศึกษา 2559

ลายมือชื่อนักศึกษา

ลายมือชื่ออาจารย์ที่ปรึกษา

ลายมือชื่ออาจารย์ที่ปรึกษาร่วม

PIUS VICTOR CHOMBO : POWER TRANSFORMER OIL DIAGNOSIS BY USING DISSOLVED GAS-IN-OIL ANALYSIS MEASURING RESULTS. THESIS ADVISOR : ASSOC. PROF. THANATCHAI KULWORAWANICHPONG, PhD., 226 PP.

POWER TRANSFORMER /CONDITION MONITORING/ DISSOLVED GAS /DIAGNOSTIC METHODS

In this thesis, computer based software was developed to simplify the diagnoses of transformer health through determination of existing fault. Five diagnostic methods were implemented to interpret faults in the developed DGA software i.e. key gas method, Dornenburg ratio method; Roger's ratio method, IEC ratio method and Duval triangle method. The software also contains trending and comparison tools for further diagnoses of the transformer condition. Python programming language was used to make the interface of the developed software. It offers good code readability, multiple programming paradigms, cross platform interpreters and good memory management. Database was programmed by using MySQL language. It is a relational database management system with high reliability, better computational capability, high speed and enough storage. Database and developed software were connected together by using MySQL-python connector. Validation of the developed software was done by using fifty DGA data taken from other researchers. Duval triangle method showed the highest ability of detection

(90%) and the highest consistency (90%) compared to other diagnostic methods. In addition, Duval triangle method depicted the greatest accuracy in both cases (based on detected cases and based on total cases) of 90% followed by IEC (89%), key gas (75%), Roger's (68%) and Dornenburg (44%) when compared to other researchers' systems. Therefore, DGA software conformed to be used as the diagnostic software of DGA information.

In real application, twenty-eight samples from in-service oil-filled power transformers in Suranaree University of Technology were taken for experiment. Dissolved gas-in-oil analysis (DGA) was performed to identify types of existing faults. Interpretation of faults were based on IEEE C57.104-2008 and IEC 60599. In 28 samples, key gas method showed a percentage of detection of 43.47 % with overheated in cellulose as a major detected fault while Duval triangle showed 53.57% with thermal fault T3 as the most occurred fault. Transformers with T3 fault were advised to be highly considered in maintenance due to the harmfulness of T3 fault when extended in service. The delay in maintenance activities may result into high degradation of insulation and hence shortens transformer life.

School of Electrical Engineering — Student's Signature__________

Academic Year 2016 — Advisor's Signature__________

Co-Advisor's Signature__________

ACKNOWLEDGMENT

The author would like to thank God for giving us health and enabling me to do this work with the School of Electrical Engineering, Suranaree University of Technology here in Thailand.

Many thanks should go to my Co-Thesis Advisor, Assistant Professor Dr. Boonruang Marungsri for his guidance and support throughout my studies and the accomplishment of this thesis. Also, special thanks should go to the committee, Associate Professor Dr. Kitti Attakitmongcol (Chairperson), Associate Professor Dr. Thanatchai Kulworawanichpong (Thesis Advisor), Assistance Professor Dr. Anant Oonsivilai (Member), and Assistant Professor Dr. Padej Pao-laor (Member) for their advice.

In addition, I would like to express my grateful thanks to all graduates and staffs from the school of Electrical Engineering and other schools in the University for their support and encourages.

Special thanks should also go to my beloved wife, BhokeWilliam Saguge, for her encourages, prayers and other supports to the accomplishment of my studies.

Pius Victor Chombo

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

CHAPTER 1

INTRODUCTION

1.1 General Introduction

Power transformers are vital components in the transmission and distribution of electrical power. Insulating oil in a power transformer is commonly known as transformer oil and used to form part of transformer insulation. It is normally obtained by fractional distillation and subsequent treatment of crude petroleum. That is why this oil is also known as mineral insulating oil. The four functions of insulating oil are to provide cooling, insulation, protection against chemical attack and prevention of sludge buildup.

In several parts of the world, most of the power transformers are operated beyond their design life and with higher average loads than ever before. Thermal and Electrical stresses are major disturbances that affect the performance of a power transformer regarding insulation deteriorations. Thus, when thermal and electrical stresses are higher than minimum permissible values, they cause deterioration of cellulosic materials and insulating oil as well. It is known that whenever a power transformer undergoes abnormal thermal and electrical stresses, certain gasses are produced due to decomposition of insulating oil; and when the fault is major, the production of decomposed gasses are more, and they are collected in the Buchholz relay. Further, when abnormal thermal and electrical stresses are not significantly high, gasses due to decomposition of transformer insulating oil will get enough time to dissolve in the oil. Hence, by only depending on the Buchholz relay, it is not possible to predict the condition of the total internal healthiness of electrical power transformer. That is why it becomes necessary to analyze the level of different gasses dissolved in transformer oil in-service. Dissolved gas-in-oil analysis (DGA) has been widely accepted for condition monitoring of oil-filled power transformer. Several diagnostic methods based on DGA have been achieved for interpreting existing faults from the types and amount of dissolved gasses available in the transformer oil. Some of these are key gas, Dornenburg ratio and Roger's ratio methods which are based on IEEE C57.104-2008. Other methods are IEC ratio and Duval triangle which are based on IEC 60599-1999. From the dissolved gas-in-oil analysis (DGA), one can predict the actual condition of internal health of a transformer through a number of dissolved gasses. Based on the type of fault gasses and their proportions the type of incipient failure can be deduced.

More studies have been undertaken to understand the physical and chemical phenomena occurring in the mineral oil during thermal and electrical stresses. Fofana, Bouaicha, Farzaneh, and Sabau, (2008); Boudraa, Mokhnache, Fofana, (2014) showed that during oil decomposition not only dissolved gasses are generated but also colloidal suspensions. Yardaz and Sarathi, (2015) explained that decomposition of cellulose and insulating oil molecules might formulate new molecules and acidic residues which are hazardous for cellulosic material. The presence of by-products in oil was further explained to be harmful in the transformer insulation by causing a gradual reduction of oil quality. Colloidal suspensions were reported to support the acceleration of oxidation in liquid insulation which results in increased acidity level and hence destruction in solid insulation; and formation of insoluble sludge which reduces the cooling rate of insulation system (Boudraa, Mokhnache, Fofana, 2014).

Therefore, accelerated oxidation and formation of insoluble sludge greatly reduce the quality of transformer insulation especially liquid insulation. Liquid insulation can be used as an important, informative tool for condition monitoring with about 70% of occurred situations (Boudraa, Mokhnache, Fofana, 2014). However, previous works have demonstrated condition monitoring of transformer with the use of dissolved gasses only without considering the quality of liquid insulation. The accumulation of dissolved gasses in oil reduces the quality and ability of the insulating oil to withstand stress.

1.2 Research Objectives

The interrelated studies considered in this thesis are;

1. To develop computer software with database for storing DGA information and simplify diagnoses process,
2. To perform diagnoses by using key gas, Dornenburg ratio, Roger's ratio, IEC ratio and Duval triangle methods implemented in the DGA software,
3. To conduct condition monitoring based on the results of in-service transformers obtained from the DGA software.

1.3 Scope and limitation of the study

In this study, transformer's life diagnosis was performed by using the results of dissolved gas-in-oil analysis. Twenty-eight samples of mineral insulating oil from in-service transformers were sampled from transformers available in Suranaree University of Technology. Sampling was performed in three batches, daytimes with humidity less than 50%, kept in amber colored bottles and later transported in a sample carrier to avoid any contamination and exposure to UV lights. Seven gasses were extracted from samples i.e. H_2, CO, CO_2, CH_4, C_2H_6, C_2H_4, and C_2H_2. Then, transformer's life was diagnosed by investigating the type of existing fault and condition of the transformer.

- The expected benefits from the developed DGA software includes:
  - To enhance condition monitoring of in-service oil-filled power transformers,
  - To save time and money from outsourcing the jobs of performing condition monitoring,
  - To reduce risk of failure of in-service oil-filled power transformers,
  - To transfer technology into the society and new generation at minimum costs.

1.4 Thesis structure

An outline of this thesis is as follows:

Chapter 1 gives the general introduction, research objectives, scope and limitation of the study and thesis structure.

Chapter 2 explains fundamentals of mineral oil used as insulating oil in power transformers. The study of the aging mechanism of insulating oil is further discussed. Also, DGA computer based software used for condition monitoring of transformers have been analyzed.

Chapter 3 describes the development of DGA computer based software, its features, implementation of diagnostic methods, and implementation of the database. Validation of DGA software is further described.

Chapter 4 illustrates DGA experiment including its installation and test procedures.

Chapter 5 describes the design and application of test cell for the simulating corona in oil. It also discusses an investigation of existing relationship between individual dissolved gas and breakdown strength values from stressed samples.

Chapter 6 discusses the results obtained in DGA and their application in condition monitoring of transformers.

Chapter 7 concludes the results of condition monitoring in transformer insulation and suggests measures to be taken.

1.5 Chapter summary

This chapter described the general introduction about oil filled power transformer. Transformer insulation in general as well as liquid insulation especially mineral insulating oil has also been discussed. The major standard for interpreting existing fault inside the transformer and the methods used for diagnosis in dissolved gas-in-oil analysis have been explained. Lastly, the objectives of research including the scope and limitation of the work have been mentioned in this chapter.

CHAPTER 2

LITERATURE REVIEW

The main goal of this study is to perform power transformer's health diagnosis by evaluating dissolved gas-in-oil results of mineral insulating oil. This helps to know the health condition of the power transformer to operate safely before a total failure occurs. However, varieties of dissolved gases are generated and accumulated in the power transformer oil due to the decomposition of insulating oil during thermal, electrical and environmental stresses (Ashkezari, Saha, Ekanayake, and Ma, 2011; Muhamad, Phung, Blackburn, and Lai, 2012; Rondla, Falahi, Zhan and Goulart, 2012). The level of dissolved gasses mainly depends on the intensity of energy dissipated (Indra, Rajaram, and Gnanasaravanan, 2000). Furthermore, Sukbirr and Bandyopadhyay, (2010) showed that solubilities of fault gasses in transformer oil, as well as their temperature dependence, are also important factors for consideration in fault analyses. It should be noted that there are two orders of magnitude difference between the least soluble H_2 and the most soluble C_2H_2 gas. The majority of gasses that are indicative of faults are also those that are in general more soluble in oil. Over a temperature range of 0 – 80 °C some gasses increase in their solubility up to 79% while others decrease their solubility up to 66%. Several studies have been undertaken to study the condition based maintenance based on dissolved gasses and estimating the life of power transformers.

2.1 Fundamentals of insulating oil

Mineral insulating oil is an oil of mineral origin, refined from petroleum crude oil through fractional distillation process (Allen, 2009; Erdman, 1988). Like the crude oil themselves, insulating oils are mixtures of many hydrocarbon molecules having a variety of structures and a distribution of molecular sizes and weights. Most of these hydrocarbons fall into three broad classes: Alkanes (paraffin), Cycloalkanes (also called naphthenes, naphthenic or cycloparaffins) and Aromatics (Bartnikas, 1994). It is then treated to remove impurities and to obtain the most desirable properties to make it suitable as an insulating and cooling liquid (Erdman, 1988). Mineral oils used in electric apparatus are specified to have certain characteristics that permit reliable performance for many years (Hirschler, 2000). The most important property of an insulating liquid is the highest dielectric strength, and for cooling, the liquid is a low viscosity. Other properties are important to show the absence of impurities and to show the degree of stability for long life. Properties indicating purities are also important in consideration of the ability of oil to be with the other materials used in transformer construction (Erdman, 1988). Mineral oil is the most commonly used liquid in power transformer both for its physicochemical properties and low cost (Martinez-Vega, 2010). It provides insulation between live and grounded metal parts inside the transformer, protects solid insulation, dissipates heat energy and used as a diagnostic tool Horning, (2013).

2.1.1 Chemical composition of insulating oil

Due to the complexity of hydrocarbon mixture, mineral insulating oils are characterized in terms of molecular classes and subclasses present rather than by individual chemical species. One approach is based on the determined empirical relationship between physical properties of oil (viscosity, specific gravity and density, refractive index) and the percentage of carbon atoms in the oil present in each of the major structural organization of the hydrocarbons, which are paraffinic, aromatic and naphthenic. These relationships have been expressed algebraically and the numerical values, usually in percentage, of C_A the fraction of Aromatic, C_P of Paraffinic, and C_N, of Naphthenic, can be calculated (Rizk and Trinh, 2014). For good fresh insulating oil, it is desirable to have more of saturable paraffin, less of aromatics and/or naphthenes and none of the olefines. However, for better stability of properties, it is necessary to have optimum aromatic and/or naphthenic hydrocarbons. Such an optimum balance is struck carefully controlled during the refining process. Depending upon the predominance, oil is usually termed as a paraffinic base or naphthenic base (BHEL, 2003).

1. Paraffinic based mineral insulating oil

The paraffinic structure can be a straight or branched group of molecules. Paraffin molecules have low thermal stability and have a low solubility for water and oxidation products. The straight paraffin structure hurdles the free flow of oil at low temperatures and thus content of such type of molecules must be reduced. Figure 2.1 shows the molecular structure of Paraffinic based mineral insulating oil.

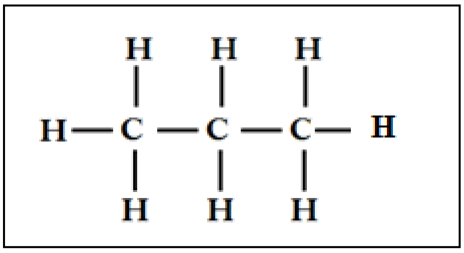


Figure 2.1 Molecular structure of Paraffinic base insulating oil (Rizk and Trinh, 2014).

2. Naphthenic based mineral insulating oil

The naphthenic structure has excellent low-temperature properties and better solvency power than paraffin. They usually have ring structures and also known as Cycloalkanes. Figure 2.2 shows the molecular structure of Naphthenic based mineral insulating oil.

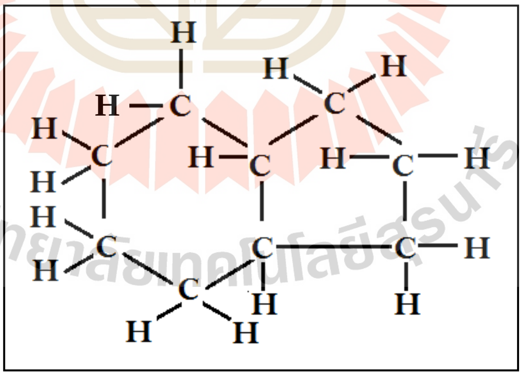


Figure 2.2 Molecular structure of Naphthenic base insulating oil (Rizk and Trinh, 2014).

2.1.2 Electrical properties of insulating oil

1. Dielectric strength

This is also defined as breakdown strength of transformer oil and is given in terms of kV. It is measured by observing at which voltage, sparking starts between a gap of two electrodes immersed in the oil. For mineral insulating oil, 30 kV is normally considered as a fair value for in-service transformers. The lower value of breakdown strength indicates the contamination of insulating oil with moisture and conducting particles. According to IEEE 60156, oil is kept in a test cell with one pair of electrodes fixed in a gap of 2.5 mm and then the voltage is slowly rose between the electrodes. The rate of rise of voltage is generally controlled at 2 kV/s and the voltage at which sparking starts between the electrodes is taken as the breakdown voltage. Measurement is performed to six times and the average value of these reading is taken as the strength of the oil.

2. Specific resistance

This indicates the ability of the insulating oil to resist current during normal and full load conditions. It is measured in ohm-cm (Ω-cm) and at a specific temperature. When temperature increases, the resistivity of oil rapidly decreases. This condition is similar to when the transformer is fully loaded; the temperature rises and may go up to 90 °C. Therefore, the resistivity of the insulating oil should be high at room temperature and good value at high temperature as well. This shows the importance of measuring the specific resistance of transformer oil at 27 °C as well as 90 °C. Minimum acceptable values of specific resistance of transformer oil at 90 °C and 27 °C is 35 × 10^{12} Ω-cm and 1500 × 10^{12} Ω-cm respectively.

3. Dielectric Dissipation Factor (tan δ)

An ideal insulating material may be considered as a non-conductor of electricity acting as a pure capacitance when built into equipment. However, in practice, conduction occurs and finite losses are produced that may be of great importance during the test and operating conditions (James and Su, 2008). Therefore, when an insulating material is subjected between live and grounded part of the electrical equipment, leakage current must flow. Since insulating material is dielectric in nature the leakage current through the insulation ideally leads the instantaneous voltage by 90^{o}. The finite losses are represented by a slight change in the current phase in relation to the applied alternating voltages. In most practical cases, insulating materials are not perfect dielectric in nature. Hence, the leakage current through the insulating material leads the instantaneous voltage with an angle little bit shorter than 90°. The tangent of that angle is called dielectric dissipation factor or sometimes tan delta of transformer oil. The leakage current through an insulating material have two component i.e. capacitive or reactive and another one is resistive or active (see Figure 2.3). Their relationship can be expressed as shown in (2.1).

$$I_R / I_C = \tan |\delta| \tag{2.1}$$

Figure 2.3 shows that the relationship between tangent delta or tan δ and components of the leakage current. From Figure 2.3 it can be seen that the value of δ is small when the resistive component of the leakage current I_R is small which indicates a high resistive property of the insulating material.

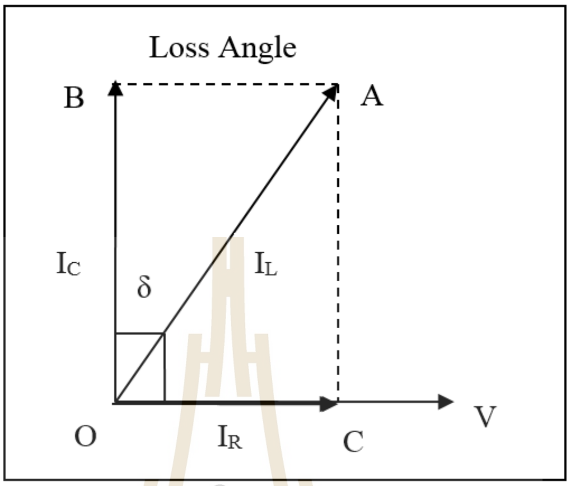


Figure 2.3 Phasor diagram of practical (lossy) dielectric

(James and Su, 2008).

High resistive insulation means that insulating material is a good insulator. Hence it is desirable to have loss angle as small as possible. When the value of tan δ is high, it indicates the presence of contaminants in transformer oil. Sometimes tan δ and resistivity of insulating oil can be closely related to each other. The lower the resistivity of the insulating oil, the higher is the value of tan δ and vice versa. Therefore, tan δ is a measure of the imperfection of dielectric nature of insulation materials like insulating oil.

2.1.3 Chemical properties of insulating oil

1. Moisture content

Paper insulator is highly hygroscopic in nature. In normal operating temperature, normally during normal loads, it absorbs the maximum amount of water from the oil. But during full load conditions, oil becomes hotter, hence the solubility of water in oil increases, as a result, the paper releases water and increase the water content in transformer oil. That is why oil temperature is of great importance during sampling. Moisture or water content in transformer oil is strongly unwanted as it affects adversely the dielectric properties of oil by supporting oxidation process. When the oxidation process occurs, acids are formulated in the oil, raise the solubility of water in the oil, and later, acids combined with water further decompose the oil forming more acid and water. The multiplicity of oxidation gradually increases the deterioration of the insulating oil. Accurate measurement of water content in a low level can be performed by the use of Coulometric Karl Fisher titrator and expressed in parts per million (ppm). The Karl Fischer titrator is shown in Figure 2.4. This method determines the amount of water by using Karl Fischer reagents such as iodine, sulfur dioxide, bases, and alcohols, which selectively and quantitatively react with water. The chemical equation for this reaction is shown in (2.2). From Figure 2.4, the method produces Iodine in proportion to the quantity of electricity according to Faraday's Law, on which the amount of water can be estimated based on the coulombs required for electrolytic oxidation as shown in (2.3)

$$I_2 + SO_2 + 3Base + ROH + H_2O \longrightarrow 2Base + HI + Base + HSO_4R \quad (2.2)$$

$$1\text{mg of water} = 10.71 \text{ coulombs} \quad (2.3)$$

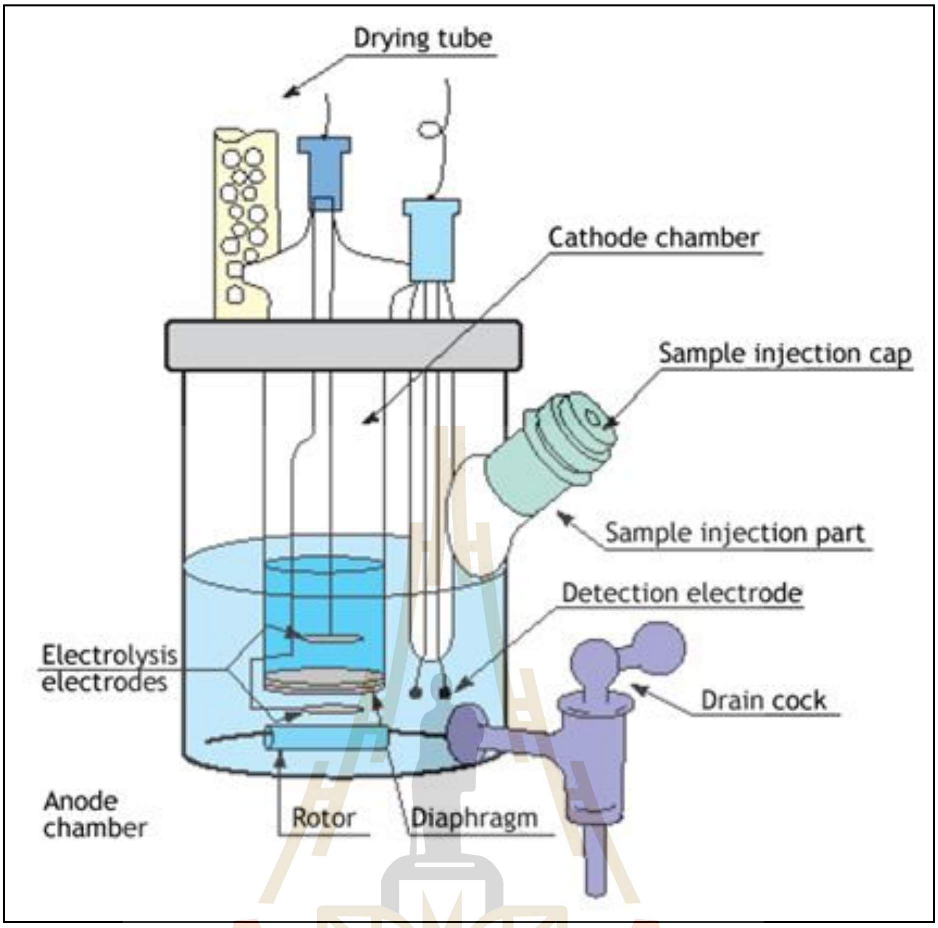


Figure 2.4 Coulometric Karl Fischer titrator

(Karl Fischer method, 2013).

2. Acidity

Acidity is a detrimental property as it causes water content to become soluble in the transformer oil. Moreover, it accelerates the oxidation process in the oil which deteriorates the insulation property of the paper in windings and causes rusting of metals in the presence of moisture. The level of acidity of transformer oil indicates the acidic constituents of contaminants and it is express in mg of KOH required to neutralize the acid present in a gram of oil. This is also known as neutralization number.

2.1.4 Physical properties of insulating oil

1. Interfacial tension

This is a measure of molecular attractive force between water and oil and is expressed in Dyne/cm or milli-Newton/meter. It is useful for deducing the presence of polar contaminants and oil decay products. Normally, good or new oil should have higher interfacial tension. Oil contaminants like those due to oxidation lower this property.

2. Flashpoint

This defines the temperature at which insulating oil can give enough vapors to produce a flammable mixture with air. With this mixture, a momentary flash is most likely to occur when a flame is applied under standard condition. This property is of very importance as it specifies the chances of fire hazard in the transformer. Therefore, transformer oils are required to have a very high flash point, of about 140 °C.

3. Pour point

This specifies the minimum temperature at which oil just start to flow under standard test condition. It is a very important property for extremely cold areas. This means that, if the temperature drops below pour point insulating oil can stop flowing and obstruct cooling system. Pour point is normally depends upon wax content in the oil. Paraffin-based oil has more wax content and higher value of pour point, compared to Naphtha based oil (Transformer insulating oil and types of transformer oil, 2017).

4. Viscosity

This is the resistance to the flow of insulating oil when subjected under normal condition. Good insulating oil must have low viscosity to offer less resistance to the conventional flow of oil. Since every liquid becomes more viscous when the temperature is decreased, for the best insulating oil viscosity should increase more slowly.

5. **Density**

BHEL (2003) described this test for special significance when the transformer is operated at a very low-temperature zone. The maximum value of density fixed at 29 °C ensures water in the form of ice present in oil remain at the bottom and does not tend to float on the oil up to a temperature of about -10 °C.

2.1.5 Functions of insulating oil

Mineral insulating oil serves the following functions when utilized in power transformers:

1. To provide dielectric strength

Kraft paper and mineral oil are the two materials that are most widely used as insulators in transformers (Horning, 2013). The two materials give the classical property when used as the dielectric strength material. An oil soaked paper gives a stronger dielectric than when oil is added arithmetically to the dielectric strength of the paper. From this combination, the dielectric strength of about 20-25% greater than just adding the dielectric breakdown strength of the two materials together can be observed. Mineral oil is widely employed as a liquid insulation due to its low cost and availability than other liquids. Oil/paper combination is affected by aging more slowly than in the case for either stand-alone material (Horning, 2013).

2. To protect paper insulator

The mechanical strength of paper insulator is mostly broken down by the effects of heat, moisture, and oxygen whereby 85 percent of all transformer failures occur because the paper has weakened to the point where it can no longer recover from stress (Horning, 2013). Therefore, the paper insulator is protected by insulating oil from the effects of heat, oxygen, and moisture. The oil helps to transfer the heat away from the paper immersed in the oil and core as it is generated. When oil deteriorates, by-products are formed in the oil and paper insulations which are very aggressive toward the paper and actively tear it apart, molecule by molecule. The mechanical strength of the solid insulation reduces drastically with the increase in by-products. Maintenance of the insulating oil can restore its ability to protect the paper.

3. To transfer generated heat

The physical properties of the oil are determined by the composition of the oil. The distribution and type of hydrocarbons present, and the molecular weight of those components determine key physical properties such as the viscosity profile, specific heat, relative density and coefficient of expansion for the oil (Horning, 2013). These key physical properties determine how well the oil will move to a point where heat can be “picked up” from the core and coils, absorb the excess heat, transport the excess heat from where it is not wanted to the shell of the transformer, and finally dissipate the excess heat to the atmosphere. As it ages in service, the oil’s characteristics with regard to the heat transfer change very slowly. For a long period of time, until solid sludge builds up in the oil, there will be basically no changes. As solid sludge forms, it will start to precipitate out of the oil. This will impede heat transfer by coating the metal surfaces where heat is dissipated by the oil to the atmosphere. Sludge from aged oil will also fill in pores and other structures in the solid insulation preventing complete passage of the oil into and out of the coil. That condition further impedes proper heat transfer. The physical properties that define the oil’s heat transfer capability do not change substantially during the aging properties, so almost all of the effects concerning heat transfer are due to sludge formation”.

4. A diagnostic tool

The internal condition of the transformer can easily be diagnosed through the use of insulating oil. It is helpful in describing existence and type of the incipient fault in the transformer.

2.2 The study of aging mechanism of insulating oil

Oil deteriorates because it oxidizes. The hydrocarbons in the oil react with the dissolved oxygen to form oxidation by-products in the oil which are sometimes called decay products, oxidation products, oxidation compounds or aging by-products in the oil. Insulating oil deteriorates and oxidizes in different rates which depend on some variables. The higher the deterioration rate, the more frequent maintenance is required, and the more is the damage done to the solid insulation if the required maintenance is delayed (Horning, Kelly, Myers and Stebbins, 2013). Referring to James and Su, (2008), aging effects are long time changes that occur in service. In Karmakar, Roy and Kumbhakar, (2011) the insulating oil of high voltage equipment was stated to be degraded due to the combination of aging processes such as electrical arcing, thermal aging and oxidation while it is operating in a long time of service. Arora and Mosch, (2011) described the aging process as a continuous process in oil filled equipment. Ingress of moisture, oxygen from the atmosphere and the presence of metallic parts of an apparatus like copper, lead, aluminum, and others under the catalytic effects of metal ions were showed to accelerate the oxidation reaction in oils at high working temperatures. The result gives the rise of oxidative products such as alcohols, ketones, esters, peroxides and acids which combine to form higher-molecular-weight compounds, both soluble and insoluble in oils; whereby insoluble oxidative products not only darken the oil color but in advance stage they lead to the formation of sludge. According to Chakravorti, Dey, and Chatterjee, (2013), factors that affect the life expectancy of the transformer are all neither known nor properly understood.

2.3 Dissolved gas-in-oil analysis

Dissolved gas-in-oil analysis has demonstrated to be extremely useful and consistent diagnostic tool in the detection and identifying incipient fault conditions in oil immersed power equipment especially transformer. It was first applied by Waddington and Allan of G.E.C. Power Engineering Ltd, Trafford Park Manchester to small distribution transformers having no Buchholz relay. The evidence of internal faults were observed by extracting dissolved gasses through vacuum treatment (Collacott, 1977). Nowadays DGA is widely used in industries as a primary diagnostic tool for transformer maintenance (Harlow, 2012). Laboratory-DGA involves taking samples of insulating oil, extracting dissolved gasses and quantifying those (Winders, 2002). DGA process use either of the two techniques when extracting dissolved gasses from samples: (i) Gas chromatography (ii) Photoacoustic spectrometry

2.3.1 Gas chromatography technique

This is a chemical analysis performed to detect and quantify organic substances from the sample. It consists of the narrow tube known as a column which is used to pass different chemical constituents of a sample through a gas stream (carrier gas) at different rates depending on their chemical, physical properties and their interaction with a specific column filling, called the stationary phase. Stationary phase separates components of the sample which cause each one to exit the column at a different time (retention time). Retention time can be altered by varying carrier gas flow rate, column length, and the temperature. The two phases which are used in the process are stationary and mobile phases. Carrier gasses usually used are Helium, Hydrogen, and Nitrogen. Figure 2.5 shows the arrangement for gas chromatography.

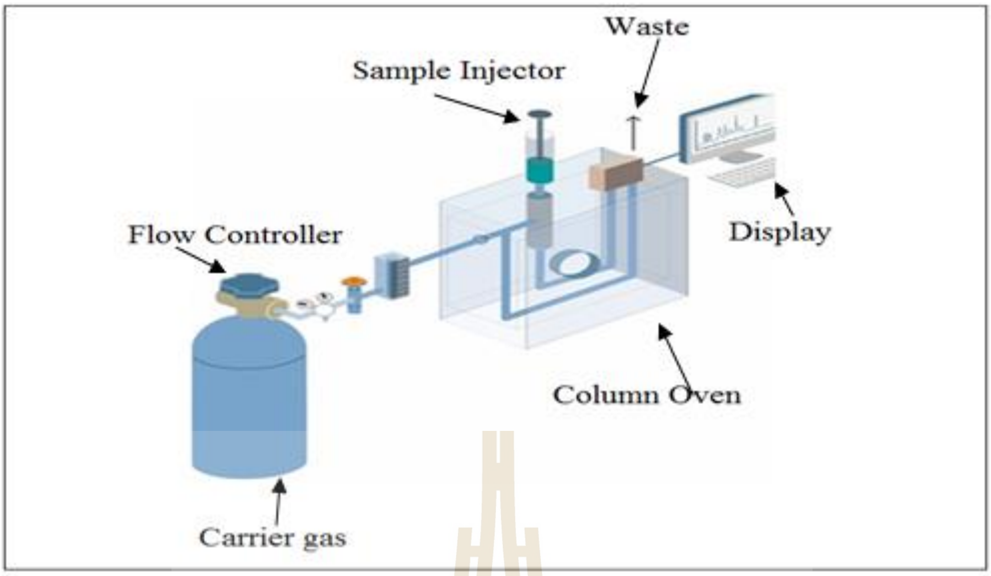


Figure 2.5 Gas chromatography technique

(Gas chromatographyy, 2017)

In Gas chromatographyy, (2017) the components of the gas chromatography were summarized as shown below:

1. Autosamplers

It provides the means to introduce a sample automatically into the inlets. Automatic insertion provides better reproducibility and time-optimization (Gas chromatography, 2017). Syringes are used manually by professionals to inject the sample into the gas chromatography. The use of an auto-sampler is a more robust approach to reproducibly inject samples into the gas chromatograph systems (Michelle and John, 2013). Currently, the automatic sampler is being used as this provides more accurate moves and saves on time.

2. Inlets

The column inlet (or injector) provides the means to introduce a sample into a continuous flow of carrier gas. The inlet is a piece of hardware attached to the column head. Types of injectors include split/splitless injector, programmed temperature vaporizer injector, and purge –and trap system (P/T).

3. Carrier gas

The choice of carrier gas depends on the type of detector that is used and the components that are to be determined. Carrier gasses for chromatography must be of high purity and chemically inert towards the sample e.g., helium (He), argon (Ar), nitrogen (N_2), carbon dioxide (CO_2) and hydrogen (H_2). The carrier gas system can contain a molecular sieve to remove water or other impurities. H_2 has a range of flow rates that are comparable to helium in efficiency. However, helium may be more efficient and provide the best separation if flow rates are optimized. Helium is non-flammable and works with a greater number of detectors and older instruments. Therefore, helium is the most common carrier gas used. However, the price of helium has gone up considerably over recent years, causing an increasing number of chromatography to switch to hydrogen gas (Gas chromatography, 2017).

4. Detectors

Thermal Conductivity Detector (TCD) is among the most commonly used measuring devices in gas chromatography for monitoring substances separated in the column. This detector measures the changes in the thermal conductivity of the carrier gas caused by the presence of the eluted substances (Jiri, 1975). The most employed detectors are the flame ionization detector (FID) and the thermal conductivity detector (TCD). They are sensitive and work over a wide range of components and concentrations respectively. FIDs are sensitive primarily to hydrocarbons components; however, they cannot detect water. TCD is non-destructive, but it can be used in-series before FID (destructive), thus providing complementary detection of the same analytes (Gas chromatography, 2017).

2.3.2 Photoacoustic spectroscopy technique

In Photoacoustic spectroscopy technique, an infrared is absorbed by a gas substance and transformed into kinetic energy (by the energy exchange process). Normally, interruption of this process produces a series of pressure waves (sound) that can be detected by microphones. By measuring the sound at different wavelengths, the photoacoustic spectrum of a gas sample can be recorded, see Figure 2.6 and 2.7. This spectrum can then be used to identify components of the sample.

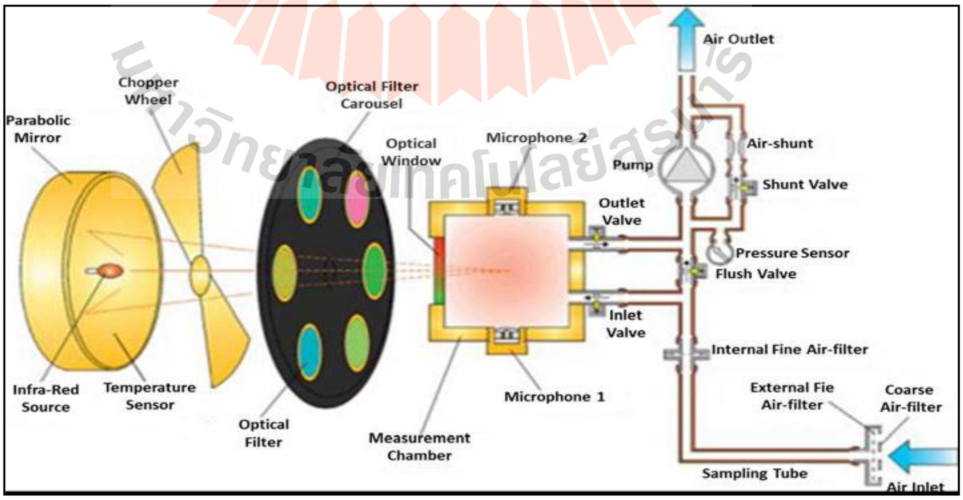


Figure 2.6 The arrangement of PAS based DGA system (Donal, 2010).

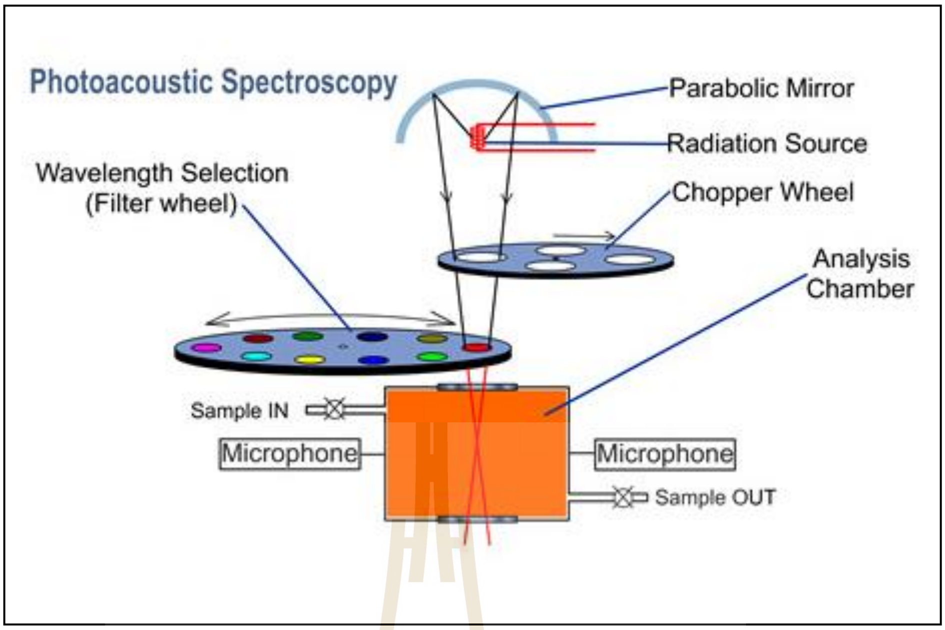


Figure 2.7 The absorption characteristic of fault gasses (Donal, 2010)

When the target gasses are specified like in the DGA application, infrared filters can be applied to select regions of the infrared spectrum that overlap with the target gas absorption spectrum. Broadband infrared is normally generated by a black body infrared source (Donal, 2010). With this method, a variety of gasses can be detected. Various researchers have discussed photoacoustic spectroscopy as a very stable diagnostic technique and useful for monitoring critical transformers. Since, each dissolved gas absorbs the infrared light at a specific wavelength, selecting the center wavelength is a critical process. Incorrect center wavelength will cause mutual interference between gasses involved.

2.4 Diagnostic methods

Different organizations have established variety diagnostic methods for interpreting the levels of dissolved gasses. Among of them are IEEE, IEC, CIGRE, ASTM etc. According to IEEE C57.104-2008 fault gasses produced during abnormal condition are Hydrogen H_2, Carbon monoxide CO, Carbon dioxide CO_2, Methane CH_4, Ethylene C_2H_4, and Ethane C_2H_6. Under IEEE and IEC, the following diagnostic methods are used for interpretation of dissolved gasses:

2.4.1 Key gas method

The presence of fault gasses within transformer oil depends on the temperature or energy that breaks the chemical bonding of the insulating oil and hence reduce its dielectric strength. The key gas method considers the individual concentration of gasses produced during faults. Key gas method interprets DGA results by referring to the four common transformer faults, which are partial discharge (PD), arcing, overheated of oil and overheated of cellulose. The significant gasses involved with each fault are shown in Figure 2.8. As per IEEE Standard C57.104-2008, H_2 is the key gas for PD, C_2H_2 for arcing and C_2H_4 for overheated oil and CO for overheated cellulose.

2.4.2 Dornenburg ratio method

The Dornenburg method suggests the existence of three general faults (thermal faults, electrical fault-low discharge, and electrical fault-high discharge) (IEEE C57.104-2008). This procedure requires significant levels of the gasses to the present in order for the diagnosis to be valid. It utilizes four calculated gas ratios to indicate a single fault type from three general fault types. Moreover, it uses five

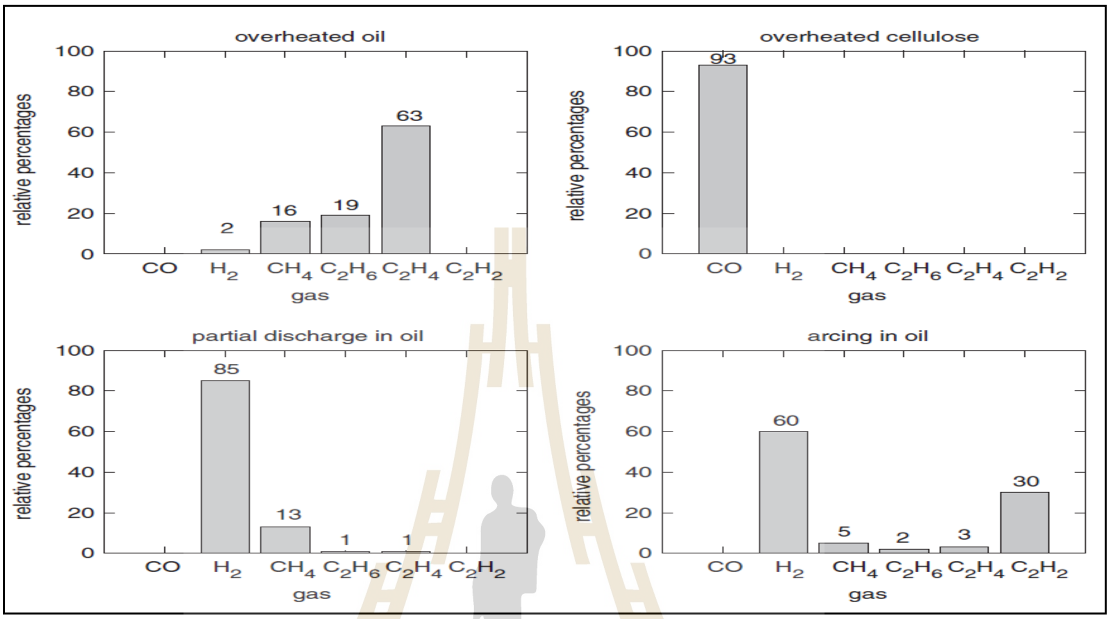


Figure 2.8 Evaluation of key gasses method (IEEE C57.104-2008)

individual gasses H_2, CH_4, C_2H_2, C_2H_4, and C_2H_6 or four-key gas ratios, which are Ratio 1 (R1), Ratio 2 (R2), Ratio 3 (R3) and Ratio 4 (R4). The values for these gasses are first compared to special concentrations—L1 in Table 2.1 below to ascertain the need for the test.

Table 2.1 Limit concentrations of dissolved gas (IEEE C57.104-2008).

Key gas	**Concentrations L1** [µl / l (ppm)]
Hydrogen (H_2)	100
Methane (CH_4)	120
Carbon monoxide (CO)	350
Acetylene (C_2H_2)	1
Ethylene (C_2H_4)	50
Ethane (C_2H_6)	65

2.4.3 Roger's ratio method

This method uses the 4-digit ratio code generated from the 5 fault gasses (H_2, CH_4, C_2H_6, C_2H_4, and C_2H_2) to determine 15 diagnosis rules for transformer conditions (Ali Saeed, 2012). According to Sherif, Ghoneim, and Sayed (2012) Roger's ratio method uses four gas ratios which are CH_4/H_2, C_2H_6/CH_4, C_2H_4/C_2H_6 and C_2H_2/C_2H_4 for diagnosis. In the refined Roger's method, the ratio C_2H_6/CH_4 only indicates a limited temperature range of decomposition but does not assist in further identification of a fault. Roger's ratio method and IEC 599 have gained popularity in industrial practices. However, it may give no conclusion in some cases.

2.4.4 IEC ratio method

This method originated from Roger's ratio method, except that the ratio C_2H_6/CH_4 was dropped since it only indicated a limited temperature range of decomposition (Ali Saeed, 2012). IEC analysis uses an advanced scenario of fault matrix where faults were classified as 7 different types partial discharge, low energy discharge, high energy discharge, thermal fault temperature <300 °C, 300 to 700 °C, and temperature > 700 °C.

2.4.5 Duval triangle

Duval triangle was first developed in 1974. It uses three hydrocarbon gasses only CH_4, C_2H_4 and C_2H_2. These three gasses correspond to the increasing levels of energy necessary to generate gasses in transformers (Duval, 2005). C_2H_2 and C_2H_4 are used in all interpretation methods to represent high energy faults such as arcs and high-temperature faults. H_2 is preferred in several of these methods to represent very low energy faults such as PDs, where it is produced in large quantities. CH_4, however, is also representative of such faults and always formed in addition to H_2 in these faults, in smaller but still large enough amounts to be quantified. CH_4 has been chosen for the triangle because it not only allows to identify these faults but provides better overall diagnosis results for all the other types of faults than when using H_2 (Duval, 2005).

2.5 Transformer conditions

The health of the oil-filled transformer insulation can be easily monitored by monitoring the levels of dissolved gas-in oil. The volume of dissolved gas-in-oil shows the magnitude of the incipient fault. From the estimated magnitude of the incipient fault, one can plan for maintenance activity or asset management. Most of the dissolved gases taken into account are Hydrogen H_2, Carbon monoxide CO, Carbon dioxide CO_2, Methane CH_4, Ethylene C_2H_4, and Ethane C_2H_6. The amount of dissolved gases in oil are computed as shown in (2.4) (IEEE C57.104-2008).

$$TDCGV = FG(V) / 1000000 \tag{2.4}$$

Where

FG = sum of H_2, CO, CH_4, C_2H_4, C_2H_2 and C_2H_6 in parts per million (ppm)

V = volume of oil in the transformer (liters)

$TDCG_V$ = total dissolved combustible gas volume (liters)

This method is used for completely oil-filled power transformer with a conservator tank. The usefulness of this method is encountered in the presence of significant volume of dissolved gas-in-oil. The results of the volume of total combustible dissolved gas measured can be used to categorize the condition of the transformer insulation and hence the life of the transformer. According to IEEE C57.104-2008 four conditions have been established to define the health of the transformer and action(s) to be taken. The following are the four conditions based on the total combustible dissolved gases as shown in Table 2.2.

Table 2.2 Dissolved gas concentrations (IEEE C57.104-2008).

Condition	**Dissolved key gas concentrations limits (ppm)**							
	H_2	**CH_4**	**C_2H_2**	**C_2H_4**	**C_2H_6**	**CO**	**CO_2**	**TDCG**
1	100	120	1	50	65	350	2500	720
2	101-701	121-400	2-9	51-100	66-100	351-570	2500-4000	721-1920
3	701-1800	401-1000	10-35	101-200	101-150	571-1400	4001-10000	1921-4630
4	>1800	>1000	>35	>200	>150	>1400	>10000	>4630

From Table 2.2, the levels of dissolved gas-in-oil are given in parts per million (ppm). This Table 2.2 assumes no previous DGA history of the transformer have been made or the transformer has not been reconditioned. It considers the use to large power transformers with large volume of insulating oil (several thousands of gallons) oil because with the small volume of insulating oil in small transformer it can give higher levels of dissolved gas-in-oil (IEEE C57.104-2008). Also, Table 2.2 does not find application in small distribution transformers, voltage regulators and other apparatus with load breaker switch operate under the oil. This is due to the generation of non-fault dissolved-gas-oil resulted from the operation of internal expulsion switches or load breaker switches. CO_2 is not considered in the volume of total combustible gas since it is a combustible gas. The interpretations of the conditions in Table 2.2 are given in Table 2.3.

Table 2.3 Transformer conditions (IEEE C57.104-2008).

Condition	**TDCG Level (μl/l)**	**TDCG rate (μl/l/day)**	**Sampling interval and operation procedures for gas generation rates**	
			Sampling intervals	**Operating procedures**
4	>4630	>30	Daily	Consider removal from service
		10 – 30	Daily	Advice manufacturer
		<10	Weekly	Exercise extreme caution Analyze for individual gases Plan outage Advice manufacturer
3	1921-4630	>30	Weekly	Exercise extreme caution Analyze for individual gases Plan outage Advice manufacturer
		10 – 30	Weekly	
		<10	Monthly	
2	721-1921	>30	Monthly	Exercise caution Analyze for individual gases Determine load dependence
		10 – 30	Monthly	
		<10	Quarterly	
1	≤ 720	>30	Monthly	Exercise caution Analyze for individual gases Determine load dependence
		10 – 30	Quarterly	Continue normal operation
		<10	Annual	

The four groups of transformer conditions have been made to classify the risk of transformer in order to allow a safe continued operation of the transformer. Below are the descriptions about transformer conditions:

- **Condition 1**: At this condition the volume of dissolved gas-in-oil is less than or equals to 720 ppm. This condition indicates that the transformer is operating satisfactorily. Also, this means the health of transformer insulation and can operate safely over a sufficient long service time.
- **Condition 2**: At this condition the volume of dissolved gas-in-oil is between 720 and 1920 ppm which is above normal combustible gas level. For any individual dissolved which is over the specified level shown in Table 2.2 should prompt for additional investigation. From the levels obtained, resample is strongly advised while the sampling period is determined with the rate of gas generation as shown in (5). The value of gas generation determines the sampling intervals as defined in Table 2.2.
- **Condition 3**: At this condition the volume of dissolved gas-in-oil is between 1921 and 4630 ppm which indicates a high level of decomposition above normal combustible gas level. For any individual dissolved which is over the specified level shown in Table 2.2 should prompt for additional investigation. From the levels obtained, resample is strongly advised while the sampling period is determined with the rate of gas generation as shown in (5). The value of gas generation determines the sampling intervals as defined in Table 2.2.In addition, the gas trend (ppm versus time) should be done.

- **Condition 4**: At this condition the volume of dissolved gas-in-oil is above 4630 ppm which indicates an excessive decomposition of combustible gases. A continued operation could result into a failure. For any individual dissolved which is over the specified level shown in Table 2.2 should prompt for additional investigation. From the levels obtained, resample is strongly advised while the sampling period is determined with the rate of gas generation as shown in (2.5). The value of gas generation determines the sampling intervals as defined in Table 2.2.

$$R = (S_T - S_0) \times V \times 10^{-6} / T \tag{2.5}$$

Where

R = rate per day (liters/day)

S_0 = first sample (ppm)

S_T = second sample (ppm)

V = volume of oil in the transformer (liters)

T = time (days)

2.6 DGA based software for transformer condition monitoring

In past few decades, researchers have been working to understand the knowledge of traditional diagnostic methods like key gas, ratio methods (Dornenburg, Roger's and IEC ratio methods), Duval triangle, etc. and utilize them to diagnose the internal health of transformer. From previous researchers, not only diagnoses were aimed but also the knowledge about performances and limitations of traditional methods. Among the limitations studied from traditional methods includes accuracies of the algorithm employed in the individual diagnostic method. Each diagnostic method was seen to have its accuracy in interpreting DGA data due to the different algorithm used by each diagnostic method. Since the accuracy of traditional diagnostic methods varies, diagnosis of DGA data should not rely on the results from one diagnostic method. For many years diagnoses of transformers involving traditional methods have been performed graphically; which consumes time and reduce accuracy due to human errors when diagnosing a large number of transformers. Recently, researchers have been developing DGA computer software to simplify diagnoses of transformer condition. In these software, one or more individual diagnostic methods are incorporated to perform interpretations. Advantages of DGA software encompass time-saving, minimized human and graphical errors; and easy assessment of diagnostic method. Some of the developed DGA software are described below:

Sukbirr and Bandyopadhyay, (2010) presented a Duval triangle: A Noble technique for DGA in Power Transformers. They described the use of MATLAB to implement Duval triangle method as DGA computer software. Graphical formulation of major gasses, their ratios, computations, and procedures on how to use the Duval triangle were given. The study also discussed two methods mainly used for computation in Duval triangle i.e. (i) by using total accumulated gasses and (ii) by using total increase between conjugate samples. In their work, the method of using total accumulated gas was used on graphical presentation of the Duval triangle. The triangle, as well as fault zones, were taken in Cartesian coordinates when plotted in MATLAB. The similar DGA software was developed by Akbar, Setayeshmehr, Borsi, and Gockenbach, (2008) based on Duval triangle except that they use Java programming language. Also, all six fault zones described in Duval triangle were defined in Cartesians through Java language. The software displayed the ability to identify DGA fault within fault zones, and furthermore the ability to identify faults that are falling between fault zones.

Ali Saeed, Nor Asia and Abubakar, (2012); and Abubakar, (2013) demonstrated computer based software which combine four DGA diagnostic methods; Roger's ratio method, IEC ratio method, Duval triangle method, and key gas method. The user interface was designed by using Microsoft Visual C# (programming language) in Microsoft Windows XP/7 (operating system). Each fault was represented by a fault code (*F1–F6*). The process of the fault decision in this computer- based software was based on a general percentage mechanism, where each of the methods had the same percentage of accurate prediction.

Abubakar, Nor Asia and Nouruddeen, (2013); introduced the hybrid DGA interpretation software as an effective power transformer management tool which used four DGA diagnostic methods such as Roger's ratio method, IEC ratio method, Duval triangle method, and key gas method. The software was programmed by using

C# programming language in Windows 7. In the interface of the software, the user can input DGA information and then the software can diagnose the same DGA data through four diagnostic methods. It has no option for user to choose the diagnostic method at any time. The DGA software was validated for accuracy based on the predicted cases A_p. The number of predictions from each diagnostic method was obtained from 117 transformer samples available in TNB network in Malaysia. Transformers were rated between 300 kVA to 1000 kVA insulated by Hyrax Hypertrans mineral insulating oil. The software was able to predict 108 out of 117 cases which is equals to 92.31%. Quick performance of the developed software was observed through results four diagnostic methods provided at once.

Sherif and Sayed, (2012) presented a software codes with the use of logic functions to identify the type of existing fault in oil immersed power transformer. The software employed expert method based dissolved gas-in-oil for performing diagnoses. It was made to give the cause of the transformer fault from software through expert system. Classical methods such as key gas method, Dornenburg ratio method, Roger’s ratio method, IEC ratio method and Duval triangle method were programmed through Microsoft Excel to first identify the existing fault from the given dissolved gas-in-oil. The real transformer cases and other researchers’ cases were obtained for testing the performance of the proposed diagnostic technique. From the comparison made between the software made, laboratory data and from the literatures, the software showed the close agreement in performance.

Hell, D'Angelo and Costa Jr, presented another useful methodology which was used for performing diagnosis in oil-filled power transformer. This methodology was based on fuzzy neural approach technique. The training of the approach used the real gases from the database available in Energitic Company of Minas Gerais (CEMIG), Brazil. In their development, they employ the use of Roger's ratio method to initialize the architecture of the membership function of the fuzzy subsets in the Fuzzy Neural Network system. The developed system had only one Fuzzy Neural Network due to the fact that Roger's ratio method and other DGA techniques like key gas method, Dornenburg ratio method, IEC ratio method and Duval triangle method has the capability to detect only one internal activity inside the transformer oil. Two layers of their developed system included fuzzy layer and output layer.

Indra, Rajaram, and Gnanasaravanan, (2000), developed power transformer faults identification using fuzzy based dissolved gas analysis method. The approach involved fuzzy Roger's system for detecting fault gases obtained from the dissolved gas analysis of oil-filled transformers. It involves the use of ratios computed based on Roger's ratio method to predict the condition of the transformer. The ratios were taken as input parameters whereas the interpretations based on the combination of ratios were taken as output parameters. The analysis involved the use of five dissolved gases such as ethylene, methane, acetylene, hydrogen and ethane for predicting the condition of the transformer. For testing the proposed system, 551 samples of insulating oil were taken Electricity board of Tamilnadu, India. According to the investigations made, they found that out of 551 samples collected from different substations and tested for dissolved gas-in-oil analysis in the laboratory of Electricity board test lab, 329 samples were found to have no fault while 137 were recommended for resampling. Therefore, 137 samples were taken and tested with Roger's ratio method and fuzzy Roger's ratio system in order to compare the performance between the two methods. The fuzzy Roger's ratio system resulted into 91 percentage of successful prediction compared to 77 percent of Roger's ratio method.

Dhote and Helonde, (2012), performed diagnosis of power transformer faults based on five fuzzy ratio methods. The system was developed to overcome the drawbacks of traditional DGA methods like key gas method, Dornenburg ratio method, Roger's ratio method, IEC ratio method and Duval triangle to detect only one internal activity. In real practice each traditional method has its own algorithm and diagnostic criteria, therefore, when the detection is out the criteria an error occurs or results into no detection. The system was developed in MATLAB programming software. The condition for the diagnosis was based on the rate of evolution of total combustible gases. For a transformer which was not degassed, an evolution rate of 2.8 liters/day was taken as a normal condition and no diagnosis was performed. For evolution rate above 2.8 liters/ day, dissolved gas analysis was performed to check if gases are above the permissible limits for performing diagnosis. The DGA techniques were applied if the dissolved gases were above the limits to diagnose the transformer fault. The proposed system was used if all DGA techniques give different diagnoses. This was due to the expectation of existence of multiple faults. The developed system was tested with 100 transformer cases from power companies CPRI, BHEL and NTPC, India. The system showed better performance with increased predictions compared to traditional DGA techniques.

Taha, Zaini, and Ghoneim, (2015), presented a comparative study between Dornenburg and Roger's ratio methods for transformer fault diagnosis based on Dissolved Gas Analysis using MATLAB Simulink tools. It was built according to the most traditional Dornenburg and Roger's ratio method for studying their performance and accuracy. The Simulink models were used for constructing traditional Dornenburg and Roger's ratio method and they were based on the interpretation method in IEEE standard. In Simulink, tools like relational operator, constant, multiplex scalar, vector signal (Mux), sink, source and switches were used for implementation. The input datasets to the model were applied through MATLAB M-files. The input dataset which were DGA data, were taken from other researchers and were considered as accurate data for reference. The total number of 77 cases were used for testing the developed model whereby Roger's showed 79 percent of agreement compared to Dornenburg with 52 percent.

2.6.1 Software language

Several programming languages have been used to program diagnostic methods for interpreting DGA data. The applicability of a certain language depends on its effectiveness, efficiency, ability to interact with the real world and cost in solving a particular problem. Among of these languages, Python has chosen for programming this DGA software.

Python was found in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica, Netherland (Python, 2015). It is a general purpose, open source computer programming language. It is optimized for software quality, developer productivity, program portability and component integration. It supports a remarkably simple, readable, and maintainable syntax: integration with external components coded in other languages (Lutz, 2010). According to Lutz, (2010) the following have been stated as the areas where python programming language can be used:

- It is used all over the world for Internet scripting, systems programming, user interfaces, product customization, numeric programming, etc. with applications domains in the range from system administration,
- website design, cell phone programming, and education to hardware testing,
- investment analysis, computer games and spacecraft control.

Python differs somehow with other programming languages like Java, C, C^{++}, Microsoft Visual Basics, Microsoft C# etc. It offers the following advantages:

- It provides good code readability – uses whitespace indentation to delimit code blocks instead of using curly braces,
- It allows programmers to express concepts in fewer lines compared to other languages,
- It supports multiple programming paradigms including object-oriented, imperative, functional programming and procedural styles,
- It has large and comprehensive standard library,
- Its interpreters are available in many operating systems (cross platform),
- It has cycle detecting garbage collector for memory management
- It has fewer syntactic exceptions and special cases compared to other programming languages like C and Pascal (Python, 2015)

Python comes with varieties of versions from Python 2.x to Python 3.x. The first release was Python 2.0 and it was released on 2000. The mostly used version with compatibility to large number of libraries is Python version 2.7. The End of Life of Python 2.7 was first set to 2015 but later was postponed to 2020 (Python, 2015).

2.6.2 Database language

According to Dyer, (2005) MySQL is an open source, multi-threaded, relational database management system created by Michael “Monty” Widenius in 1995. In 2000, MySQL was released under a dual license model that permitted the public to use it for free under the GNU Public License (GPL); this caused its popularity to soar. MySQLwas considered as a leading database because of its reliability, performance, and features. Many features contribute to MySQL’s standing as a superb database system; speed is one of its most prominent features and the storage engine which manages queries and interfaces between a user’s SQL commands the database’s backend storage.

2.7 Chapter summary

This chapter has presented the fundamentals of mineral insulating oil including their composition, properties and functions. The study of aging mechanism of mineral insulating oil have also presented in detail. The two major techniques used for performing dissolved gas-in-oil analysis have been explained. Four conditions resulted from the existing faults inside the liquid insulation have been described. Finally, the recent works concerning development of DGA based software as well as the features of Python and MySQL programming languages are mentioned.

CHAPTER 3

DEVELOPMENT OF DGA COMPUTER BASED SOFTWARE

3.1 Introduction

This DGA computer based software was programmed and compiled on Windows 7 to allow it to be used on Windows Operating System as a majority computer system. Its inputs were eight dissolved gasses and other non-gaseous data which are displayed in the input form before being saved into the database. The database was programmed by using MySQL 5.5 (database programming language). MySQL is the open source software released under a dual license to allow it to be used with the public under Public License. The open source software, speed, storage and computational capability made it the best choice for storage and execution of DGA data (Dyer, 2008).

3.2 Installation of the Python installer.

Since python is an open source, it is easily available through online searching. It can be downloaded through Google search engine by using the official python website. All python versions from 2.*x* to 3.*x* are available in the official python website.

3.2.1 The official website of the python language

The official website of python programming language is https://www.python.org. Through Google search engine user can reach the official website of python programming language. Figure 3.1 shows the snapshot of the official website with the address in red dashed-box.

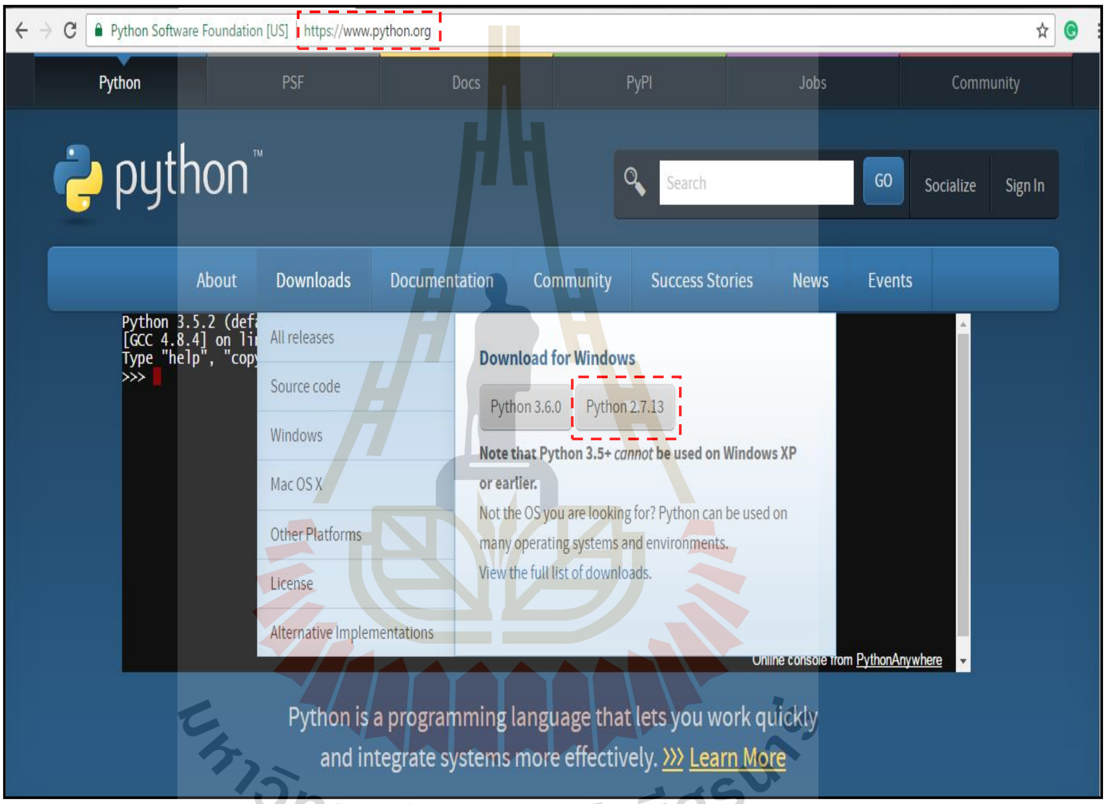


Figure 3.1 Snapshot of the python official website.

User has to select the python installer that suits its operating system such as Windows operating system, Linux operating system, Mac operating system, etc. From the python official website as shown in the snapshot of Figure 3.1, for Windows operating system user has to click directly on the python version that he want. Figure 3.1 shows python 3.6.0 and python 2.7.13. Once any version is clicked it will be downloaded into the download folder of the respective computer. The version 3.6.0 is the current or an updated version of python language but current versions lack some of the important libraries (not all libraries have been updated into current versions).

3.2.2 Python installation files

Figure 3.2 shows python installation files for Windows operating system. The python installer with names Windows×86 are installers for Windows operating systems with 32 bit system type while Windows×86-64 for 64 system type.

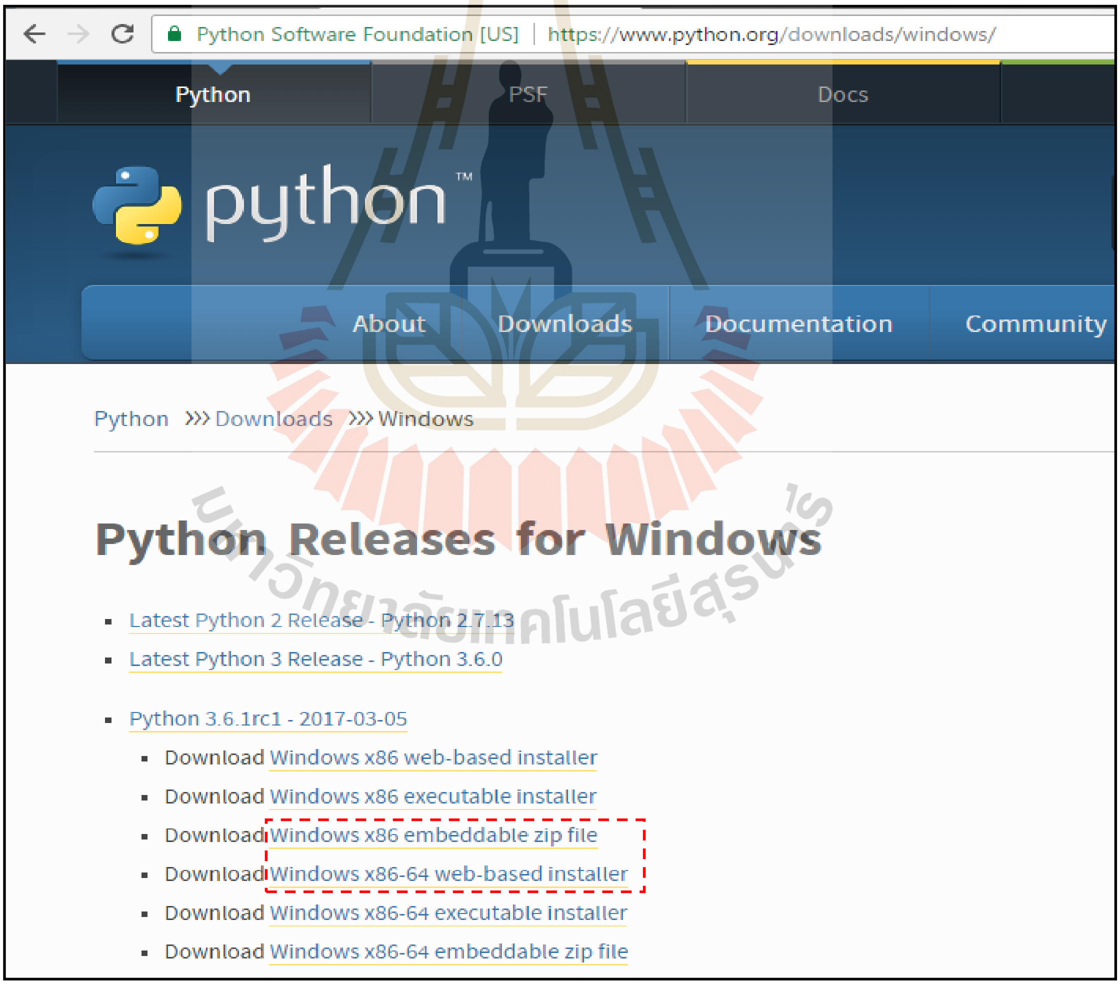


Figure 3.2 Python installer files for Windows operating system

3.2.3 Installing Python file

When the downloaded file with .msi extension is opened, the installation proceeds as shown from the snapshots of Figure 3.3 (a) to (d).

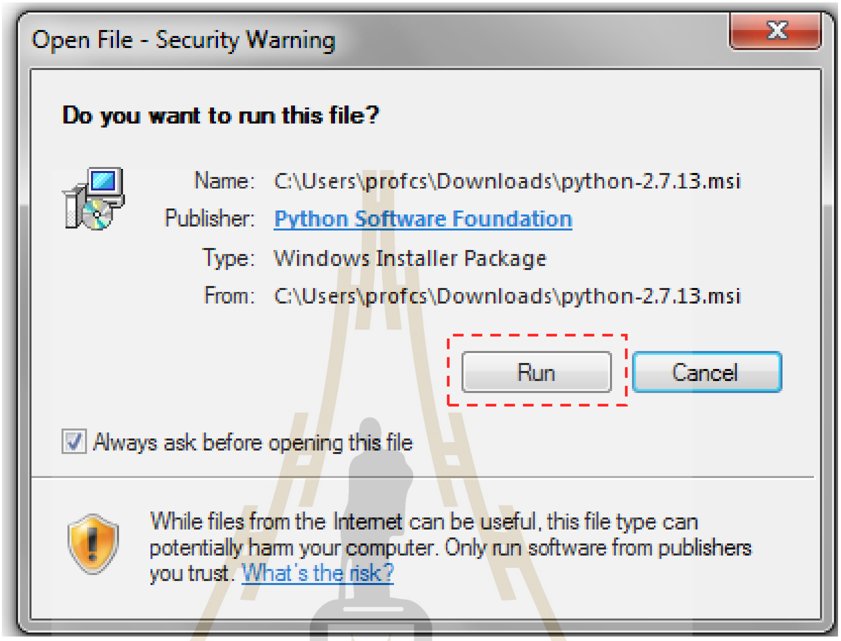


Figure 3.3 (a) running the downloaded file with .msi extension

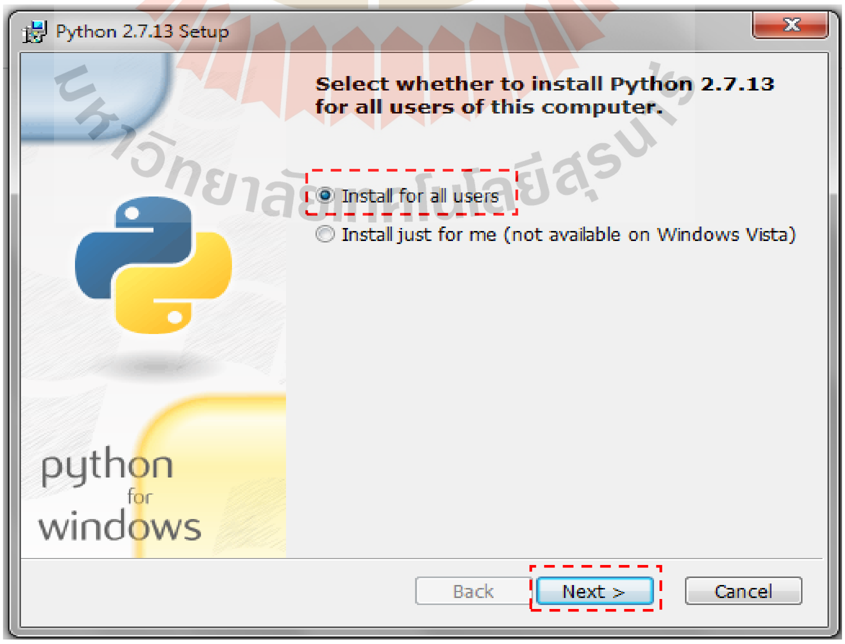


Figure 3.3 (b) choosing installation mode

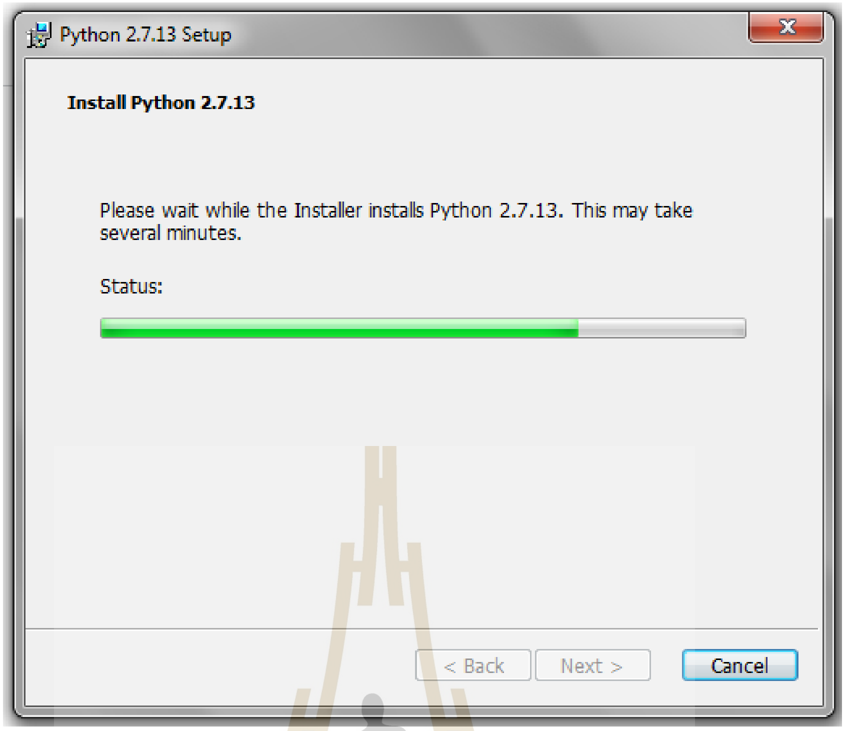


Figure 3.3 (c) Extracting compressed files

Figure 3.3 (d) Finishing installation process

3.2.4 Opening Python software

Once finished to be installed, python software can be searched through computer search option as shown in Figure 3.4 (a). Programming codes are written in IDLE (Python GUI) after being clicked from shown snapshot. When IDLE is clicked it opens the python shell where user can choose for a new python editor to write the new codes. The python shell is shown in Figure 3.4 (b). The new python editor was opened shown in Figure 3.4 (c).

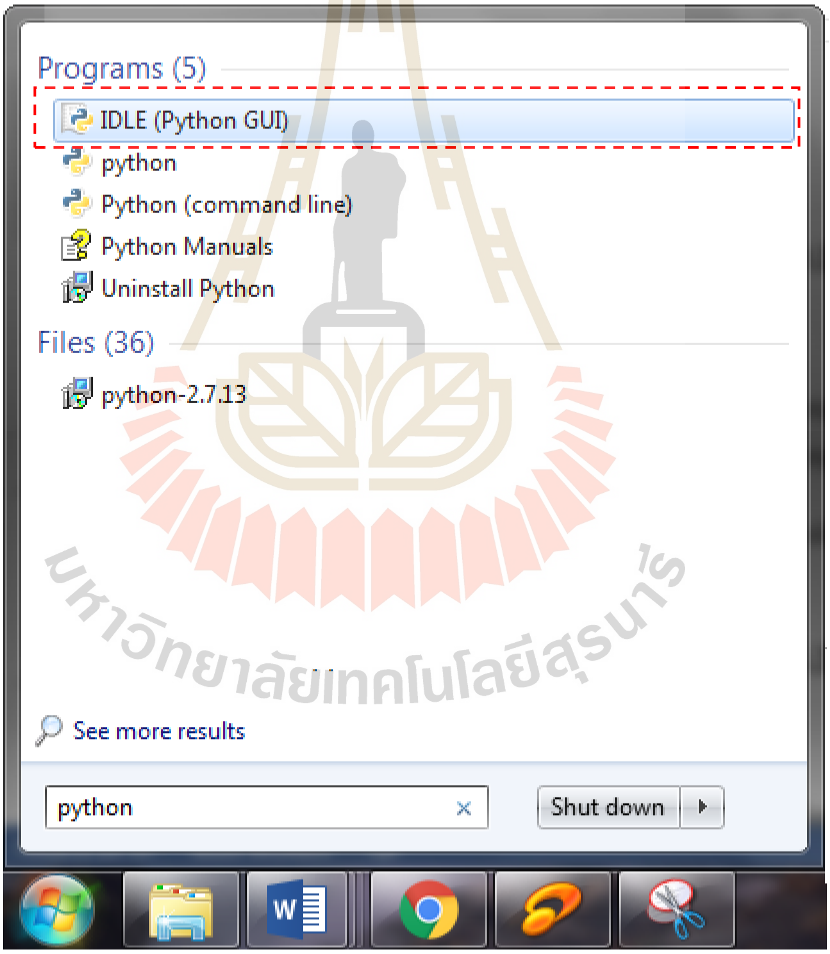


Figure 3.4 (a) Opening python software

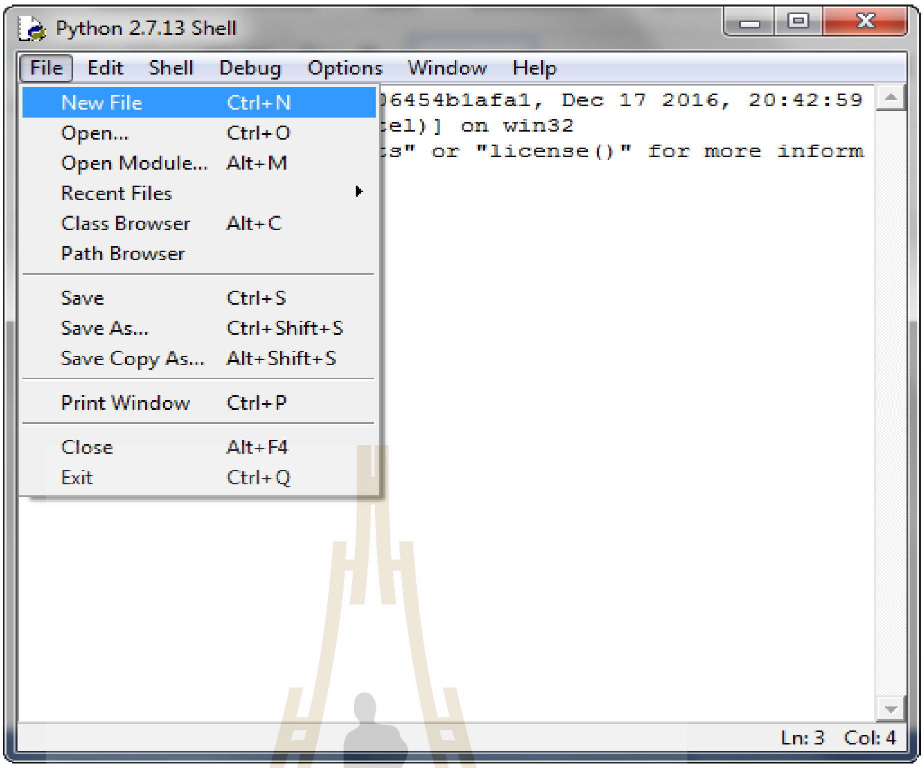


Figure 3.4 (b) Python shell

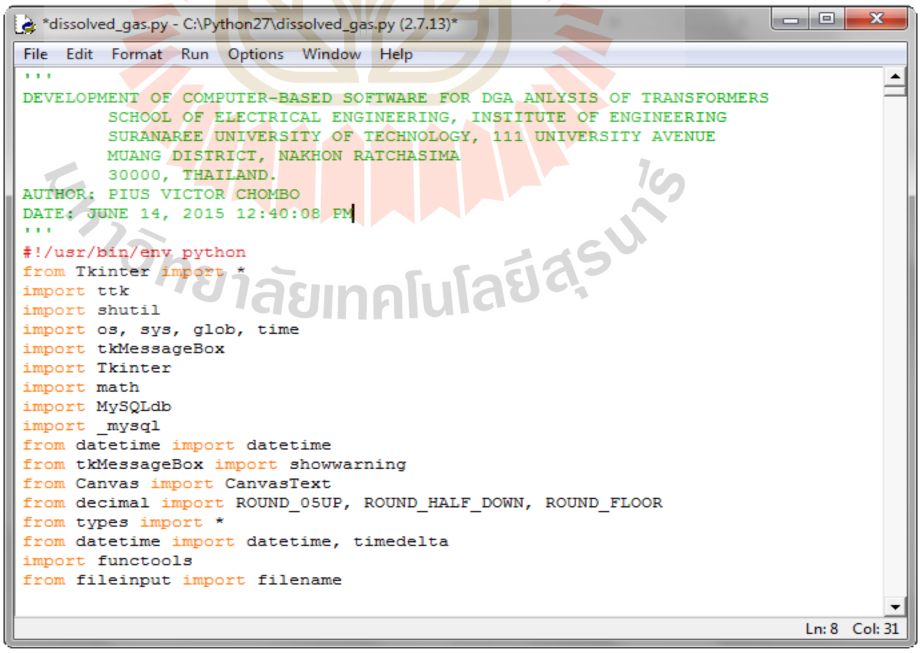


Figure 3.4 (c) Python script file

3.2.5 Python libraries

Python imports useful libraries for performing required tasks. The major library is Tkinter. Libraries are added into the script by import command as shown in Figure 3.4 (c). Below is the list of libraries used in python programming language.

- Requests()
- Scrapy()
- wxPython()
- Pillow()
- SQLAlchemy()
- BeautifulSoup()
- Twisted()
- NumPy()
- SciPy()
- Matplotlib()
- Pygame()
- Pyglet()
- pyQT()
- pyGtk()

3.3 Developed DGA software.

3.3.1 Flowchart of the developed DGA software

Figure 3.5 shows the flowchart of the DGA software with five diagnostic methods, gas comparison, and gas trending options. Five diagnostic methods are key gas method, Roger's ratio method, Dornenburg ratio method, IEC ratio method and Duval triangle method. It also has 16 fields for user to input DGA information.

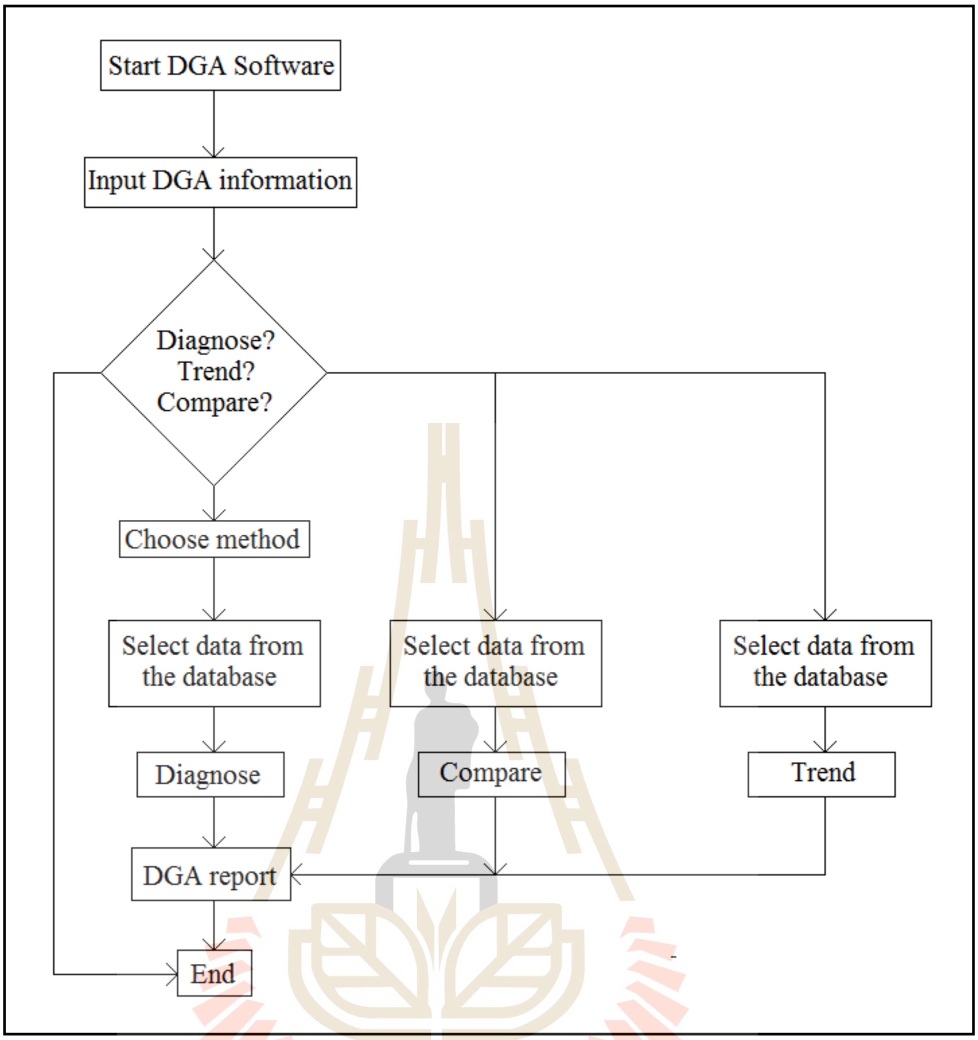


Figure 3.5 Flowchart for the DGA software.

The targeted inputs were dissolved gasses in parts per million (ppm) such as H_2, CO, CO_2, N_2, CH_4, C_2H_6, C_2H_4, and C_2H_2. Other non-gaseous data involved Power (kVA), Voltage (kV), oil temperature (°C), humidity (%), altitude (km), sampling date (date), transformer location (name of the location) and transformer number. Table 3.1 shows some of the tools used in the development of DGA software. Figure 3.6 shows the snapshot of the python codes used to develop the DGA interface. Figure 3.7 shows the developed interface of the DGA software.

Pius.py - C:/Users/profcs/Desktop/Pius.py (2.7.13)

File Edit Format Run Options Window Help

```
from Tkinter import *

fields1 = ["HYDROGEN  H2", "CARBON MONOXIDE CO", "CARBON DIOXIDE CO2", "NITROGEN N2", "METHANE CH4", '
entries = []
def makeform(root, fields1):
    for field in fields1:
        row = Frame(root)
        lab = Label(row, relief=RAISED, width=30, height=1, text=field)
        entt=StringVar()
        ent = Entry(row, textvariable=entt, relief=RAISED, width=32)
        row.pack(side=TOP, anchor=NW)
        lab.pack(side=LEFT)
        ent.pack(side=RIGHT, expand=NO)
        ent.delete(0, END)
        ent.insert(END, 'Input your value here')
        ent.focus()
        ent.bind('<Return>', (lambda event: fetch()))
        entries.append(entt)
    return entries

root=Tk()
root.minsize(800, 600)
root.maxsize(800, 600)

toolbar = Frame(root, bd=1, relief= FLAT)
but1 = Button(toolbar, text='KEY GAS')
but2 = Button(toolbar, text='DORNENBURG')
but3 = Button(toolbar, text='ROGERS')
but4 = Button(toolbar, text='IEC ')
but5 = Button(toolbar, text='DUVAL')
but6 = Button(toolbar, text='COMPARISON')
but7 = Button(toolbar, text='TRENDING')
but1.config(bd=1, width=15, font = ('times', 10, 'bold'))
but2.config(bd=1, width=15, font = ('times', 10, 'bold'))
but3.config(bd=1, width=15, font = ('times', 10, 'bold'))
but4.config(bd=1, width=15, font = ('times', 10, 'bold'))
but5.config(bd=1, width=15, font = ('times', 10, 'bold'))
but6.config(bd=1, width=15, font = ('times', 10, 'bold'))
but7.config(bd=1, width=15, font = ('times', 10, 'bold'))
but1.pack(side=LEFT, padx=1, pady=1)
but2.pack(side=LEFT, padx=1, pady=1)
but3.pack(side=LEFT, padx=1, pady=1)
but4.pack(side=LEFT, padx=1, pady=1)
but5.pack(side=LEFT, padx=1, pady=1)
but6.pack(side=LEFT, padx=1, pady=1)
but7.pack(side=LEFT, padx=1, pady=1)
toolbar.pack(side=TOP, fill=X)

root.title('DGA FOR POWER TRANSFORMERS')
widget = Label(root, text='DISSOLVED GAS-IN-OIL ANALYSIS FOR')
widget.config(fg='blue',font=('times', 15, 'bold'),height=1, width=20)
widget.pack(fill=BOTH)
widget = Label(root, text='POWER TRANSFORMERS')
widget.config(fg='blue', font=('times', 20, 'bold'), height=1, width=20)
widget.pack(side=TOP, fill=BOTH)
widget = Label(root, text='(C) SUT 2015 \n INSTITUTE OF ENGINEERING, SURANAREE UNIVERSITY OF TECHNOLOG
widget.config(fg='gray', font=('times', 10, 'bold'), height=3, width=20)
widget.pack(side=BOTTOM, fill=BOTH)
widget = Label(text='    DGA INPUT INFORMATION')
widget.config(bg='lightgreen', fg='dark green', font=('times', 12, 'bold'),height=1, width=25)
widget.pack(side=LEFT, anchor=N)
menu = Menu(root)
root.config(menu=menu)
filemenu = Menu(menu)
menu.add_cascade(label="File", menu=filemenu, underline=0)
viewmenu = Menu(menu)
menu.add_cascade(label="View", menu=viewmenu, underline=0)
helpmenu = Menu(menu)
menu.add_cascade(label="Help", menu=helpmenu, underline=0,)
ents = makeform(root, fields1)
root.bind('<Return>', (lambda event: fetch(ents)))
widget1 = Button(root, text='SAVE TO DATABASE')
widget1.pack(side=RIGHT, padx=180, pady=5, anchor=N)
widget1.config(bd=3, relief=RAISED,bg='gray', fg='white',font=('times', 12, 'bold'))
root.mainloop()
```

Ln: 38 Col: 18

Figure 3.6 Snapshot of the python codes used to develop the DGA interface

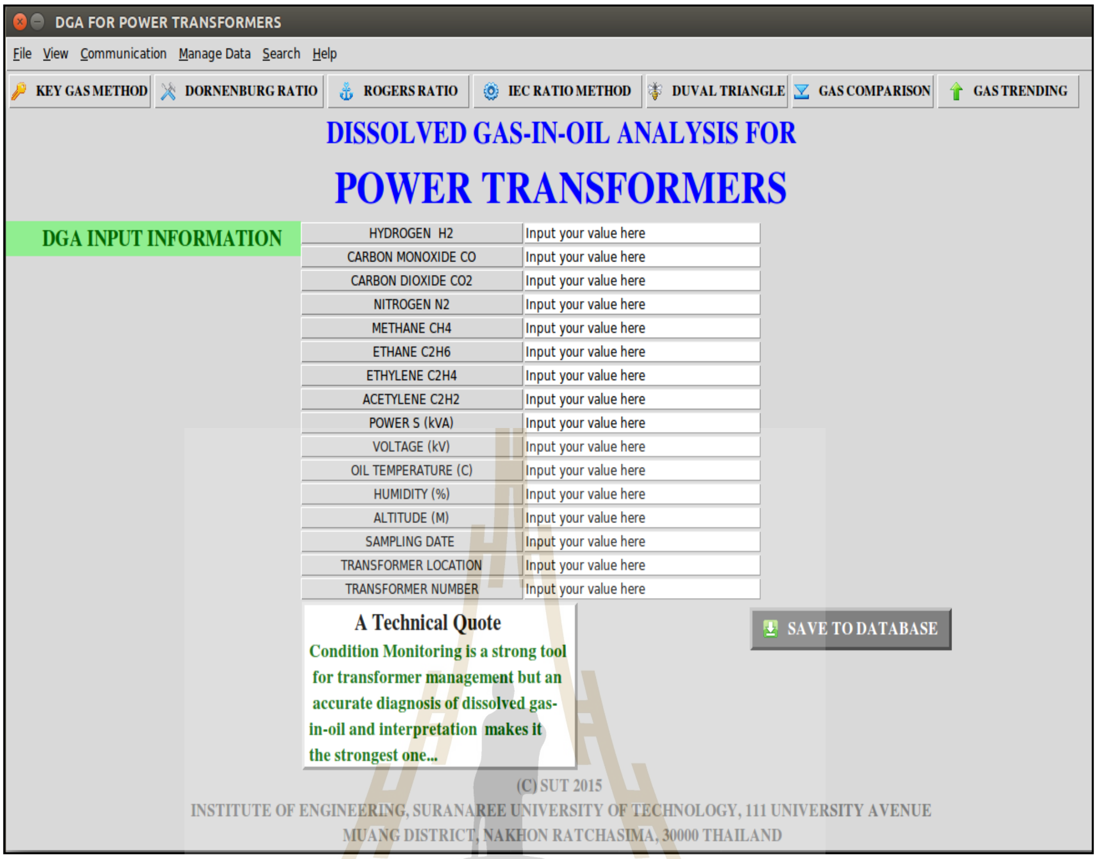


Figure 3.7 Interface of the developed DGA interface

Table 3.1 Tools used for the development of DGA-software

Tool	Type
Computer	HP 550 Duo Core
Operating System	Windows 7
Programming Language	Python 2.7
Compiler	Eclipse
Database	MySQL 5.5

3.3.2 Size and color of the interface

The interface took the same size of the computer regarding width and height. The width and height of the computer used were 1200 × 760 pixels. The gray color was selected to suit the background color.

3.3.3 Input form and details

The input form contained 15 fields of DGA information for the user to input; and a Save to Database button for saving data to the database.

3.3.4 Menu items

The menu items allow the user to access database to view DGA data, a summary of the report and other necessary information about DGA software.

3.3.5 Toolbar items

Toolbar items contained diagnostic methods, comparison and trending tools for gasses. The user can choose any item and through it, the user can also retrieve any DGA data from the database to perform the task.

3.3.6 Dropdown Lists from File Menu

The drop down menu from the File Menu is shown in Figure 3.8. The Open Menu opens Diagnosis summary of previously performed interpretation processes done by any of the five diagnostic methods. In Comparison summary, it opens a summary of comparison processes performed, and in Trending summary, it opens the summary of trending processes done. The Close App closes the DGA software.

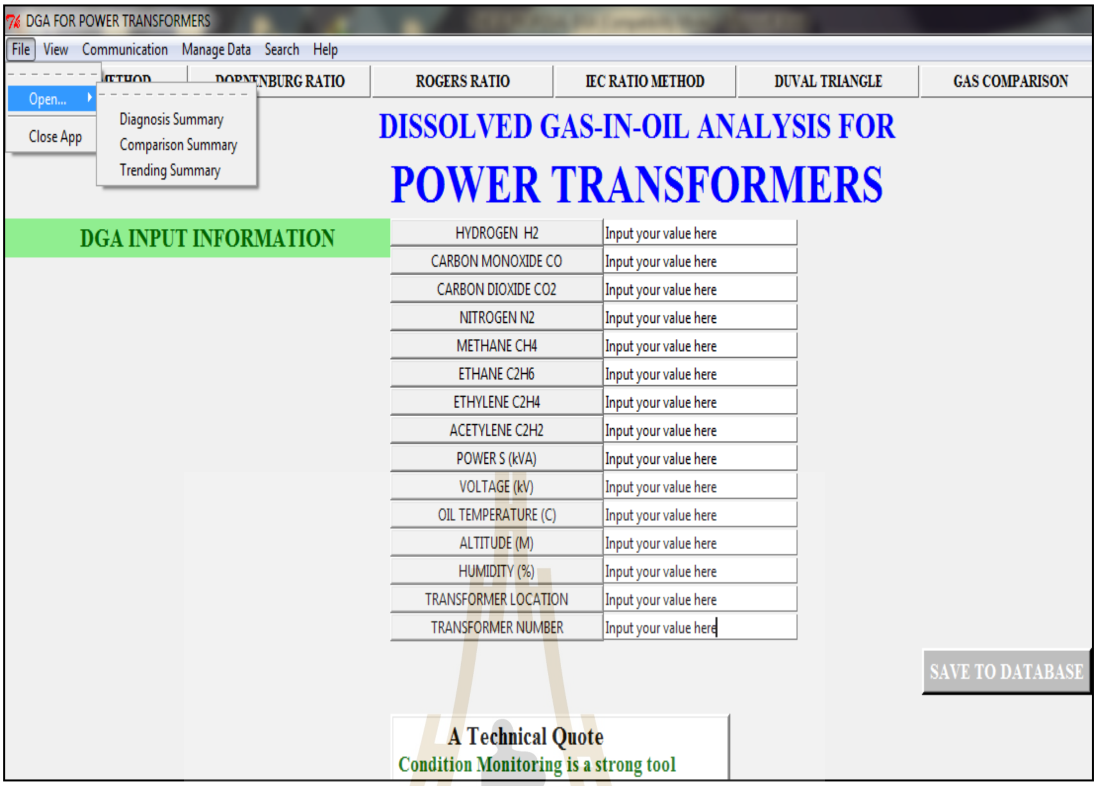


Figure 3.8 Drop-down lists in File Menu of DGA software.

3.3.7 Dropdown lists from View menu

Drop down lists from the View menu is shown in Figure 3.9. Database connection checks the existence of a connection between the database and DGA software whereas Last Entry and Database Entries retrieves the last stored entry and all entries respectively.

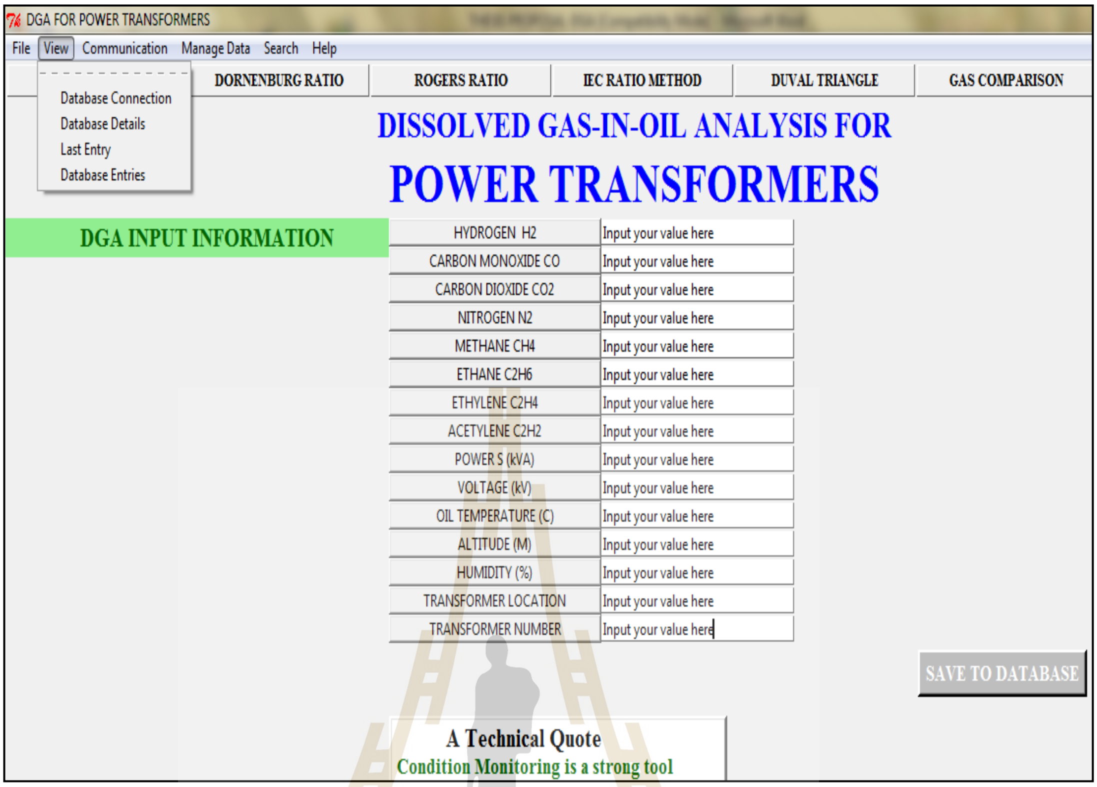


Figure 3.9. Drop-down lists in the View Menu.

3.4 Implementation of diagnostic methods

3.4.1 Key gas method (KGM)

This method is based on the proportions of principal gasses over Total Dissolved Combustible Gases (TDCG). The principal gasses are H_2, CO, CH_4, C_2H_2, C_2H_4, and C_2H_6. For interpreting the condition and then existing fault, the proportion of each principal gas is given in (3.1)-(3.7) as a reference to TDCG obtain in (3.1). Before performing diagnosis by using key gas method, the levels of dissolved gas are compared to the limited levels of dissolved gas-in-oil. These levels are shown in Table 3.2. When dissolved gas-in-oil exceed these limits, the diagnosis is suited to be performed. The faults corresponding to the obtained proportions are shown in Figure

$$TDCG = H_2 + CO + CH_4 + C_2H_2 + C_2H_4 + C_2H_6. \tag{3.1}$$

$$\% H_2 = (100 \times H_2) / TDCG \tag{3.2}$$

$$\% CO = (100 \times CO) / TDCG \tag{3.3}$$

$$\% CH_4 = (100 \times CH_4) / TDCG \tag{3.4}$$

$$\% C_2H_2 = (100 \times C_2H_2) / TDCG \tag{3.5}$$

$$\% C_2H_4 = (100 \times C_2H_4) / TDCG \tag{3.6}$$

$$\% C_2H_6 = (100 \times C_2H_6) / TDCG \tag{3.7}$$

Table 3.2 Limits of key gas (Reddy, 2014)

Gas	**H_2**	**CO**	**C_2H_4**	**C_2H_2**
Concentration	150	500	20	12

Table 3.3 Diagnostic criteria for key gas method

Fault	**Key gas**	**Criteria**	**Gas percent amount**
Arcing	C_2H_2	Mainly H_2 and C_2H_2	H_2: 60%,C_2H_2: 30%
Corona (PD)	H_2	Large amount of H_2	H_2: 85%, CH_4: 13%
Overheating of Oil	C_2H_4	Large amount of C_2H_4	C_2H_4: 63%, C_2H_6: 20%
Overheating of Cellulose	CO	Large amount of CO and CO_2.	CO: 90%

2.9 (see section 2.6.1). In Table 3.3, criteria used for diagnosing the fault are given. Figure 3.10 shows the flowchart for performing the diagnosis by using key gas method. In Appendix B python codes used for performing diagnosis by using key gas method in DGA software have been shown.

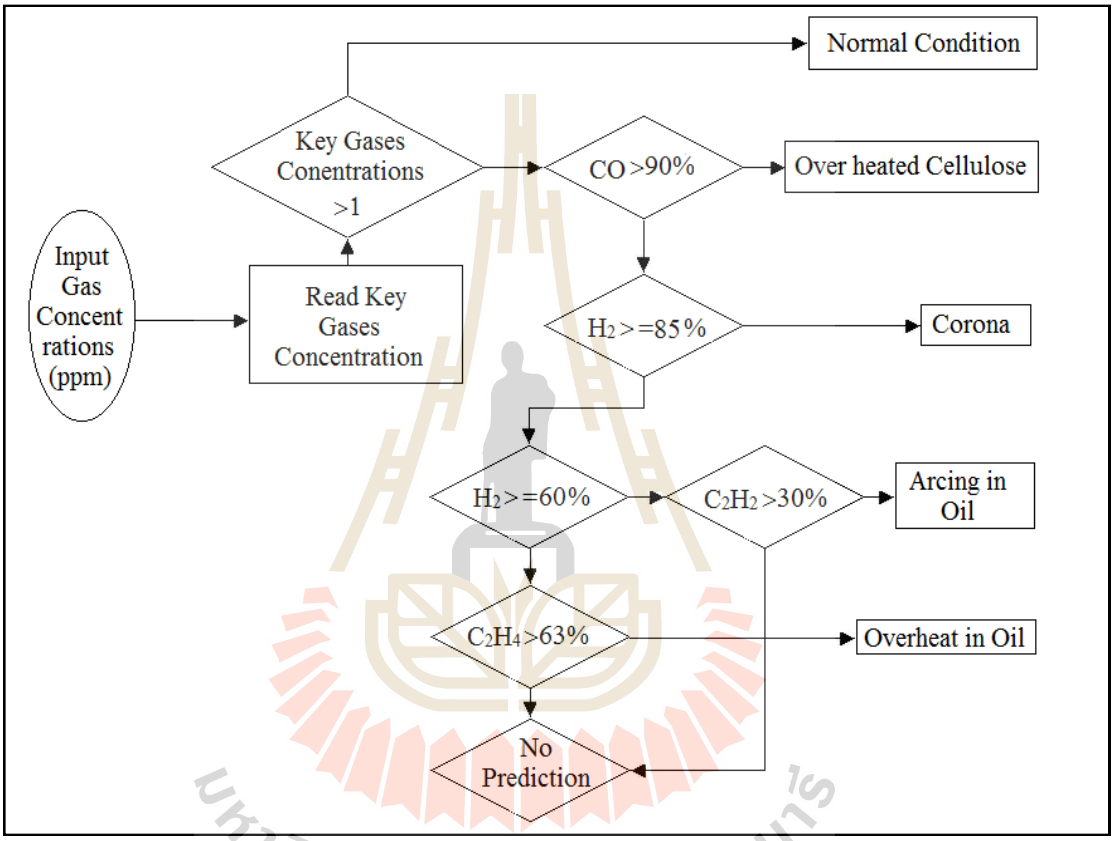


Figure 3.10 Flowchart of diagnosis by key gas method (Reddy, 2014).

In the script of python codes first the limits of key gas is checked by comparing with gas values in Table 3.2. If the conditions are satisfied for performing diagnosis, computation is done as per (3.1) to (3.7). The obtained results are then compared with the diagnostic criteria shown in Table 3.3 in order to identify the existing fault. Figure 3.10 shows the flowchart for performing the diagnosis by using key gas method.

Figure 3.11 shows the diagnosis done by the implemented key gas in DGA software. On top of the interface of Figure 3.11 is the combo box with ID NUMBER for user to select and retrieve the DGA information stored in the database. In the left side of the interface is the search button which retrieve the DGA data from the database with id number entered by user in the combo box. Below the combo box are the DGA data retrieved from the database. From the retrieved DGA data, id number and the dissolved gas-in-oil are displayed in ppm. Also, the computation of TDCG and the proportions of individual dissolved gasses are shown. Details about the current date and time, opening time and computing time (CPU time) are given in the right side of the interface. The remaining part of the interface shows the plots of proportions of the individual dissolved gas in the histogram and analysis of the existing faults.

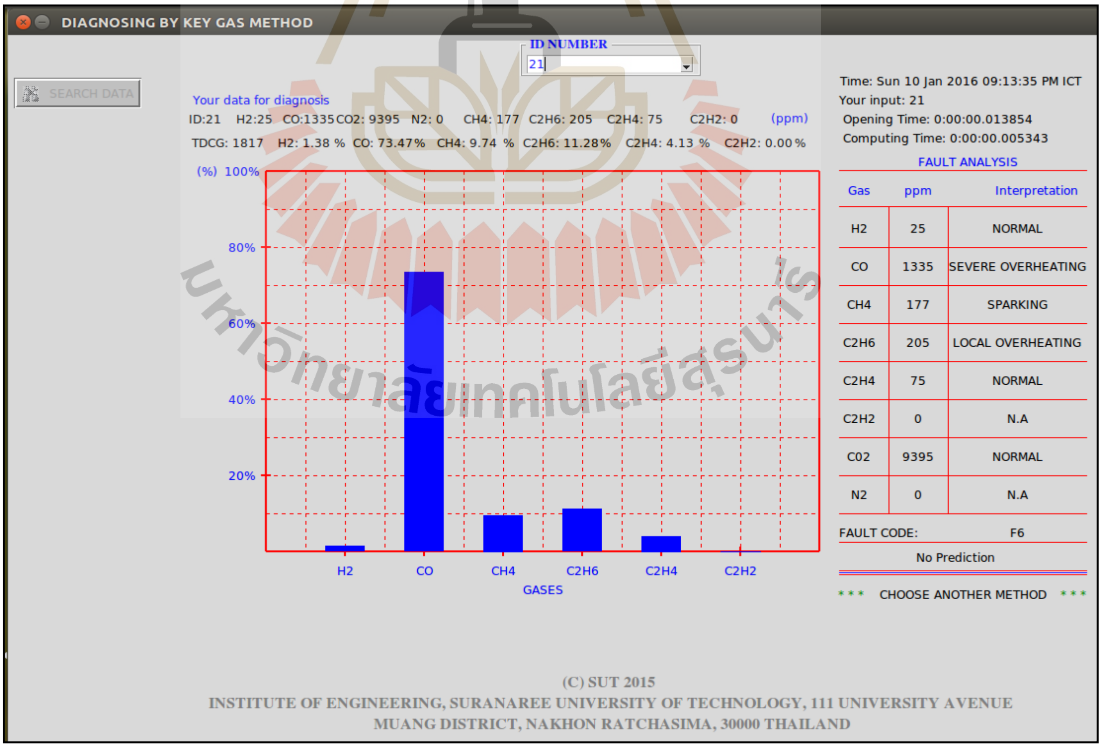


Figure 3.11 Diagnosis of the key gas method in the DGA software.

3.4.2 Dornenburg ratio method

The values of dissolved gasses are first compared to special concentrations—L1 in Table 2.1 (see section 2.6.2) to check the existence of the problem. It checks whether any of the dissolved gasses doubles the L1 norm in Table 2.1 and if any two gasses exceed L_1 norm. If satisfied, Ratio 1 (R1), Ratio 2 (R2), Ratio 3 (R3) and Ratio 4 (R4) are computed as shown in (3.8)-(3.11). The criteria used to diagnose fault are shown in Table 3.4. Figure 3.12 depicts the flow chart for performing diagnosis by using Dornenburg ratio method.

$$\text{Ratio 1 (R1)} = CH_4/H_2 \tag{3.8}$$

$$\text{Ratio 2 (R2)} = C_2H_2/C_2H_4 \tag{3.9}$$

$$\text{Ratio 3 (R3)} = C_2H_2/CH_4 \tag{3.10}$$

$$\text{Ratio 4 (R4)} = C_2H_6/C_2H_2 \tag{3.11}$$

Table 3.4 Ratios for Dornenburg method (IEEE C57.104-2008)

Suggested fault diagnosis	**Ratio 1 (R1) CH_4/H_2**		**Ratio 2 (R2) C_2H_2/C_2H_4**		**Ratio 3 (R3) C_2H_2/CH_4**		**Ratio 4 (R4) C_2H_6/C_2H_2**	
	Oil	Gas space	Oil	Gas space	Oil	Gas space	Oil	Gas space
Thermal decomposition	>1.0	>0.1	<0.75	<1.0	<0.3	<0.1	>0.4	>0.2
Partial discharge (low-intensity)	<0.1	<0.01	Not significant		<0.3	<0.1	>0.4	>0.2
Arcing (high-intensity)	>0.1 to <1.0	>0.01 to <0.1	>0.75	>1.0	>0.3	>0.1	<0.4	<0.2

In the script of python codes given in Appendix B (page 151) first the limits of dissolved gas-in-oil were checked by comparing with gas values in Table 2.1 (see section 2.6.2). If the conditions are satisfied for performing diagnosis, computation is done as per on (3.8)-(3.11). The obtained results are then compared with the diagnostic criteria shown in Table 3.4 in order to identify the existing fault. Figure 3.12 shows the flowchart for performing the diagnosis by using Dornenburg ratio method. Figure 3.13 shows the diagnosis done by the implemented Dornenburg ratio method in the DGA software. On top of the interface of Figure 3.13 is the combo box with ID NUMBER for user to select and retrieve the DGA information stored in the database. In the left side of the interface is the search button which retrieve the DGA data from the database with id number entered by user in the combo box. Below the combo box are the DGA data retrieved from the database. From the retrieved DGA data, id number and the dissolved gas-in-oil are displayed in ppm. Also, the computation of Ratio 1 (R1), Ratio 2 (R2), Ratio 3 (R3) and Ratio 4 (R4) computed based on (3.8)-(3.11) are shown. Details about the current date and time, opening time and computing time (CPU time) are given in the right side of the interface. The remaining part of the interface shows two tables whereby the first table shows the comparisons of the individual dissolved gas with the limit *L1* and whether it is useful to do the diagnosis. If it is required to do the test, the second table displays the ratios and the analysis of the existing faults. The existing faults is determined by matching the ratios obtained in (3.8)-(3.11) with the diagnostic criteria in Table 3.4.

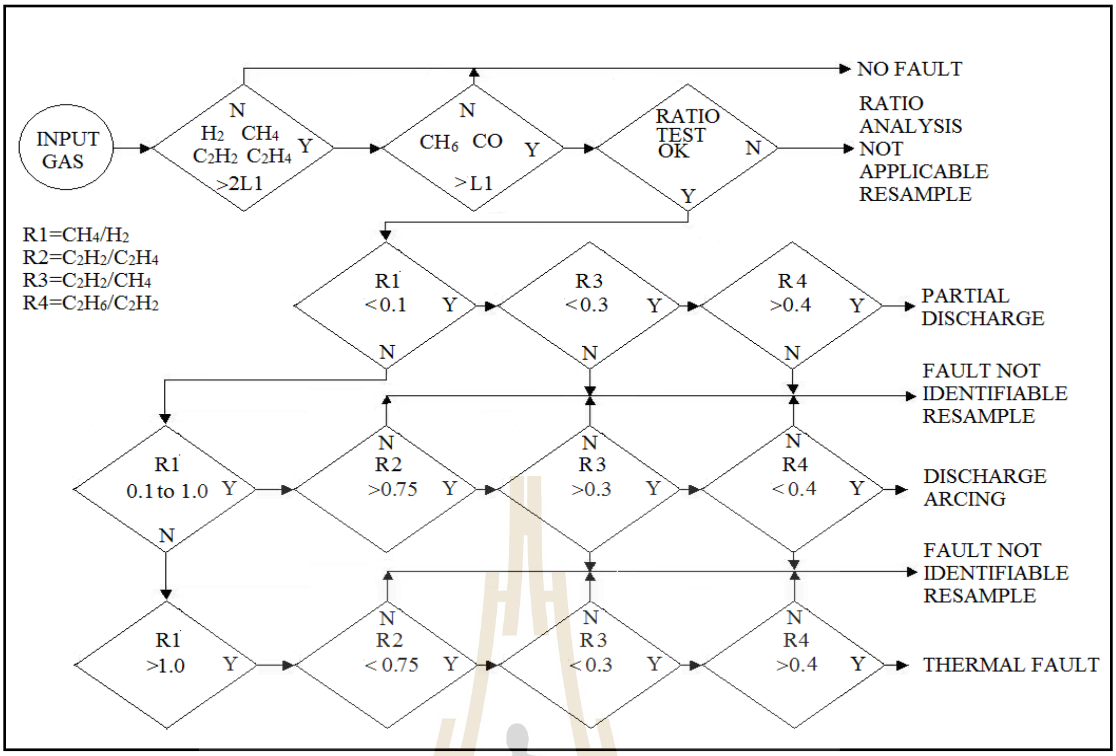


Figure 3.12 Flowchart of Dornenburg ratio method (IEEE C57.104, 2008).

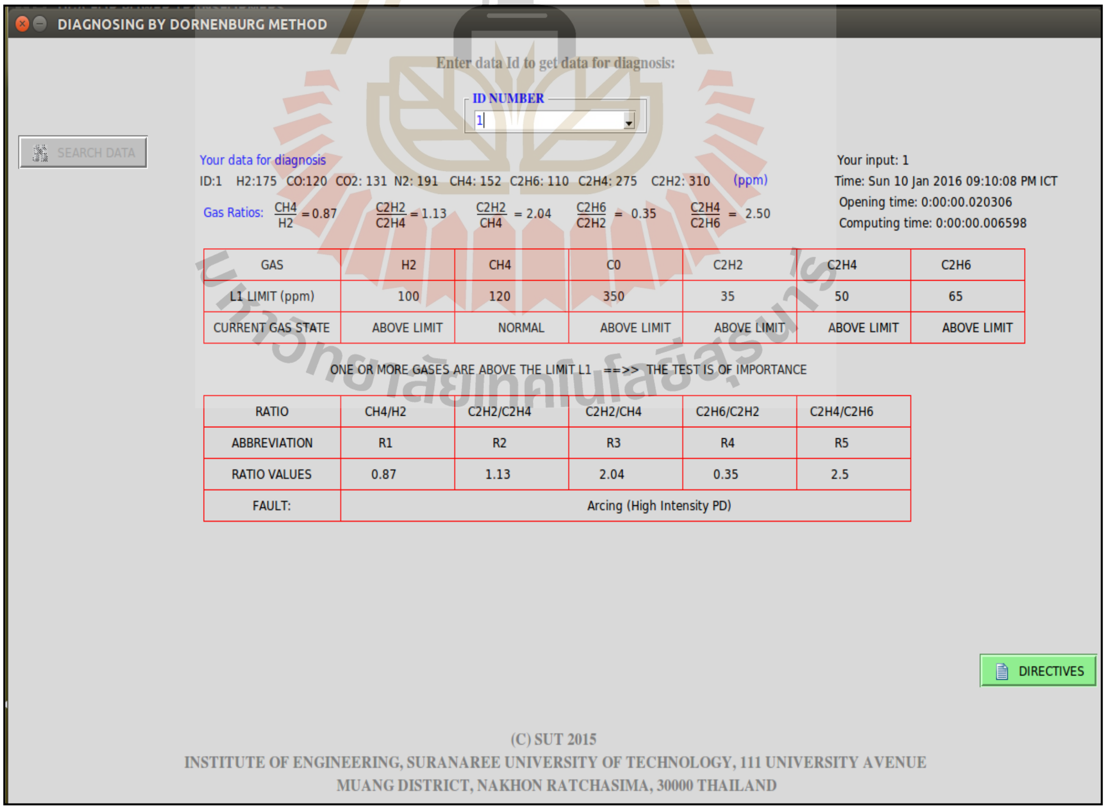


Figure 3.13 Diagnosis of Dornenburg ratio method in the DGA software.

3.4.3. Roger's ratio method

This method is based on ratios R1, R2, R3 and R4 as shown in (3.8)-(3.11). The codes of identification, ratio codes and diagnostic criteria are given in Tables 3.5-3.7. Figure 3.14 shows the flowchart of diagnosis by using Roger's ratio method.

Table 3.5 Code of identification (Sherif, Ghoneim, and Sayed, 2012)

Ratio	**Code**
CH_4/H_2	1
C_2H_6/ CH_4	2
C_2H_4/ C_2H_6	3
C_2H_2/C_2H_4	4

Table 3.6 Roger's ratio codes (Sherif, Ghoneim, and Sayed, 2012)

1	**2**	**3**	**4**	**Ranges**
0	5	0	0	<0.1
1	0	0	0	0.1 to 1.0
1	1	1	1	1.0 to 3.0
2	2	2	1	>3.0

The codes given in Table 3.5 were used to represent ratios of dissolved gas-in-oil. The codes were later used to match with the ranges given in Table 3.6. The arrangement of codes are also shown in Table 3.7 for diagnosing the existing fault.

Table 3.7 Fault diagnostic criteria-Roger's ratios (Sherif, Ghoneim, and Sayed, 2012)

Diagnostic				Faults
0	0	0	0	No fault-Normal deterioration
0	5	0	0	Partial discharge of low energy density or hydrolysis
1	5	0	0	Partial discharge of high density, possibly with tracking
0	5	1	0	Coincidental partial discharges and conductor overheating
0	5	0	1	Partial discharge of increasing energy density
1>2	0	0	0	Low discharge energy: flashover without power follow through
1>2	0	1	0	Low discharge energy: continuous sparking to floating potential
1>2	0	2	0	High energy discharge: arc with power follow through
0	0	1	0	Insulated conductor overheating
0	0	1	1	Complex thermal hotspot and conductor overheating
1	0	0	1	Coincidental thermal hotspot and low energy discharge
0	1	0	0	Thermal fault of low temperature <150°C
0	0>2	0	1	Thermal fault of temperature range 100°C to 200°C
0	1	1	0	Thermal fault of temperature range 150°C to 300°C Overheating of copper due to eddy currents.

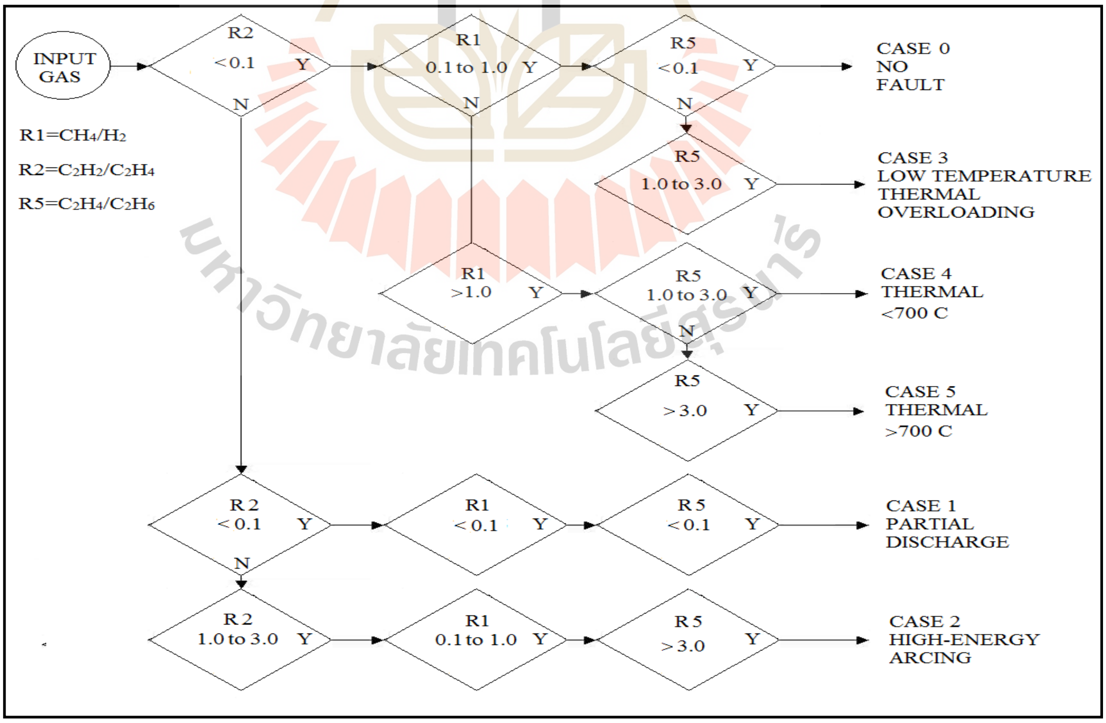


Figure 3.14 Flowchart of diagnosis by Roger's ratio method (IEEE C57.104, 2008).

In the script of python codes given in Appendix B, the computation was done as per (3.8) to (3.11). The obtained results were then compared with the diagnostic criteria shown in Table 3.7 in order to identify the existing fault. Figure 3.14 shows the flowchart for performing the diagnosis by using Roger's ratio method. Figure 3.15 shows the implemented Roger's ratio method in the DGA software with combo box for user to select and retrieve the DGA information stored in the database. The search button retrieves the DGA data from the database with id number entered by user in the combo box. Below the combo box are the DGA data retrieved from the database. The interface shows two tables whereby the first table shows the codes identified while the second table displays the ratios and the analysis of the existing faults.

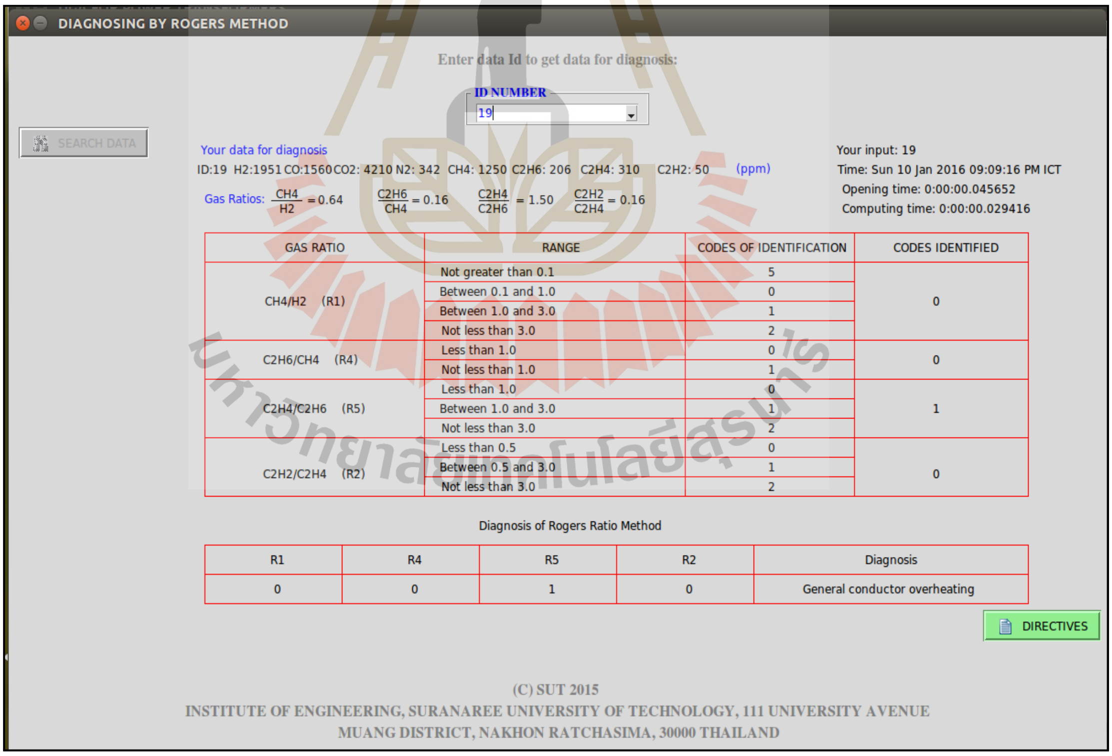


Figure 3.15 Diagnosis of Roger's ratio method in the DGA software.

3.4.4 IEC ratio method

This method is based on ratios R2, R1, R3, and R5. Tables 3.8-3.10 show codes of identification, ratio codes and diagnostic criteria used in IEC ratio method. Figure 3.16 shows the diagnosis from the implemented IEC ratio method.

Table 3.8 Code of identification (Ali Saeed, 2012)

Ratio	**Code**
C_2H_2/ C_2H_4	1
CH_4/H_2	2
C_2H_4/ C_2H_6	3

Table 3.9 IEC ratio codes (Ali Saeed, 2012)

Case	**Characteristic fault**	**1**	**2**	**3**
PD	Partial discharges	NS	<0.1	<0.2
D1	Discharge of low energy	>1	0.1 – 0.5	>1
D2	Discharge of high energy	0.6 – 2.5	0.1 - 1	>2
T1	Thermal fault t<300°C	NS	>1 but NS	<1
T2	Thermal fault 300°C<t< 700°C	<0.1	>1	1 - 4
T3	Thermal fault t>700°C	<0.22	>1	<4

Table 3.10 IEC diagnostic criteria (James and Su, 2008)

No.	**Fault type**	**C_2H_2/ C_2H_4**	**CH_4/H_2**	**C_2H_4/ C_2H_6**
0	No fault	0	0	0
1	Partial discharge of low energy density	0 but NS	1	0
2	Partial discharge of high energy density	1	1	0
3	Discharge of low energy	1 or 2	0	1 or 2
4	Discharge of high energy	1	0	2
5	Thermal fault of low temperature, <150°C	0	0	1
6	Thermal fault of low temperature, 150-300°C	0	2	0
7	Thermal fault of medium temperature, 300-700°C	0	2	1
8	Thermal fault of high temperature, >700°C	0	2	2

In the script of python codes for IEC ratio method given in Appendix B, the computation was done as per (3.8) to (3.11). The obtained results were then compared with the diagnostic criteria shown in Table 3.10 in order to identify the existing fault. Figure 3.16 shows the implemented IEC ratio method in the DGA software with combo box for user to select and retrieve the DGA information stored in the database. The search button retrieves the DGA data from the database with id number entered by user in the combo box. Below the combo box are the DGA data retrieved from the database. The interface shows two tables whereby the first table shows the codes identified while the second table displays the ratios and the analysis of the existing faults.

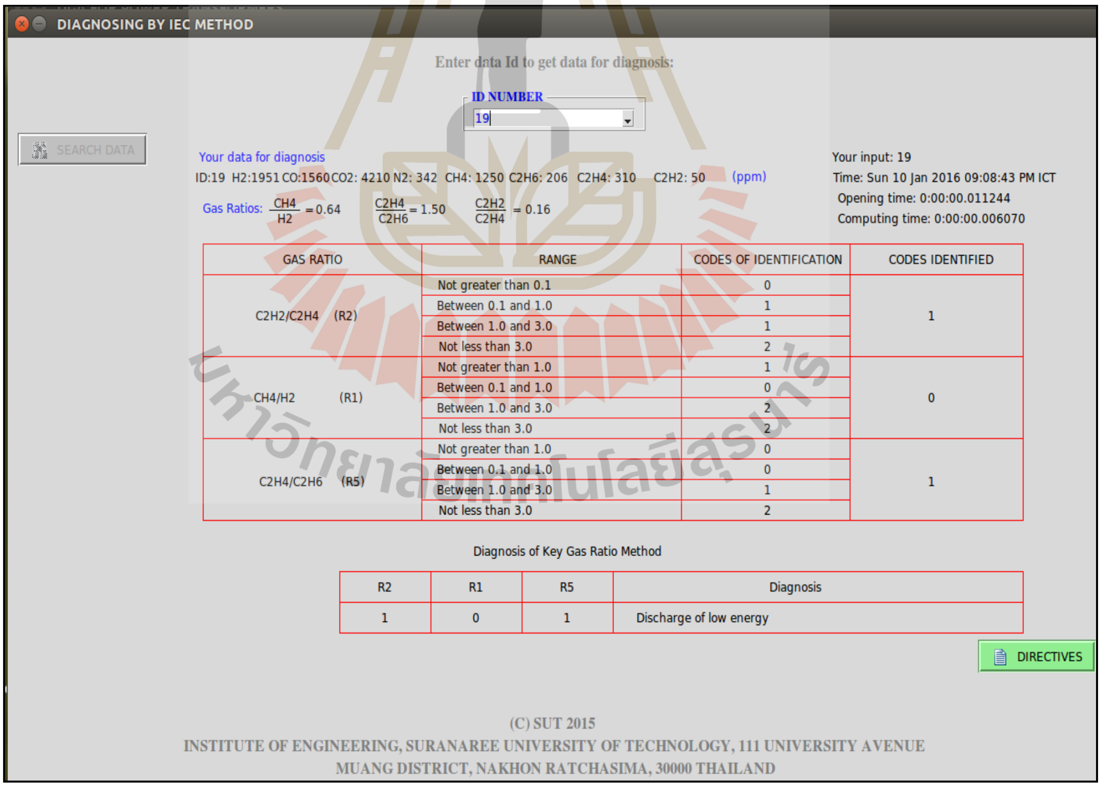


Figure 3.16 Diagnosis of IEC method in the DGA software.

3.4.5 Duval triangle method

This method uses three hydrocarbon gasses namely CH_4, C_2H_4, and C_2H_2. It employs proportions of these three gasses to identify the fault. The triangle method is indicated in Figure 3.17. In addition to the 6 zones of individual faults mentioned in Table 3.11 (PD, D1, D2, T1, T2 or T3), an intermediate zone DT has been attributed to mixtures of electrical and thermal faults in the transformer (Duval, 2005). This method uses the principal of intersection of three hydrocarbon gasses such as CH_4, C_2H_4, and C_2H_2. The triangle shown in Figure 3.17 is an equilateral triangle with each side expressed in terms of percentage. The percentages are arranged in such a way that the highest percentage of one hydrocarbon gas is cross to the lowest percentage of another hydrocarbon gas. The fault is identified by the intersection of three hydrocarbon gasses and the respective fault zone. In order to locate the intersection point, proportions of the three hydrocarbon gasses are first computed.

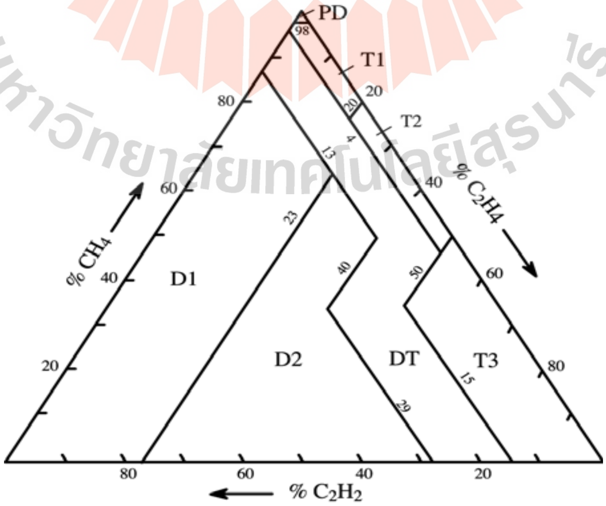


Figure 3.17 Duval triangle (Duval, 2005)

Table 3.11 Examples of faults detectable by DGA (Duval, 2005)

Symbol	Fault	Examples
PD	Partial discharges	Discharges of the cold plasma (corona) type in gas bubbles or voids, with the possible formation of X-wax in the paper.
D1	Discharge of low energy	Partial discharges of the sparking type, inducing pinholes, carbonized punctures in the paper. Low energy is arcing inducing carbonized perforation or surface tracking of paper or the formation of carbon particles in the oil.
D2	Discharge of high energy	Discharges of paper or oil, with power follow-through, resulting in extensive damage to paper or formation of carbon particles in oil, metal fusion or tripping of equipment and gas alarms.
T1	Thermal fault T<300°C	Evidenced by paper turning brownish (>200° C) or carbonized (>300° C).
T2	Thermal fault 300< t<700° C	Carbonization of paper, the formation of carbon particles in the oil.
T3	Thermal fault T>700° C	Extensive formation of carbon particles in oil, metal coloration (>800° C) or metal fusion (>1000° C)

In order to display a DGA result in the triangle, one must start with the concentrations of the three gases, (CH_4) = *A*, (C_2H_4) = *B* and (C_2H_2) = *C*, in ppm. First, the sum of these three values is calculated:

$$CH_4 + C_2H_4 + C_2H_2 = S \tag{3.12}$$

then, relative proportion of the three gases are estimated in %:

$$P_1 = \%\ CH_4 = 100\ (A/S) \tag{3.13}$$

$$P_2 = \%\ C_2H_4 = 100\ (B/S) \tag{3.14}$$

$$P_3 = \%\ C_2H_2 = 100\ (C/S) \tag{3.15}$$

X, Y, and Z necessarily lie between 0, and 100%, and ($X + Y + Z$) should always be equal to 100 %. By plotting X, Y and Z in the triangle provide only one point in the triangle (Sukbirr and Bandyopadhyay, 2010). After obtaining the percentages of fault gasses, the line is drawn from % CH_4 parallel to % C_2H_2, % C_2H_4 parallel to % CH_4 and % C_2H_2 parallel to % CH_4. The triangle coordinates are calculated by using normal trigonometric function to obtain the location of points A, B, and C on the selected length $\overline{AB} = L$ and the initial point A (A_x, A_y). The length L is expressed in terms of pixels (500 pixels) and point A (A_x, A_y) was initialized as A (120, 550) pixels in computer drafting. Triangular coordinates and allocation of points are shown in Figure 3.18.

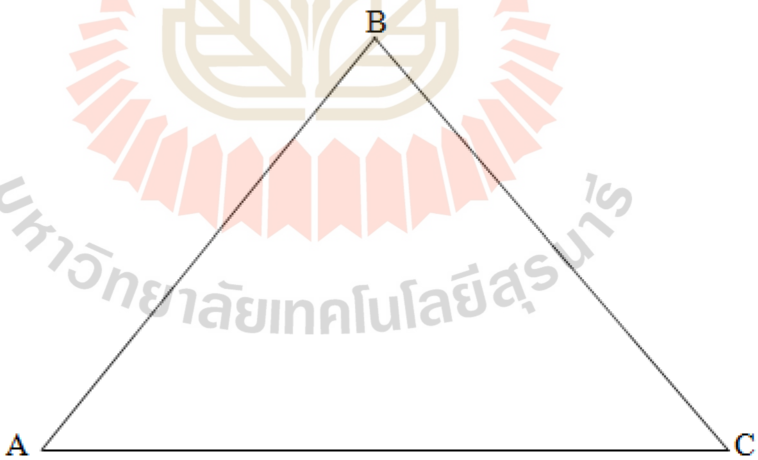


Figure 3.18 Allocation of triangular coordinates.

The coordinate A was described as A (A_x, A_y) and can be anywhere in the coordinate system of computer drafting. Through (21) to (26), a coordinate point in each edge was calculated. Therefore, by considering the fault point as point F (F_x, F_y) and $P1$, $P2$ and $P3$ as calculated above in (17)-(20), the fault zone can be identified. Figure 3.19 shows the allocation of fault point whereas Table 3.12 shows a list of triangular coordinates of fault zones in a Duval triangle. Figure 3.20 shows the flowchart for performing diagnosis by Duval triangle. The implemented Duval triangle is depicted in Figure 3.21.

$$B = (B_x, B_y) \tag{3.16}$$

$$B_x = A_x + 0.5L \quad \text{and} \tag{3.17}$$

$$B_x = 120 + 275 = 395 \text{ pixels}$$

$$B_y = A_y - L\cos 30^\circ = 550 - 476.3 = 73.7 \text{pixels} \tag{3.18}$$

And

$$C = (C_x, C_y) \tag{3.19}$$

$$C_x = A_x + L \tag{3.20}$$

$$C_x = 120 + 500 = 620 \text{ pixels}$$

$$C_y = A_y = 550 \text{ pixels} \tag{3.21}$$

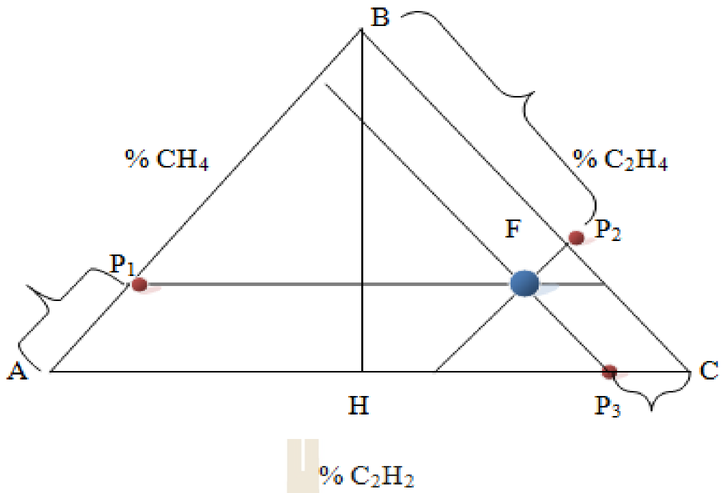


Figure 3.19 Allocation of fault in a triangle

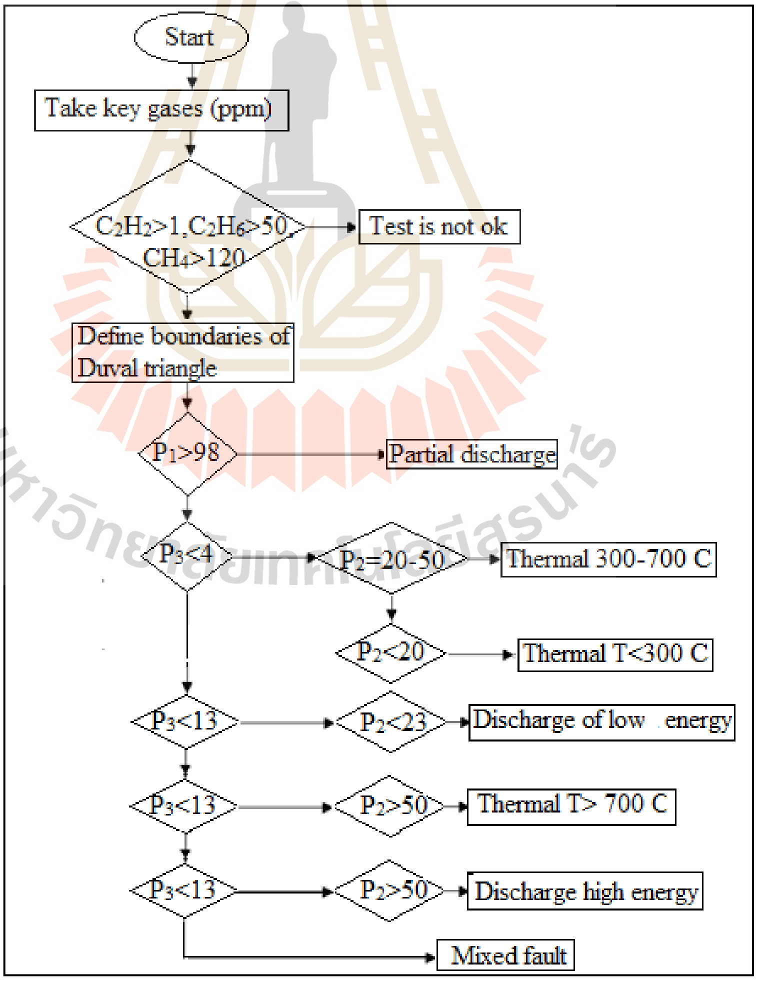


Figure 3.20 Flowchart of diagnosis by Duval triangle method

Table 3.12 Triangular coordinates for the Duval triangle and fault zones (Sukhbir and Bandyopadhyay, 2010)

Area	Points	CH_4	C_2H_4	C_2H_2
PD	PD1	98	2	0
	PD2	100	0	0
	PD3	98	0	2
DT	DT1	0	71	29
	DT2	0	85	15
	DT3	35	50	15
	DT4	46	50	4
	DT5	96	0	4
	DT6	87	0	13
	DT7	47	40	13
	DT8	31	40	29
T1	T11	76	20	4
	T12	80	20	0
	T13	98	2	0
	T14	98	0	2
	T15	96	0	4
T2	T21	46	50	4
	T22	50	50	0
	T23	80	20	0
	T24	76	20	4
T3	T31	0	85	15
	T32	0	100	0
	T33	50	50	0
	T34	35	50	15
D1	D11	0	0	100
	D12	0	0	77
	D33	64	23	77
	D14	87	0	13
D2	D21	0	23	77
	D22	0	71	29
	D23	31	40	29
	D24	47	40	13
	D25	64	23	13

In the script of python codes for Duval triangle given in Appendix B first the proportions of three hydrocarbon gasses were computed as per (3.13) to (3.15). The three proportions were obtained on the base of their summation given in (3.12). Before plotting, the triangle was plotted with coordinates computed in (3.16) to (3.21). Then the fault zones were plotted inside the triangle by considering the sides of the triangle as boundaries. The coordinates of fault zones are shown in Table 3.12. Figure 3.21 shows the implemented Duval triangle with plotted fault zones. Zones were differentiated by colors for easy identification of the intersection. On top of the interface of Figure 3.21 is the combo box with ID NUMBER for user to select and retrieve the DGA information stored in the database. In the left side of the interface is the search button which retrieve the DGA data from the database with id number entered by user in the combo box. Below the combo box are the DGA data retrieved from the database. From the retrieved DGA data, id number and the dissolved gas-in-oil are displayed in ppm. Also, the computation of P_1, P_2 and P_3 computed based on (3.13)-(3.16) are shown. Details about the current date and time, opening time and computing time (CPU time) are given in the right side of the interface. The remaining part of the interface shows the implementation of the triangle, fault zones and intersection of P_1, P_2 and P_3. On the right side is the table that shows the interpretation of the existing fault according intersection point and fault zone. Then the existing faults is determined by the type of fault zone on which intersection has fall on it as shown in Figure 3.21.

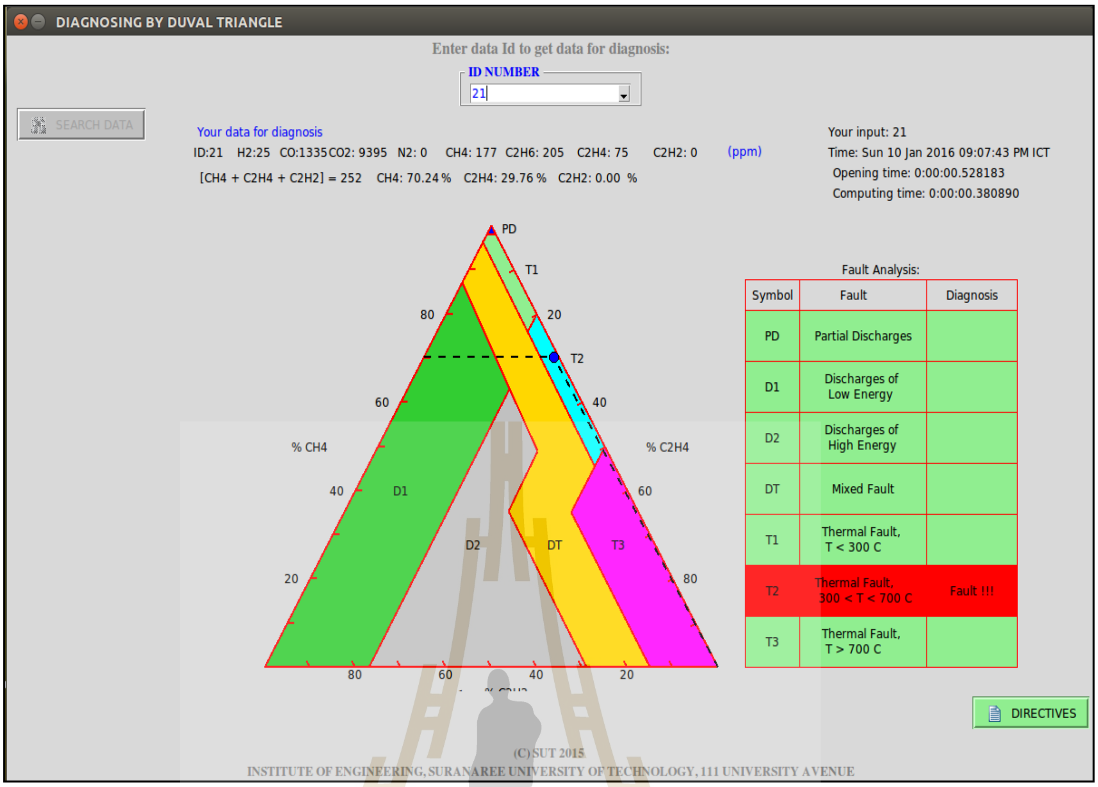


Figure 3.21 Diagnosis of Duval triangle in the DGA software.

3.4.6 Gas comparison

The comparison option co pares two DGA data with different gas levels obtained in different dates but from the same transformer. This identifies the gas increment within a specific period. Figure 3.22 shows the flowchart of performing comparison in the DGA software. The implementation of comparison is shown in Figure 3.23.

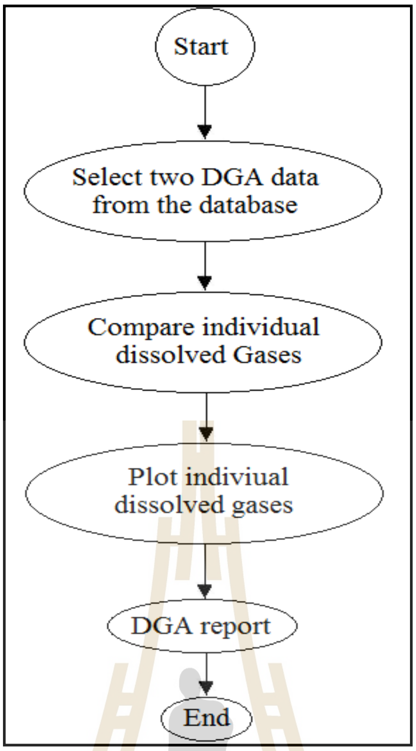


Figure 3.22 Flowchart of performing comparison in the DGA software.

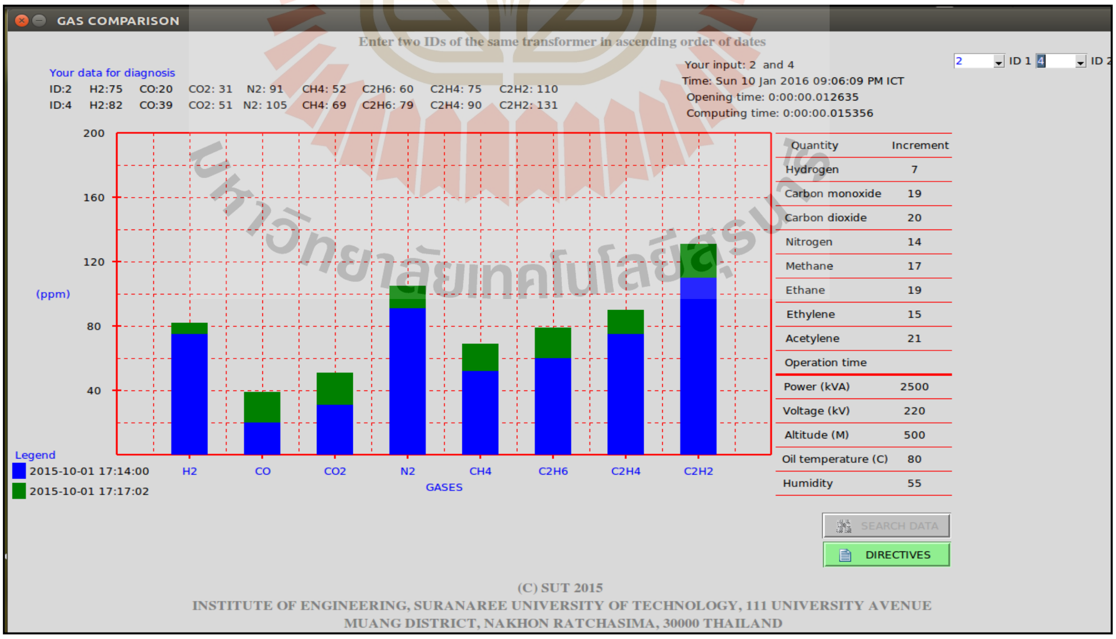


Figure 3.23 Gas comparison in the DGA software

3.4.7 Gas trending

Various gas data, five in numbers from the same transformer in ascending order can be trended to see the increment of gas generation over a long time. Figure 3.24 shows the flowchart for performing trending in DGA software. Figure 3.25 shows the implementation of trending option.

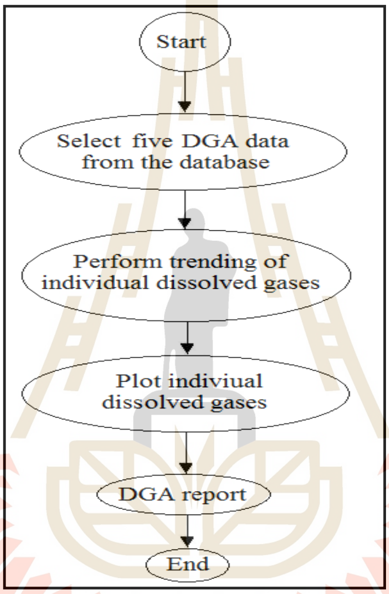


Figure 3.24 Flowchart of performing trending in the DGA software.

3.4.8 Report summary

The summary report for every diagnosis, comparison, and trending was saved in the report folder within the computer. The flowchart for preparation of the report summary is depicted in Figure 3.26 (a). The process starts after user has finished to perform either diagnosis, comparison, or trending. It involves opening the summary file for diagnosis, comparison, or trending and then write the current summary. The command for opening the file with sample codes is shown below:

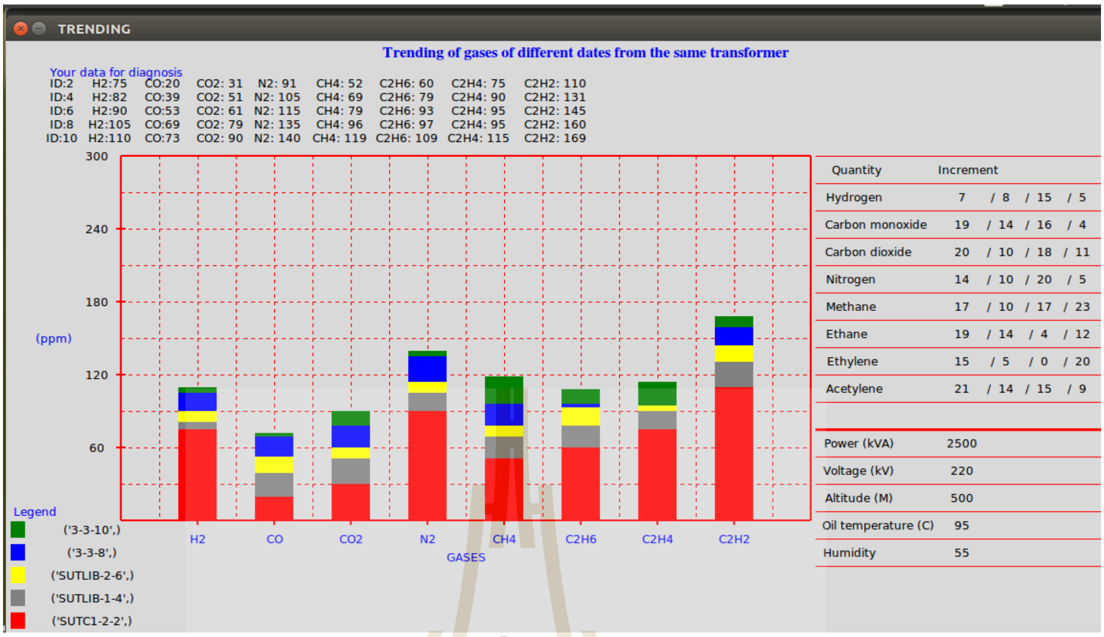


Figure 3.25 Implementation of gas trending in the DGA software

- with open('.\summary.txt', mode='a+') as fl:

  lines_of_text = [" "]

  fl.writelines (lines_of_text)

  fl.close ()

where

- *open* () = command for opening existing file or creating a new non-existing file,
- *mode* = specifies the mode of opening file *r* = read only, *w* = write only and *a* = append mode,
- *lines_of_text* [" "] = specifies the contents to be written in the file,
- *writelines* () = command for writing the contents in the file,

- *fl.writelines (lines_of_text)* = writes the content specified in the *lines_of_text* [" "] to the file,
- *fl.close* = closes the file after finishing the task.

Figure 3.26 (b) shows the sample codes for writing the summary of diagnosis at the end of diagnosis by using key gas method. At the end of each diagnosis process the summary report is prepared and written in the text file. Figure 3.26 (c) shows the snapshot of the diagnosis done by Duval triangle and key gas method written in the text file. The summary shows the name of the diagnostic method, date, id number, DGA data, computation and diagnosis from the computation.

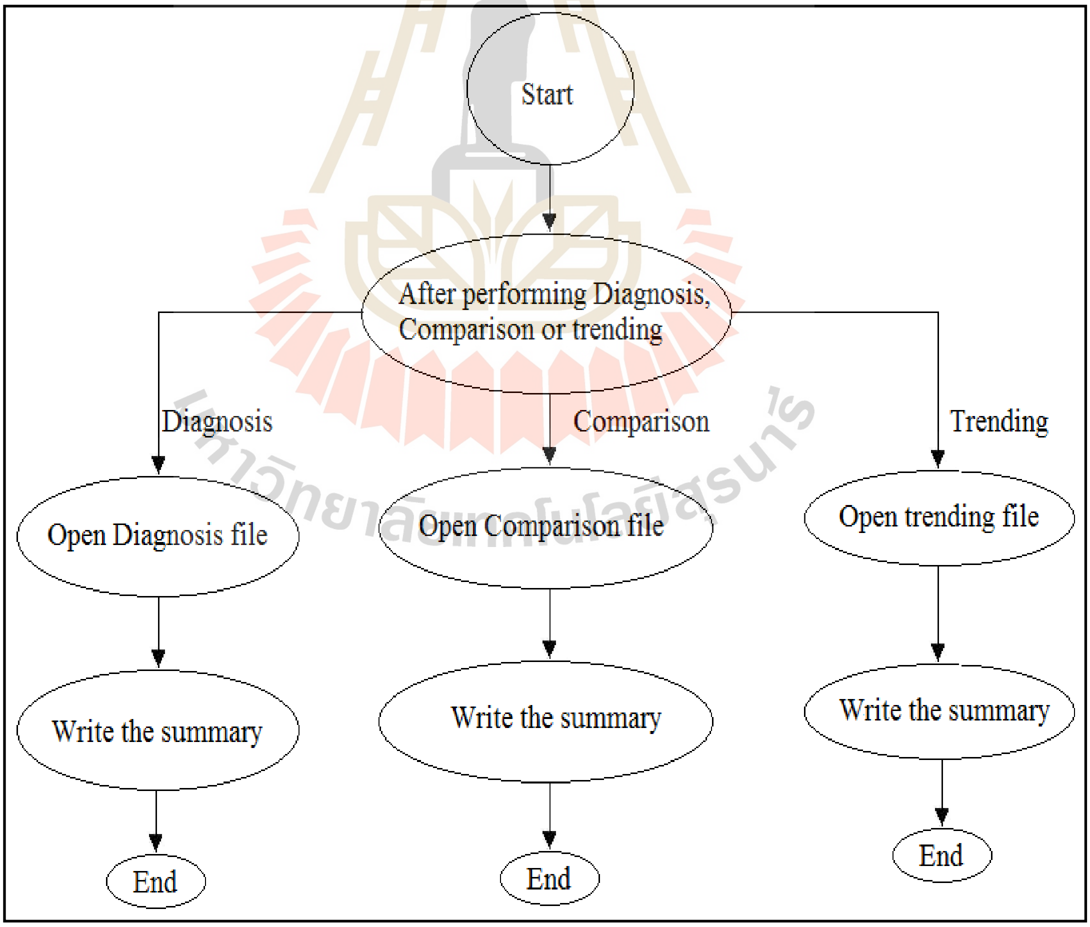
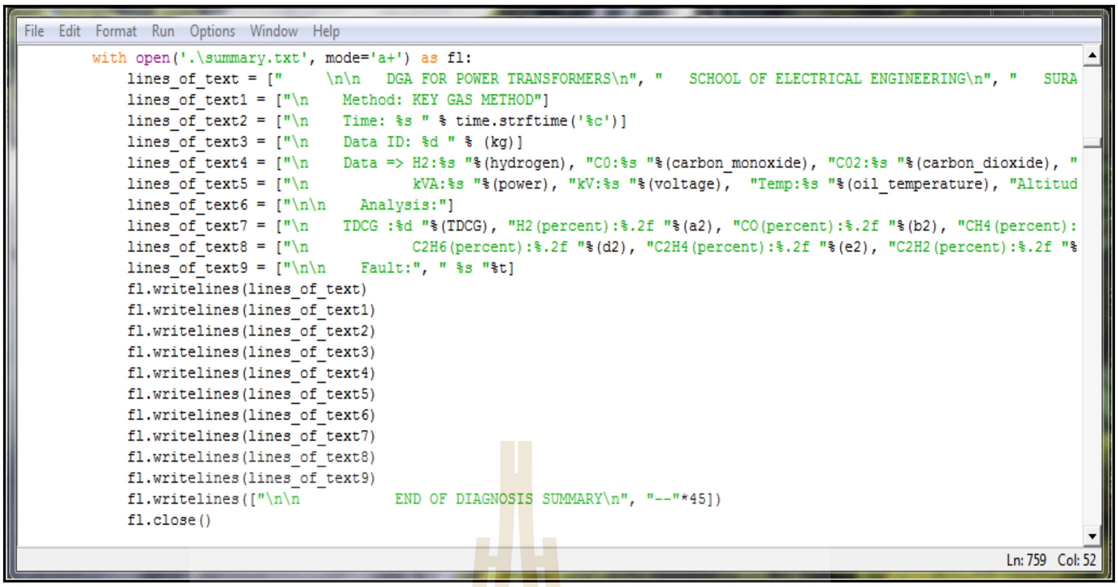
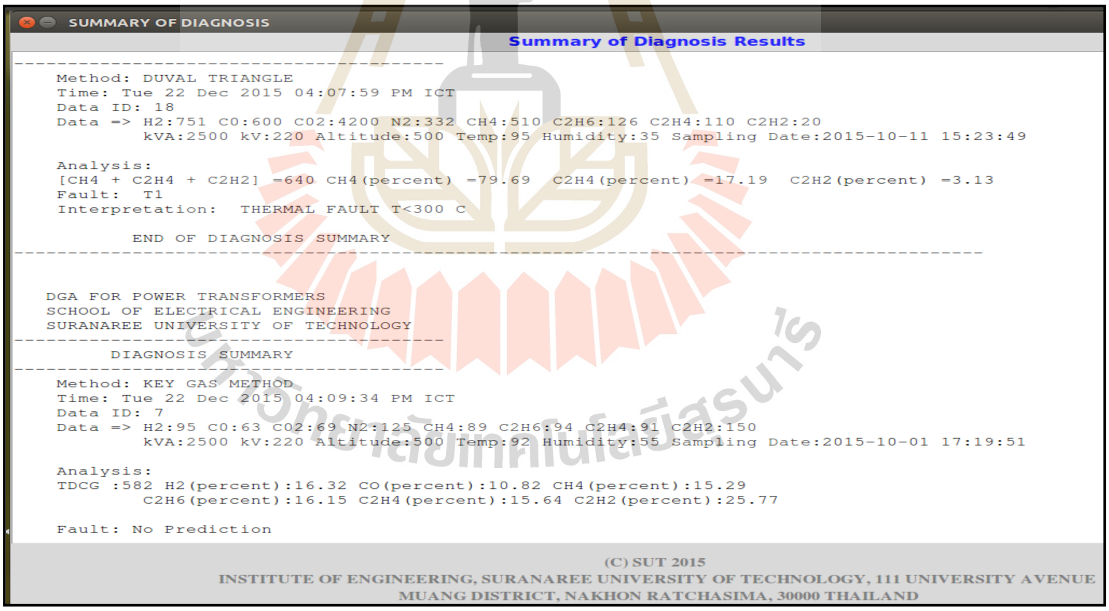


Figure 3.26(a) Flowchart of the report summary.



```
File  Edit  Format  Run  Options  Window  Help
        with open('.\summary.txt', mode='a+') as fl:
            lines_of_text = ["     \n\n   DGA FOR POWER TRANSFORMERS\n", "    SCHOOL OF ELECTRICAL ENGINEERING\n", "    SURA
            lines_of_text1 = ["\n     Method: KEY GAS METHOD"]
            lines_of_text2 = ["\n     Time: %s " % time.strftime('%c')]
            lines_of_text3 = ["\n     Data ID: %d " % (kg)]
            lines_of_text4 = ["\n     Data => H2:%s "%(hydrogen), "CO:%s "%(carbon_monoxide), "CO2:%s "%(carbon_dioxide), "
            lines_of_text5 = ["\n             kVA:%s "%(power), "kV:%s "%(voltage),  "Temp:%s "%(oil_temperature), "Altitud
            lines_of_text6 = ["\n\n     Analysis:"]
            lines_of_text7 = ["\n     TDCG :%d "%(TDCG), "H2(percent):%.2f "%(a2), "CO(percent):%.2f "%(b2), "CH4(percent):
            lines_of_text8 = ["\n             C2H6(percent):%.2f "%(d2), "C2H4(percent):%.2f "%(e2), "C2H2(percent):%.2f "%
            lines_of_text9 = ["\n\n     Fault:", " %s "%t]
            fl.writelines(lines_of_text)
            fl.writelines(lines_of_text1)
            fl.writelines(lines_of_text2)
            fl.writelines(lines_of_text3)
            fl.writelines(lines_of_text4)
            fl.writelines(lines_of_text5)
            fl.writelines(lines_of_text6)
            fl.writelines(lines_of_text7)
            fl.writelines(lines_of_text8)
            fl.writelines(lines_of_text9)
            fl.writelines(["\n\n           END OF DIAGNOSIS SUMMARY\n", "--"*45])
            fl.close()
                                                                                          Ln: 759  Col: 52
```

Figure 3.26(b) Sample codes for preparing the report summary.



```
SUMMARY OF DIAGNOSIS
                              Summary of Diagnosis Results
----------------------------------------
    Method: DUVAL TRIANGLE
    Time: Tue 22 Dec 2015 04:07:59 PM ICT
    Data ID: 18
    Data => H2:751 CO:600 CO2:4200 N2:332 CH4:510 C2H6:126 C2H4:110 C2H2:20
            kVA:2500 kV:220 Altitude:500 Temp:95 Humidity:35 Sampling Date:2015-10-11 15:23:49

    Analysis:
    [CH4 + C2H4 + C2H2] =640 CH4(percent) =79.69  C2H4(percent) =17.19  C2H2(percent) =3.13
    Fault:  T1
    Interpretation:  THERMAL FAULT T<300 C

           END OF DIAGNOSIS SUMMARY
------------------------------------------------------------------------------------------


   DGA FOR POWER TRANSFORMERS
   SCHOOL OF ELECTRICAL ENGINEERING
   SURANAREE UNIVERSITY OF TECHNOLOGY
----------------------------------------
         DIAGNOSIS SUMMARY
----------------------------------------
    Method: KEY GAS METHOD
    Time: Tue 22 Dec 2015 04:09:34 PM ICT
    Data ID: 7
    Data => H2:95 CO:63 CO2:69 N2:125 CH4:89 C2H6:94 C2H4:91 C2H2:150
            kVA:2500 kV:220 Altitude:500 Temp:92 Humidity:55 Sampling Date:2015-10-01 17:19:51

    Analysis:
    TDCG :582 H2(percent):16.32 CO(percent):10.82 CH4(percent):15.29
            C2H6(percent):16.15 C2H4(percent):15.64 C2H2(percent):25.77

    Fault: No Prediction

                                      (C) SUT 2015
   INSTITUTE OF ENGINEERING, SURANAREE UNIVERSITY OF TECHNOLOGY, 111 UNIVERSITY AVENUE
              MUANG DISTRICT, NAKHON RATCHASIMA, 30000 THAILAND
```

Figure 3.26 (c) Combined summary report of diagnostic works in one text file.

3.5 Database programming

Since MySQL programming language is an open source, it is also easily available through online searching. It can be downloaded through Google search engine by using the official MySQL website. MySQL versions are available from their official website.

3.5.1 The official website of the MySQL software

The official website of MySQL programming language is https://www.mysql.com. Through Google search engine user can reach the official website of MySQL programming language. Figure 3.27 shows the snapshot of the official website with the address in red dashed-box.

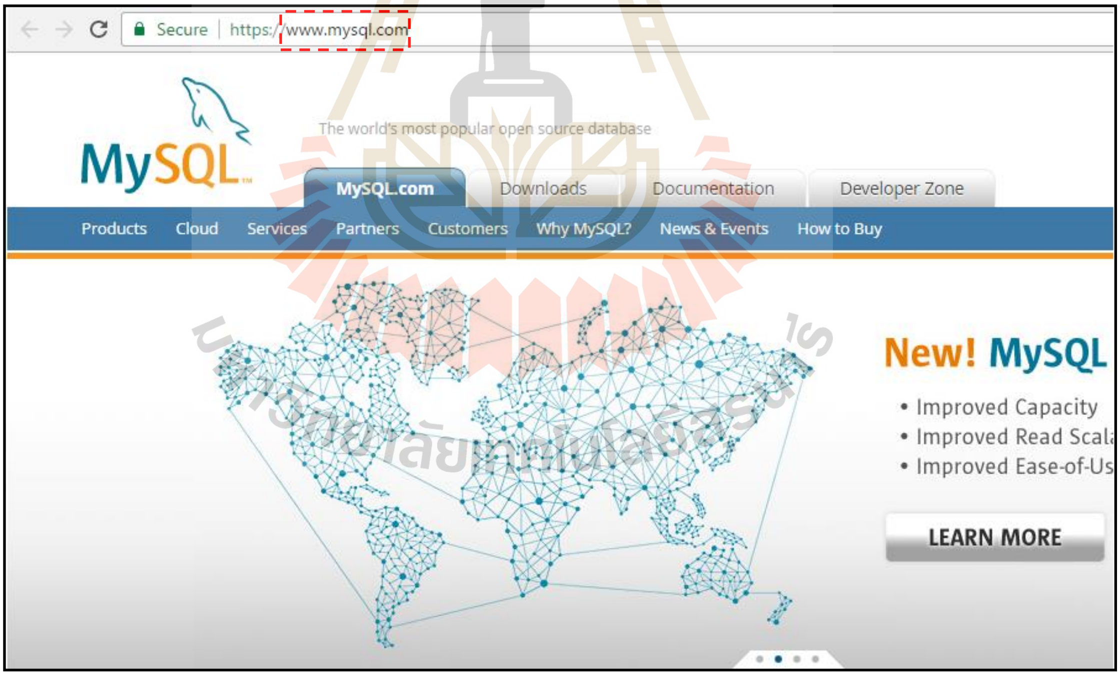


Figure 3.27 The official website of MySQL

User has to select the MySQL installer that suits its operating system such as Windows operating system, Linux operating system, Mac operating system, etc. From the MySQL official website as shown in the snapshot of Figure 3.27, for Windows operating system user has to click in the Download tab to choose the MySQL version that he want. Figure 3.28 shows MySQL versions for Windows operating system with system type 32 bit and 64 bit. To reach directly the link of Figure 3.28, one has to type direct the link https://dev.mysql.com/downloads/windows/installer/. Once MySQL installer is clicked it opens the new window for user to agree and download the current version of the installer as shown in Figure 3.29. Figure 3.30 shows the last stage for downloading the MySQL installer by clicking in the red dashed box.

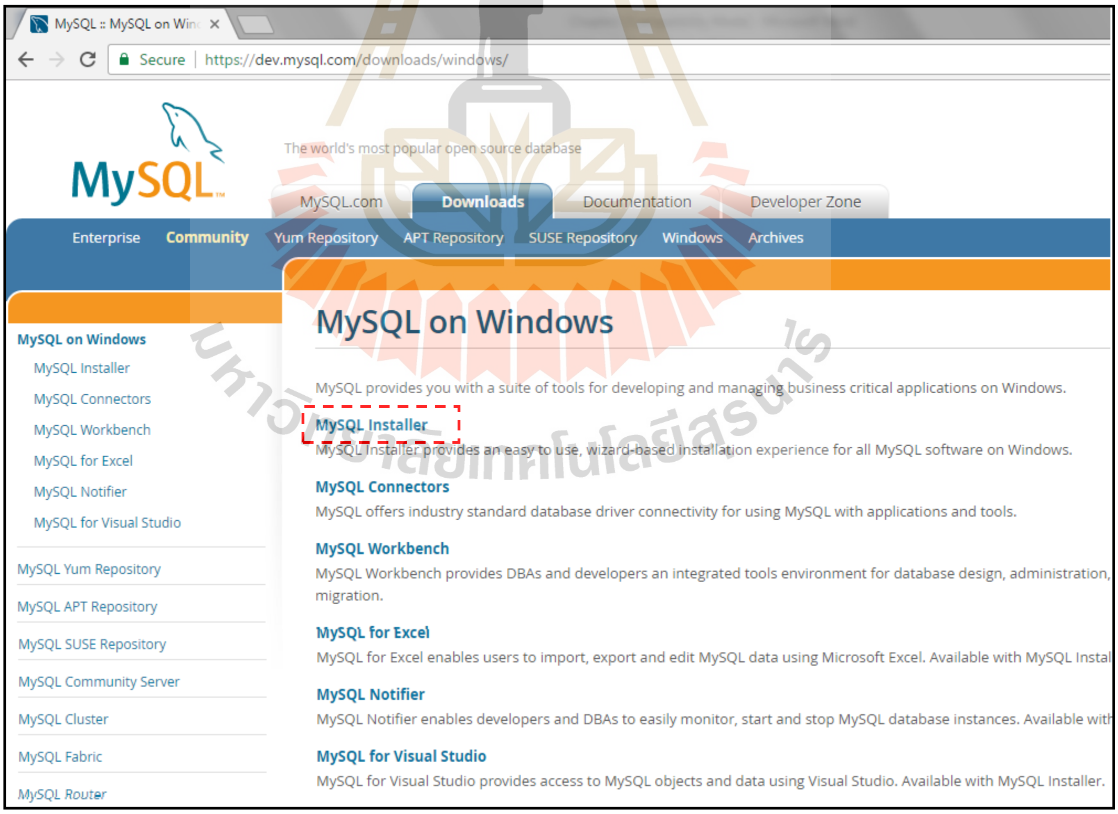


Figure 3.28 The MySQL installer for Windows OS

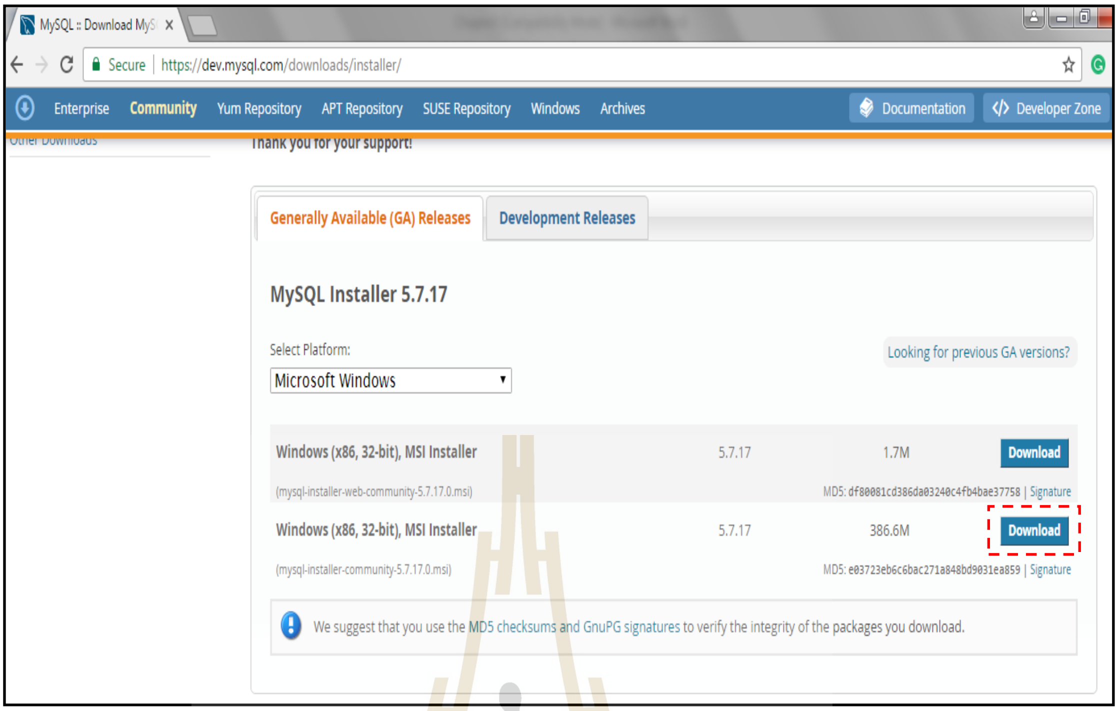


Figure 3.29 The MySQL installer version 5.7.17 for Windows 32-bit

Figure 3.30 The final MySQL installer link

3.5.2 Installing MySQL setup

When the downloaded file with .msi extension is opened, the installation proceeds as shown from the snapshots of Figure 3.31 (a) to (d).

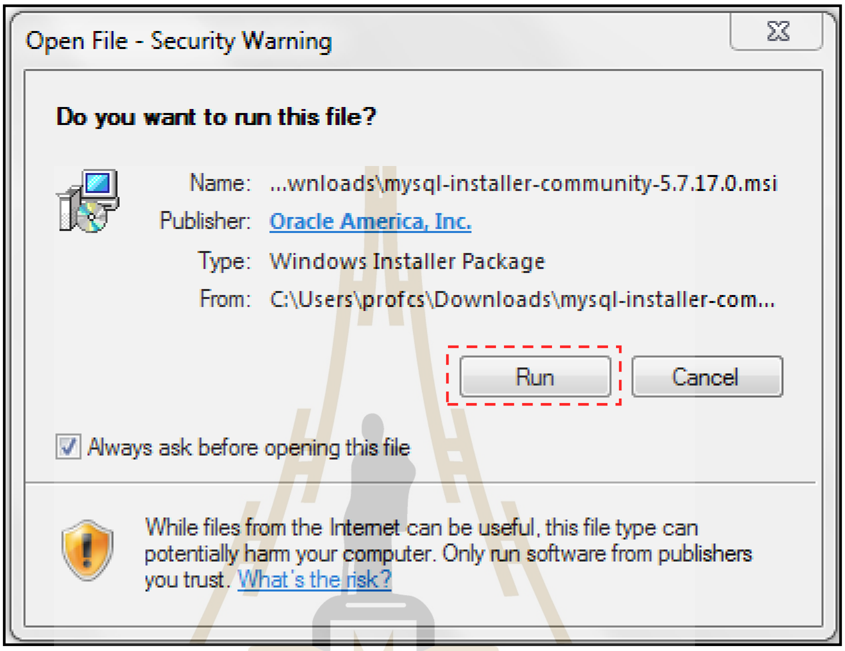


Figure 3.31 (a) running the downloaded file with .msi extension

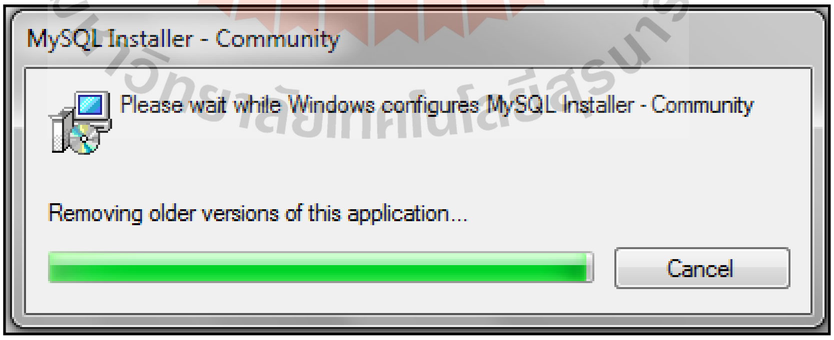


Figure 3.31 (b) Installing the downloaded file

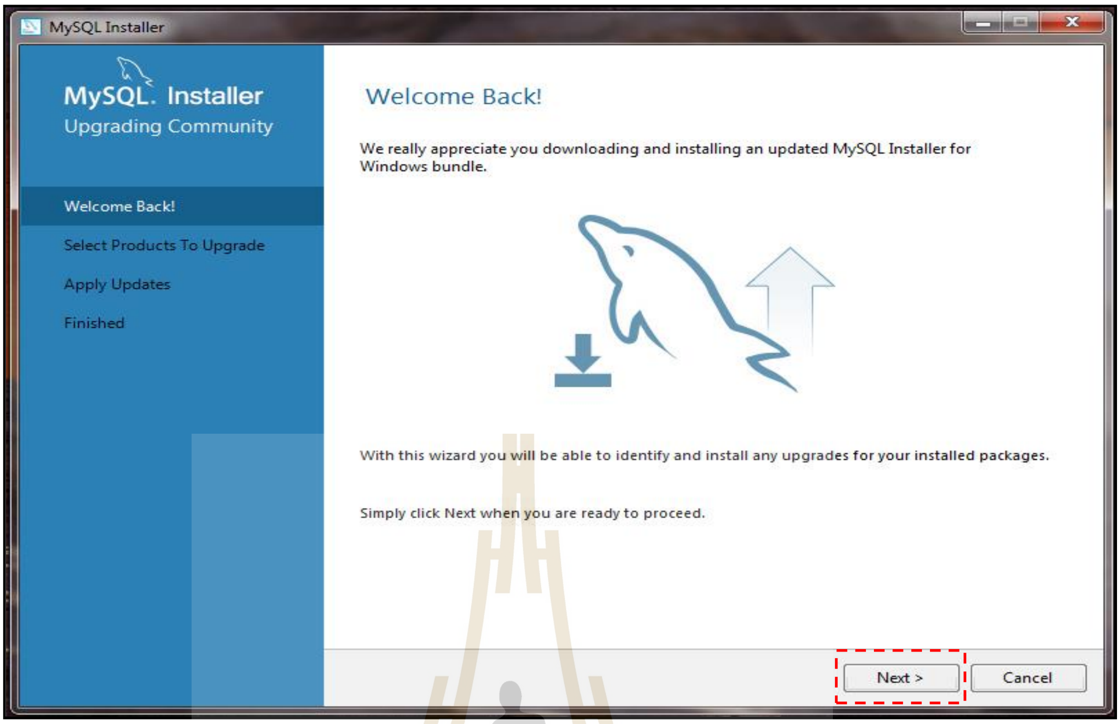


Figure 3.31 (c) Launching the mysql wizard

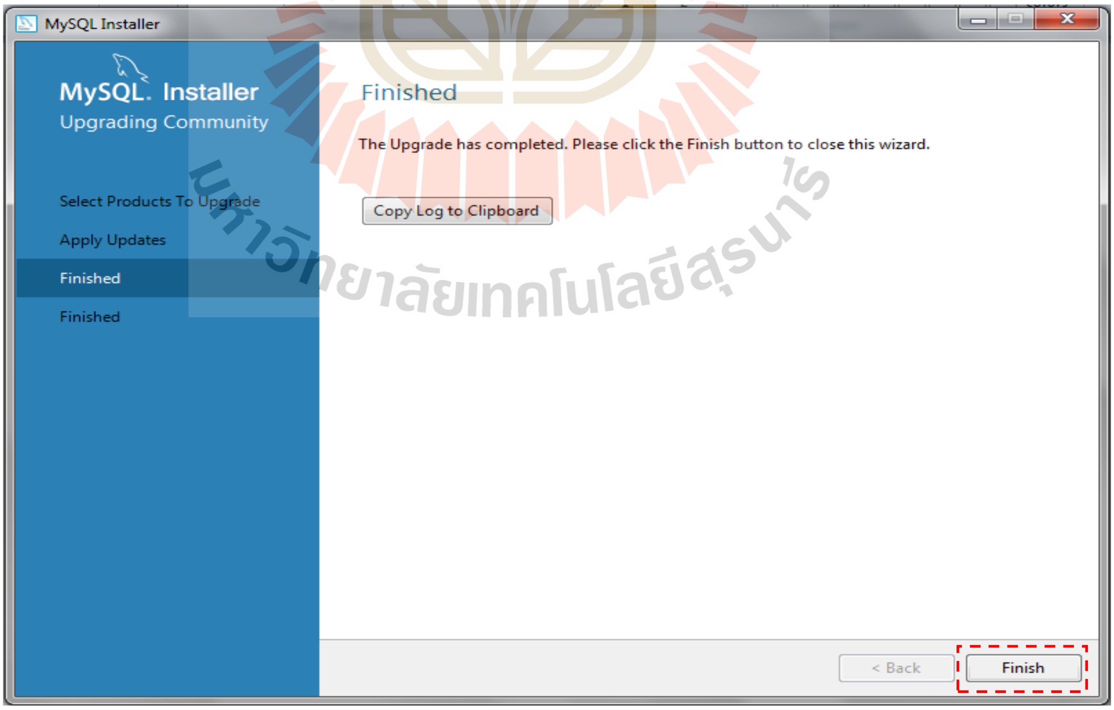


Figure 3.31 (d) Finishing the mysql wizard

3.5.3 Programming MySQL database

Another alternative of programming database is by using XAMPP software which can be obtained through https://sourceforge.net/projects/xampp/. Once finished to be installed, MySQL software can be searched through computer search option as shown in Figure 3.32. This XAMPP software comes with the different databases. To create database one needs first to know the location of the XAMPP folder which contains mysql folder and bin folder. Figure 3.33 shows the opening of the command window through computer search option. Since the command window shows the default address in it, one needs to change the address to suit the location of the XAMPP folder, mysql folder and then bin folder as shown in Figure 3.34.

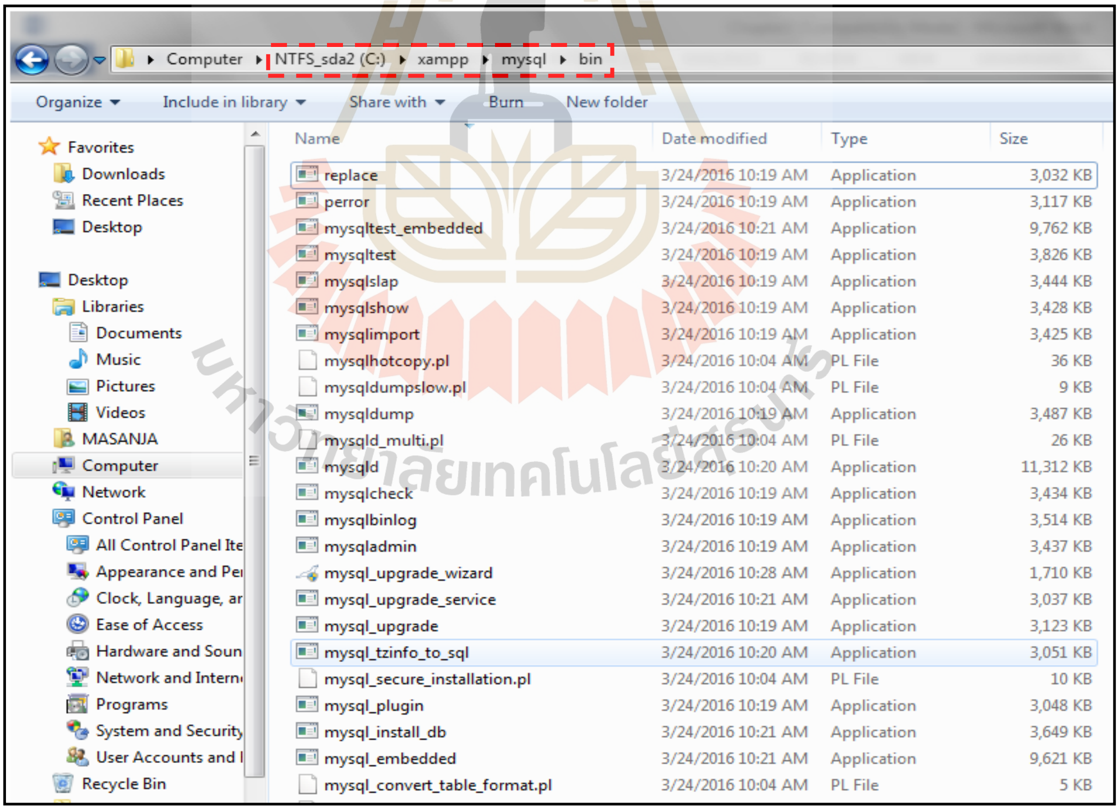


Figure 3.32 Location of the mysql database in XAMPP folder

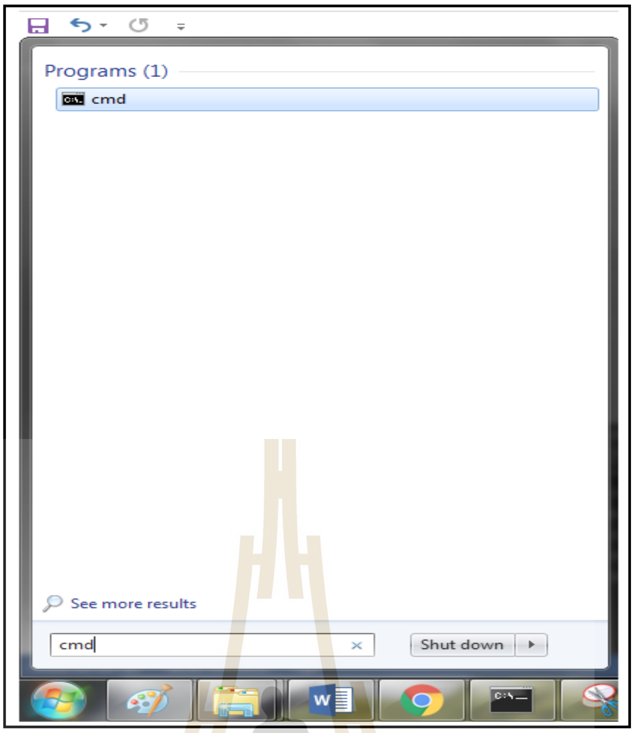


Figure 3.33 Opening the command window

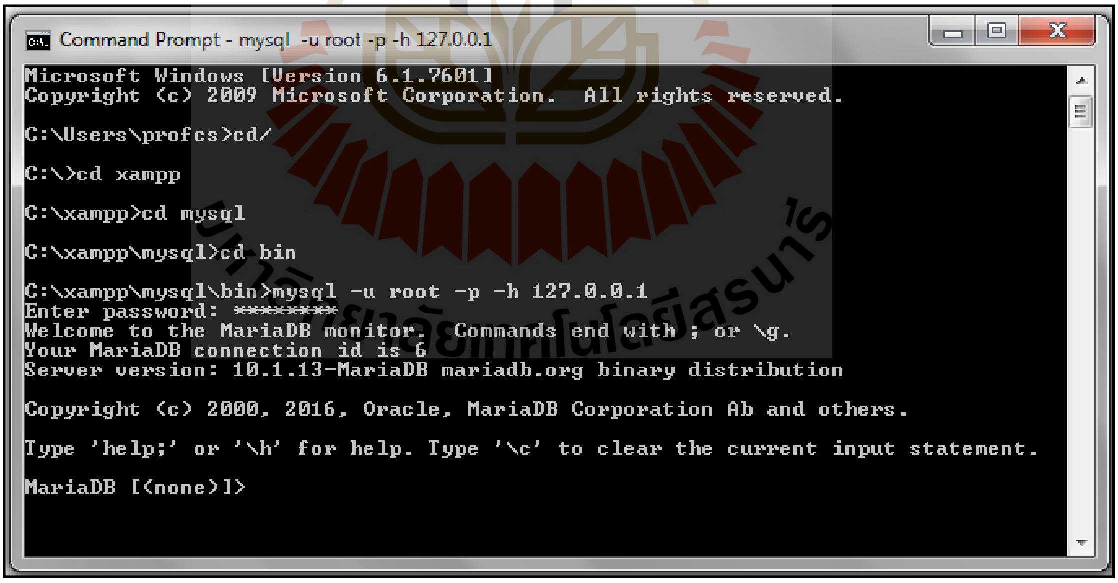


Figure 3.34 Login into the mysql database

As shown in the Figure 3.34 above, after opening the command window the default folder address is to local disk C. For this case, XAMPP folder was out of program files folder. Then user has to change the default address up to bin folder by typing the following:

- cd / (enter)
- cd xampp (enter)
- cd mysql (enter)
- cd bin (enter)

The last address of C:\ xampp\mysql\bin shows that the current location is to the bin folder. User can login to continue with database creation by typing the following address;

- mysql –u root –p –h 127.0.0.1

where

mysql = type of database

-u = username (by default is *root*)

-p = password (by default *no password*)

-h = host (by default is *localhost* or 127.0.0.1)

When it is required to input password again before login (press enter only). Then user will receive a welcome note as shown in Figure 3.34. From Figure 3.35, it can be seen that there are default databases which are ready made in the software. To view the list of databases the command below was used:

- show databases;

Then the new database was ready to be made for storing DGA data. It was stored within the same computer. The new database was named 'dissolved' to reflect the process of dissolved gas-in-oil analysis. The database was created by using the command below:

- create database dissolved;

Figure 3.36 shows a list of databases created and stored in inbuilt computer server including ‘dissolved’ database. One table was created in it as shown in Figure 3.37.

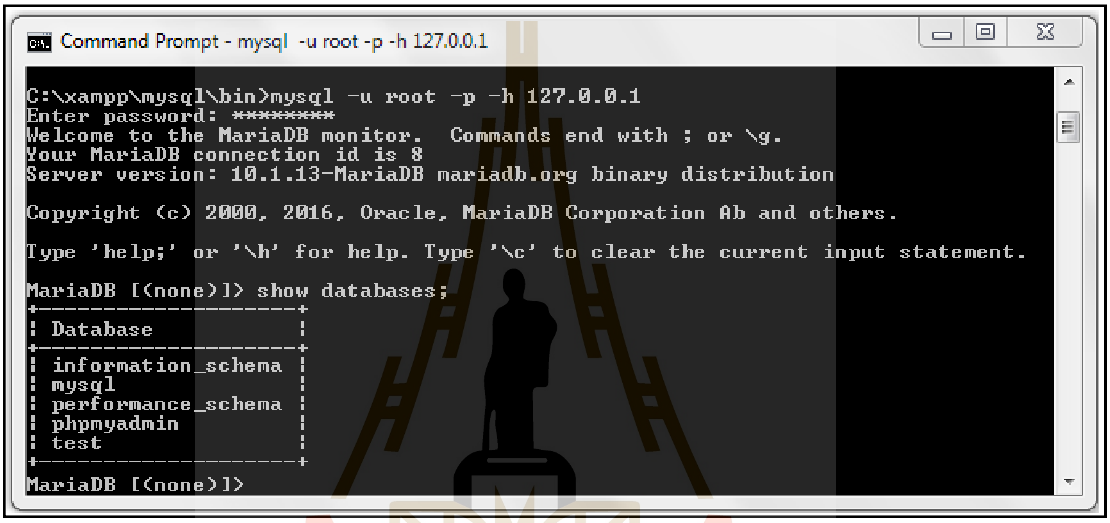


Figure 3.35 List of database

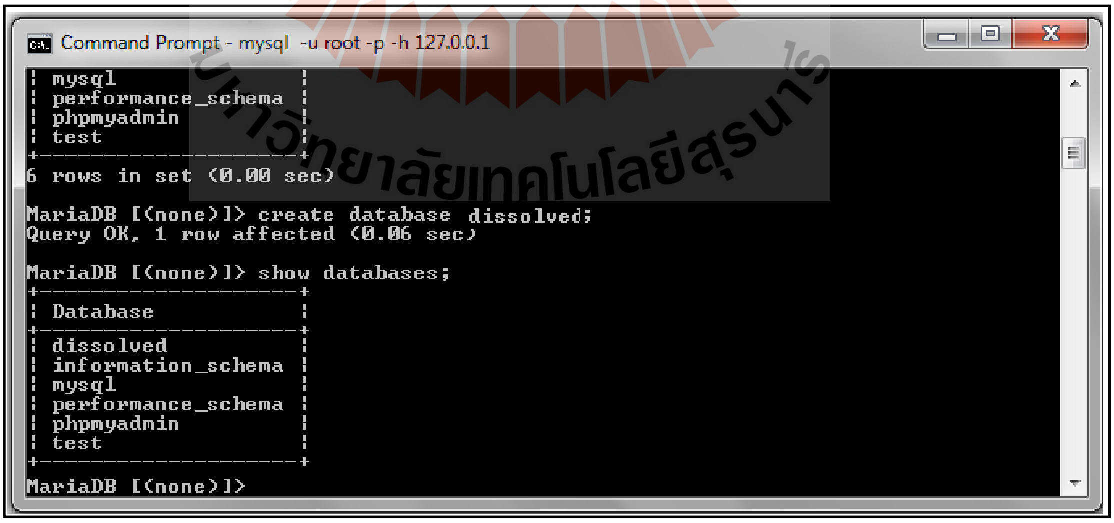


Figure 3.36 Created database (dissolved)

Before creating the table in the database, the database '*dissolved'* was selected for storing this table. The selection of the database was done by the command:

- use *dissolved*;

whereby '*dissolved*' is the name of the database created to store a table with DGA data. Table in the database was created for storing DGA data. The table was named '*new*' in this database and was created by using the following command:

- create table *new* values(*id* (int(6) *primary_key*, *auto_increment*, *hydrogen* varchar(6), *carbon_monoxide* varchar(6), *carbon_dioxide* varchar(6), *nitrogen* varchar(6), *methane* varchar(6), *ethane* varchar(6), *ethylene* varchar(6), *acetylene* varchar(6), *power* varchar(6), *voltage* varchar(6), *oil_temperature* varchar(6), *humidity* varchar(6), *altitude* varchar(6), *m_location* varchar(6), *m_number* varchar (6)).

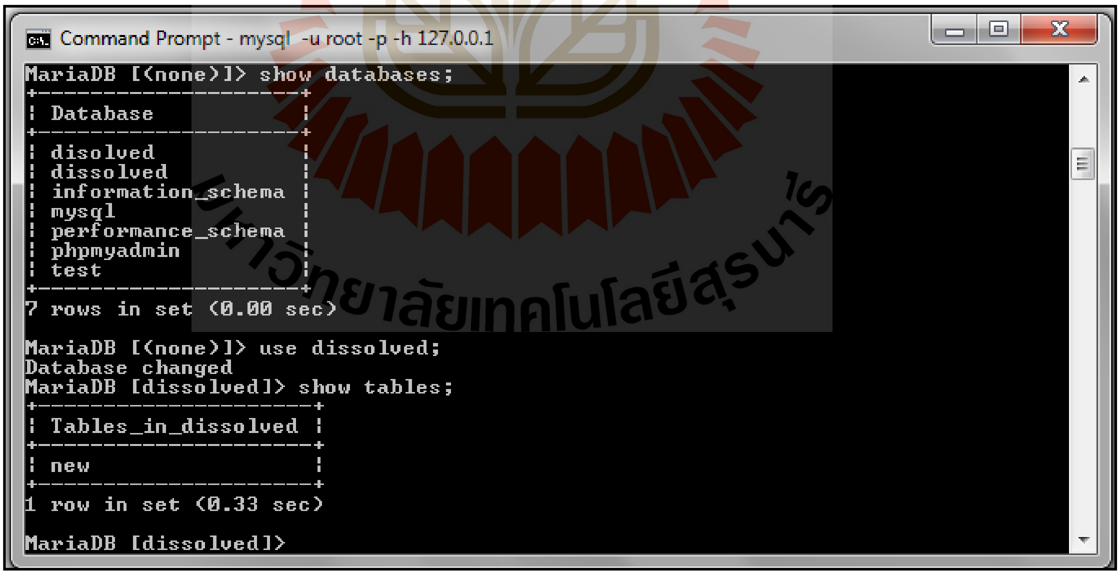


Figure 3.37 Created table in the database.

Where

- *new* = name of the table in the database
- *id* = identification number for differentiating DGA data and easy retrieve
- *int* = data type i.e. integer, 6 is the number of digits specified
- *varchar* = variable character (this data type can hold letters and number)
- *primary_key* = a unique identifier of the specific table
- *auto_increment* = automatic generation of identification number
- *hydrogen* = hydrogen
- *carbon_monoxide* = carbon monoxide
- *carbon_dioxide* = carbon dioxide
- *nitrogen* = nitrogen
- *methane* = methane
- *ethane* = ethane
- *ethylene* = ethylene
- *acetylene* = methane
- *power* = power of the transformer in kVA
- *voltage* = voltage of the transformer in high voltage side, kV
- *oil_temperature* = temperature of the transformer oil during sampling, °C
- *humidity* = relative humidity at the atmosphere before sampling in %
- *altitude* = altitude where transformer is installed, M
- *m_location* = location of the transformer, name
- *m_number* = number of the transformer, number

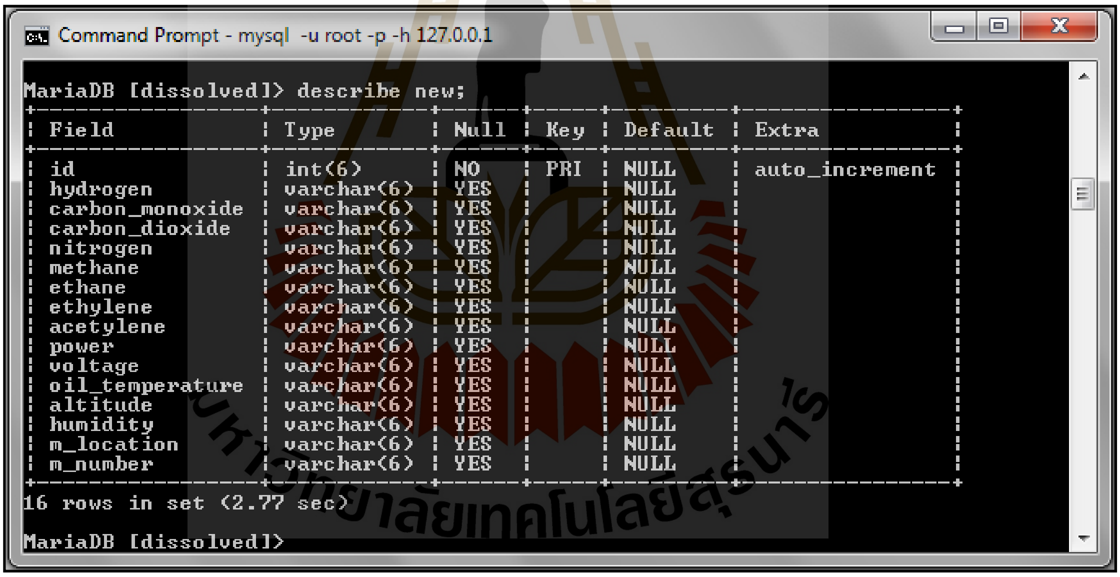
The parameters defined in the command of creating the table contains parameters that match to those shown in the input form of the interface (see Figure 3.7). The details of the created table are requested in the mysql by the following command:

- describe *new*;

where

- *new* = name of the table

The details are shown in the Figure 3.38. For inserting the data in the table, the input must match with the table fields. The following command was used to insert data in the table created inside the dissolved database:



```
Command Prompt - mysql  -u root -p -h 127.0.0.1
MariaDB [dissolved]> describe new;
+------------------+------------+------+-----+---------+----------------+
| Field            | Type       | Null | Key | Default | Extra          |
+------------------+------------+------+-----+---------+----------------+
| id               | int(6)     | NO   | PRI | NULL    | auto_increment |
| hydrogen         | varchar(6) | YES  |     | NULL    |                |
| carbon_monoxide  | varchar(6) | YES  |     | NULL    |                |
| carbon_dioxide   | varchar(6) | YES  |     | NULL    |                |
| nitrogen         | varchar(6) | YES  |     | NULL    |                |
| methane          | varchar(6) | YES  |     | NULL    |                |
| ethane           | varchar(6) | YES  |     | NULL    |                |
| ethylene         | varchar(6) | YES  |     | NULL    |                |
| acetylene        | varchar(6) | YES  |     | NULL    |                |
| power            | varchar(6) | YES  |     | NULL    |                |
| voltage          | varchar(6) | YES  |     | NULL    |                |
| oil_temperature  | varchar(6) | YES  |     | NULL    |                |
| altitude         | varchar(6) | YES  |     | NULL    |                |
| humidity         | varchar(6) | YES  |     | NULL    |                |
| m_location       | varchar(6) | YES  |     | NULL    |                |
| m_number         | varchar(6) | YES  |     | NULL    |                |
+------------------+------------+------+-----+---------+----------------+
16 rows in set (2.77 sec)

MariaDB [dissolved]>
```

Figure 3.38 Details of the created table

- insert into *new*(*hydrogen, carbon_monoxide, carbon_dioxide, nitrogen, methane, ethane, ethylene, acetylene, power, voltage, oil_temperature, humidity, altitude, m_location, m_number*) *values*(12, 25, 30, 5, 2, 5, 3, 5, 1500, 22, 65, 40, 500, Suknives, 9);

where

- *insert into* = mysql command for inserting data into the table
- *new* = name of the table
- *values* = the values of each field inside the table

3.5.4 Connecting MySQL database and Python interface

For DGA data entered by user to be stored in the database, the database must be connected with the interface. There are several connectors used to connect mysql and python. The connector can be obtained through the official website of mysql software as shown in Figure 3.39 in the red dashed box.

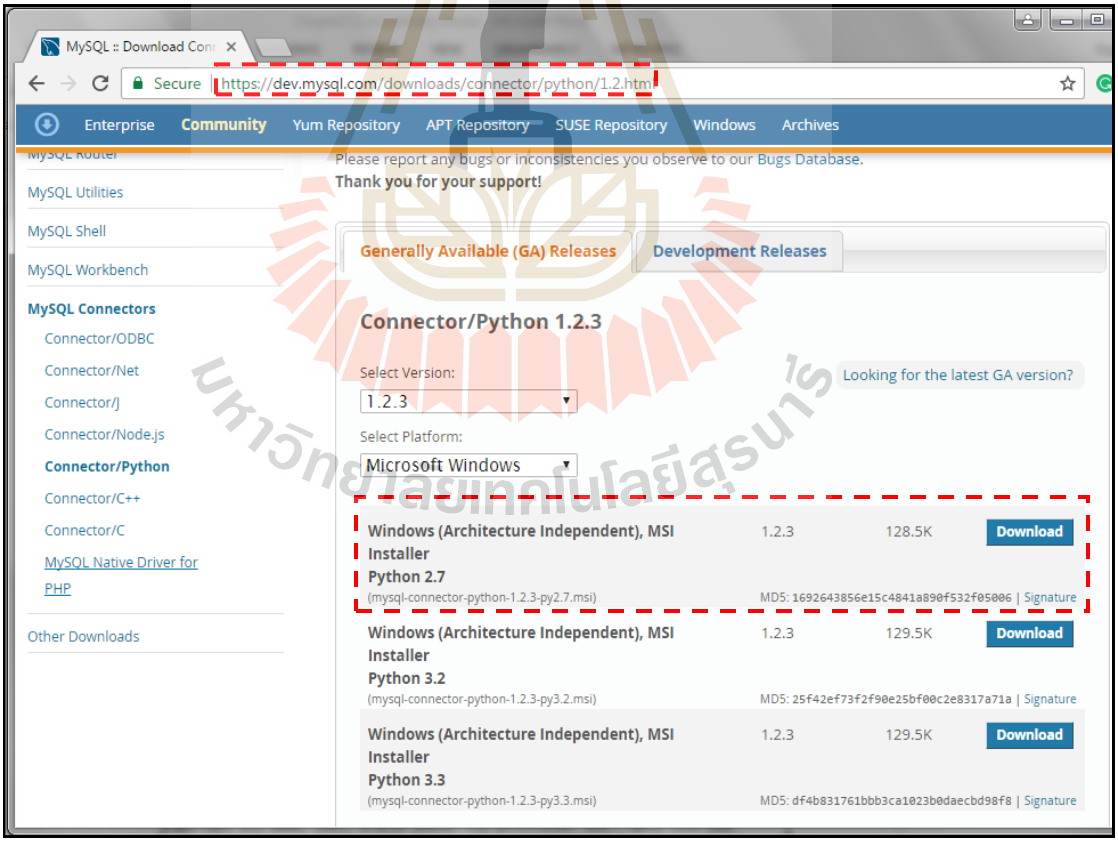


Figure 3.39 MySQL-python connector

MySQL database was connected with the DGA software by using the mysql-python connector. It is a standardized MySQL driver for linking the communication between python and MySQL database. For testing the functionality of the communication after installing the connector, one has to type the following command in the python script:

- import *MySQLdb*

If the connector is not successfully installed it shows an error in the python shell as shown in Figure 3.40 (a). Figure 3.40(b) shows the installation of the connector.

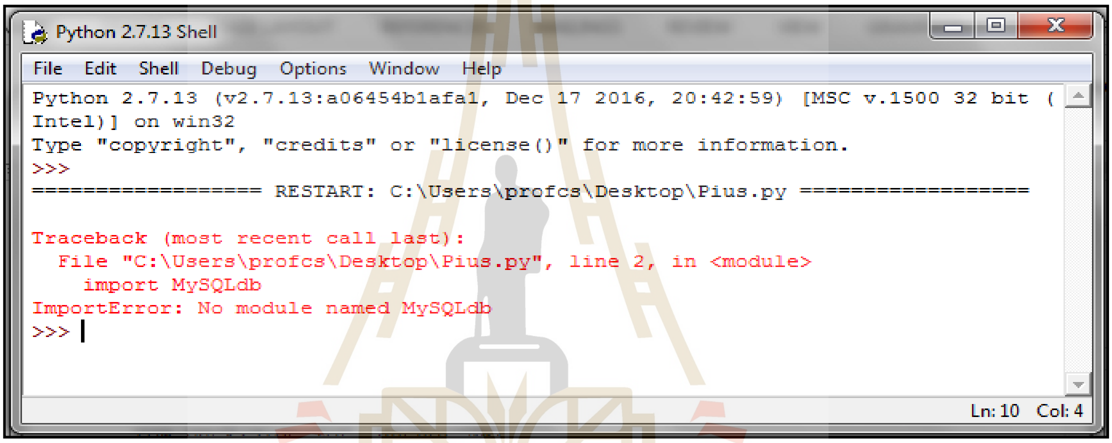


Figure 3.40(a) Unsuccessful installation of MySQL-python connector

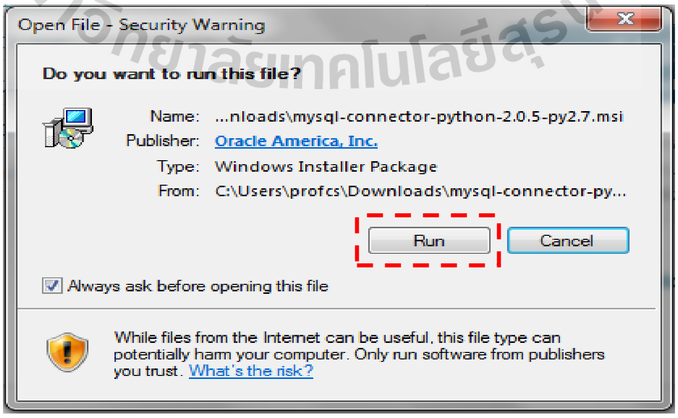


Figure 3.40(b) Running of the MySQL-python connector

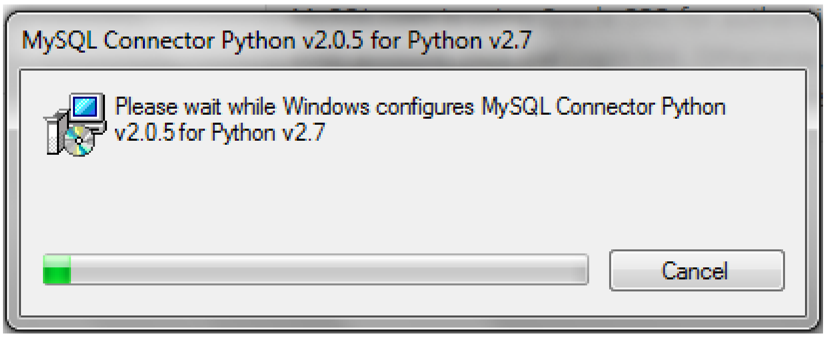


Figure 3.40(c) Installation of MySQL-python connector

After finishing to install the mysql-python connector, a message box was programmed to make sure that user knows the status of the connection. Figure 3.41 shows the status of successful connection from the drop-down list of view menu in the python interface (see Figure 3.9) after running the XAMPP control panel.

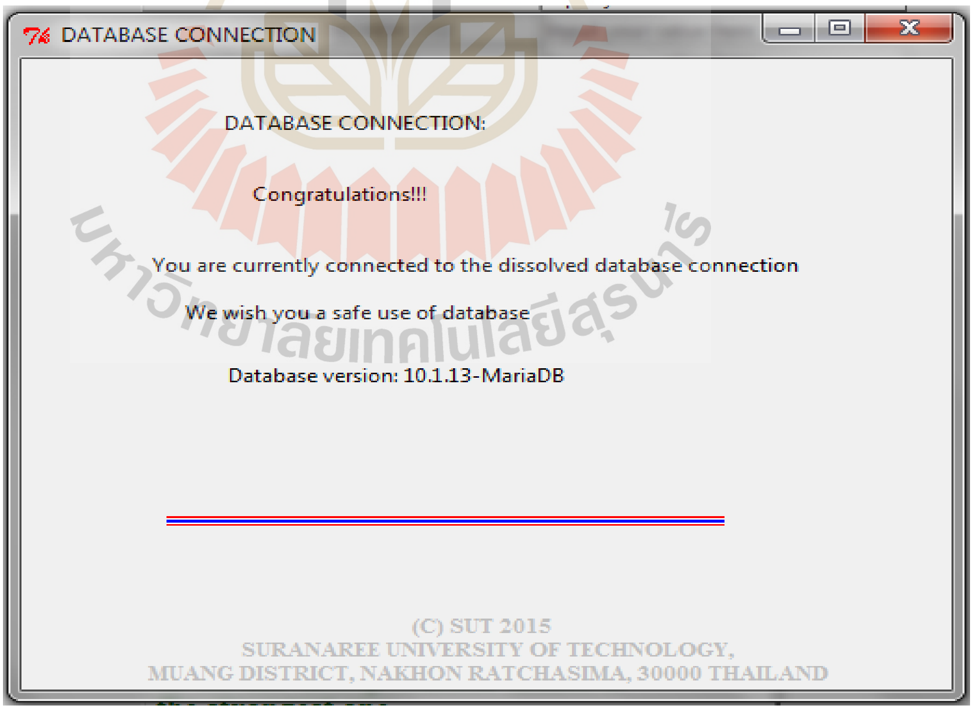


Figure 3.41 Status of the database connection

3.6 Validation of DGA software

DGA data were taken from other researchers, stored in the software database and then interpreted by using five diagnostic methods developed in the software. The total number of 50 DGA data and results were used for comparison with the results from the DGA software. The data from Gouda, Saleh and EL-Hoshy, (2016) and Sherif, Ghoneim, and Sayed, (2012) are shown in Table 3.13.

Table 3.13 DGA data from other researchers (Gouda, Saleh and EL-Hoshy, 2016; Sherif, Ghoneim, and Sayed, 2012)

No	Gasses in ppm						
1	H_2	CH_4	C_2H_2	C_2H_4	C_2H_6	CO	CO_2
2	41	65	0	58	22	130	1453
3	17	66	0	17	9	268	1362
4	20	48	0	186	36	143	533
5	179	306	0	579	73	496	1050
6	73	177	0	52	37	1767	5398
7	210	43	187	102	12	167	1070
8	39	7	0	65	12	1469	2692
9	37	96	0	57	19	132	926
10	46	147	0	150	27	184	2678
11	18	63	0	51	20	390	2633
12	200	30	98	60	9	138	308
13	28	62	0	76	16	538	1206
14	68	78	0	136	103	550	1720
15	54	143	0	101	23	1358	4497
16	19	47	0	62	27	229	1114
17	678	70	237	89	31	768	1909
18	86	277	0	338	63	323	2574
19	20	19	0	77	45	571	1854
20	25	49	0	58	42	424	1814
21	53	150	0	379	64	312	987
22	114	241	0	574	80	1309	5675
23	99	298	0	5376	628	783	2819
24	34	83	0	136	100	645	3197
25	25	132	0	17	61	712	2576
26	13	138	0	16	83	1046	5197
27	762	93	126	54	38	459	5346
28	43	116	0	139	65	718	2326
29	179	306	0	579	73	496	1050

Table 3.13 DGA data from other researchers (Gouda, Saleh and EL-Hoshy, 2016; Sherif, Ghoneim, and Sayed, 2012) (Continued)

No	Gasses in ppm						
	H_2	CH_4	C_2H_2	C_2H_4	C_2H_6	CO	CO_2
30	57	141	0	51	38	1405	5484
31	40	8	0	15	34	2723	3929
32	35	283	0	222	121	1475	11034
33	15	159	0	87	29	1769	11263
34	55	159	0	493	114	1309	11298
35	37	123	0	52	67	869	3896
36	723	191	288	293	110	459	5346
37	7	15	0	58	78	687	4647
38	30	51	0	54	12	803	3566
39	31	56	0	77	33	771	5175
40	109	226	0	192	68	1150	5240
41	137	279	0	505	66	897	3085
42	59	119	0	70	36	735	4738
43	151	242	0	232	68	1076	5418
44	870	77	14	54	73	459	5346
45	376	575	0	1092	146	674	2160
46	17	40	0	86	11	78	823
47	191	47	0	15	43	634	6623
48	154	11	3	8	14	487	3395
49	36	245	0	332	144	538	2326
50	86	187	0	363	136	26	198

3.6.1 Fault codes

Fault codes used from other researchers were F1, F2, F3, F4, F5, F6 and F7. They used them to increase the strength of interpretation and minimize the number of 'No Detection' from diagnostic methods as shown in Table 3.14

Table 3.14 DGA fault codes used by other researchers (Gouda, Saleh and El-Hoshy, 2016)

Method	F1	F2	F3	F4	F5	F6	F7
Duval	Normal	thermal fault <300 °C	thermal fault 300-700°C	thermal fault >700°C	low and high energy discharge	*PD* with mix thermal and electrical	out of code fault

Table 3.14 DGA fault codes used by other researchers (Gouda, Saleh and El-Hoshy, 2016)

<table>
<tr><td rowspan="2">Dorne nburg</td><td rowspan="2">Normal</td><td>out of code</td><td colspan="2" rowspan="2">thermal decomposition</td><td rowspan="2">arcing</td><td rowspan="2">partial discharges</td><td rowspan="2">out of code fault</td></tr>
<tr><td>thermal fault of low temp. < 150 °C</td></tr>
<tr><td rowspan="3">IEC</td><td rowspan="3">Normal</td><td rowspan="2">thermal fault of low temp. 150-300 °C</td><td rowspan="3">thermal fault of medium temp. 300-700 °C</td><td rowspan="3">thermal fault of high temp. > 700 °C</td><td>discharge with low energy</td><td rowspan="3">partial discharge with low energy density</td><td rowspan="3">out of code fault</td></tr>
<tr><td>sparking</td></tr>
<tr><td>thermal fault of low temp. < 150 °C</td><td>discharge with high energy, arcing</td></tr>
<tr><td rowspan="4">Roger's</td><td rowspan="4">Normal</td><td>thermal fault of low temp. 150-200 °C</td><td>Winding circulating current</td><td rowspan="4">insulated conductor overheat</td><td>arc with power follow through</td><td>partial discharge</td><td rowspan="4">out of code fault</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td rowspan="2">thermal fault of low temp. 200-300 °C</td><td rowspan="2">core/tank circulating current</td><td>sparking</td><td rowspan="2">partial discharge with tracking</td></tr>
<tr><td>flashover</td></tr>
<tr><td>Key gas</td><td>Normal</td><td>out of code fault</td><td>overheating of oil</td><td>hotspot</td><td>arcing fault</td><td>partial discharge (corona)</td><td>out of code fault</td></tr>
</table>

Abbreviations of faults detected by each diagnostic method of DGA software are shown in Table 3.15. These abbreviations are the only detected faults from DGA software.

Table 3.15 Abbreviations of faults detected by DGA software

Diagnostic Method	Fault types used for interpretation
Key Gas Method	O.C – overheating in cellulose N.D – no detection
Dornenburg Ratio Method	Normal – normal deterioration Arc. H – arcing (High-Intensity PD) Therm – thermal decomposition PD. L – PD (Low-Intensity PD)
Rogers Ratio Method	Normal – normal deterioration Arc. H – arcing (High-Intensity PD) T<700 – thermal fault T<700 °C T>700 – thermal fault T<700 °C
IEC Ratio Method	Dis. L – discharge of low energy T<150 – thermal fault T<150 °C 150-300 – thermal fault 150<T<300 °C 300-700 – thermal fault 300<T<700 °C PD. L – PD (Low-Intensity PD)
Duval Triangle Method	Dis. L – discharge of low energy Dis. H – discharge of high energy T<300 – thermal fault T<300 °C 300-700 – thermal fault 300<T<700 °C T>700 – thermal fault T<700 °C Mixed – mixed fault (electrical and thermal)

The summary of results from DGA software as compared from others results are shown in Table 3.16.

Table 3.16 Comparison of results from DGA software and other researchers

No	Other researchers					DGA software				
	KGM	DON	ROG	IEC	DUV	KGM	DON	ROG	IEC	DUV
1	F7	F1	F3	F3	F3	ND	ND	T<700	ND	300-700
2	F7	F1	F7	F3	F1	ND	Normal	T<700	300-700	T<300
3	F7	F1	F3	F4	F4	ND	ND	T>700	ND	T>700
4	F7	F3, F4	F3	F4	F4	ND	ND	T>700	ND	T>700
5	F7	F1	F3	F3	F3	ND	ND	T<700	ND	300-700
6	F7	F1	F7	F5	F5	ND	Arc. H	Arc. H	Dis. L	Dis. H
7	F4	F1	F7	F7	F4	O.C	ND	ND	ND	T>700
8	F7	F1	F3	F3	F3	O.C	ND	ND	ND	300-700

Table 3.16 Comparison of results from DGA software and other researchers

(Continued)

9	F7	F1	F7	F4	F4	ND	ND	T>700	T>700	300-700
10	F7	F1	F7	F3	F3	ND	ND	T<700	300-700	300-700
11	F7	F1	F7	F5	F5	ND	Arc.H	Arc. H	Dis. L	Dis. H
12	F7	F1	F3	F4	F4	ND	ND	T>700	ND	T>700
13	F7	F3, F4	F7	F3	F4	ND	ND	T<700	ND	T>700
14	F7	F3, F4	F3	F4	F3	ND	ND	T>700	ND	300-700
15	F7	F1	F3	F3	F4	ND	ND	T<700	ND	T>700
16	F7	F5	F7	F5	F5	ND	ND	NP	Dis. L	Dis. L
17	F7	F1	F7	F4	F4	ND	ND	T>700	T>700	T>700
18	F4	F1	F7	F2	F4	ND	ND	LTTO	T<150	T>700
19	F7	F1	F3	F3	F4	ND	ND	T<700	ND	T>700
20	F7	F1	F3	F4	F4	ND	ND	T>700	ND	T>700
21	F7	F3, F4	F3	F4	F4	ND	ND	T>700	ND	T>700
22	F7	F3, F4	F3	F4	F4	ND	ND	T>700	ND	T>700
23	F7	F3, F4	F3	F4	F4	ND	Therm	T>700	ND	T>700
24	F7	F3, F4	F7	F3	F4	ND	ND	T<700	ND	T>700
25	F7	F1	F2	F2	F3	ND	ND	ND	150-300	T<300
26	F7	F1	F2	F2	F3	ND	ND	ND	150-300	T<300
27	F7	F5	F7	F5	F5	ND	Arc.H	ND	Dis. L	Dis. L
28	F7	F3, F4	F3	F3	F4	ND	ND	T<700	ND	T>700
29	F7	F3, F4	F3	F4	F4	ND	ND	T>700	ND	T>700
30	F7	F1	F3	F3	F3	O.C	ND	T<700	ND	300-700
31	F4	F1	F7	F1	F1	O.C	ND	Normal	ND	T>700
32	F7	F3, F4	F7	F3	F3	ND	ND	T<700	300-700	300-700
33	F7	F1	F7	F3	F3	ND	ND	ND	300-700	300-700
34	F7	F3, F4	F3	F4	F4	O.C	ND	T>700	ND	T>700
35	F7	F1	F2	F2	F3	ND	ND	ND	150-300	300-700

Table 3.16 Comparison of results from DGA software and other researchers

(Continued)

36	F7	F5	F7	F5	F5	ND	Arc.H	ND	Dis. L	Mixed
37	F7	F1	F2	F2	F4	ND	ND	ND	ND	T>700
38	F7	F1	F3	F4	F4	ND	ND	T>700	ND	300-700
39	F7	F1	F3	F3	F4	ND	ND	T<700	ND	T>700
40	F7	F3, F4	F3	F3	F3	ND	ND	T<700	ND	300-700
41	F7	F3, F4	F3	F4	F4	ND	ND	T>700	ND	T>700
42	F7	F1	F3	F3	F3	O.C	ND	T<700	ND	300-700
43	F7	F3, F4	F3	F4	F3	ND	ND	T>700	ND	300-700
44	F7	F6	F6	F7	F5	ND	PD. L	ND	PD. L	300-700
45	17	F3, F4	F3	F4	F4	O.C	ND	T>700	ND	T>700
46	F7	F1	F3	F4	F4	ND	ND	T>700	ND	T>700
47	Dis L	NP	NP	NP	300-700	ND	ND	Normal	ND	300-700
48	NP	NP	NP	NP	Arch H	ND	PD. L	ND	PD. L	300-700
49	NP	NP	NP	363	T>700	ND	ND	T<700	300-700	T>700
50	NP	NP	NP	NP	T>700	ND	ND	T<700	ND	T>700

3.6.2 Percentage of successful detection and consistency

The number of detections done by five diagnostic methods from other researchers with DGA data in Table 3.13 was used to calculate percentages of successful interpretation (*P*) and consistencies (*C*) and summarized in Table 3.17. The formulas are shown in equation (3.22) and equation (3.23) as referred in Muhamad, Phung, Blackburn, and Lai, (2007).

$$\%\ \text{Success detection} = 100 \times \text{Number of detections} / \text{Number of cases} \quad (3.22)$$

$$\%\ \text{Consistency} = 100 \times \text{Number of correct detections} / \text{Number of detections} \quad (3.23)$$

As taking an example of interpretations done by diagnostic methods in Table 3.13, calculations of percentage successful detections as per equation (3.22) were done and summarized in Table 3.17:

For key gas method:

$$\% \text{ Successful detection} = (100 \times 9) / 50 = 18\%$$

For Dornenburg ratio method:

$$\% \text{ Successful detection} = (100 \times 46) / 50 = 92\%$$

For Roger's ratio method:

$$\% \text{ Successful detection} = (100 \times 30) / 50 = 60\%$$

For IEC ratio method:

$$\% \text{ Successful detection} = (100 \times 44) / 50 = 88\%$$

For Duval triangle method:

$$\% \text{ Successful detection} = (100 \times 50) / 50 = 100\%$$

Also taking an example of interpretations done by diagnostic methods in Table 3.13, calculations of percentages of consistency as per equation (3.23) were done and summarized in Table 3.17:

For key gas method:

$$\% \text{ Consistency} = (100 \times 3) / 4 = 75\%$$

For Dornenburg ratio method:

$$\% \text{ Consistency} = (100 \times 20) / 46 = 43\%$$

For Roger's ratio method:

$$\% \text{ Consistency} = (100 \times 18) / 30 = 60\%$$

For IEC ratio method:

$$\% \text{ Consistency} = (100 \times 38) / 44 = 86\%$$

For Duval triangle method:

$$\% \text{ Consistency} = (100 \times 44) / 50 = 88\%$$

Table 3.17 Summary of diagnosis performance from other researchers

Method	**Predictions**	**% Prediction, *P***	**Correct prediction**	**% Consistency, *C***
Key gas	9	18	7	77
Dornenburg	46	92	20	43
Roger's	30	60	18	60
IEC	44	88	38	86
Duval triangle	50	100	44	88

In DGA software, calculations of successful detection and consistency were done as per equation (3.22) and (3.23) respectively. The summary of interpretation performed in five diagnostic methods is shown in Table 3.18.

For key gas method:

$$\% \text{ Successful detection} = (100 \times 7) / 50 = 14\%$$

For Dornenburg ratio method:

$$\% \text{ Successful detection} = (100 \times 8) / 50 = 16\%$$

For Roger's ratio method:

$$\% \text{ Successful detection} = (100 \times 38) / 50 = 76\%$$

For IEC ratio method:

$$\% \text{ Successful detection} = (100 \times 18) / 50 = 36\%$$

For Duval triangle method:

$$\% \text{ Successful detection} = (100 \times 50) / 50 = 100\%$$

And their consistencies as per (28) were calculated as follows:

For key gas method:

$$\% \text{ Consistency} = (100 \times 5) / 7 = 71\%$$

For Dornenburg ratio method:

$$\% \text{ Consistency} = (100 \times 3) / 8 = 37\%$$

For Roger's ratio method:

$$\% \text{ Consistency} = (100 \times 21) / 38 = 55\%$$

For IEC ratio method:

$$\% \text{ Consistency} = (100 \times 15) / 18 = 83\%$$

For Duval triangle method:

$$\% \text{ Consistency} = (100 \times 42) / 50 = 84\%$$

Table 3.18 Summary of diagnosis performance from DGA software

Method	**Detected**	**% Detection, *P***	**Correct detection**	**% Consistency, *C***
Key gas	7	14	5	71
Dornenburg	8	16	3	37
Roger's	38	76	21	55
IEC	18	36	15	83
Duval triangle	50	100	42	84

3.6.3 Accuracies of diagnostic methods of DGA software

Furthermore, the accuracy of each diagnostic method from DGA software was evaluated. This defines the degree to which the result of a measurement, calculation or specification conforms to the correct value or standard. According to Muhamad, Phung, Blackburn, and Lai, (2007) accuracy was divided into two categories. The first category involves accuracy based only on detected cases (A_p) while the second one involves only the total number of cases (A_T). Both cases were calculated by using equation (3.24) and (3.25) respectively.

% Accuracy based on detection = 100×correct detections / Number of detections

$$A_P = (100 \times T_R) / T_P \tag{3.24}$$

% Accuracy based on total cases = 100×correct detections / Number of cases

$$A_T = (100 \times T_R) / T_C \tag{3.25}$$

As taking an example of interpretations by done diagnostic methods in Table 3.18, calculations of accuracy based on detection and total cases as per (3.24) and (3.25) respectively were done as follow:

For key gas method:

$$A_P = (100 \times 3) / 4 = 75\%$$

$$A_T = (100 \times 4) / 50 = 8\%$$

For Dornenburg ratio method:

$$A_P = (100 \times 4) / 9 = 44\%$$

$$A_T = (100 \times 9) / 50 = 18\%$$

For Roger's ratio method:

$$A_P = (100 \times 26) / 38 = 68\%$$

$$A_T = (100 \times 38) / 50 = 76\%$$

For IEC ratio method:

$$A_P = (100 \times 16) / 18 = 89\%$$

$$A_T = (100 \times 18) / 50 = 36\%$$

For Duval triangle method:

$$A_P = (100 \times 45) / 50 = 90\%$$

$$A_T = (100 \times 50) / 50 = 100\%$$

Table 3.19 shows the summary of accuracies of the diagnostic method from DGA software.

Table 3.19 Summary of accuracies of diagnostic methods from DGA software

	Key gas	**Dornenburg**	**Roger's**	**IEC**	**Duval**
Total Cases, T_C	50	50	50	50	50
No Prediction, T_{NP}	46	41	12	32	0
Number of Predictions, T_P	4	9	38	18	50
Correct Predictions, T_R	3	4	26	16	45
Incorrect Predictions, T_W	1	5	12	2	5
Accuracy (Predicted cases), A_P	75	44	68	89	90
Accuracy (Total cases), A_T	8	18	76	36	100

The accuracies A_P of each diagnostic method of DGA software were plotted in the chart and shown in Figure 3.42. The accuracy A_p indicates how accurate the method can detect the fault from the levels of given dissolved gasses.

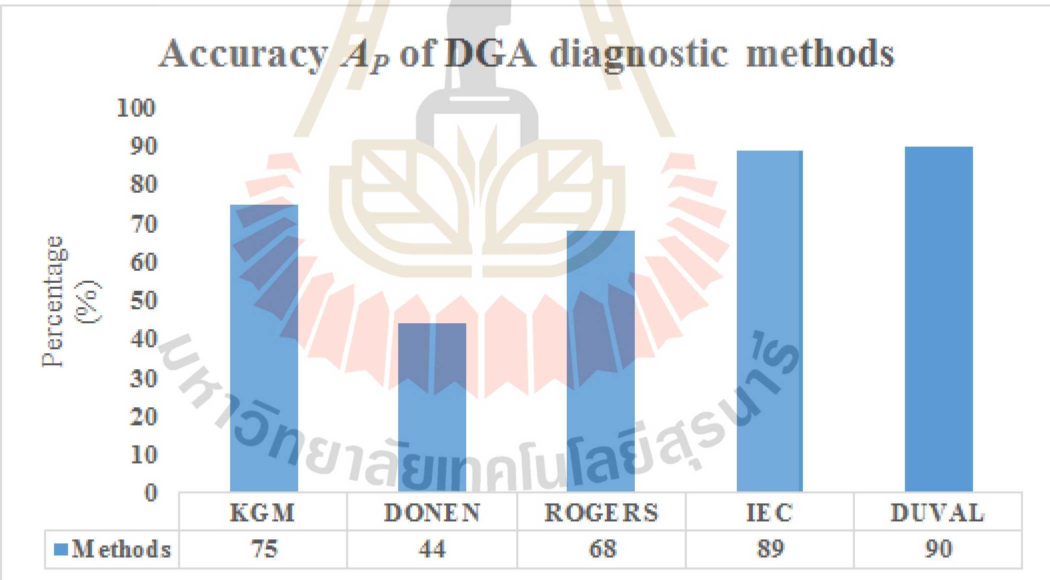


Figure 3.42 Accuracies A_P of diagnostic methods of DGA software

The comparison of accuracies between diagnostic methods of DGA software and from others researchers are shown in Table 3.20. The initials in Table 3.20 can be compared from the previous Table 3.19. As seen from Table 3.20, DGA represents results of DGA software while OR represents those of other researchers.

Table 3.20 Comparison of accuracies of DGA and other researchers

	Key Gas		Dorn		Rog		IEC		Duval	
	DGA	OR	DGA	OR	DGA	OR	DGA	OR	DGA	OR
T_C	50	50	50	50	50	50	50	50	50	50
T_{NP}	46	46	41	4	12	20	32	6	0	0
T_P	4	4	9	46	38	30	18	44	50	50
T_R	3	3	4	20	26	18	16	38	45	44
T_W	1	1	5	14	12	12	2	6	5	6
A_P	75	75	44	43	68	60	89	86	90	88

The summary of comparison of accuracies A_P from Table 3.20 was plotted in the chart as shown in Figure 3.43 for each diagnostic method from DGA software and other researchers systems.

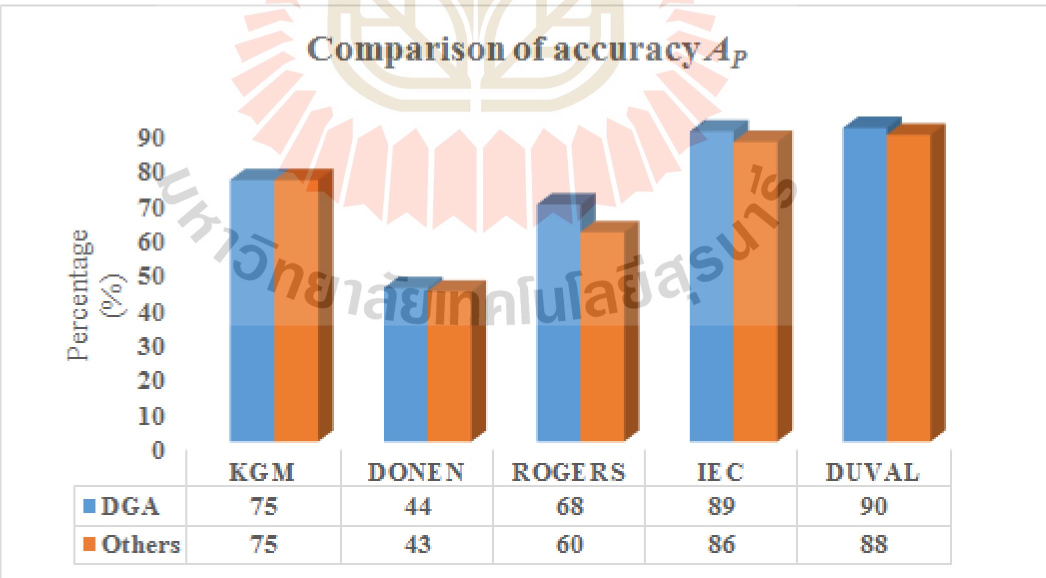


Figure 3.43 Comparison of accuracy A_P of DGA software and other researchers

Table 3.21 Errors of DGA diagnostic methods

Method	$A_{P,DGA}$	$A_{P,OR}$	$(A_{P,DGA} - A_{P,OR}) / A_{P,OR}$, **(%)**
Key gas	75	75	0.000
Dornenburg	44	43	0.023
Roger's	68	60	0.133
IEC	89	86	0.035
Duval triangle	90	88	0.023

3.7 Conclusion of the validation

From Figure 3.43 Duval triangle was seen to have greatest accuracy A_P based on detected faults followed by IEC (89%), key gas (75%), Roger's (68%) and Dornenburg (44%) when compared to other researchers systems. Also, four out of five diagnostic methods showed better in terms of errors as shown in Table 3.21. Therefore, the DGA software was conformed to be used for real practices.

3.8 Compiling the software

After finishing to make python codes, they need to be compiled to be an executable file. An executable file is a file which is intended to perform a specified task (s) since it contains compiled codes. In order to make an executable file, first, inside the codes the first line must be a '*shebang line*' or sometimes called '*bang line*'. It is nothing but an absolute path to the bash interpreter. It starts with a number sign and an exclamation mark (#!) followed by the full path to the interpreter like /bin/ as shown in the Figure 3.44. Then, the python script should be saved in the same folder with of python installation. Another file named '*setup.py*' should be made and saved in the same python folder for guiding the process of making executable file. An icon of 16×16 or 32×32 pixels may be useful for making an image of executable file.

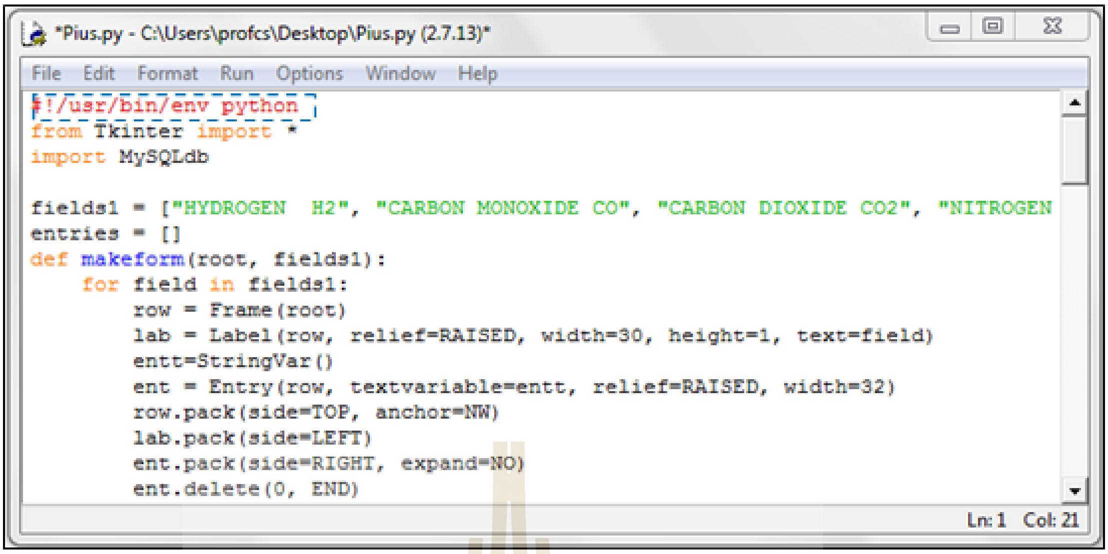


Figure 3.44 Shebang line in the python script

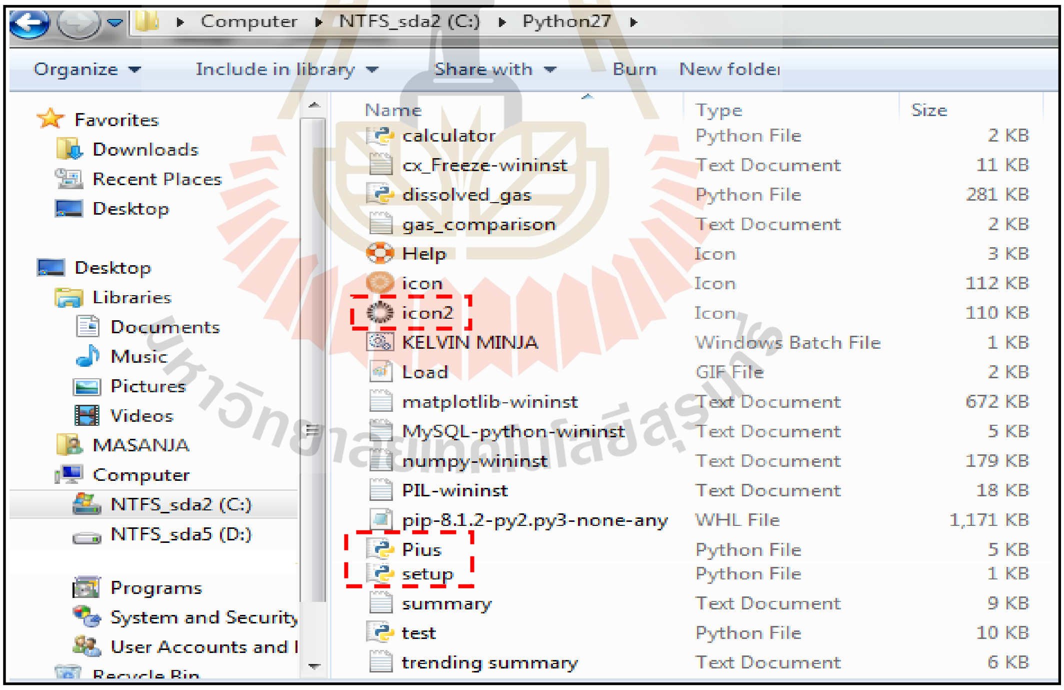


Figure 3.45 Preparing a script, setup file and icon in the python folder

An icon named *icon2* with *.ico* extension was used as an icon for the compiled script. There are various website where icons can be found. One of the useful website include https://www.iconfinder.com. The setup file contains some details about the script to be compiled especially names and location as well as an icon to be included. The contents of the used *setup.py* file are shown in Figure 3.46. As seen in Figure 3.46, the file contains the title of the script, script file, icon name, author's name,

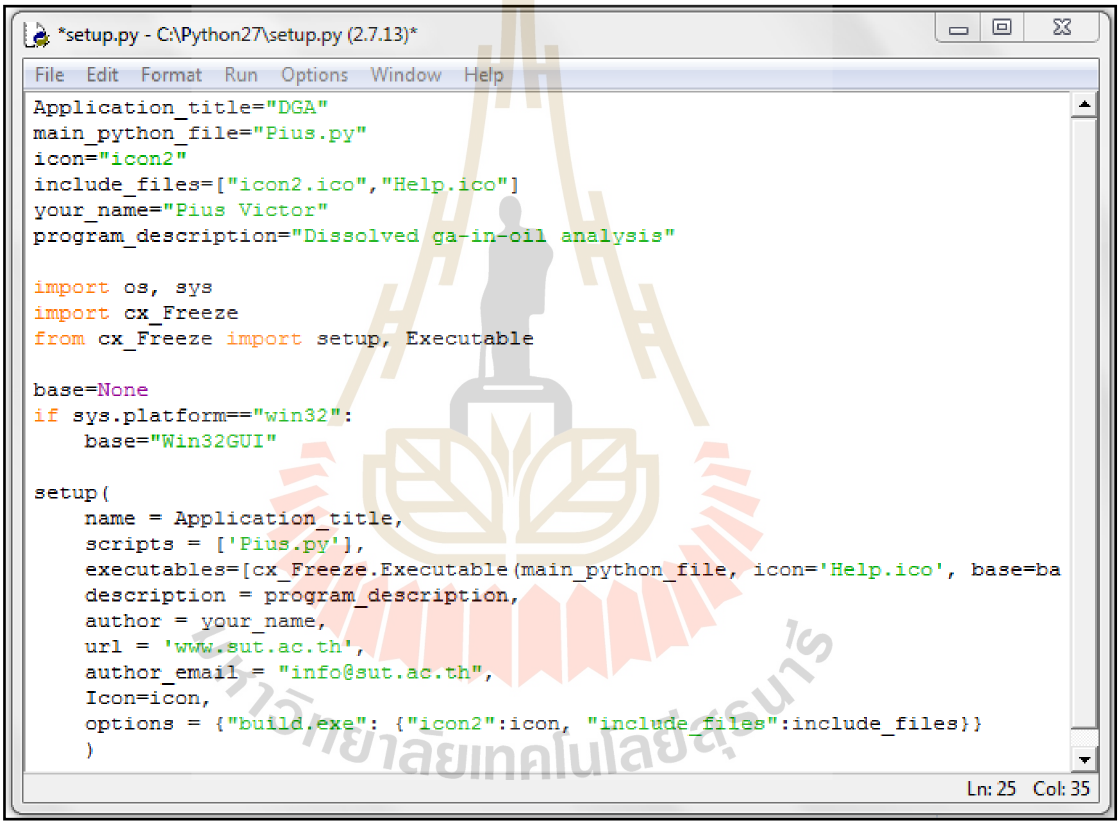


Figure 3.46 Codes of the setup file

name of the program, system type (win 32) and options to include other files. Another file included is a '*cx_Freeze*'. This file contains a set of scripts and modules which find useful in freezing python codes into an executable file. There are variety of files performing the similar task like *cx_Freeze* such as *py2exe* and *pyapp*. The only advantage of *cx_Freeze* is the cross platform application. It works in any platform where python can work. The official website for downloading *cx_Freeze* is shown in Figure 3.47. It comes with a variety of versions depending on the system type (32-bit or 64-bit) and file extension needed (*.exe* or *.whl*). If one has downloaded *.exe* file

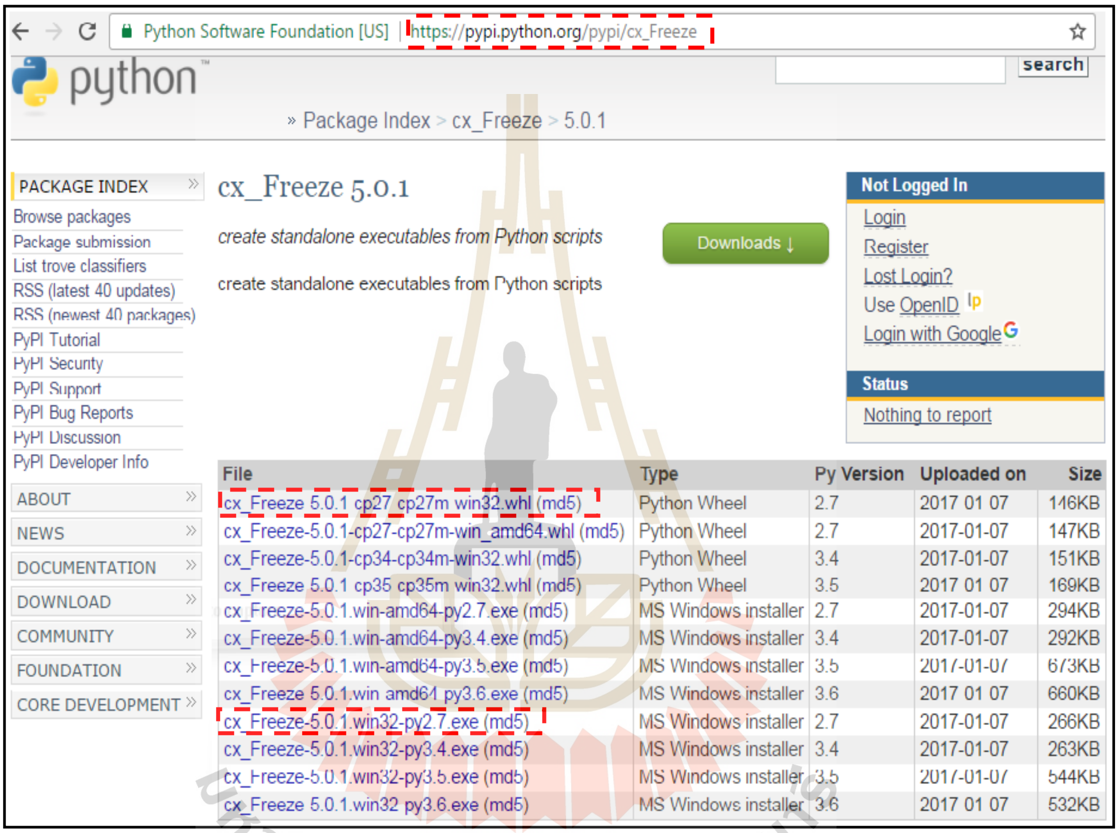


Figure 3.47 Downloading the *cx_Freeze* file

the file should be run as an administrator and it is automatically installed in the python folder. For *.whl* one must use *pip* to install it. After installing the *pip*, through the command window, one must type the following command to install the *.whl* *cx_Freeze*:

- *pip* install (file location) + (file name).*whl*   (enter)

For this case the *cx_Freeze* file downloaded was saved in the download folder of the computer with name *cx_Freeze-4.3.4-cp27-none-win32.whl*. The command entered in the command window is shown below:

- pip install C:\Users\profcs\Downloads cx_Freeze-4.3.4-cp27-none-win32.whl (enter)

Then the cx_Freeze file was ready to be used to compile the python script. With the use of *.exe* file of *cx_Freeze* it provides the simple installation shown in Figure 3.48.

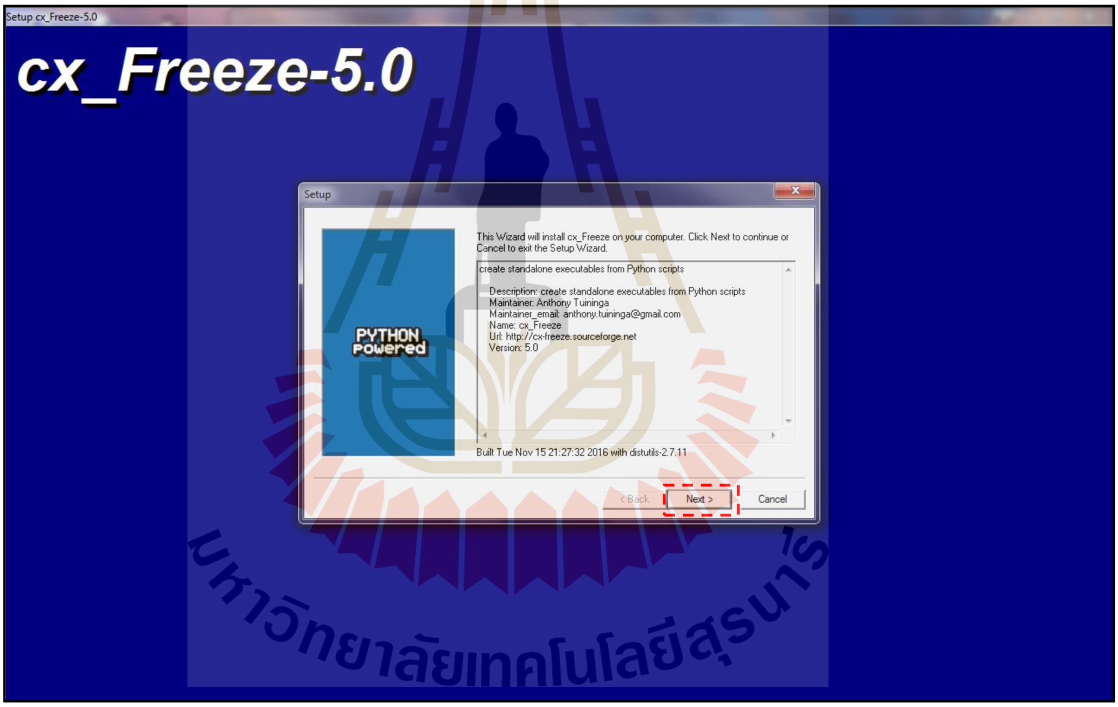


Figure 3.48 Running the *cx_Freeze* with *.exe* extension

From the python file the command window was opened in order to get the direct address of the python folder. This was simply done by opening the python folder, clicking the python script, hold the shift key in the keyboard and right click the mouse to get the option to open the command window as shown in the Figure 3.49. As it is

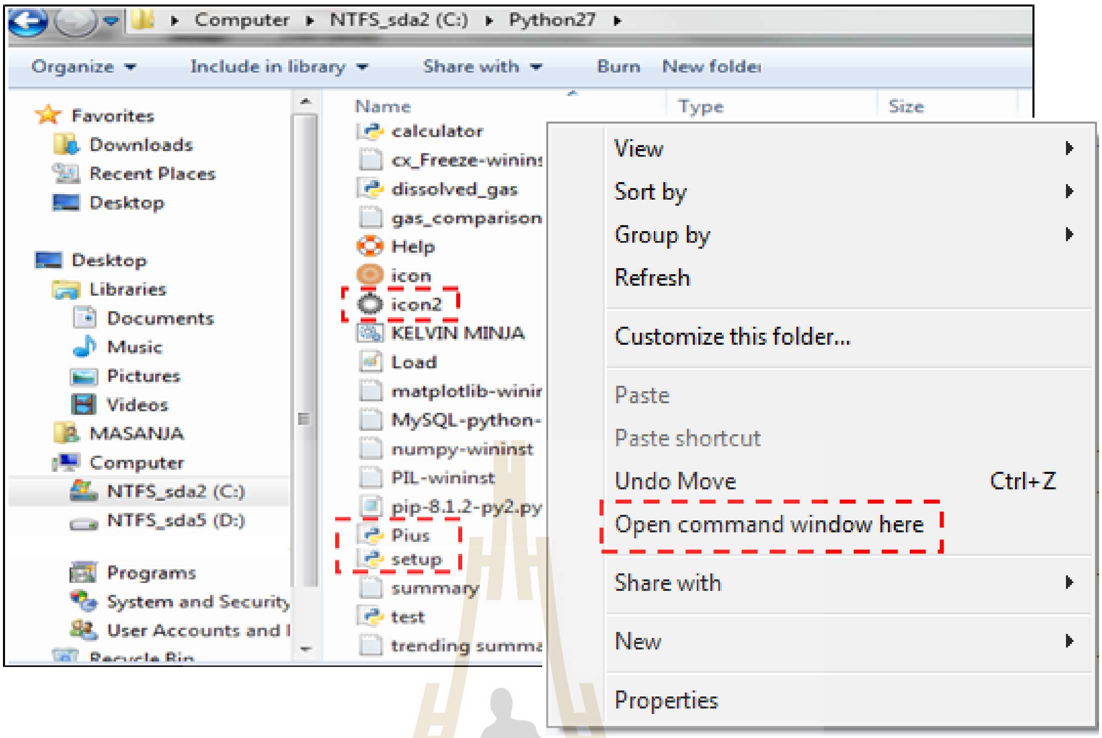


Figure 3.49 Opening the command window in the python folder

shown in the Figure 3.49, the *setup.py* file was highlighted and right-clicked to open the command window. The command used to compile the python codes is shown in

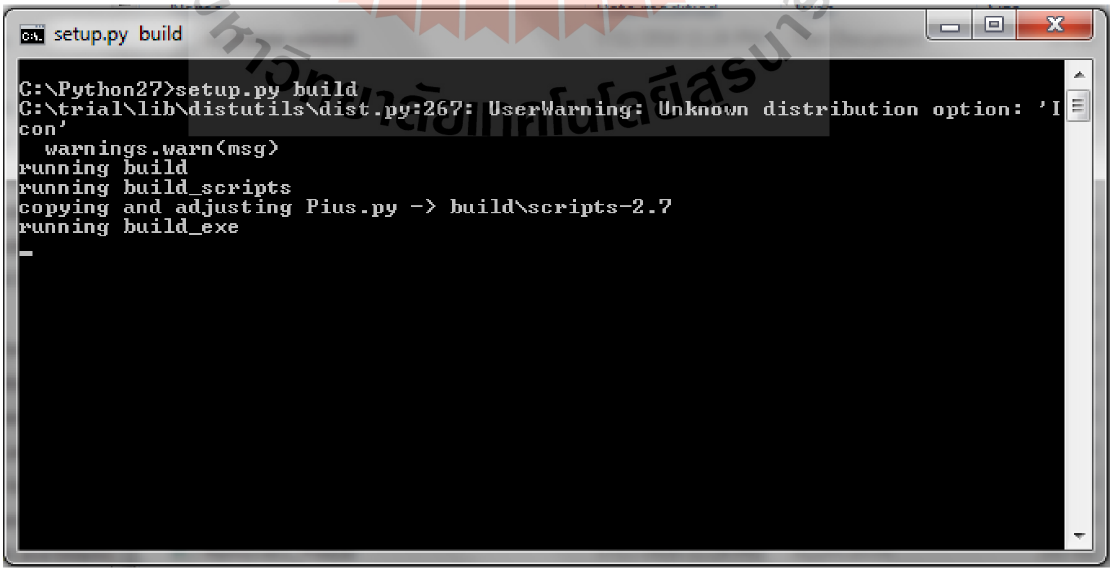


Figure 3.50 Compiling *setup.py*

Figure 3.50. The command used is shown below:

- *setup.py build*

where by the *setup.py* is the script used to guide the compilation of python codes. Figure 3.50 shows the initial process while compiling the setup.py in the command window. Figure 3.51 shows the last step of compiling the setup.py file in the command window.

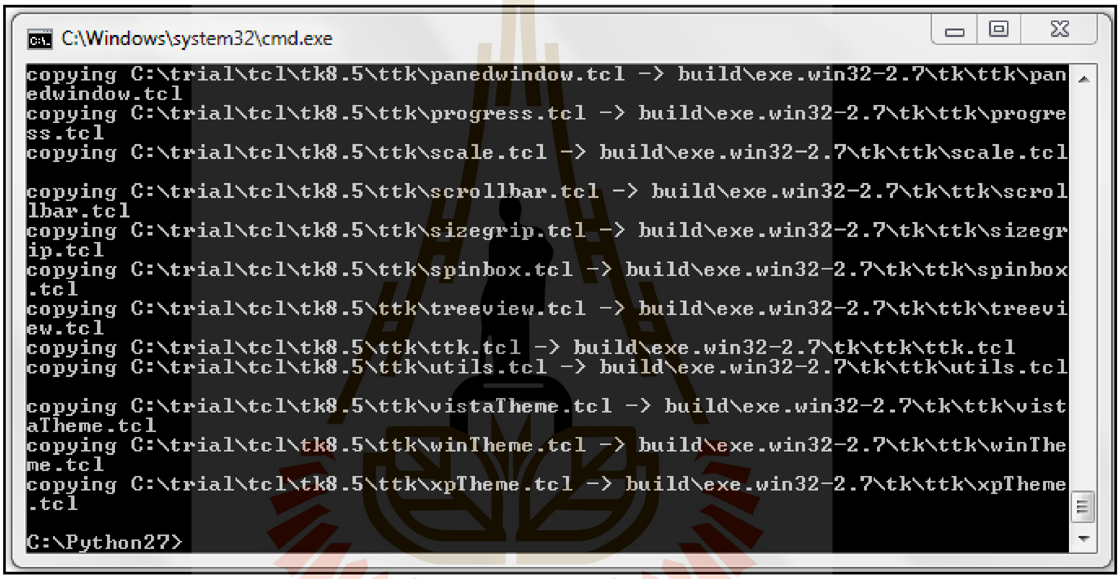


Figure 3.51 End of compilation python script

The compiled files are kept in the new folder named *build*. The *build* folder is shown in Figure 3.52 in red dashed-box. Inside the *build* folder two folders are occupied such as *exe.win*32-2.7 folder and *script*-2.7 folder. The compiled python script file is kept inside the *exe.win*32-2.7 folder. In order for the compiled script to be used as an application, compiled script must be compressed. The mostly used application for compressing files is *win-rar* application. The official website for obtaining the win-rar application is shown in Figure 3.53. The setup was downloaded and installed for

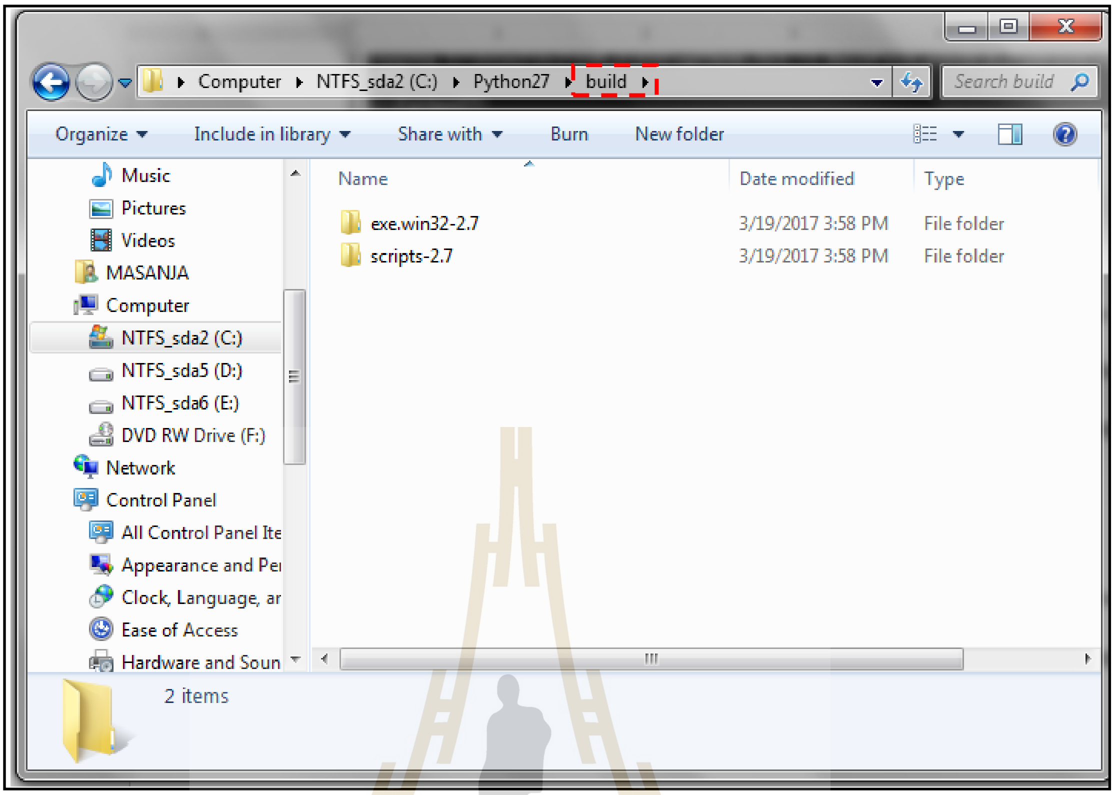


Figure 3.52 Folders compiled in the *build* folder

compressing files. First, all files in the *exe.win*32-2.7 folder were highlighted and added in the archive as shown in the Figure 3.54. By compressing the files it allows the application to be bounded together with all necessary libraries. Figures 3.54-3.61 show the whole process of compressing compiled python scripts. The complied python files can be copied to any computer and can work satisfactorily. The setup of the DGA software when copied to the desktop of this computer is shown in Figure 3.62. Figure 3.63 shows the interface of the DGA software when opened on this computer.

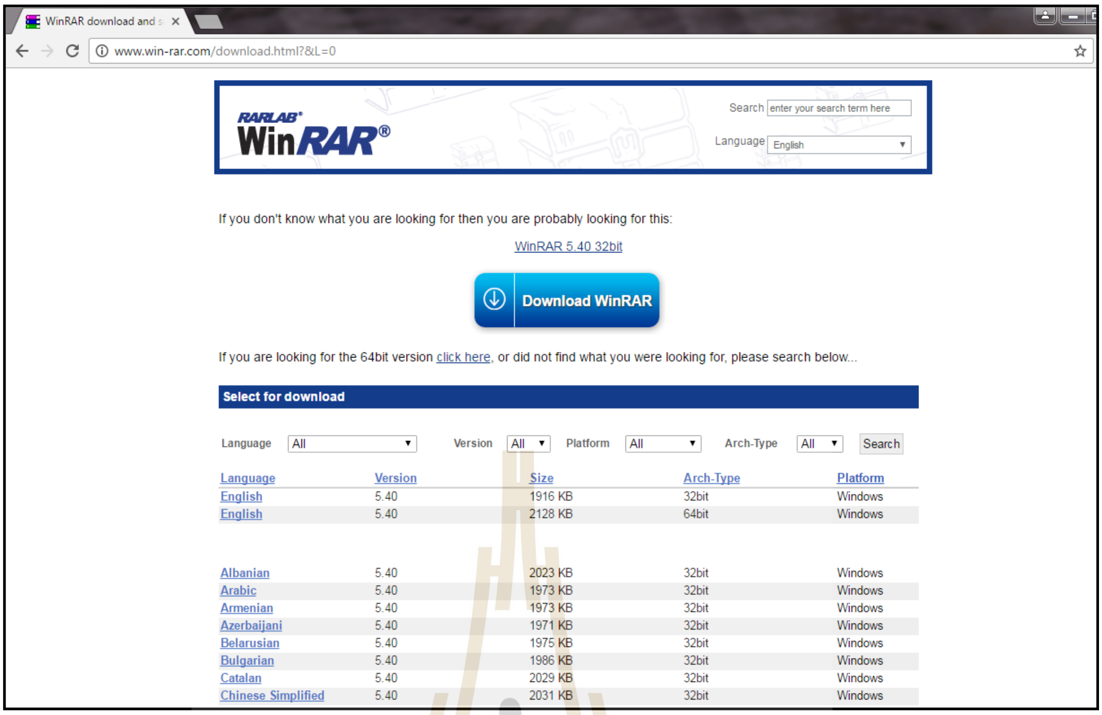


Figure 3.53 Official website of the win-rar software

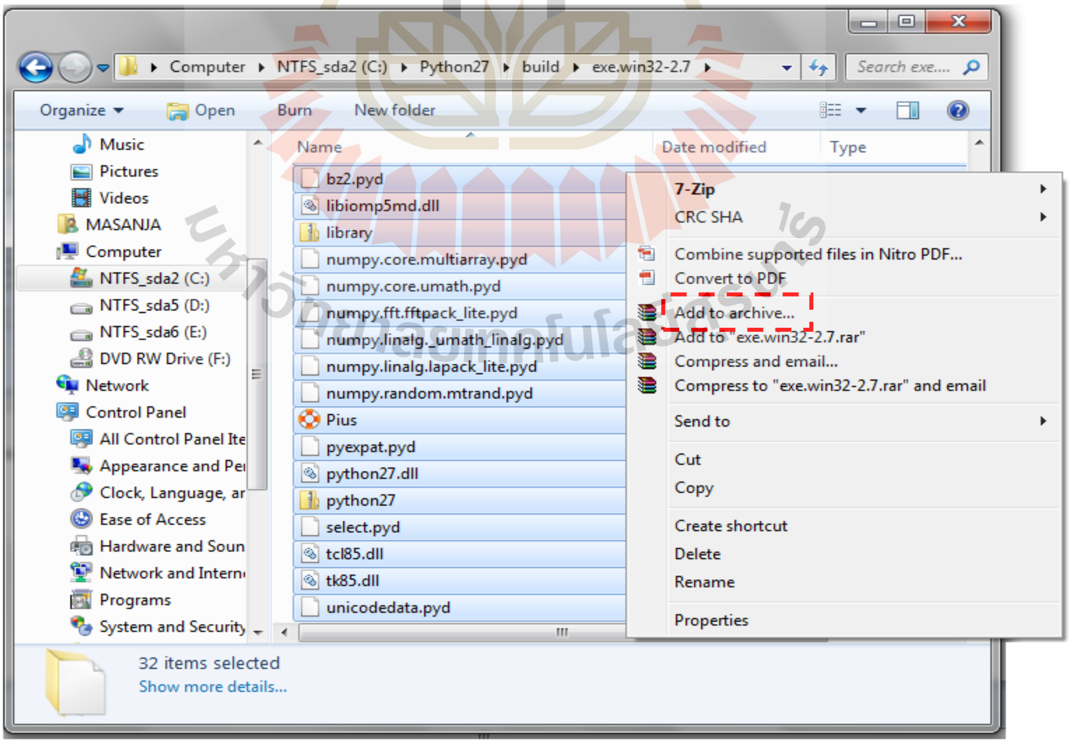


Figure 3.54 Adding compiled files to archive in the win-rar software

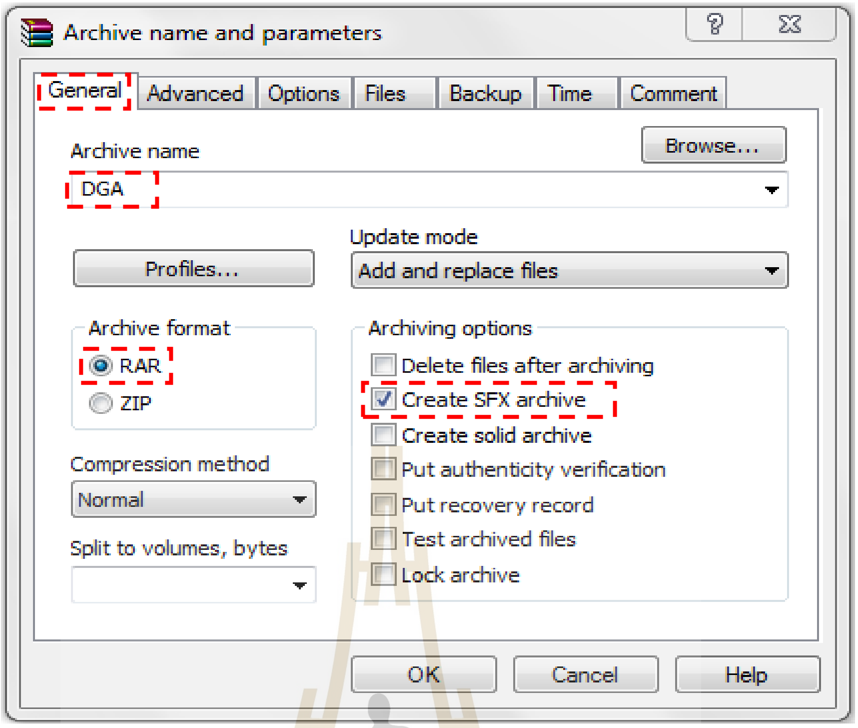


Figure 3.55 Creating SFX archive

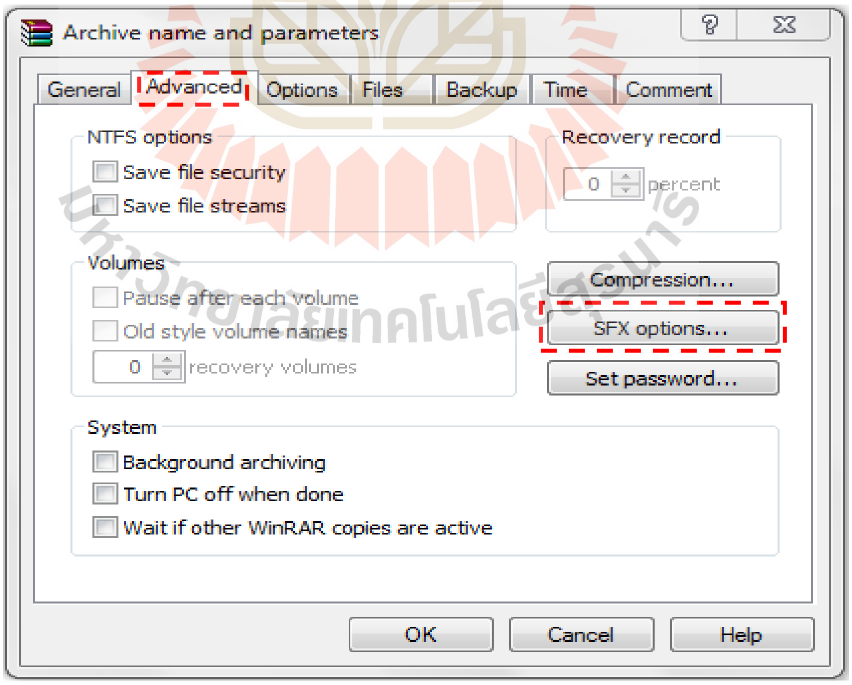


Figure 3.56 Advanced options for SFX archive

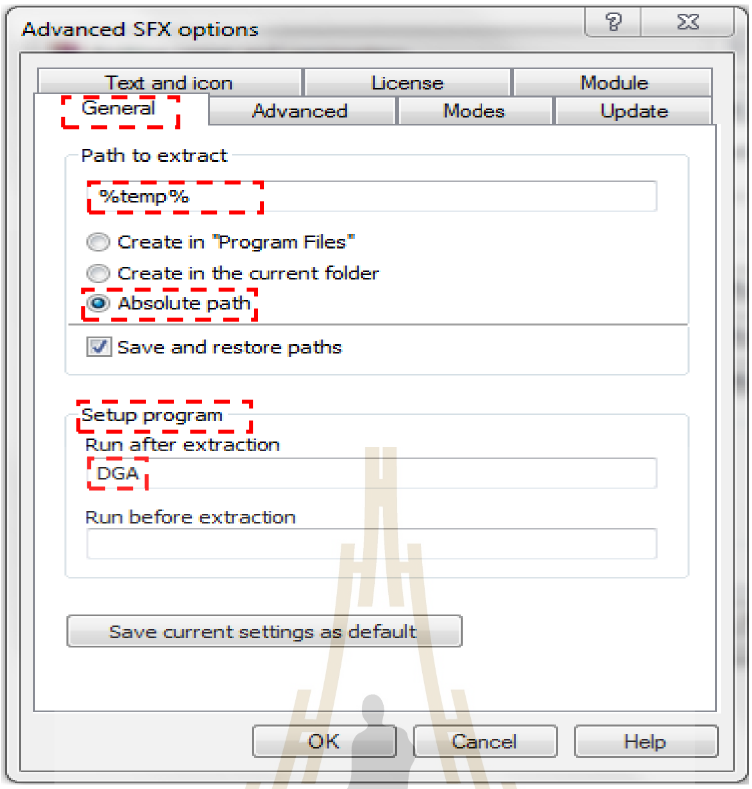


Figure 3.57 General options for SFX archive

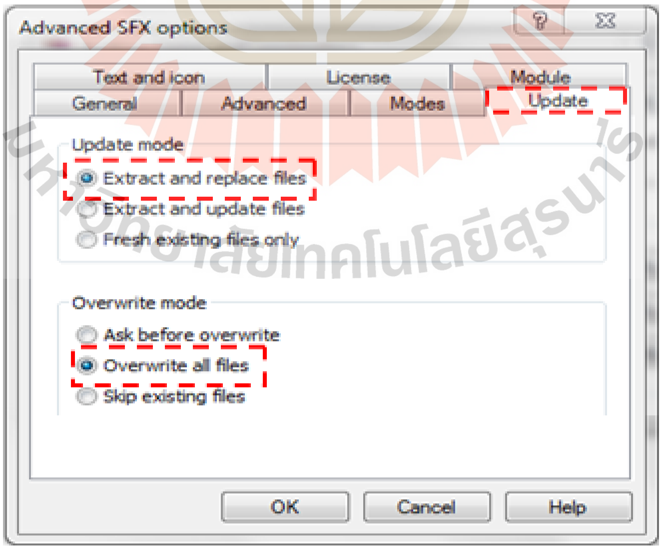


Figure 3.58 Update options for SFX archive

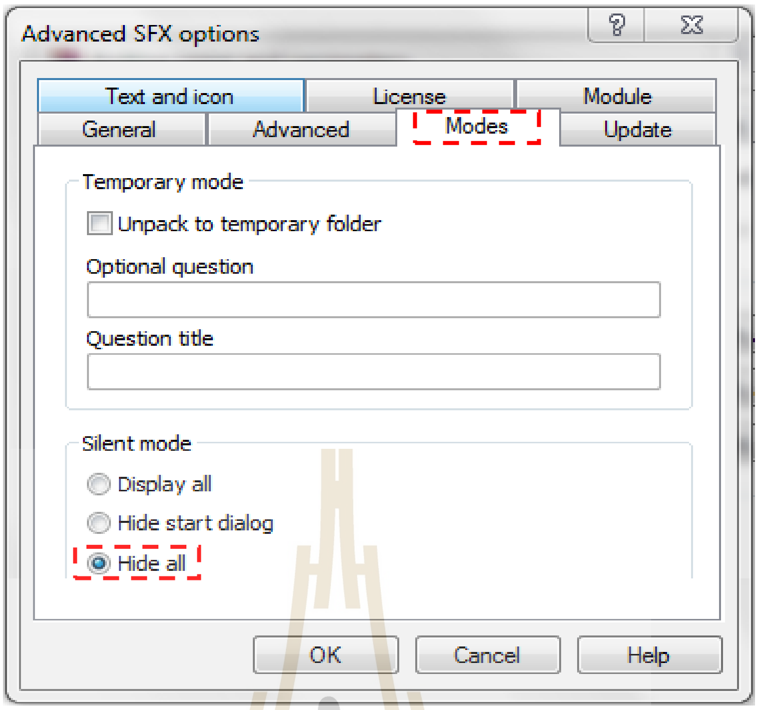


Figure 3.59 Option for hiding other files

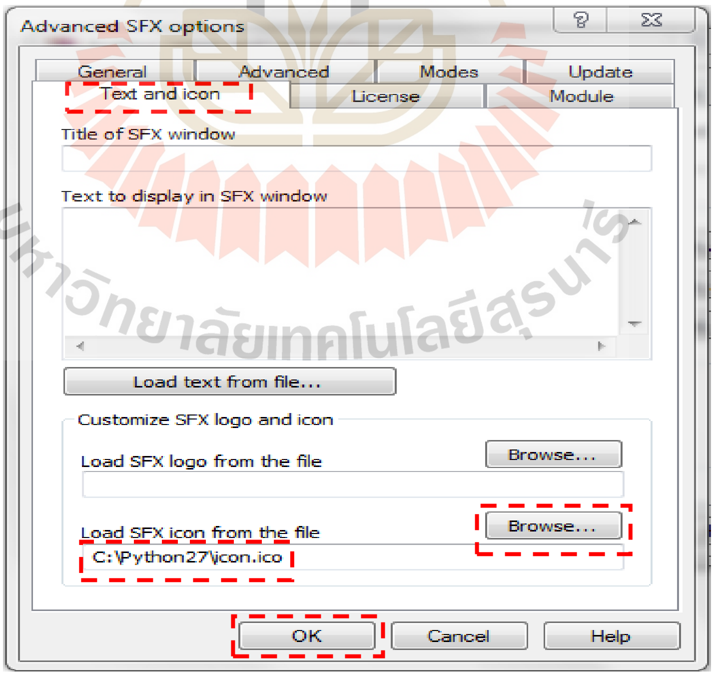


Figure 3.60 Adding icon in the win-rar software

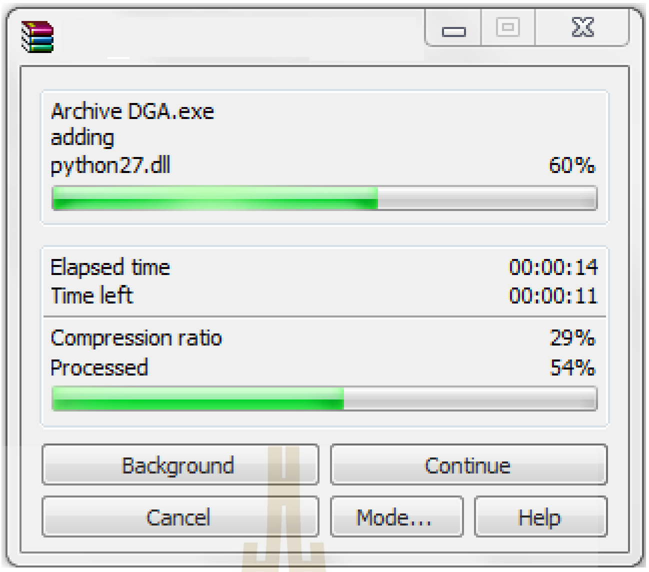


Figure 3.61 Compressing python files with win-rar software

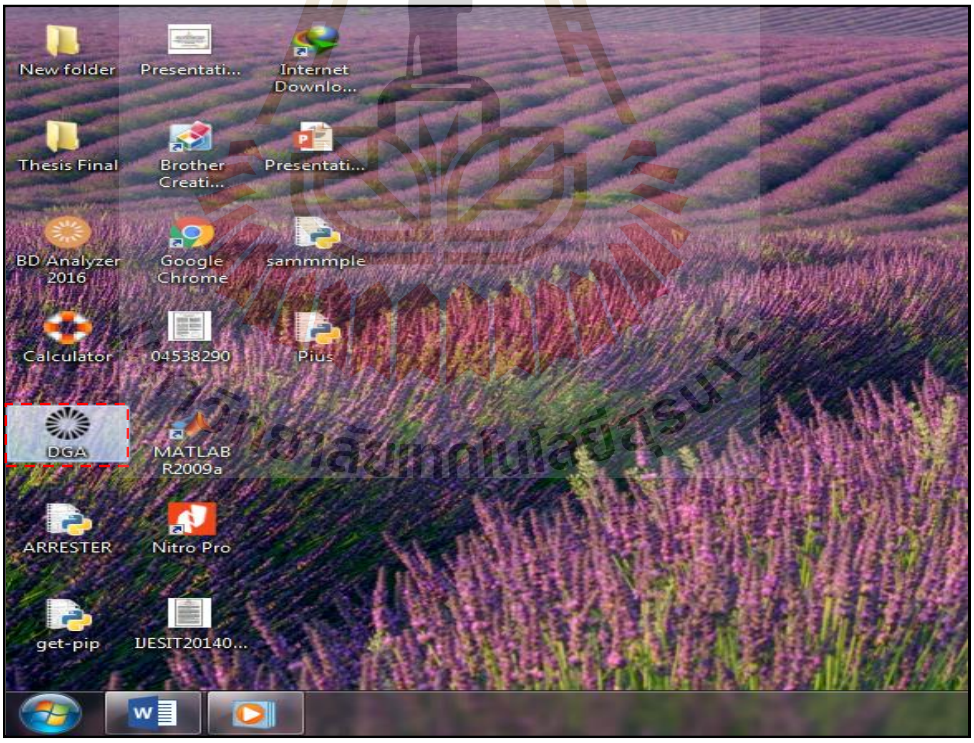


Figure 3.62 The icon of the ready-made DGA software

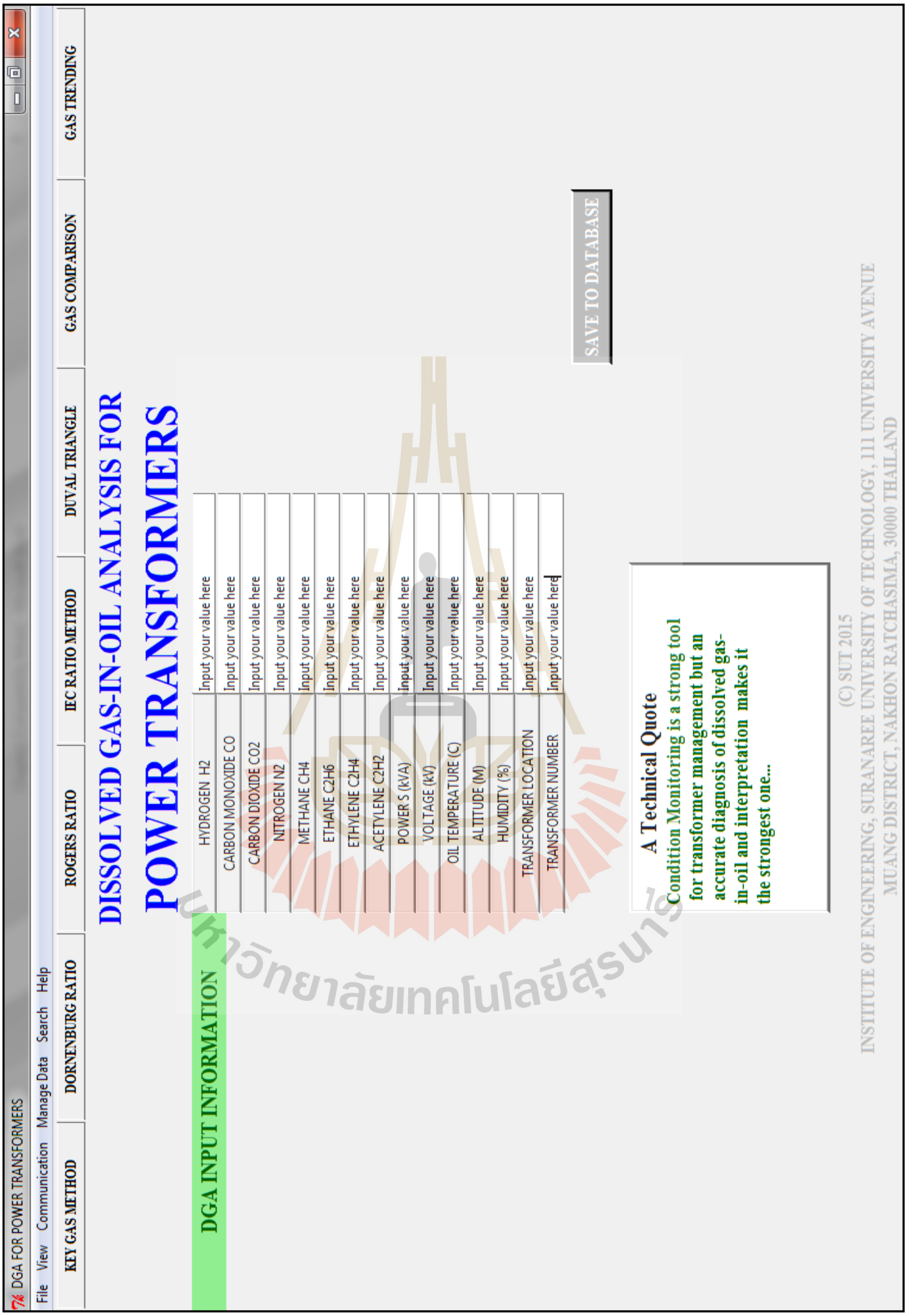


Figure 3.63 The ready-made DGA software

3.9 Chapter summary

This chapter has discussed about development of DGA based software. The use of Python for developing graphical user interface and MySQL language for programming database have been clearly described. Also the validation of the DGA software with 50 DGA data from other researchers have been explained. The use of detections for computation of percentage of prediction, percentage of consistency, accuracy based on detections and accuracy based on total cases have been discussed. The comparison of the performance of diagnostic techniques of DGA software and from other researchers have been discussed. Compilation of python scripts is detailed in this chapter including official websites, downloading the setups, and installation.

CHAPTER 4

EXPERIMENTS

4.1 Materials

In this study an experiment namely, dissolved gas-in-oil with mineral insulating oil from in-service transformers was performed. Samples were taken from twenty-eight in-service power transformers available in Suranaree University of Technology, Nakhon Ratchasima Thailand. Approximately 100 cc from the sample of each transformer was taken for performing this experiments.

4.1.1 Locations of power transformers in Suranaree University of Technology

Samples of mineral insulating oil were taken from in-service power transformers powering different areas in Suranaree University of Technology. Records of the estate management of the university show the number of connected transformers of more than one hundred. Since there is a large number of transformer serving the university, only sampled transformers are highlighted from the university map. Figure 4.1 shows the locations of sampled in-service power transformers. As depicted from the Figure 4.1, the locations have been named similar to buildings' names.

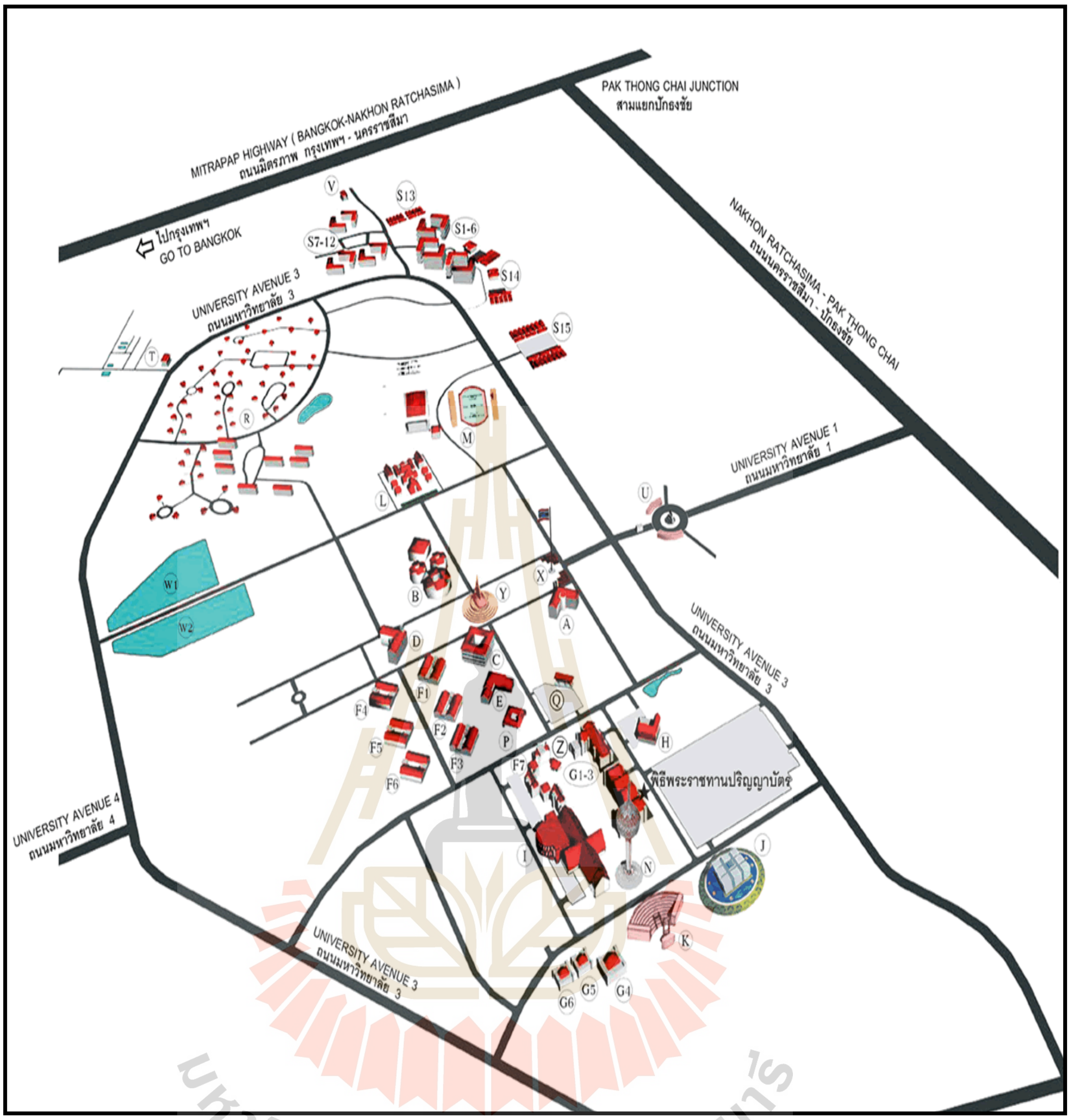


Figure 4.1 Map of Suranaree University of Technology to locate sampled power transformers.

Where

(**A**) Cassava Research building

(**R**) Sukanives 1, Sukanives 2, Sukanives 3, Sukanives 4, Sukanives 7, Sukanives 8

S Suranives 1, Suranives 5, Suranives 7-8, Scaffolding Suranives 9, Suranives 9-10, Suranives 11-12, Suranives 14.

T Animal building, Animal feeding building, Farm office

V Plumbing building

M Sports and Health center, Football field

H Lawn at Surasamanakhan hotel, Scaffolding at Surasamanakhan hotel

L Sukkhawithi 2, Sukkhawithi 3, Sukkhawithi 6

4.1.2 Sample names

Samples were named similar to buildings' names for easy follow-up of transformer condition. Table 4.1 shows the samples taken from several transformers. The list of samples in their order of testing is given in the Appendix D.

Table 4.1 Sample names and their locations

Sample number	Sample name	Location
1	Suknives 1	Suknives building 1
2	Sukkhawithi 2	Sukkhawithi building 2
3	Suknives 3	Suknives building 3
4	Sukkhawithi 3	Sukkhawithi building 3
5	Suknives 7	Suknives building 7
6	Animal building	Animal building
7	Suknives 2	Suknives building 2
8	Suranives 14	Suranives building 14
9	Suknives 4	Suknives building 4
10	Sukkhawithi 6	Sukkhawithi building 6

Table 4.1 Sample names and their locations (Continued)

11	Suknives 8	Suknives building 7
12	Suranives 2	Suranives building 2
13	Surasamanakhan	Lawn at Surasamanakhan hotel
14	Animal feeding	Animal feeding building
15	Football field	Football field
16	Scaffolding Surasamanakhan	Scaffolding Surasamanakhan
17	Sport center	Sport and Health center
18	Farm office	Farm office
19	Suranives 2	Suranives building 2
20	Suranives 14-01	Suranives building 14-01
21	Scaffolding Suranives 9	Scaffolding Suranives building 9
22	Suranives 11-12	Suranives building 11-12
23	Plumbing building	Plumbing building
24	Suranives 9-10	Suranives building 9-10
25	Suranives 7-8	Suranives building 7-8
26	Sport center	Sport and Health center
27	Cassava research	Cassava research building
28	Suranives 5	Suranives building 5

4.1.3 Sample questionnaire

Details of sampled transformers were recorded in a sample questionnaire. It helps to clarify some information from the diagnosed transformer before making decisions. A sample of the questionnaire is shown in Figure 4.2.

4.2 Experiment

4.2.1 Dissolved gas-in-oil analysis

This involved the use of commercial DGA equipment from MORGAN SCHAFFER. It follows ASTM D3612 to analyze dissolved gasses in oil and IEEE C57-106-2008 with the use of Roger’s ratio method and IEC 60599-1999 with the use of Duval triangle method for interpretation

SAMPLE QUESTIONNAIRE FOR MINERAL INSULATING OIL

PROPERTY OWNERSHIP

Owner: SURANAREE UNIVERSITY Location: ....

SAMPLING INFORMATION

Sample No.: ... Date of sampling: .... Oil Temperature: .... °C

Loading: .... % (If 0%) Time of Out of Service: ....

Have oil pumps? ☐ Yes ☐ No Number of oil-pumps: ....

Sampling Vessels: ☐ Syringe ☐ Amber-colored bottle Index number: ....

Measures taken: ☐ Filtering/degassing ☐ Oil regeneration ☐ Oil change. Date ....

Manufacturer: .... Type of oil: .... Oil Weight: .... kg

Oil sample location: ☐ Main Tank ☐ TC

Purpose of sampling: ☐ Routine inspection ☐ Oil treatment ☐ Topping up

Voltage: .... kV Power: .... MVA ....

Frequency: ...... Hz Year of Manufacture: ......Taken into operation: ....

Cooling system: .... Application: ☐ System/Distribution ☐ Generator

Figure 4.2 Example of a sample questionnaire

4.2.1.1 DGA equipment

The DGA equipment from Morgan Schaffer Corporation has Myrkos – Portable DGA MicroGC (for gas extraction) offering repeatability better or equal to ASTM D 3612 or IEC 60567. It measured separately seven gasses most significantly fault gasses within 120 seconds as shown in Figure 4.3 and air components (O_2 and N_2) in a separate kit. The levels of dissolved gasses extracted

Figure 4.3 MYRKOS gas extractor.

from MYRKOS were read to the computer for interpretation. Syringes of 100 cc used were used for carrying samples during measurement.

4.2.1.2 Installation procedures

The Helium gas was used as a carrier gas properly connected to the gas extractor through pressure regulators and pressure gauges. The pressure gauge was set to 2000 psi and pressure regulator 80±2 psi before switching on MYRKOS gas extractor. Helium gas must first be opened to flow to the MYRKOS before turning it on. An Ethernet cable was used to connect MYRKOS and computer with the aid of PPM Report. After powering MYRKOS, the PPM report installed in the computer lead a step by step program into the calibration of MYRKOS, preparation of samples and later measurement of the oil sample. The arrangement of DGA components in the laboratory for gas extraction is shown in Figure 4.4.

4.2.1.3 Testing procedures

The MYRKOS equipment was allowed to warm up for 45 minutes for its channel temperatures to reach their set points necessary for system stabilization. Calibration of gasses inside the MYRKOS was agreed to ±2%. Furthermore, CO and CO_2 were calibrated to less than 10 ppm while other gasses were less than detectable limits (LDL).

Figure 4.4 Arrangement of DGA components in the laboratory

A 100 cc syringe was first cleaned by Acetone $(CH_3)_2CO$ and later by samples to be measured. The samples were then filled up into the syringes and set as required in the procedure including the addition of CO_2 and shake up. After all procedures, finally, gas levels were measured and read through the computer. Figure 4.5 shows the cleaning of the syringe with the sample to be measured during the gas extraction process for quantifying the levels of dissolved gasses.

Figure 4.5 Cleaning a syringe for gas extraction

4.3 Chapter summary

This chapter discussed an experiment namely, dissolved gas-in-oil with mineral insulating oil from twenty-eight in-service transformers available in Suranaree University of Technology, Nakhon Ratchasima Thailand. Installation procedures and testing procedures for DGA experiment is detailed in this chapter.

CHAPTER 5

CORONA IN OIL

5.1 Introduction

This chapter describes the development of test cell that can simulate the condition of oil corona in power transformer. The need of this chapter follows the requirement from the thesis committee. When electrical stress, thermal stress or both occur in a transformer, multiple activities grow in transformer's insulation i.e. partial discharge (corona), arcing or overheating (Rondla, Falahi, Zhan and Goulart, 2012; Ashkezari, Saha, Ekanayake and Ma, 2011; Nemeth, Laboncz and Kiss, 2009; Muhamad, Phung, Blackburn and Lai, 2012). Oil corona is among of the early occurring activities inside the liquid insulation. It is referred as a type of localized discharge ionization phenomenon that occurs in gaseous media around the energized conductors when the stress exceeds a critical value and causes deterioration in insulation systems (IEC 60270, 2000; Bartinikas, 1979; Rohwein, 1996). Moreover, Oswald, James, and Plymouth, (1964) explained corona as an outcome of an accelerated ionization taking place in a gaseous media under the influence of electric field. Therefore, the presence of corona in the liquid insulation can be due to the presence of water vapors, bubbles (Oswald, James, and Plymouth, 1964), metallic protrusions in oil (Ghaffarian and Edin, 2010) etc. In Ghaffarian and Edin, (2010) corona has been distinguished from partial discharge as the activities accompanied by visual phenomena while those which are not visible i.e. occur inside of materials or devices are termed as partial discharge. The existence of corona in oil can cause serious problems. Among of them are power loss as it consume high voltage and low current (Oswald, James and Plymouth, 1964; Izeki, Kurahashi and Matsuura, 1971), erosion on the surface of solid insulation, disruption and change of its atomic or molecular structure due to bombardment of electrons combined with the intensifying effect of heating (Ghaffarian and Edin, 2010), production of dissolved gases (Oswald, James and Plymouth, 1964; Ghaffarian and Edin, 2010; Rohwein, 1996) and bubbles (Izeki, Kurahashi and Matsuura, 1971) absorbed in the oil and reduction of corona inception voltage (CIV) when the level of dissolved gases increases. Furthermore, Yadav and Sarathi, (2015); Fofana, Bouaicha, Farzaneh, and Sabau, 2008); Boudraa, Mokhnache, and Fofana, (2014) explained that decomposition of liquid insulation molecules may formulate new molecules and acidic residues which are hazardous for the solid insulation. In liquid insulation, when corona exists in oil, the dielectric strength of insulating oil decreases drastically as corona continues to persist until breakdown occurs (Rohwein, 1996). For oil filled transformer where minimum acceptable levels of corona resistance and dielectric strength are monitored, the liquid insulation can be improved to acceptable levels by reconditioning before failure. In the absence of proper maintenance plan, failure is more likely to happen when oil corona accelerates. Failure of liquid insulation results to the failure of solid insulation and hence the end of life of power transformer is reached. With the designed test cell, an investigation was performed to study existing relationship between dissolved gas in oil and breakdown strength of mineral oil.

5.2 Test cell

The test cell was made by using a cylindrical acrylic with the 200 mm height, 120 mm diameter and 5 mm thickness. It has four supporters, white cylindrical rods in each corner with plastic nuts, made by Teflon 20 mm diameter and 240 mm height. The internal space of the test cell was made to support electrodes and oil sample. Figure 5.1 (a) shows a complete test cell and Figure 5.1 (b) shows an arrangement of the test cell. Moistures generated during the stressing process was removed by silica gel equipped in an external transparent plastic bottle. The silica gel bottle has dimensions of 20 mm diameter, 40 mm height and 1.5 mm thickness with a tight cover attached on the top cover of the test cell with four small holes (1.5 mm diameter) in its base. It has a gap adjuster to vary the distance between the high voltage electrode and ground electrode. The test cell was made possible to accommodate one needle as well as four needles when needed.

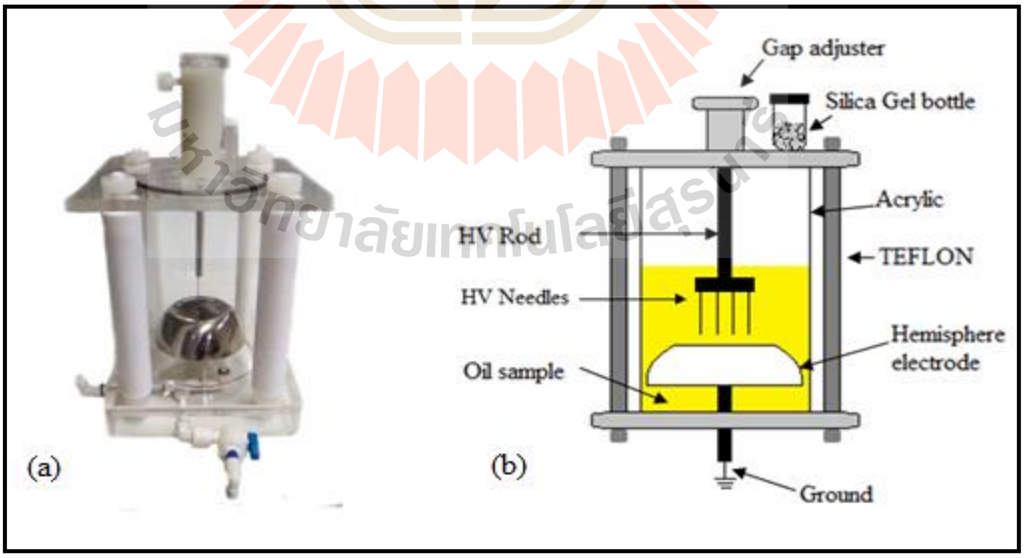


Figure 5.1 Test cell (a) complete test cell (b) its arrangement

5.2.1 Electrodes

The configuration of electrodes used was needle-to-hemisphere configuration. Two kinds of high voltage electrode were employed into the test cell i.e. one needle and four needles. Four needles comprised of three pieces i.e. vertical rod, horizontal rod, and needles. Vertical and horizontal rods were made up with stainless steel 8 mm diameter, 90 mm height for vertical rod and 40 mm width for the horizontal rod. Its four needles were made from steel having lengths of 42 mm, 1 mm diameter, a tip radius of 15 μm and tip curvature of 30°. The distance between one pin and the other one was 10 mm. The distance between the first or fourth pin and the side way was 5 mm. Figure 5.2 (a) shows a one needle high voltage electrode. Figure 5.2 (b) shows the four needles high voltage electrode with a vertical rod, horizontal rod, and four needles.

The ground electrode was made from the hemispherical shaped bowl with stainless steel material having 80 mm lower diameter, 100 mm upper diameter and 1 mm thickness. The distance between the lower diameter of the ground electrode and upper diameter was 25 mm and the distance between the lower diameter of the ground electrode and the acrylic base was 55 mm. HV needles and ground electrode were separated by a gap of 5 mm. Figure 5.2 (c) shows four needles and hemisphere (ground) electrode configuration in the test cell.

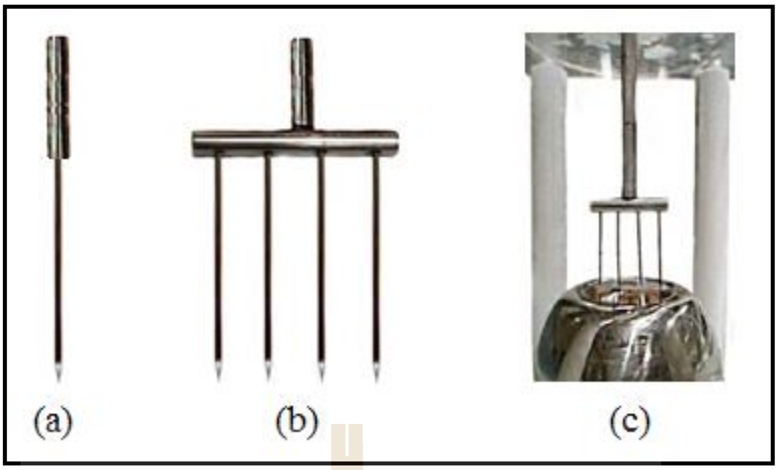


Figure 5.2 Electrodes (a) one needle (b) four needles (c) four needles-to-hemisphere configuration

5.3 Installation procedures

5.3.1 Preparing a sample

The sample of insulating oil was supplied by the Provincial Electricity Authority of Thailand (PEA) for simulating corona in oil. For preparation, first, the gap between electrodes was adjusted to 5.0 mm by using the gap adjuster. Then the test cell was rinsed with acetone and later by a portion of the sample to be stressed. After rinsing, the unwanted oil was drained and stored properly for disposal. Figure 5.3 (a) shows the acetone reagent used for cleaning the test cell while Figure 5.3 (b) shows the draining process of unwanted oil after cleaning the test cell. The sample to be stressed was then poured gently into the test cell without generating bubbles. A sample was left for about five minutes to allow bubbles to varnish.

(a) (b)

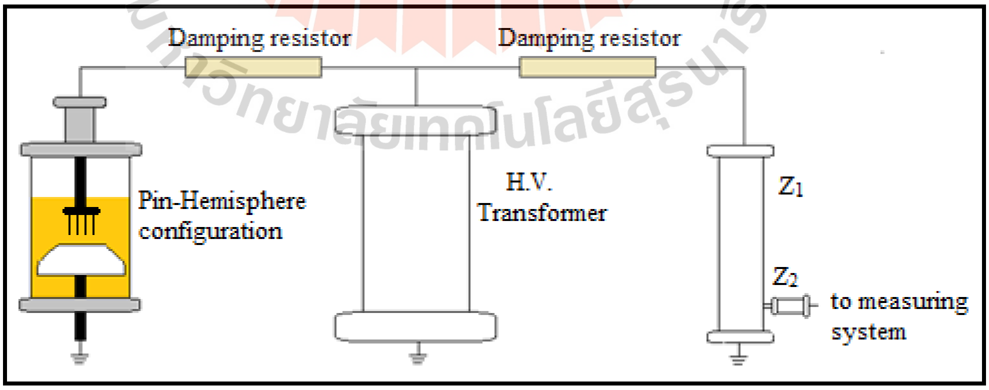
Figure 5.3 Cleaning the test cell (a) acetone (b) draining oil after cleaning test cell

5.3.2 Connection of the stressing circuit

The prepared sample was stressed with 30 kVA 22 kV_{rms} 50 Hz transformer. The input was connected via autotransformer necessary for varying stress voltage. Figure 5.4 shows a high voltage power supply with (a) autotransformer used to vary the input of high voltage transformer and (b) high voltage transformer for stressing the oil sample.

Figure 5.4 High voltage source (a) autotransformer (b) high voltage transformer

Figure 5.5 shows the circuit configuration for setting up of oil corona. Damping resistor was used to limit breakdown current from destroying the test cell. The right side of the HV transformer in Figure 5.4 (a) shows the arrangement used for measuring the voltage applied to the test cell. It employed voltage divider through HV impedances Z_1 and Z_2. Figure 5.5 (b) shows an experimental set up for oil corona in the HV lab.



(a)

(b)

Figure 5.5 Oil corona set up (a) circuit configuration (b) experimental set up with 1-test cell, 2-damping resistor, 3-HV transformer, 4-HV impedance Z_1, 5-HV impedance Z_2

5.4 Testing procedures

After arranging the circuit as in Figure 5.5 with 5 mm gap between electrodes in the test cell, the voltage was applied to the electrodes by varying voltage from the autotransformer. By varying voltage in autotransformer, the stress voltage from HV transformer was also varied. Stress voltage was increased from zero up to when oil corona was observed either visually or audibly in between electrodes. The corresponding stress voltage was recorded as corona inception voltage. Again stress voltage was increased from zero up to when the breakdown occurred. The corresponding stress voltage was recorded as breakdown voltage. Measurements were done through voltage divider connected across HV transformer with HV impedances Z_1 and Z_2. The measured CIV and breakdown voltages were 5.3 kV and 10.2 kV respectively. Stress voltage for oil corona was set 1.2 times above CIV. Three oil samples were kept under this stress voltage for 24 hours, one week and two weeks respectively. After 24 hours, one week and three weeks oil samples were measured for dissolved gasses and breakdown strengths. Figure 5.6 shows the preparation of stressed sample for measurement of breakdown strength.

Figure 5.6 Preparation of stressed sample for measurement of breakdown strength.

Mean, range, the percentage of range to mean and standard deviation were computed based on six tests for checking the validity of the test. Validity was checked from the test results obtained in each sample through (5.1) to (5.4).

$$\text{Range} = \text{Highest value} - \text{Lowest value} \tag{5.1}$$

$$\text{Mean, } \bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{5.2}$$

$$\% \frac{R}{\bar{X}} = \frac{R}{\bar{X}} \times 100 \tag{5.3}$$

$$\text{Deviation, } \sigma = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X})}{n-1}} \tag{5.4}$$

where X_i = test results,

n = number of tests per sample (n=6)

R = range

$\bar{X}$ = mean of test results

$\% \frac{R}{\bar{X}}$ = percentage of range to mean

For the valid tests, (5.3) and (5.4) had values less than or equal to 92 percent and 4 percent respectively.

5.5 Relationship of dissolved gas-in-oil and breakdown strength

The test was based on IEC 60156. The breakdown strength equipment used for measuring breakdown strength after oil corona test is shown in Figure 4.6 (chapter 4). The test cell with mushroom shaped polished brass electrodes, a 2 mm gap, 0-60 kV 50 Hz of test voltage and 2 kVs-1 rate of rise of test voltage was employed. A part of stressed samples from 24 hours, one week and two weeks were taken for measurement of dissolved gasses. Table 5.1 shows the results of breakdown strength test for a virgin oil sample and stressed oil samples. Validation of test results of breakdown strength was done based on the percentage of range to mean (≤ 92 %) and standard deviation (≤ 4%). The levels of dissolved gasses were related to the results of breakdown strength test. Based on IEC 60156, for a breakdown strength test to be valid % Range to mean must be less than 92%. Hence breakdown strength tests were valid. Table 5.2 shows the levels of dissolved gasses in oil of stressed samples.

Table 5.1 Results of breakdown strength tests of stressed samples

Sample	Period (hours)	Mean (kV)	% Range/Mean (%)
Virgin	0	43.60	69.81
Sample 1	24	28.35	36.24
Sample 2	168	20.52	24.00
Sample 3	336	18.77	80.66

Scatter plots of breakdown values versus dissolved gasses are shown in Figure 5.7. Each dissolved gas was plotted against breakdown values from the virgin to stressed oil sample. From the plotted graphs, the relationship of each dissolved gas and breakdown value was studied. As observed in Figure 5.7, only CO and CO_2 showed the existence of a relationship with breakdown strength. The remained

Table 5.2 Results of dissolved gas-in-oil of stressed samples

Sample	Period	H_2	CO	CO_2	CH_4	C_2H_6	C_2H_4	C_2H_2
Virgin	0	0	0	363	0	0	0	0
Sample 1	24hrs	0	3	422	0	0	0	0
Sample 2	168 hrs	0	12	438	0	0	0	0
Sample 3	336 hrs	0	16	474	0	0	0	0

dissolved gas and breakdown value was studied. As observed in Figure 5.7, only CO and CO_2 showed the existence of a relationship with breakdown strength. The remained dissolved gasses showed non-existence of the relationship. From this result, the existence of a negative relationship between CO, CO_2 and breakdown strength does not strongly sound since these gasses can also be generated during transportation and preparation in the laboratory due to exposure to ultraviolet light. Therefore, stressing insulating oil for a short period may lead to false results. It was advised to stress the insulating oil over an extended period for better results.

5.6 Chapter summary

This chapter has discussed the design of new test cell for simulating corona in oil as an internal activity inside the oil filled transformer. The test cell made from acrylic material with mushroom shaped polished brass electrodes was discussed. Four samples of mineral insulating oil stressed at 5.0 mm gap, supply voltage of 1.2 times CIV, 50 Hz were explained. Two tests i.e. dissolved gas in oil and breakdown strength from stressed samples were shown. The results of the study of existing relationship between dissolved gas-in-oil and breakdown values was also described.

Table 6.1 Interpretation of dissolved gasses (Continued)

22	25	129	0	37	841	307	8552	ND	ND	ND	ND	ND
23	0	0	9	0	0	35	2165	ND	ND	ND	ND	ND
24	0	693	0	0	199	183	3768	OC	ND	ND	ND	T3
25	0	0	0	3	0	100	4686	ND	ND	ND	ND	T3
26	0	0	0	18	0	36	1974	OC	ND	ND	ND	D2
27	0	0	13	22	0	324	2751	OC	ND	ND	ND	T3
28	0	0	0	3	0	89	4585	OC	ND	ND	ND	T3

6.1.1 Conditions of power transformers

Since the condition of the transformer especially condition 1 can affect the interpretation done by using diagnostic methods, they were first studied from the results given in Table 6.1. Table 6.2 shows the summary of interpreted conditions of power transformers. This was done based on the IEEE C57.104-2008. Figure 6.1

Table 6.2 Conditions of power transformers

Conditions	Number of transformers
1	25
2	2
3	0
4	1

shows the number of power transformer and their specific conditions. From Figure 6.1 it can be seen that a large number of power transformers (25 transformers) are in condition 1. According to IEEE C57.104-2008, these power transformers are operating satisfactorily. But the rate of evolution of dissolved gases should be carefully monitored in order to examine any incipient fault. It can also be observed that one transformer is in condition 4. The rate of evolution of dissolved gasses should be quickly identified in order to exercise extreme caution.For those in condition 2 an action must be taken to identify the evolution rate and exercise caution.

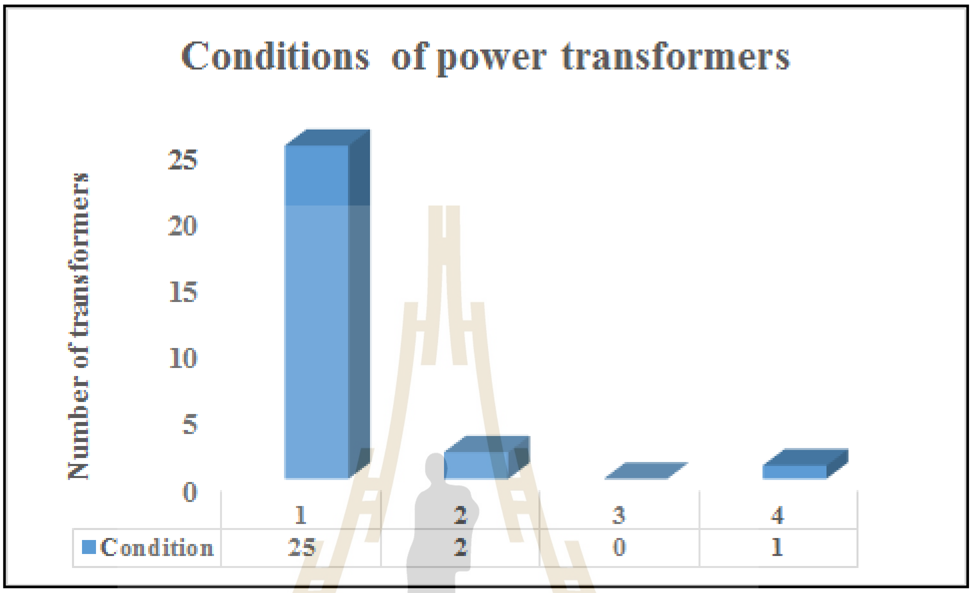


Figure 6.1 Conditions of power transformers

6.2 Diagnosis of key gas method

From Table 6.1, the diagnoses of key gas method can be summarized as shown in Table 6.3. As observed from Table 6.3, key gas method succeeded to diagnose all cases. But almost half of the cases were resulted into no detection. The percentage of detection of this method was computed and found to be 42.86%.

Table 6.3 Diagnoses of key gas method

Cases	28
Successful detections	28
Faults detected	12
No detections	16
Percentage of detection (%)	$(100 \times 12)/28 = 42.86$

The faults predicted by the key gas method were plotted for studying the health of the transformer due to the severity of the existing faults in the insulation. Figure 6.2. As shown in Table 6.1, the only one and most detected fault was overheated oil. Overheated oil is a typical thermal fault and a moderate serious fault. The existence of this fault deteriorate the liquid insulation and if it stays over a long time may cause a serious damage.

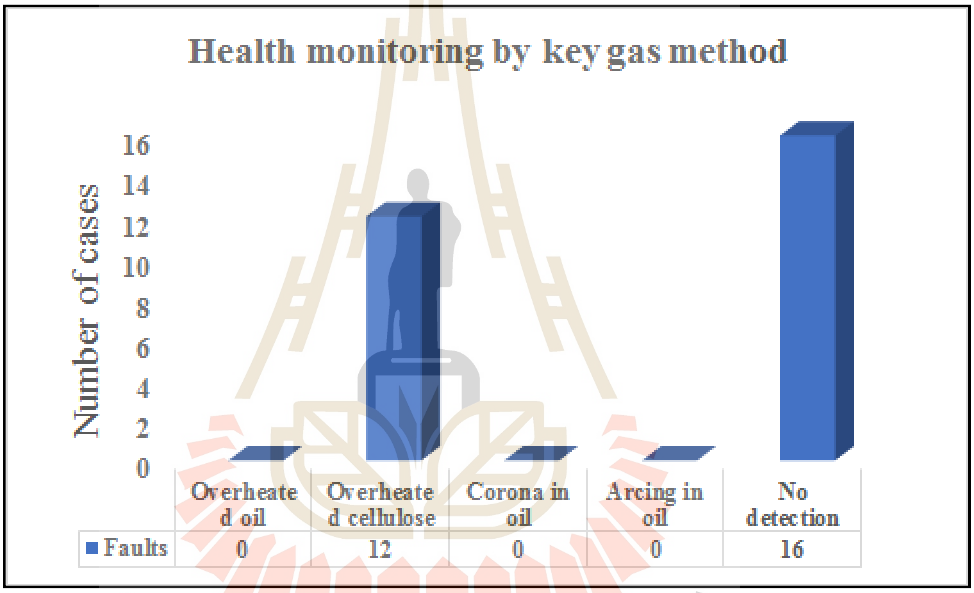


Figure 6.2 Diagnoses of key gas method

6.3 Diagnosis of Dornenburg ratio method

From Table 6.1, the diagnoses of Dornenburg ratio method can be summarized as shown in Table 6.4. Observation from Table 6.4 showed that Dornenburg ratio method did not succeed to diagnose in any case. In all 28 cases it resulted into no detection. The computed percentage of detection of this method was around 0.00%.

Table 6.4 Diagnoses of Dornenburg ratio method

Cases	28
Successful detections	28
Faults detected	0
No detections	28
Percentage of detection (%)	(100 × 0)/28 = 0.00

Figure 6.3 shows no case detected by Dornenburg ratio method. This is because most of the hydrocarbon gases have zero ppm which indicates they in condition 1.

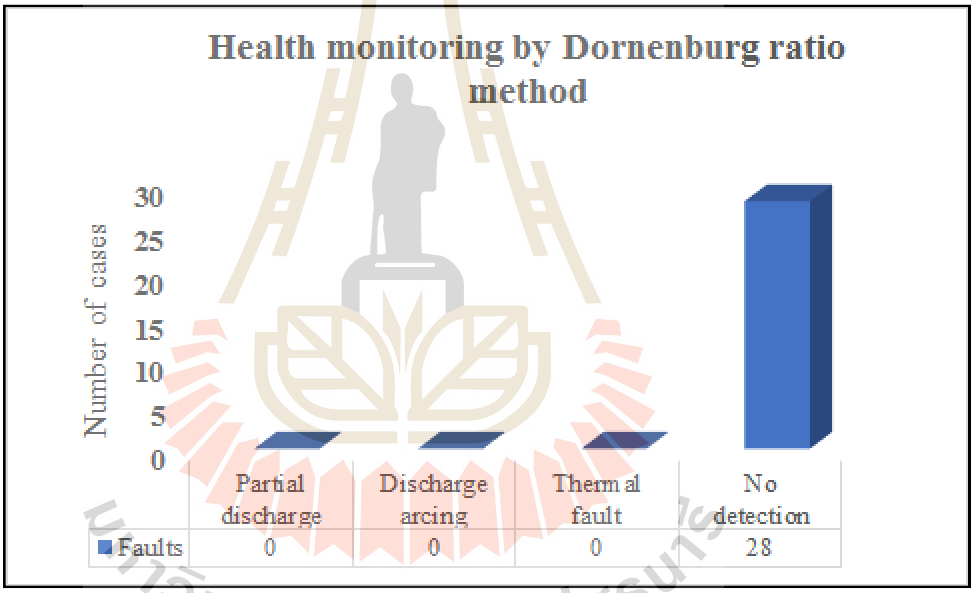


Figure 6.3 Diagnoses of Dornenburg ratio method

6.4 Diagnosis of Roger's ratio method

From Table 6.1, the diagnoses of Roger's ratio method can be summarized as shown in Table 6.5. Roger's did not succed to predict any fault in 28 cases. The computed percentage of detection of this method was 0.00%. Figure 6.4 shows the

Table 6.5 Diagnoses of Roger's ratio method

Cases	28
Successful detections	28
Faults detected	0
No detections	28
Percentage of detection (%)	(100 × 0)/28 = 0.00

numbers of faults for Roger's ratio method. This is because most of the hydrocarbon gases have zero ppm which indicates they in condition 1.

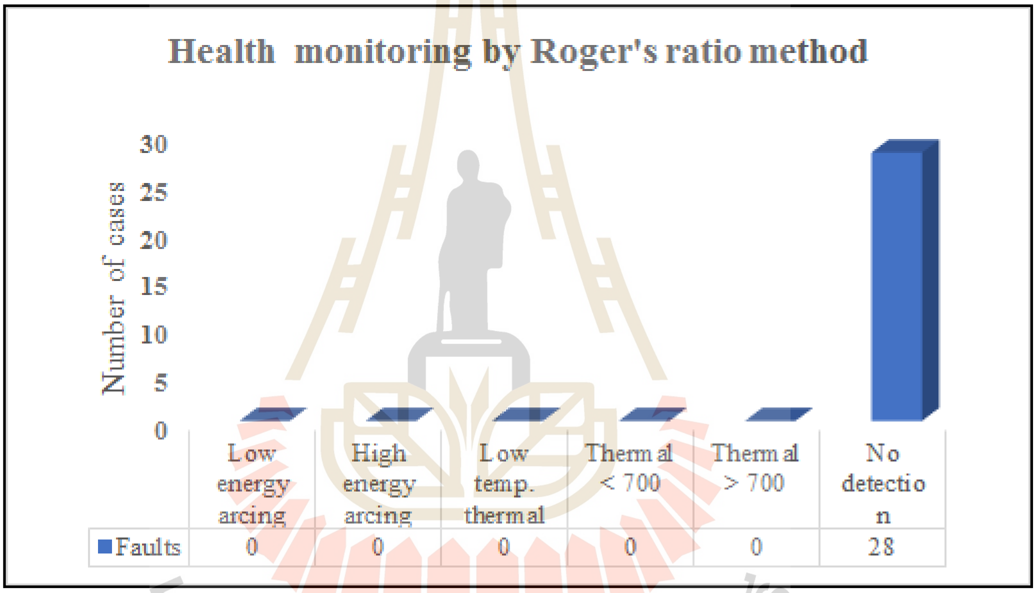


Figure 6.4 Diagnoses of Roger's ratio method

6.5 Diagnosis of IEC ratio method

From Table 6.1, the diagnoses of IEC ratio method can be summarized as shown in Table 6.6. Figure 6.5 shows one case detected by IEC ratio method. The detected case was thermal fault of temperature between 150 °C and 300 °C. This fault is less serious to due to the range of temperature it existed.

Table 6.6 Diagnoses of IEC ratio method

Cases	28
Successful detections	28
Faults detected	1
No detections	27
Percentage of detection (%)	$(100 \times 1)/28 = 3.57$

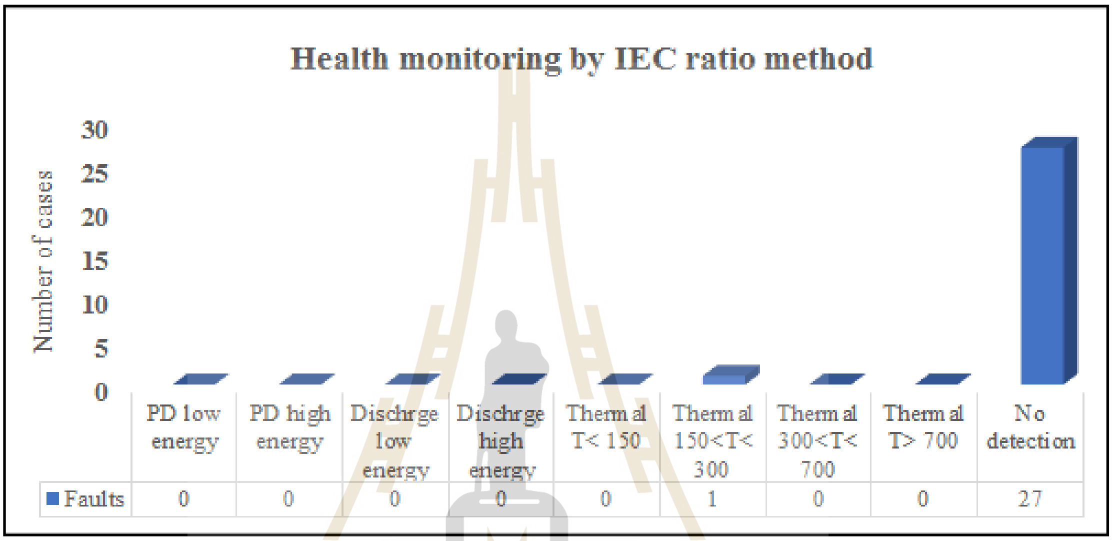


Figure 6.5 Diagnoses of IEC ratio method

6.6 Diagnosis of Duval triangle method

From Table 6.1, the diagnoses of Duval triangle method can be summarized as shown in Table 6.7. The list of detected faults by Duval triangle is shown in Table 6.8. Figure 6.6 shows the case detected by Duval triangle method. The percentage of detection was 53.57%. This indicates the greatest ability of this method to detect faults.

Table 6.7 Diagnoses of Duval triangle method

Cases	28
Successful detections	28
Faults detected	15
No detections	13
Percentage of detection (%)	(100 × 15)/28 = 53.57

Table 6.8 Detected faults in Duval triangle

No	Detectable Fault	Number of occurrences
1	Partial Discharge (PD)	0
2	Discharge of Low Energy Intensity (D1)	0
3	Discharge of High Energy Intensity (D2)	2
4	Mixed Fault (DT)	0
5	Thermal Fault T < 300 °C (T1)	1
6	Thermal Fault 300 < T < 700 °C (T2)	1
7	Thermal Fault T > 700 °C (T3)	11
8	No detection	13
	Total	28

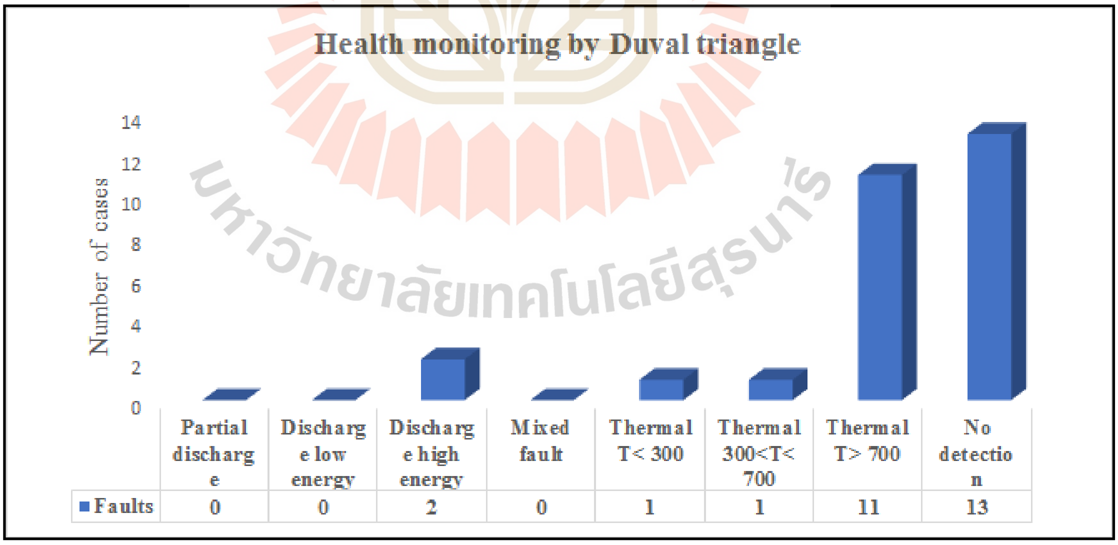


Figure 6.6 Diagnoses of Duval triangle method

From Figure 6.6, it can be seen that T3 is the most occurring fault among the other faults. Thermal fault T3 is the serious fault if maintenance is delayed over a long service time. In Duval (2003) T3 fault was described to be caused by circulating current in core windings which cannot be cleared by changing or reconditioning of insulating oil.

6.7 Performance of diagnostic methods

Figure 6.7 shows the comparison of performances of diagnostic methods. These perfomances have been computed from Table 6.3 to Table 6.8 in terms of percentage of predictions. Duval triangle showed the ability to predict faults by 53.57%, followed by key gas method 43.47%. Ratio methods like Dornenburg, Roger’s and IEC ratio methods failed to perform diagnosis because most of transformers were in condition 1.

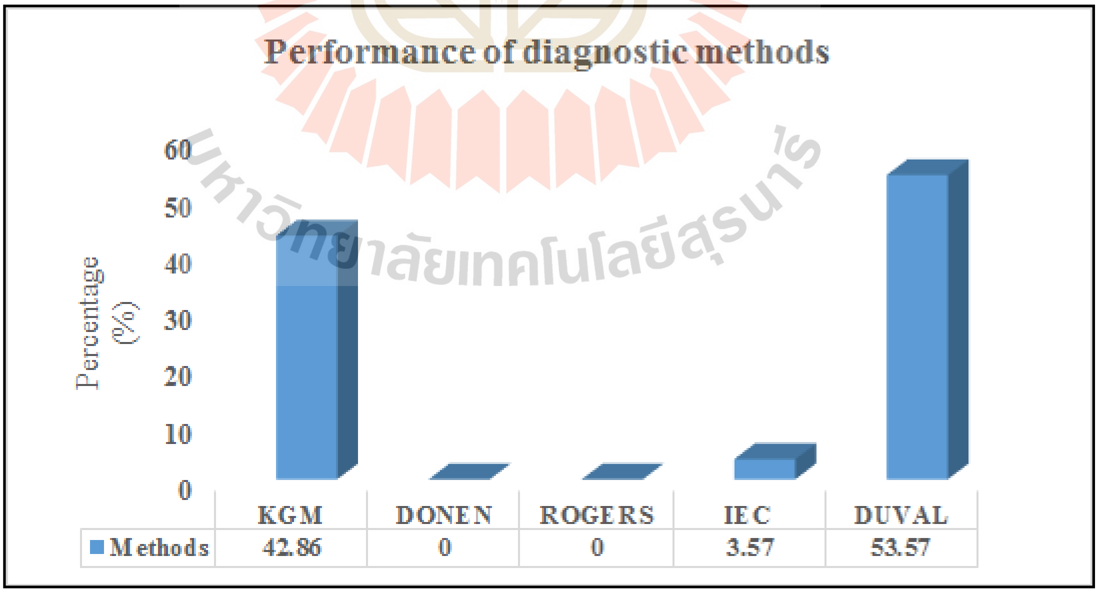


Figure 6.7 Performance of diagnostic methods

CHAPTER 6

RESULTS AND DISCUSSION

6.1 Analysis of dissolved gasses

Dissolved gasses were interpreted by using DGA technique. Five diagnostic methods often employed in DGA i.e. key gas, Roger's ratio, Dornenburg ratio, IEC ratio and Duval triangle were used for interpreting results of dissolved gasses. The results of interpretation are shown in Table 6.1.

Table 6.1 Interpretation of dissolved gasses

No	Gasses in ppm							Interpretation				
	H_2	CH_4	C_2H_2	C_2H_4	C_2H_6	CO	CO_2	KGM	DOR	ROG	IEC	DUV
1	0	0	0	0	0	0	643	ND	ND	ND	ND	ND
2	0	0	0	0	0	84	2715	OC	ND	ND	ND	ND
3	0	0	0	0	0	33	3095	OC	ND	ND	ND	ND
4	0	0	0	4	6	102	6019	OC	ND	ND	ND	T3
5	0	0	7	0	23	48	4889	ND	ND	ND	ND	ND
6	0	0	0	0	0	57	5160	OC	ND	ND	ND	ND
7	0	0	7	0	20	17	4073	ND	ND	ND	ND	ND
8	0	0	0	2	2	94	6427	OC	ND	ND	ND	T3
9	10	0	0	44	24	365	9087	ND	ND	ND	ND	T3
10	0	0	0	0	0	18	1684	OC	ND	ND	ND	ND
11	0	0	0	2	0	126	4153	OC	ND	ND	ND	T3
12	0	0	0	48	12	308	8078	ND	ND	ND	ND	T3
13	5	0	0	69	21	266	7977	ND	ND	ND	ND	T3
14	0	0	0	0	45	64	2477	ND	ND	ND	ND	ND
15	16	14	0	0	112	49	3061	ND	ND	ND	ND	ND
16	0	0	0	4	0	60	4302	OC	ND	ND	ND	T3
17	0	0	0	0	9	166	7165	ND	ND	ND	ND	ND
18	0	6019	0	0	1246	27	1879	ND	ND	ND	ND	ND
19	17	0	9	0	21	304	14219	ND	ND	ND	ND	D2
20	0	0	4	5	0	10	1876	ND	ND	ND	ND	T1
21	15	21	0	4	114	67	4244	ND	ND	ND	150-300	T2

6.8 Chapter summary

This chapter has discussed the interpretation of dissolved gasses interpreted by using DGA technique. The conditions of power transformers have also been interpreted and discussed. Performance of five diagnostic methods often employed in DGA i.e. key gas, Roger's ratio, Dornenburg ratio, IEC ratio and Duval triangle were used for interpreting results of dissolved gasses were explained. Their abilities in terms of percentage of predictions were given.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

This work developed computer based software for performing condition monitoring of oil filled power transformers. The interface of the developed DGA software, progrmmed by using python language, was made to enable user to input DGA information, save to database and retrieve them for performing diagnoses. MySQL programming language was then used to create database necessary for storing DGA information. The developed DGA software was lastly connected to the database by the use of mysql-python connector. Five diagnostic methods i.e. key gas method, Dornenburg ratio method; Roger's ratio method, IEC ratio method and Duval triangle method were implemented to interpret existing faults. In validating the developed software from results of fifty DGA data from other researchers, Duval triangle method showed the highest percentage of detection and consistency (90%) compared to other diagnostic methods. Moreover, Duval triangle showed the greatest accuracy A_P and A_T (90%) when compared to other researchers' systems. Therefore, the developed DGA software was conformed for condition monitoring practices.

For using the developed software, twenty-eight samples of mineral insulating oil sampled from in-service transformers were taken from Suranaree University of Technology for condition monitoring. From the results, it was seen that overheated in cellulose was the only and most detected fault with key gas method. This technique showed 43.47 % of detection. Thermal fault of temperature greater than 700°C (T3) was the most detected fault in Duval triangle method. Ratio methods failed to detect faults because the majority of transformers were detected to be in Condition 1. This is due to the non-existence of hydrocarbon gasses. In their presence, sometimes ratio methods are not able to detect faults because of the multiple faults existing in the transformer. The existence of multiple faults inside the transformer is indicated by different interpretation from diagnostic methods.

Since overheated in cellulose and thermal fault of temperature greater than 700 °C (T3) are both thermal fault, but T3 is the most serious fault due to the amount of temperature beyond 700 °C. If this fault exist for a long time may weaken the liquid insulation and hence shorten the life of the transformer. The group of transformers with thermal fault T3 should be highly considered in maintenance.

7.2 Future work

The main considerations for the future works should include the following,

(a) To combine diagnostic methods in the developed DGA software with other artificial techniques to improve the diagnostic performance to detect multiple activities occurring inside the transformer liquid insulation.

(b) The results of dissolved gas-in-oil analysis shown in section 6.1 and breakdown strength test (see Table C1 in Appendix C) obtained from twenty-eight samples of mineral insulating oil were used to establish a model for estimating the remained life of the transformer. The results may be useful for future studies if long service DGA data are available.

REFERENCES

Abubakar, S. (2013). Introducing the Hybrid-DGA Interpretation Software as an Effective Power Transformer Management Tool. **4th International Conference on Power Engineering, Energy, and Electrical Drives**, (pp. 1410-1414).

Akbar, A., Setayeshmehr, A., Borsi, H. and Gockenbach, E., (2008), A Software Implementation of the Duval Triangle Method, **IEEE International Symposium on Electrical Insulation**, (pp. 124-127).

Ali Saeed, A., Nor Asia, M., and Abubakar, S. (2012), DGA Interpretation of Oil Filled Transformer Condition Diagnosis, **Trans. Elec. Ele. Mat.**, 13(5): 229-232.

Allen Filters Inc, (2009), An Allen White Paper: Transformer Oil, Part 1, Maintenance of Mineral Transformer Oil. Retrieved from http://cdn2.hubspot.net/hub/272197/file-251812186-pdf/Whitepapers/afi-wp-transoil1.pdf

Ashkezari, A. D., Saha, T.K., Ekanayake, C., and Ma, H. (2011), Evaluating the accuracy of different DGA techniques for improving the transformer oil quality interpretation, **AUPEC**. (pp. 1-6).

Arora, R. and Mosch, W., (2011), **High Voltage and Electrical Insulation Engineering**, USA: Wiley-IEEE Press.

Bartnikas, R., (1994), **Engineering Dielectrics: Electrical Insulating Liquids**, Chelsea: ASTM International.

Bartinikas, R. (1979). **Engineering Dielectrics**. Baltimore: ASTM.

BHEL, (2ed), (2003), **Transformer**, New Delhi: Tata Mc Graw-Hill.

Boudraa, S., Mokhnache, L. and Fofana, I. (2014), A Comparative Study of Impact of Electrical stress and thermal aging on transformer oil, (pp.1-6).

Bräsel, E. and Sasum, U., Online Transformer Gas Diagnostics on the Basis of IEC 60567/60599.

Brian, S., (2003), **Gas Chromatography**, UK: LGC.

Chakravorti, S., Dey, D. and Chatterjee, B., (2013), **Recent Trends in Condition Monitoring of Transformers: Theory, Implementation, and Analysis**, London: Springer.

Chudnovsky, B., (2013), **Electrical Power Transmission and Distribution-Ageing and Life Extension Techniques**, UK: CRC Press

Collacott, R., (1977), **Mechanical Fault Diagnosis and Condition Monitoring**, London: Chapman and Hall Ltd.

Dasgupta, I., (2002), **Design of Transformers**, New Delhi: Tata Mc Graw-Hill.

Dhote, N. K. and Helonde, J. B. (2012), Diagnosis of power transformer faults based on five fuzzy ratio methods, **WSEAS Transactions on Power Systems**, 7(3) 114-125

Donal, S. (2010), **Photo-Acoustic Spectroscopy for dissolved gas analysis**

Duval, M., (2005), Dissolved Gas Analysis and the Duval Triangle.

Duval, M. and dePablo, A. (2001), Interpretation of gas-in-oil analysis using new IEC publication 60599 and IEC TC 10 databases, **IEEE Elec. Ins. Mag**., 17:31-41.

Duval, M. (2002), A Review of Faults Detectable by Gas-in-Oil Analysis in Transformers, **IEEE Elec. Ins. Mag**., 18(3).

Duval, M. (2013), Smart Grid Monitoring of Transformers by DGA", **CIGRE Thailand,** Bangkok.

Dyer, R., (2005), **MYSQL in a Nutshell: A Desktop Quick Reference**, USA; O'Really Media, Inc.

Erdman, H. G., (1988), **Electrical insulating oils, Issue 998**, USA: ASTM International.

Facilities Instructions, Standards, and Techniques (FIST) Volume 3-31, Transformer Diagnostics, June 2013, Bureau of Reclamation, Retrieved from www.usbr.gov/power/data/fist/fist3_31/fist3-31.pdf

Fofana, I., Bouaicha, A., Farzaneh, M., and Sabau, J. (2008). Aging Behavior of Mineral Oil and Ester Liquids: a Comparative Study, **Annual Report Conference on Electrical Insulation Dielectric Phenomena**, (pp. 87-90).

Fofana, I., Wasserberg, V., Borsi, H., and Gockenbach, E., (1999). Determination of the Electrical Properties of Mixed Transformer Fluids under Thermal Ageing, **High Voltage Engineering Symposium**, (pp. 360-363).

Gas chromatograph. (Feb 2017). Gas chromatograph. Retrieved from http://en.wikipedia.org/wiki/Gas_chromatograph

Ghaffarian, M.N. and Edin, H. (2010). Corona in Oil as a Function of Geometry, Temperature, and Humidity. **Annual Report on Electrical Insulation and Dielectric Phenomena**. pp. 1-4. doi:10.1109/CEIDP.2010.5724061

Gill, P., (2ed), (2008), **Electrical Power Equipment Maintenance and Testing**, UK: CRC Press.

Gouda, O., Saleh, S. and El-Hoshy, S. (2016), Power Transformer Incipient Faults Diagnosis Based on Dissolved Gas Analysis, **TELEKOMNIKA Indonesian J. Ele. Eng**. 17 (1).

Harlow, J., (3ed), (2012), **Electrical Power Transformer Engineering**, US: CRC Press.

Hell, M. B., D'Angelo, M. S. F. V., Costa Jr, P. P., Power Transformer Fault Diagnosis Based on Dissolved Gas Analysis Using a Fuzzy Neural Network Approach in a Real Data Base.

Hirschler, M. M., (2000), **Electrical Insulating Materials: International Issues, Issue 1376**, Baltimore: ASTM International.

Horning, M., Kelly, J., Myers, S., and Stebbins, R., (2013), **Transformer Maintenance eGuide**, USA: SD Myers, Inc.

Huo-Ching S., Yann-Chang H., and Chao-Ming H. (2012), A Review of Dissolved Gas Analysis in Power Transformers, **Energy Proc**. 14: 1220 – 1225.

IEEE C57.104-2008, (2009), **IEEE Guide for the Interpretation of Gases Generated in Oil-Immersed Transformers**, IEEE Power & Energy Society.

IEC 60270. (2000). **High Voltage Test Techniques – Partial Discharge Measurements**. IEC.

Indra, G., Rajaram, M. and Gnanasaravanan, A., (2000), Power Transformer Faults Identification Using Fuzzy Based Dissolved Gas Analysis Method, **J. Elec. Eng.** 1: 1-5.

IS 1866:2000 – **Code of Practice for Electrical Maintenance and Supervision of Mineral Insulating Oil in equipment**. Retrieved from http://law.resource.org /pub/in/bis/S05/is.1866.2000.pdf

Izeki, N., Kurahashi, A. and Matsuura, K. (1971). The behavior of Oil Corona and Damage of Transformer Insulation. **IEEE Transactions on Power**

Apparatus and System. Vol. PAS-90, Issues: 5, pp. 2330-2338, doi: 10.1109/TPAS.1971.293081

James, R.E., and Su, Q., (2008), **Condition Assessment of High Voltage Insulation in Power System Equipment**, UK: IET.

Jiang, Y., (2006). **Condition-based Hazard Rate Estimation and Optimal Maintenance Scheduling**, Pro Quest,

Jiri, S. (1975), Detectors in Gas Chromatography: **J. Chromatography Library Volume4**, Amsterdam: Elsevier Scientific Publishing Company.

Karl Fischer method. (2013, May 11). Karl Fischer method. Retrieved from http://www.dins.jp/4kf/kf.htm

Lutz, M., (2010), **Powerful Object Oriented Programming; Programming Python**, USA: O'Really Media, Inc.

Michelle, G.C., John, R.D. (2013), **Forensic Application of Gas Chromatography**, UK: CRC Press.

Muhamad, N. A., Phung, B. T., Blackburn, T. R., and Lai, K. X.(2012), Comparative Study and Analysis of DGA Methods for Transformer Mineral Oil, **IEEE Pow. Tech**. pp. 45-50.

Nemeth, B., Laboncz S. and Kiss, I. (2009). Condition monitoring for power transformers using DGA and fuzzy logic, **IEEE Elect. Ins. Conf**., pp. 373-376, June 2009.

Norazhar, A. and Abu Siada, A., (2013), High Voltage Power Transformer Dissolved Gas Analysis, **HV Maintenance Forum-IDC Technologies**.

Oswald, X. H., James, G. B. and Plymouth, M. (1964). Corona Measuring Techniques and Their Use in Insulation System Evaluation. **IEEE Transactions on Component Parts**, Vol. 11, Issue: 2, pp. 299-316. doi:10.1109/TCP.1964.1134981

Python, (2015), Python (programming language), Retrieved from http://www.en.wikipedia.org/wiki/Python_(programming_language)

Rahul P., and Deshpande, M.T., (2012), Dissolved Gas Analysis of Mineral Oil Used in Transformer, **J. Appl. Innova. Eng. Manag**.1: 208-212.

Reddy, N. V., (2014), Dissolved Gas Analysis – An early identification of faults in high voltage power equipment using MATLAB GUI, Master Thesis, National Institute of Technology, Rourkela, India. Retrieved from http://www.nitrkl.ac.in/

Rizk, A. M., and Trinh, N. G., (2014), **High Voltage Engineering**, UK: CRC Press.

Rogers, R. R., (1978), IEEE and IEC codes to interpret incipient faults in transformers, using gas in oil analysis, **IEEE Trans. Electr. Ins**.

Rohit, K. A., (2013), Different DGA Techniques for Monitoring of Transformers, **J. Ele. Ele. Eng**.

Rohwein, G. J. (1996). Corona Processing in Insulating Oil. **22nd International Power Modulator Symposium.** pp. 141-144, doi: 10.1109/MODSYM.1996.564471

Rondla, P., Falahi, M., Zhan, W. and Goulart A., (2012), A regression algorithm for transformer fault detection, **IEEE Pow. En. Soc**. (pp. 7126-7130).

Serveron Corporation, (2007), Serveron White Paper: DGA Diagnostic Methods, Retrieved from http://www.serveron.com.

Sherif, S., Ghoneim, M. and Sayed, A. W. (2012), Dissolved Gas Analysis as a Diagnostic Tool for Early Detection of Transformer Faults, **Adv. Ele. Eng. Sys**. 1: 152-156.

Shrivastava, K., and Choubey, A. (2012), A Novel Association Rule Mining with IEC Ratio based Dissolved Gas Analysis for Fault Diagnosis of Power Transformers" **J. Adv. Comp. Res.**, 2: 2249-7277.

Sukbirr, S., and Bandyopadhyay, M.N. (2010), Duval Triangle: A Noble Technique for DGA in Power Transformers, **J. Ele. Pow. Eng**., 4: 193-197.

Taha, I. B. M., Zaini, H. G. and Ghoneim, S. S. M. (2015), Comparative study between Dornenburg and Roger's ratio methods for transformer fault diagnosis based on Dissolved Gas Analysis using MATLAB Simulink tools, 363-367.

Transformer insulating oil and types of transformer oil. (2017, Feb). Retrieved from http://www.electrical4u.com/transformer-insulating-oil-and-types-of-transformer-oil/.

Urquiza, D., García, B., Burgos, J.C. and Pirez J.L. **DGA: reference values from Power Transformers in Spain**.

Xin, Y., **Electrical Insulating Liquids**, ASTM International: Baltimore

Yadav, K. S., and Sarathi, R. (2015), Influence of thermally aged barrier on corona discharge activity in transformer oil under AC voltages, **IEEE Trans. Diel. Elect. Ins**, 22: 2415-2423.

APPENDIX A

A Study of Condition Monitoring of Oil Immersed Transformer via Results of Breakdown Strength and Dissolved Gas In-Oil of Mineral Insulating Oil

Pius Victor Chombo and Boonruang Marungsri* and Thanatchai Kulworawanichpong

School of Electrical Engineering, Institute of Engineering, Suranaree University of Technology
111 University Avenue, Nakhon Ratchasima 30000, Thailand, Tel. 098-202-9237, Email: piusvictor2013@gmail.com
*Corresponding Author: Tel. 089717-7065, Email: bmshvee@sut.ac.th

***Abstract*—Condition monitoring of oil immersed transformer has become a crucial study among researchers. Current techniques especially DGA use only dissolved gases to interpret the condition of oil filled transformer while internal transformer faults result into dissolved gases and loss of oil quality. This work presents investigation of existing relationship between breakdown strength and dissolved gas in-oil of mineral oil. Observations showed unclear relationship existing between breakdown strength and individual dissolved gas. It was suggested to investigate existing relationship over a long service time.**

***Index Terms*—Oil Immersed transformer, Condition monitoring, Dissolved gases, Breakdown strength**

I. Introduction

Condition based monitoring has been widely employed as a tool for monitoring and hence planning the maintenance of transformers in the power system. Transformer is most important and expensive element in which its performance can affect reliability of the power system. Under condition based monitoring incipient faults can be easily detected and therefore minimize unnecessary cost due to unplanned failures in the transformer [1].

During occurrence of stresses (electrical or thermal stress [2], [3], [6]) in a transformer, multiple activities occur in transformer's insulation i.e. partial discharge (corona), arcing or overheating [4], [6]. Due to stresses insulating oil and/or cellulose are degraded into fault gases which totally or partly dissolved in oil [4-5]. It is normally known that incipient faults inside the transformer produce excessive temperature which decomposes mineral insulating oil (liquid insulation). Liquid insulation especially mineral oil is composed of hydrocarbon components. When decomposition of liquid insulation is undertaken electrochemical stability of hydrocarbons breaks down [7-9]. Yardaz [8] explained that breakdown of solid and liquid insulation molecules may formulate new molecules and acidic residues which are hazardous for cellulose. Furthermore, in [7], [9] the breakdown of hydrocarbons was explained not only to produce dissolved gases but also colloidal suspensions. Apart from dissolved gases, the negative side of colloidal suspensions includes acceleration of oxidation in liquid insulation which results into increased acidity level and hence destruction in solid insulation; and formation of insoluble sludge which reduces the cooling rate of insulation system [7], [9]. Therefore, accelerated oxidation and formation of insoluble sludge greatly reduce the quality of transformer insulation especially liquid insulation. Liquid insulation can be used as an important informative tool for condition monitoring with about 70% of occurred situations [7]. However, previous works have demonstrated condition monitoring of transformer with the use of dissolved gases only []without considering the quality of liquid insulation. The accumulation of dissolved gases in oil reduces the quality and ability of insulating oil to withstand stress.

This work performed an investigation to study existing relationship between dissolved gas in oil and breakdown strength of mineral oil. From the obtained results, individual dissolved gas was related with the breakdown strength.

II. Experiments

A. Samples.

24 Samples of mineral insulating oil were collected from in-service power transformers with 1.25 MVA 22/0.4/023 kV 50 Hz in Suranaree University. The collection was done during daytimes at humidity less than 50% to make sure contamination of moisture is within limited level. Then samples were taken for measurement of dissolved gas in oil and breakdown strength.

B. Dissolved Gas In-Oil.

Syringes of 100 cc were used for measurement of dissolved gases in oil. The commercial DGA equipment Myrkos – Portable DGA MicroGC was used for extracting and measuring dissolved gases from samples. It measured separately seven gases - most significantly fault gases within

120 seconds. Interested dissolved gases were H_2, CO, CO_2, CH_4, C_2H_6, C_2H_4 and C_2H_2. Other air components (O_2 and N_2) were also measured in a separate kit but not included in the investigation. All dissolved gases were expressed in parts per million (ppm).

C. *Breakdown Strength.*

This test was done according to IEC 60156. Test cell used had mushroom shaped electrodes with 2.0mm gap. Each sample was tested 6 times by applying alternating voltage from 0 V in a rate of 2kV/s until breakdown occurred. All samples were tested in daytimes at humidity less than 50%. Fig.1 shows equipment used for testing breakdown strength.

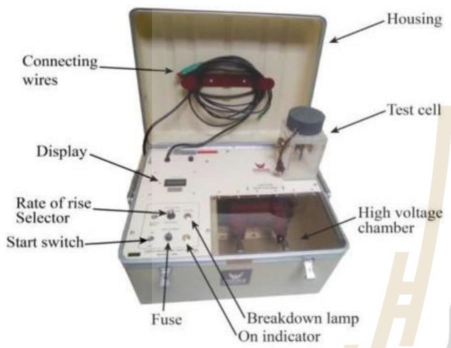


Fig. 1. Breakdown strength equipment

Obtained results of each sample were used in computation of mean, percentage of range to mean and deviation as per (1) to (3) respectively. For a valid test, (2) and (3) should give a value less than or equal to 92% and 4% respectively.

$$\text{Mean} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad (1)$$

$$\%\frac{\text{Range}}{\text{Mean}} = \frac{\text{Range}}{\text{Mean}} \times 100 \quad (2)$$

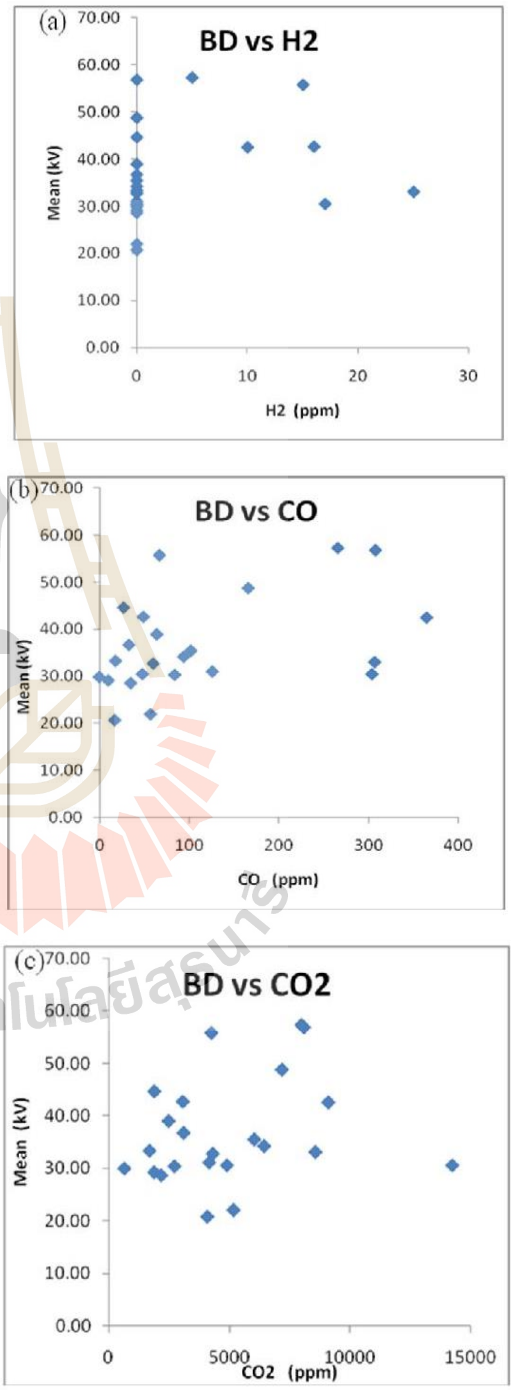
$$\text{Deviation} = \sqrt{\frac{\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2}{n-1}} \quad (3)$$

x_i = test results for i=1, 2, ..., n, $\bar{x}$ =mean obtained in (1)

n=6 (number of tests per sample)

III. Results and Discussion

Mean value of each valid test was plotted with individual dissolved gas. Fig.2 shows the results of mean values for all measured breakdown strength plotted against individual gases. BD represents breakdown strength.



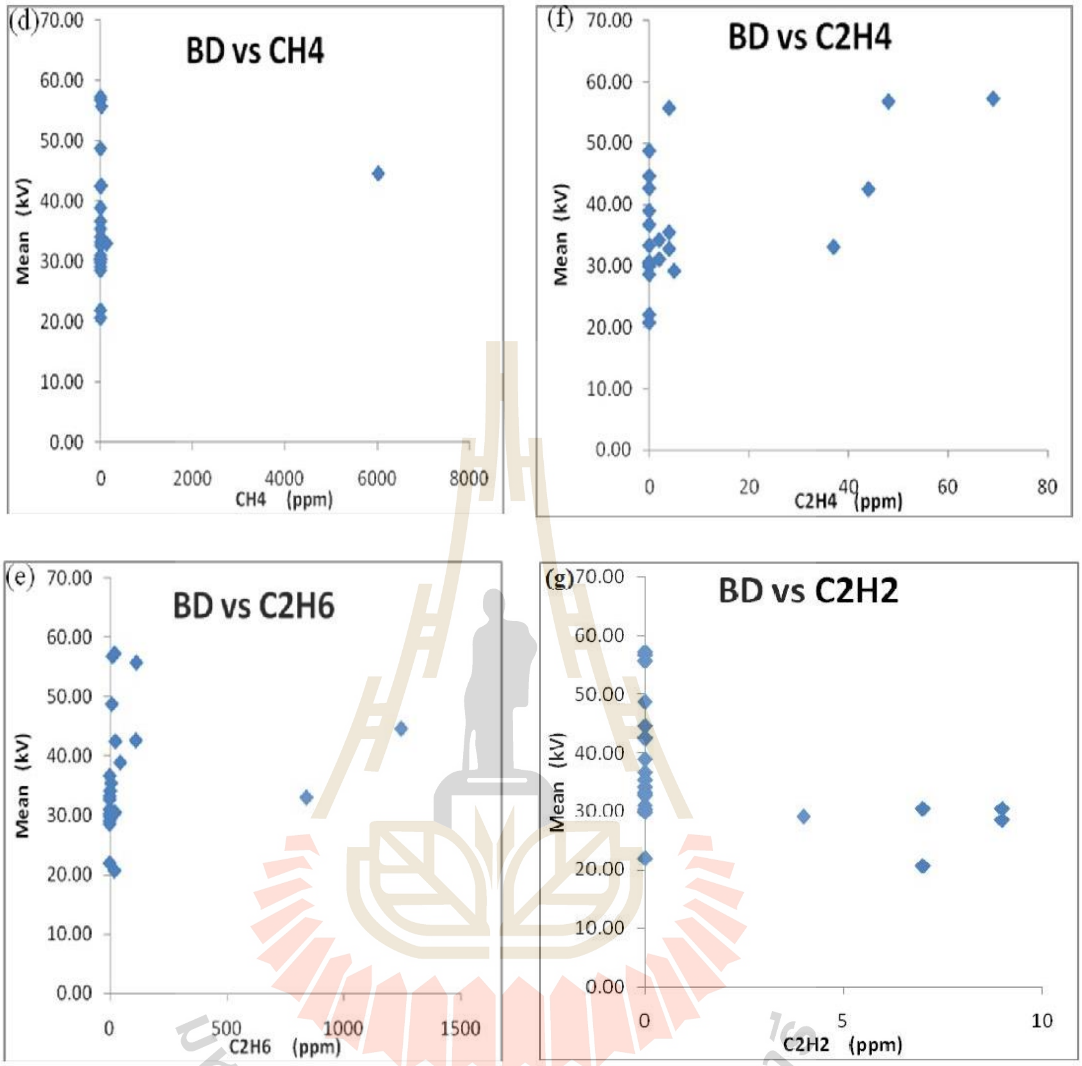


Fig. 2. Plots of mean of breakdown strength of samples and individual dissolved gas (a) H_2 (b) CO (c) CO_2 (d) CH_4 (e) C_2H_6 (f) C_2H_4 (g) C_2H_2

In order to investigate existing relationship between breakdown strength and dissolved gases, each individual gas was related with breakdown strength values (see Fig.2 (a)-(g)).

From Fig.2, it can be observed that mean values of measured breakdown strength was varying between 20kV and 60kV. In the subplot of H_2 (see Fig.2a) it can be seen that in the absence of H_2 the breakdown values of 20kV and 60kV were obtained. The same occurred to CH_4, C_2H_6, C_2H_4 and C_2H_2.

This gives unclear meaning when interpreting the initial behavior of dissolved gas to breakdown strength. However, the similar values were also observed to occur in higher levels of ppm (see in subplots of H_2, CO, CO_2 and C_2H_4). It indicates that breakdown strength can either be affected by high level of individual dissolved gas or not.

For the case of CH_4, in absence of CH_4 there was a significant change in breakdown strength (60kV to 20kV) compared to its presence (say 6000 ppm) whereby the breakdown strength was dropped only from 60kV to 50kV.

IV. Conclusion

This work was performed to investigate the existence of relationship between breakdown strength and dissolved gases in mineral insulating oil of power transformers. The breakdown strength was based on IEC 60156 with mushroom shaped electrodes separated by a gap of 2.0mm. Mean values of breakdown strength from valid test results were related with individual dissolved gases. In most of individual gases the mean value of breakdown strength was seen to range between 20kV and 60kV even in absence of dissolved gases. However, the highest value of breakdown strength was occurred in higher level of dissolved gases. It was concluded that no clear relationship observed to exist between breakdown strength and individual dissolved gas. Furthermore, the investigation was suggested to be performed over a long service time.

Acknowledgment

This work was supported by Laboratory of High Voltage Insulation Technologies of Suranaree University of Technology, Thailand.

References

[1] I.B. Taha, H.G. Zaini and S.S. Ghoneim"Comparative study between dornenburg and rogers methods for transformer fault diagnosis based on dissolved gas analysis using matlab simulink tools," IEEE Conf. En. Conv.(CENCON),2015,pp: 363-367.

[2] P. Rondla, M. Falahi, W. Zhan, and A.Goulart, "A regression algorithm for transformer fault detection ," IEEE Pow. Ener. Soc. pp. 7126-7130, 2012.

[3] A. D. Ashkezari, T.K. Saha, C. Ekanayake, and H.Ma, "Evaluating the accuracy of different DGA techniques for improving the transformer oil quality interpretation ," AUPEC. pp. 1-6, 2011.

[4] B.Nemeth, S. Laboncz and I.Kiss, "Condition monitoring for power transformers uding DGA and fuzzy logic," IEEE Elect. Ins. Conf., pp. 373-376, June, 2009.

[5] N. K. Dhote and J.B.Belonde, "Diagnosis of power transformer faults based on five fuzzy ratio method," WSEAS Trans. Pow. Sys. vol. 7, Issue. 3, pp. 114-125, July, 2011.

[6] N. A. Muhamad, B.T. Phung, T.R. Blackburn and K.X.Lai, "Comparative study and analysis of DGA methods for transformer mineral oil," IEEE Power Tech., pp. 45-50, January, 2016

[7] S. Boudraa, L. Mokhnache and I. Fofana, "A Comparative Study of Impact of Electrical Stress and Thermal Aging on Transformer Oil," 11th IEEE International Multi-Conference on Systems, Signals & Devices (SSD14), pp.1-6, 2014.

[8] K. S. Yadav and R. Sarathi, "Influence of Thermally Aged Barrier on Corona Discharge Activity in Transformer Oil Under AC Voltages," IEEE Transactions on Dielectrics and Electrical Insulation, Vol. 22, No.5, pp.2415-2423, 2015.

[9] I. Fofana, A. Bouaicha, M. Farzaneh and J. Sabau, "Ageing Behaviour of Mineral Oil and Ester Liquids: a Comparative Study," Annual Report Conference on Electrical Insulation and Dielectric Phenomena, pp.87-90, 2008..

Implementation of Computer based Software for Oil Immersed Power Transformer Condition Monitoring via Dissolved Gas In-oil Results

Pius Victor [1], Boonruang Marungsri [2*]
[1] School of Electrical Engineering, Suranaree University of Technology, Thailand
111 University Avenue, Nakhon Ratchasima 30000, Tel. 093-643-8346, Email: piusvictor2013@gmail.com
[2*] Corresponding Author Tel. 089717-7065-, Email: bmshvee@sut.ac.th

Abstract—**The vital role of power transformers in the power system has attracted the consideration of Condition based Maintenance for many years. Dissolved gas in-oil analysis (DGA) has gained popularity in condition monitoring and diagnosis through its diagnosing methods like Key Gas, Dornenburg, Rogers, IEC and Duval Triangle. This paper presents the implementation of computer based software with Database for transformer condition monitoring with the use of Dissolved Gas in-oil Analysis. It involves database for storage of DGA information in which they can be retrieved by the software for fault diagnosis through the use of five diagnostic methods (Key Gas method, Dornenburg ratio method, Roger's ratio method, IEC ratio method and Duval Triangle method). 50 samples from other researchers were taken, stored and tested for proving the performance of the DGA software. The DGA software showed the accuracy of more than 90 percent in fault interpretation (with the use of Duval Triangle). The implementation involved the use of database made with MySQL 5.5 for storage of DGA information, DGA software made with Python 2.7 and compiled with Eclipse in Ubuntu 14.04 (computer operating system).**

Index Terms— **Condition monitoring, Database, DGA, Power transformers.**

INTRODUCTION

REGARDING the importance of power transformer in the power system and the dynamic nature of faults that occur in it, condition based maintenance has been much considered as the best maintenance practice. This is due to the fact that most of the faults are accumulated for a certain period of time. Degradation of insulation in oil filled transformer is a normal process, it occurs even normal load conditions but abnormal conditions accelerates the rate of degradation which affects electrical, chemical and physical properties of insulating oil [6]. For the case of distribution networks with a wide range of voltages like in Thailand [2] power transformers are available in large number, the use of database can reduce tedious works of inputting DGA information for interpretation. Some of the DGA software developed are described in [2], [3], [4], [5] but they do not have database i.e. they request user input for every diagnostic. All these cannot make reference with the previous DGA information. This leads to unnecessary time consumption when user wants to compare or to know the trend of internal condition in power transformers. Five Condition assessment tools for power transformers through DGA (Key Gas, Dornenburg, Rogers, IEC and Duval) have been reviewed and implemented in software bases. Paper also opens the way forward to the access and use of database raw data for diagnosis; evaluate method(s) used, data comparison, gases trending, saved work summary for future evaluation.

An easy graphical user interface was made by using Python programming language. According to [7], it is defined to be a general purpose, open source computer programming language, optimized for software quality, developer productivity, program portability and component integration. Also Python is said to support external components coded in other languages; a scripting language as it makes it easy to utilize and direct other software components.

As in [8], MySQL was stated as an open source, multi-threaded, relational database management system created by Michael "Monty" Widenius in 1995, and in 2000, MySQL was released under a dual license model that permitted the public to use it for free under the GNU Public License (GPL); which caused its popularity to be soaring. The success of MySQL as a leading database was defined due not only to its price-after all, other cost-free and open source databases are available-but also its reliability and performances. Most features contributed to MySQL's standing as a superb database system include speed and the storage engine.

In this paper, DGA software features and basics of condition monitoring of power transformer are detailed in Section 2; revision of Condition assessment tools for DGA, their implementation, usage of software for diagnosis, gas comparison and trending, report and work summary are explained in Section III. The results of diagnosis by different assessment method in comparison with other researchers are described in Section IV and conclusion is given in Section V. The diagnosis and evaluation of DGA works are according to IEEE and IEC standards.

II. FEATURES OF DGA SOFTWARE

A. *Graphical User Interface Layout*

A simple GUI consists of three parts. (i) User Input Form-for saving gas details (H_2, CO, CO_2, N_2, CH_4, C_2H_6, C_2H_4 and C_2H_2) and non-gas details (Power-kVA, Voltage-kV, Altitude-M, Humidity-%, Oil Temperature-C, Location-name, Sampling date-date) for further evaluation and decision making about power transformer. (ii) Toolbar with five diagnosing methods (KGM, Dornenburg, Rogers, IEC, Duval), gas comparison and gas trending. (iii) Menu bar with options for customizing software and database details. After saving gas and non-gas data to the database, user can easily pick any data by data ID from the DGA software, and diagnose by using any diagnosing method. All diagnoses give gas details, execution, analysis and graphical representation which are easy to learn and understand.

B. *Database*

A MySQL database was created and connected to the software. A database has a table in it with eleven columns named ID (primary key); hydrogen, carbon_monoxide, carbon dioxide, nitrogen, methane, ethane, ethylene, acetylene, power, voltage, altitude, oil temperature, humidity, sampling_date, machine_location, machine_number. User has nothing to do with the database rather than saving and picking data which is done through the use of DGA software.

C. *Development Tools and Algorithm of DGA Software*

DGA Software was developed in Ubuntu 14.04 (operating system) and database connected with it. Other details are shown in Table I below. The software works effectively in this operating system. Figure 1 shows the flow chart of DGA software in which user enters DGA information and can choose whether to diagnose by any of the five methods, compare or trend. After each operation (diagnosis, comparison or trending) DGA software displays a report which includes method execution, graphical representation and analysis. It also saves the report in a plain text document which can easily be viewed by document viewer. Other plain text documents can be seen through the software while (for further evaluations) others can be viewed by computer document viewer (for report purposes).

TABLE I TOOLS USED FOR THE DEVELOPMENT OF DGA-SOFTWARE

Tool	Type
Computer	HP 550 Duo Core
Operating System	Ubuntu 14.04
Programming Language	Python 2.7
Compiler	Eclipse
Database	MySQL 5.5

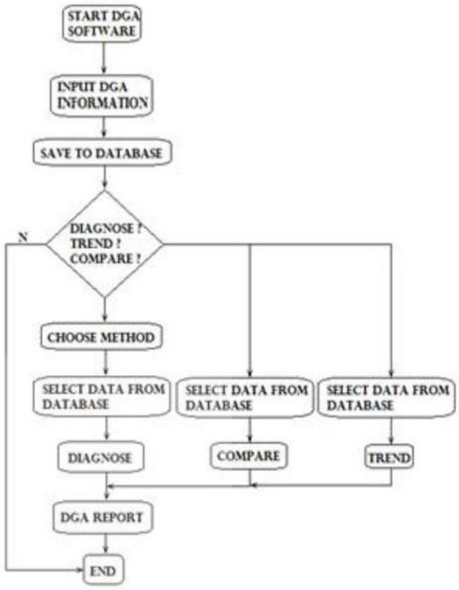


Fig. 1. Algorithm of DGA software

D. *Condition Classification for DGA Methods*

Five methods were used to interpret DGA information and analyze the condition of power transformer. This DGA Software gives user an opportunity to select any data from database and use any DGA method to diagnose it at a time as interpretation varies with the method.

E. *Key Gas Method*

This method employs principal gases to detect the presence of faults in the power transformer; these gases are H_2, CO, CH_4, C_2H_6, C_2H_4 and C_2H_2 whereby the significant and proportion of the gases are called "key gases". The fault is identified according to the presence and percentage of each key gas [9], [10]. Table II below shows the interpretation of different faults by key gas method.

TABLE II COMBUSTIBLE GAS IN KEY GAS ANALYSIS [9]

Gas	Normal	Abnormal	Interpretation
H_2	< 150 ppm	> 1,000 ppm	Arcing, Corona
CH_4	< 25 ppm	> 80 ppm	Sparking
C_2H_6	< 10 ppm	> 35 ppm	Local Overheating
C_2H_4	< 20 ppm	> 100 ppm	Severe Overheating
CO	< 500 ppm	> 1,000 ppm	Severe Overheating
CO_2	< 1,000 ppm	> 15,000 ppm	Severe Overheating
N_2	1 – 10%	NA	NA
O_2	0.2 – 3.5%	NA > 0.5%	Combustibles

F. *Dornenburg Ratio Method*

This method first examines individual gas, and recommends a test if any gas is beyond the set up limit in parts per million as shown in Table III. Three faults can be detected by this method while five gases and four ratios are used for detecting faults as shown in Table IV. Dornenburg interpretation is described in Table V.

TABLE III
DORNENBURG LIMITS FOR INDIVIDUAL GAS, L_1 [10], [11]

Gas	Limit, ppm
H_2	100
CH_4	120
C_2H_4	50
C_2H_2	35
C_2H_6	1
CO	350

TABLE IV
DORNENBURG GAS RATIOS [10], [11], [12]

Ratio	Gas
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

TABLE V
DORNENBURG FAULT INTERPRETATION [10], [11]

Fault diagnosis	CH_4 / H_2	C_2H_2/C_2H_4	C_2H_2/ CH_4	C_2H_6/C_2H_2
Thermal Decomp.	> 1.0	< 0.75	< 0.3	> 0.4
Corona	< 0.1	-	< 0.3	> 0.4
Arcing	0.15 – 1.0	> 0.75	> 0.3	< 0.4

G. Roger's Ratio Method

In previous, this method was using four gas ratios which were CH_4/H_2, C_2H_6/CH_4, C_2H_4/C_2H_6 and C_2H_2/C_2H_4 for diagnosis whereby the ratio C_2H_6/CH_4 only indicated a limited temperature range of decomposition, but did not assist in further identification of fault and hence was deleted. The improved Roger's method uses the remained three ratios to analyze six conditions of the power transformer [9], [10], [11], [12]. Two tables are used in improved Roger's method: one defined the code of the ratio, and the other defined the diagnosis rule as shown in Table VI and Table VII respectively [12].

TABLE VI ROGER'S GAS RATIOS [11]

Ratio	Gas
Ratio 1	CH_4 / H_2
Ratio 2	C_2H_2 / C_2H_4
Ratio 5	C_2H_4 / C_2H_6

TABLE VII ROGER'S FAULT INTERPRETATION [10], [11]

Case	Fault diagnosis	C_2H_2/C_2H_4	CH_4 / H_2	C_2H_4 / C_2H_6
0	Unit Normal	< 0.1	>0.1 to <1.0	<1.0
1	Low energy density arcing	< 0.1	< 0.1	<1.0
2	Arcing-High energy discharge	0.1 – 3.0	0.1 to 1.0	>3.0
3	Low temperature-thermal	< 0.1	>0.1 to <1.0	1.0 to 3.0
4	Thermal < 700 °C	< 0.1	>1.0	1.0 to 3.0
5	Thermal > 700 °C	< 0.1	>1.0	>3.0

H. IEC Ratio Method

Five gases, four gas ratios and seven fault types are involved in this method as described in Table VIII and Table IX below.

TABLE VIII IEC GAS RATIOS [11]

Ratio	Gas
Ratio 1	CH_4 / H_2
Ratio 2	C_2H_2 / C_2H_4
Ratio 5	C_2H_4 / C_2H_6

TABLE IX IEC FAULT INTERPRETATION [11]

Case	Fault diagnosis	C_2H_2/C_2H_4	CH_4 / H_2	C_2H_4 / C_2H_6
PD	Partial Discharge	-	< 0.1	< 0.2
D1	Low Energy Discharges	>1.0	0.1 to 0.5	>1.0
D2	High Energy Discharge	0.6 – 2.5	0.1 to 1.0	>2.0
T1	Thermal Fault <300 °C	-	>1.0	<1.0
T2	Thermal 300 - 700 °C	< 0.1	>1.0	1.0 to 4.0
T3	Thermal > 700 °C	< 0.2	>1.0	>4.0

I. Duval Triangle Method

This method was developed by Michel Duval in 1974; it employs three hydrocarbon gases (CH_4, C_2H_4 and C_2H_2) in triangular map and detects seven fault types corresponding to the increasing in energy levels of gas formation in transformers in service [3], [12].

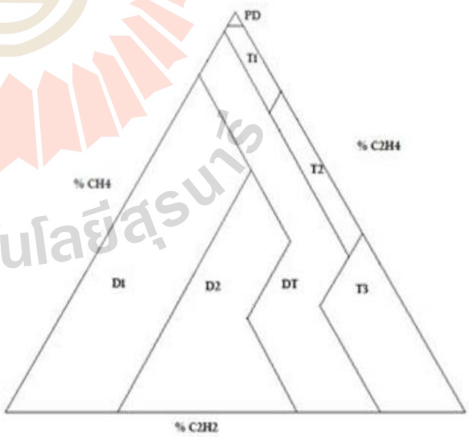


Fig. 2. Duval Triangle

The fault zone is identified by an intersection point between parallel lines obtained from the percentages of fault gases as shown in equation (1), (2) and (3).

$$\%CH_4 = 100xCH_4/(CH_4 + C_2H_2 + C_2H_4) \quad (1)$$

$$\%C_2H_4 = 100xC_2H_4/(CH_4 + C_2H_2 + C_2H_4) \quad (2)$$

$$\%C_2H_2 = 100xC_2H_2/(CH_4 + C_2H_2 + C_2H_4) \quad (3)$$

III. Implementation of DGA Computer Software

A MySQL database was created and named as "dissolved" and connected to the DGA software. It was saved in a computer server as shown in a list of database in Figure 3.

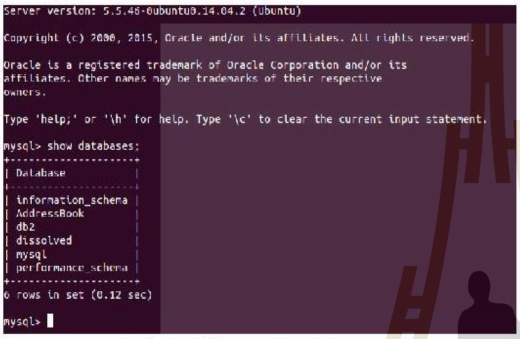


Fig. 3. List of Databases in a computer server

A database has two tables but the one named "dga" will be used as shown in Figure 4. It has eleven columns named ID (primary key); hydrogen, carbon_monoxide, carbon dioxide, nitrogen, methane, ethane, ethylene, acetylene, power, voltage, altitude, oil temperature, humidity, sampling_date, machine_location, machine_number as shown in Figure 5. User has nothing to do with the database rather than saving and picking data which is done through the use of DGA software

Fig. 4. List of tables in a database

The interface of the developed computer program is shown in Figure 6 with five diagnosing methods for interpreting DGA information (fault diagnosis) and two condition monitoring tools (comparison – two gases and trending – more than two gases) in the toolbar. Every DGA information when is stored in the database, it automatically registered with a unique ID which provides an easiest way to find it by using DGA software for diagnosis.

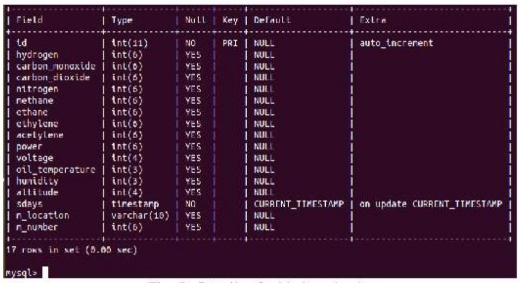


Fig. 5. Details of table in a database

DGA software was connected with the MySQL database by using mysql-python connector.

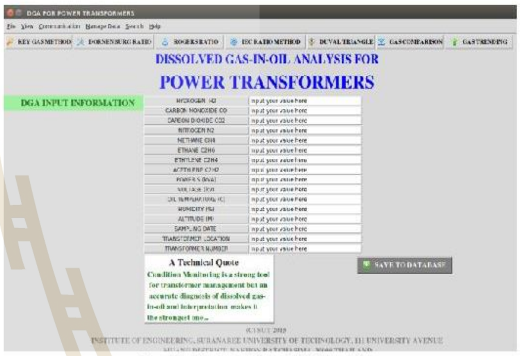


Fig. 6. The interface of DGA software

From the interface, user can opt to retrieve single or multiple DGA information from the database. Figure 7 shows single data selected from the database its ID which returns gas and non gas details.

Fig. 7. Single data retrieved from the database

Figure 8 shows multiple data selected from the database by their IDs which gives gas and non gas details.

The interface of Key Gas diagnostic method of the DGA software is shown in Figure 9.

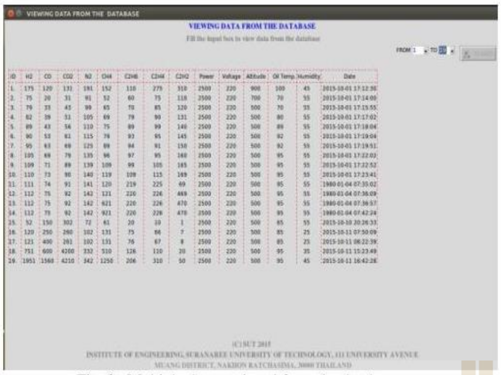


Fig. 8. Multiple data retrieved from the database

It includes user request for DGA data, DGA information, computations, graphical representation and result analysis.

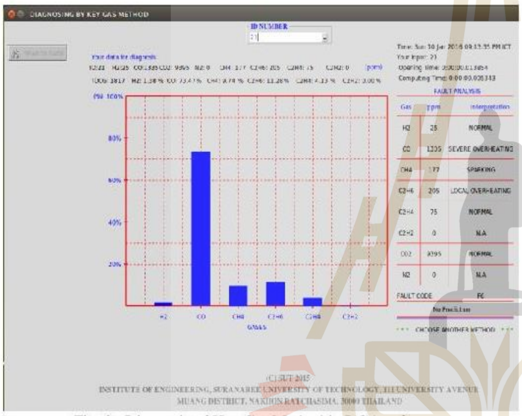


Fig. 9. Diagnosis of Key Gas Method in DGA software

The interface of Dornenburg diagnostic method in the DGA software is shown in Figure 10 including user request for DGA data, DGA information, computations and result analysis.

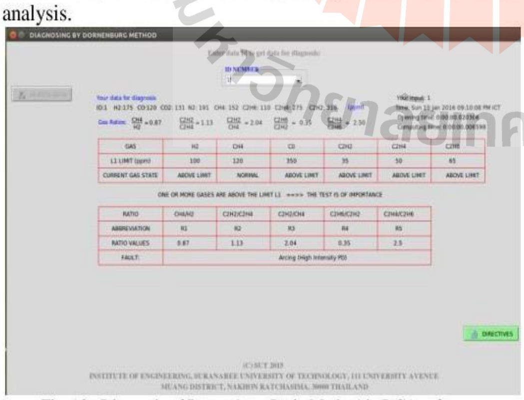


Fig. 10. Diagnosis of Dornenburg Ratio Method in DGA software

The interface of Roger's Ratio diagnostic method in the DGA software is shown in Figure 11 including user request for DGA data, DGA information, computations and analysis.

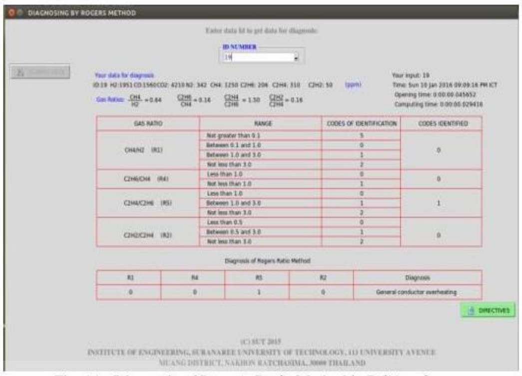


Fig. 11. Diagnosis of Roger's Ratio Method in DGA software

The interface of IEC Ratio diagnostic method of the DGA software is shown in Figure 12 including user request for DGA data, DGA information, computations and analysis.

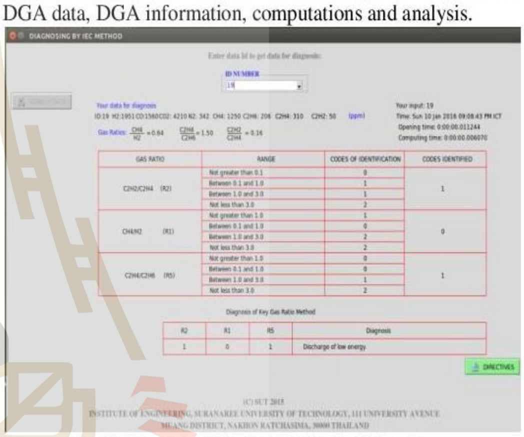


Fig. 12. Diagnosis of IEC Ratio Method in DGA software

The interface of Duval Triangle in the DGA software is shown in Figure 13 including user request for DGA data, DGA information, computations and analysis.

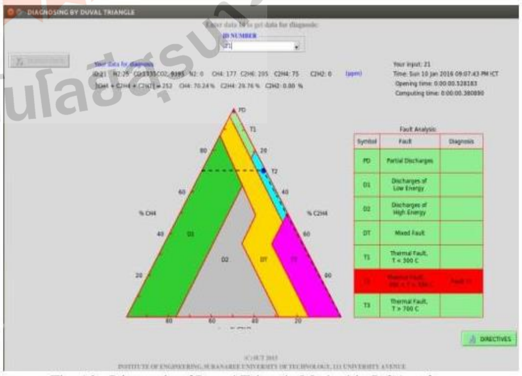


Fig. 13. Diagnosis of Duval Triangle Method in DGA software

After diagnosis DGA software saves the summary of diagnosis in plain text that can be easily viewed by a computer viewer as shown in Figure 14.

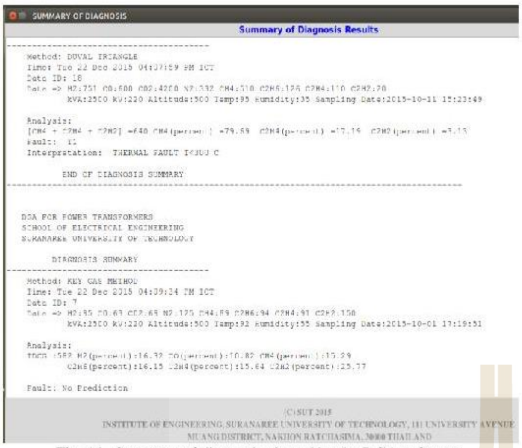


Fig. 14. Summary of diagnosis viewed by the DGA software

The DGA software is able to compare data of the same transformer sampled in different sampling periods and saved in the database as shown in Figure 15.

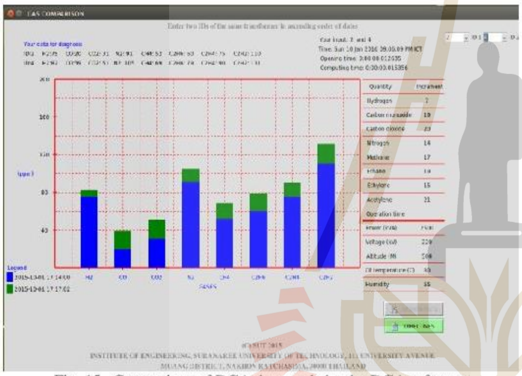


Fig. 15. Comparison of DGA data made by the DGA software

IV. RESULTS

A. *Software Validation*

50 DGA data from Oil Filled Transformers with different conditions were taken from other researchers [4], [12] for testing the DGA software. The DGA data were first stored in the DGA database and then retrieved and diagnosed by each diagnostic method.

B. *Percentage of Successful Prediction and Consistency*

The number of predictions done by five diagnostic methods with DGA data in Table X were used to calculate percentages of successful predictions (P) and consistencies (C) as shown in Table X. The formulas are shown in equation (4) and equation (5) as referred [13]. The summary of comparison of interpretation results obtained by using five diagnostic methods is also shown in Table X.

$$\%Success\,Pred. = \frac{100x Numberof\,Predictions}{NumberofCases} \tag{4}$$

$$\%Consistency = \frac{100x Correct\,Predictions}{Numberof\,Predictions} \tag{5}$$

TABLE X
INTERPRETATION SUMMARY OF FIVE METHODS FROM DGA SOFTWARE

Method	Number of Predictions	% Successful Prediction, P	Correct Predictions	% Consistency, C
Key Gas	3	6	2	67
Dornen	8	16	6	75
Rogers	38	76	29	76
IEC	18	36	14	77
Duval	50	100	48	96

C. *Accuracy of the Diagnostic Method*

Furthermore, the diagnostic method was evaluated by its accuracy. Reference [13] divided the accuracy into two categories. The first category involves accuracy based only on predicted cases (Tp) while the second one involves only the total number of cases (Tc). Both cases can be calculated by using equation (6) and (7) respectively.

$$Accuracybasedon\,Prediction = \frac{100xT_R}{T_P} \tag{6}$$

$$AccuracybasedonTotalCases = \frac{100xT_R}{T_C} \tag{7}$$

The overall accuracies between DGA software and from others researchers can be seen in Table XII for better understanding of DGA performance. The initials in Table XII can be compared with the previous Table XI. In Table XII, D represented results from DGA software while O represented those of other researchers.

TABLE XI
COMPARISON OF ACCURACIES OF DGA DIAGNOSTIC METHODS

	KGM	Dorn	Rog	IEC	Duv
Total Cases, T_C	50	50	50	50	50
No Prediction, T_{NP}	47	42	12	32	0
Number of Predictions, T_P	3	8	38	18	50
Correct Predictions, T_R	2	6	29	14	48
Incorrect Predictions, T_W	1	2	11	4	2
Accuracy (Predicted cases), A_P	67	75	76	77	96
Accuracy (Total cases), A_T	4	12	38	28	96

The summary of Table XII was plotted in a chart for easy interpretation as seen in Figure 14.

TABLE XII
COMPARISON OF ACCURACIES OF DGA AND OTHER RESEARCHERS

	KGM		Dorn		Rog		IEC		Duval	
	D	O	D	O	D	O	D	O	D	O
T_C	50	50	50	50	50	50	50	50	50	50
T_{NP}	47	46	42	4	12	20	32	6	0	0
T_P	3	4	8	46	38	30	18	44	50	50
T_R	2	3	6	20	29	18	14	38	48	44
T_W	1	1	2	14	11	12	4	6	2	6
A_P	67	75	75	43	76	60	77	86	96	88
A_T	4	6	12	40	38	36	28	76	96	88

The Figure 16 is a plot of A_P between DGA software and other researchers systems.

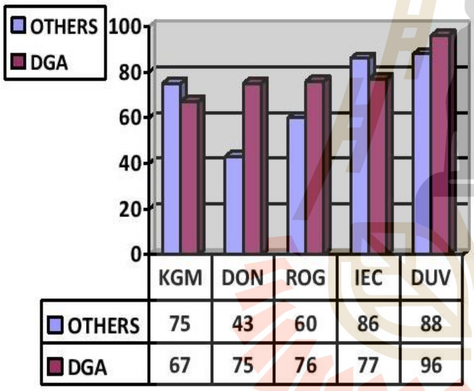


Fig. 16. Diagnosis of Duval Triangle Method in DGA software

V. Conclusion

The DGA software has successfully developed and database has connected to it. The accuracy comparison was done for DGA software and other researchers systems and found to be higher especially in Duval Triangle. The DGA software has also shown the great ability of diagnosing large number of DGA information from distribution network with many transformers with the presence of database at minimum time. The DGA data were saved only once and the diagnostic processes were able to be repeated several times by retrieving data from the database hence minimizing computational time, reduces inaccuracy in diagnosis, provides an easy and quick time-to-time condition monitoring and also it gives a safe storage of DGA data for future uses. The summary of diagnosis after interpretation and saved can be used by users in other times without rerunning again the software.

References

[1] P. Pao-la-or, A. Isaramongkolrak and T. Kulworawanichpong, "Finite Element Analysis of Magnetic Field Distribution for 500kV Power Transmission Systems," *IEEE Electr. Insulation Magn.*, vol. 21, pp. 21 – 27, 2005.

[2] A. S. Alghamdi, "DGA Interpretation of Oil Filled Transformer Condition Diagnosis", *Trans. Electrical and Electronic Materials*, vol. 13, pp. 229-232, Oct. 2012.

[3] A. Akbari, A. Setayeshmehr, H. Borsi and E. Gockenbach., "Software Implementation of the Duval Triangle Method", *in 2008 Proc. IEEE Int. Symp. Electrical Insulation*, Vancouver, pp: 124-127.

[4] O. Gouda, S. Saleh, and S. EL-Hoshy, "Power Transformer Incipient Faults Diagnosis Based on Dissolved Gas Analysis", *TELEKOMNIKA Indonesian J. Electr. Eng.*, vol. 17, no. 1, pp. 10-16 January, 2016.

[5] S. Singh and M.N Bandyopadhyay, "Duval Triangle: A Noble Technique for DGA in Power Transformers," *Int. J. Electr. Power Eng.*, vol. 4, no. 3, pp: 193 – 197, 2001.

[6] R.E. James and Q. Su, *Condition Assessment of High Voltage Insulation in Power System Equipment*, IET: UK, 2008.

[7] M. Lutz, *Powerful Object-Oriented Programming: Programming Python*, O'Really Media, Inc: USA, 2011, ch. 1.

[8] R. Dyer, *MYSQL in a Nutshell: A Desktop Quick Reference*, O'Really Media, Inc: USA, 2005, ch. 1.

[9] H. Sun, Y. Huang, and C. Huang, "A Review of Dissolved Gas Analysis in Power Transformers," *Int. Conf. Advances in Energy Eng, Energy Procedia* vol. 14, pp. 1220 – 1225, 2012.

[10] *IEEE Guide for the Interpretation of Gases Generated in Oil-Immersed Transformers*, IEEE Standard C57.104, 2008.

[11] R. Kumar, "Different DGA Techniques for Monitoring of Transformers," *Int. J. Electronics and Electrical Eng*, vol. 1, no. 4, December, pp. 299-303, 2013.

[12] S. Ghoneim and S. Ward, "Dissolved Gas Analysis as Diagnostic Tools for Early Detection of Transformer Faults," *Advances Electrical Eng. Syst*, vol. 1, no. 3, pp. 152-156, 2012.

[13] N. A. Muhamad, B. T. Phung, T. R. Blackburn, and K. X. Lai, "Comparative Study and Analysis of DGA Methods for Transformer Mineral Oil", *IEEE Power Tech.*, pp. 45-50, July, 2012.

Pius Victor was born in Dar es Salaam, Tanzania on 15 th October, 1986. He obtained his B.Eng. degree in Electrical Engineering from Dar es Salaam Institute of Technology, Tanzania in 2013. He is now a Master's Degree student in the School of Electrical Engineering, Suranaree University of Technology in Thailand. His interest research topics include High Voltage Systems Design and Monitoring, Laboratory and System Programming.

Training of Electrical Engineers for Transformer Condition Based Maintenance through Computer Based DGA

Pius Victor and Boonruang Marungsri
School of Electrical Engineering, Suranaree University of Technology, Thailand
111 University Avenue, Nakhon Ratchasima 30000, Tel. 093-643-8346, Email: piusvictor2013@gmail.com

Abstract

The vital role of power transformer in the power system has attracted the consideration of Condition based Maintenance for many years. Dissolved gas in-oil analysis (DGA) has gained popularity in condition monitoring and diagnosis through its diagnostic methods like Key Gas, Dornenburg, Rogers, IEC and Duval Triangle. As the importance of Condition based Maintenance increases, the need of advanced monitoring and diagnostic tools increases which can be accomplished only by skilled researchers. This necessitates training for young researchers (Electrical Engineers) but most of their existing environments are not user friendly for them to practice, apply and develop advanced tools from basic ones. This paper introduces computer based software with five diagnostic methods and database which enables evaluations of works done. It gives abilities to Young Electrical Engineers to practice, apply, evaluate, compare and advance basic diagnostic tools for condition monitoring of power transformers.

Keywords: Training, Electrical Engineers, Condition monitoring, Power transformer, DGA.

1 INTRODUCTION

Degradation of insulation in oil filled transformer is a normal process, it occurs even under normal load conditions but abnormal conditions accelerates the rate of degradation which affects electrical, chemical and physical properties of insulating oil [1]. The complex nature of fault in oil filled transformer has lead to advanced studies on detection, diagnose and deciding the mode of condition monitoring for transformers. The importances of transformer economically and technologically have raised an attention to researchers to further study and develop an affordable condition monitoring tool(s). Neither some of condition monitoring systems that have been developed by some utilities and being implemented to potential transformers nor ready-made DGA computer software are abundantly available for Training of young Electrical Engineers. Ready-made systems are very expensive, equipped in backbone and high potential transformers while some of available raw codes are not compatible to some of them. Utilities and research centers use highly skilled researchers while there is little effort on training young Electrical Engineers. An example can be taken to Thailand, a middle developing country, with 500kV (EHV single and double circuit), 230 kV, 115 kV, transmission line systems [2] and more than 100 Universities but there are only two DGA machines (one at Electricity Generating Authority of Thailand – EGAT and the other one is at Suranaree University of Technology – SUT handled-over on 2015). With reference to this introduction, the paper introduces a development of computer based software which is useful for training of Electrical Engineers. Five Condition assessment tools for power transformers through DGA (Key Gas, Dornenburg, Rogers, IEC and Duval) have been reviewed and implemented in software bases. Paper also opens the way forward to the access and use of database raw data for diagnosis; evaluating method(s), and saved work summary for future evaluation.

An easy graphical user interface was made by using Python programming language. According to [3], it is defined as a general purpose, open source computer programming language, optimized for software quality, developer productivity, program portability and component integration. It is further described to support a remarkably simple, readable and maintainable syntax; integration with external components coded in other languages; a scripting language as it makes it easy to utilize and direct other software components.

As in [4], MySQL is stated to be an open source, multi-threaded, relational database management system created by Michael "Monty" Widenius in 1995, and in 2000, MySQL was released under a dual license model that permitted the public to use it for free under the GNU Public License (GPL); this caused its popularity to soar. The success of MySQL as a leading database is stated due not only to its price-after all, other cost-free and open source databases are available-but also its reliability and performance. There are many features that contribute to MySQL's standing as a superb database system; speed and storage are one of its most prominent features which manages queries and interfaces between a user's SQL commands the database's backend storage.

In this paper, DGA software features and basics of condition monitoring of power transformer are detailed in Section 2; revision of Condition assessment tools for DGA and their implementation are discussed in Section 3, the results of accuracies of diagnostics methods are explained in Section 4. These results of diagnosis by different assessment method were compared with the results from other researchers. The diagnosis and evaluation of DGA works were according to IEEE and IEC standards.

2 FEATURES OF DGA SOFTWARE

A. Development Tools of DGA software

Python programming language was used to develop a DGA Software in Ubuntu 14.04 (operating system) while MySQL database programming language was used to create a database in a computer server and connected them together. Other details are shown in Table 1 below. The software works effectively in this operating system.

Table 1. Tools used for the development of DGA-Software

Tool	Type
Computer	HP 550 Duo Core
Operating System	Ubuntu 14.04
Programming Language	Python 2.7
Compiler	Eclipse
Database	MySQL 5.5
Connector	Mysql-python connector

Figure 1 shows the flow chart of DGA software in which user enters DGA information and can choose to diagnose by any of the five methods. In every diagnosis DGA software displays a report which shows diagnosing time, DGA data (gas details), some of computational method, graphical representation and analysis. It also saves the report in a plain text document which can easily be viewed by through DGA software or computer document viewer.

B. Graphical User Interface

The graphical user interface has three main parts. (i) User Input Form-for saving gas details (H_2, CO, CO_2, N_2, CH_4, C_2H_6, C_2H_4 and C_2H_2) and non-gas details (Power-kVA, Voltage-kV, Altitude-M, Humidity-%, Oil Temperature-C, Location-name, Sampling date-date) for further evaluation and decision making about power transformer. (ii) Toolbar with five diagnosing methods (KGM, Dornenburg, Rogers, IEC, Duval). (iii) Menu bar with details about DGA software and database. It has Help button with drop down menu for user to choose a category for getting enough information. After saving gas and non-gas data to the database, user can easily pick any data by data ID from the DGA software, and diagnose by using any diagnosing method. All diagnoses give gas details, execution, analysis and graphical representation which are easy to learn and understand for young Electrical Engineers.

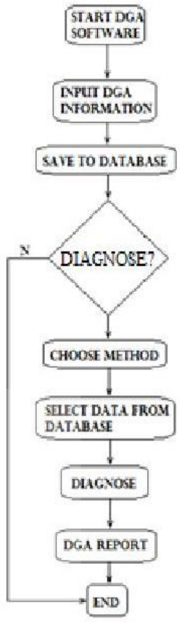


Figure 1. Algorithm of DGA Software

C. Database

For the purpose of saving DGA data, a MySQL database was created and connected to the software. A database has a table in it with eleven columns named ID (primary key); hydrogen, carbon_monoxide, carbon dioxide, nitrogen, methane, ethane, ethylene, acetylene, power, voltage, altitude, oil temperature, humidity, sampling_date, machine_location, machine_number. For reducing complexity user can only save and retrieve data from the database which is done through the use of DGA software. Every DGA information when is stored in the database, it automatically registered with a unique ID which provides an easiest way to find it by using DGA software for diagnosis.

D. Condition Classification for DGA Methods

Three IEEE methods (Key Gas Method, Dornenburg Ratio Method and Rogers Ratio Method) and IEC methods (IEC Ratio Method and Duval Triangle Method) were used to interpret DGA information and analyze the condition of power transformer.

E. Key Gas Method

Six principal gases which are used by this method to detect the presence of faults in the power transformer are H_2, CO, CH_4, C_2H_6, C_2H_4 and C_2H_2 whereby the significant and proportion of the gases are called "key gases". The fault is identified according to the presence and percentage of each key gas [5], [6]. Table 2 below shows the interpretation of different faults by key gas method.

Table 2. Combustible gases in Key Gas analysis [5]

Gas	Normal	Abnormal	Interpretation
H_2	< 150 ppm	> 1,000 ppm	Arcing, Corona
CH_4	< 25	> 80	Sparking
C_2H_6	< 10	> 35	Local Overheating
C_2H_4	< 20	> 100	Severe Overheating
CO	< 500	> 1,000	Severe Overheating
CO_2	< 1,000	> 15,000	Severe Overheating
N_2	1 – 10%	NA	NA
O_2	0.2 – 3.5%	NA > 0.5%	Combustibles

F. Dornenburg Ratio Method

As in [6] this method first examines individual gas, and recommends a test if any gas is beyond the set up limit L_1 and if any two gases are twice beyond limits L_1 in parts per million as shown in Table 3. Three faults can be detected by this method while five gases and four ratios are used for detecting faults as shown in Table 4. Dornenburg interpretation is described in Table 5.

Table 3.Dornenburg Limits for individual gas, L_1 [6], [7]

Gas	Limit, ppm
H_2	100
CH_4	120
C_2H_4	50
C_2H_2	35
C_2H_6	1
CO	350

Table 4.Domenburg Gas Ratios [6], [7], [8]

Ratio	Gas
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

Table 5.Domenburg fault interpretation [6], [7]

Diagnosis	CH_4 / H_2	C_2H_2/ C_2H_4	C_2H_2/ CH_4	C_2H_6/ C_2H_2
Therm. Decom	> 1.0	< 0.75	< 0.3	> 0.4
Corona	< 0.1	-	< 0.3	> 0.4
Arcing	0.15 – 1.0	> 0.75	> 0.3	< 0.4

G. Rogers Ratio Method

This method uses three ratios to analyze six conditions of the power transformer [5], [6], [7], [8]. Two tables are used in Roger's method: one defined the code of the ratio, and the other defined the diagnosis rule as shown in Table 6 and Table 7 respectively [8].

Table 6.Roger's Gas Ratios [7]

Ratio	Gas
Ratio 1	CH_4 / H_2
Ratio 2	C_2H_2 / C_2H_4
Ratio 5	C_2H_4 / C_2H_6

Table 7.Roger's fault interpretation [6], [7]

Case	Fault diagnosis	C_2H_2/ C_2H_4	CH_4 / H_2	C_2H_4 / C_2H_6
0	Unit Normal	< 0.1	>0.1 to <1.0	<1.0
1	Low energy density arcing	< 0.1	< 0.1	<1.0
2	Arcing-High energy discharge	0.1 – 3.0	0.1 to 1.0	>3.0
3	Low temperature-thermal	< 0.1	>0.1 to <1.0	1.0 to 3.0
4	Thermal < 700 °C	< 0.1	>1.0	1.0 to 3.0
5	Thermal > 700 °C	< 0.1	>1.0	>3.0

H. IEC Ratio Method

Five gases, four gas ratios and seven fault types are involved in this method as described in Table 8 and Table 9 below.

TABLE 8.IEC Gas Ratios [7]

Ratio	Gas
Ratio 1	CH_4 / H_2
Ratio 2	C_2H_2 / C_2H_4
Ratio 5	C_2H_4 / C_2H_6

TABLE 9.IEC Fault Interpretation [7]

Case	Fault diagnosis	C_2H_2/ C_2H_4	CH_4 / H_2	C_2H_4 / C_2H_6
PD	Partial Discharge	-	< 0.1	< 0.2
D1	Low Energy Discharges	>1.0	0.1 to 0.5	>1.0
D2	High Energy Discharge	0.6 – 2.5	0.1 to 1.0	>2.0
T1	Thermal Fault <300 °C	-	>1.0	<1.0
T2	Thermal 300 - 700 °C	< 0.1	>1.0	1.0 to 4.0
T3	Thermal > 700 °C	< 0.2	>1.0	>4.0

I. Duval Triangle Method

This method uses three hydrocarbon gases namely CH_4, C_2H_4 and C_2H_2. It employs proportions of these three gases to identify the fault in a triangular map to detect seven fault types corresponding to the increasing in energy levels of gas formation in transformers in service [8], [9]. The triangle with fault zones is shown in Figure 2.

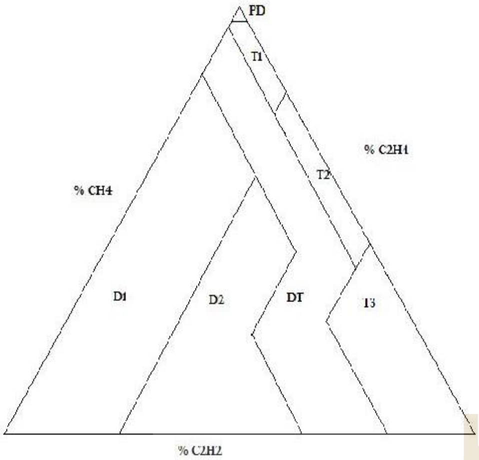


Figure 2.Duval Triangle

The fault zone is identified by an intersection point between parallel lines obtained from the percentages of fault gases as shown in equation (1), (2) and (3).

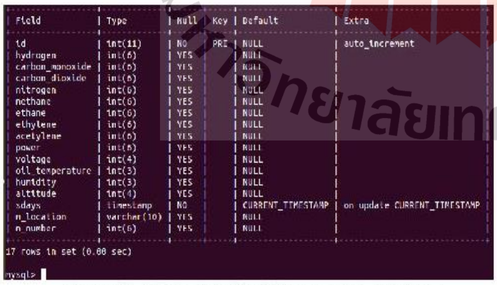
$$\%CH_4 = 100xCH_4 / (CH_4 + C_2H_2 + C_2H_4) \quad (1)$$

$$\%C_2H_4 = 100xC_2H_4 / (CH_4 + C_2H_2 + C_2H_4) \quad (2)$$

$$\%C_2H_2 = 100xC_2H_2 / (CH_4 + C_2H_2 + C_2H_4) \quad (3)$$

3 DEVELOPED DGA SOFTWARE

The created database has two tables as shown in Figure 3. For our concern the table named 'dga' will be used. This table stores all information in their respective fields. It has 17 fields for storing DGA information including data ID, hydrogen, carbon monoxide, carbon dioxide, nitrogen, methane, ethane, ethylene, acetylene, power, voltage, oil temperature, humidity, altitude, sampling date, machine location and machine number.



Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
hydrogen	int(6)	YES		NULL	
carbon_monoxide	int(6)	YES		NULL	
carbon_dioxide	int(6)	YES		NULL	
nitrogen	int(6)	YES		NULL	
methane	int(6)	YES		NULL	
ethane	int(6)	YES		NULL	
ethylene	int(6)	YES		NULL	
acetylene	int(6)	YES		NULL	
power	int(6)	YES		NULL	
voltage	int(4)	YES		NULL	
oil_temperature	int(3)	YES		NULL	
humidity	int(3)	YES		NULL	
altitude	int(4)	YES		NULL	
sdays	timestamp	NO		CURRENT_TIMESTAMP	on update CURRENT_TIMESTAMP
m_location	varchar(10)	YES		NULL	
m_number	int(6)	YES		NULL	

17 rows in set (0.00 sec)

mysql>

Figure 3. Table details of the created database

The graphical user interface of the developed computer program in Figure 4 shows five diagnostic methods for interpreting DGA information and two condition monitoring tools (comparison – for two gases and trending – for more than two gases) in the toolbar. They are clearly visible for beginners to see and select. It requests user to input gas and non-gas details to save in the database.

Figure 4. The interface of DGA Software

After saving DGA information in the database user can opt to retrieve single or multiple DGA information from the database. The developed user interface of the Key Gas diagnostic method in the DGA software is shown in Figure 5. It includes user request of DGA data from the database, retrieved DGA information (gas values in ppm), computations (TDCG and gas proportions over TDCG $\%H_2$, $\%CO$, $\%CH_4$, $\%C_2H_6$, $\%C_2H_4$ and $\%C_2H_2$), graphical representation of gas proportions, result analysis (fault analysis) and computational time (method opening time, diagnostic time and current date).

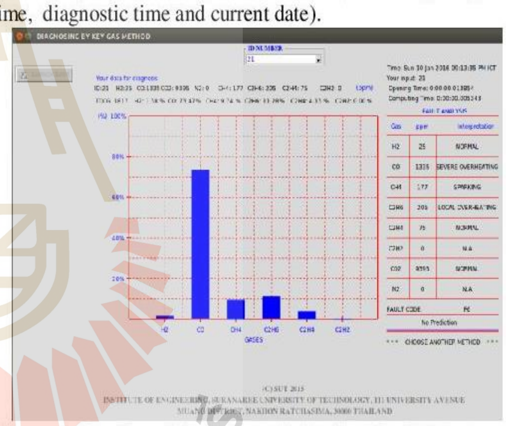


Figure 5. Interface of fault diagnosis by Key Gas Method

The developed user interface of the Dornenburg Ratio method in the DGA software shown in Figure 6 includes user request of DGA data from the database, retrieved DGA information (gases values in ppm), computations (gas comparisons with Normal Limits L_1 – if any gas is above L_1 - computations of gas ratios R_1, R_2, R_3 and R_4), result analysis (fault analysis) and computational time (method opening time, diagnostic time and current date).

For Roger's Ratio method the interface is shown in Figure 7 which includes user request of DGA data from the database, retrieved DGA information (gases values in ppm), computations (computations of gas ratios R_1, R_2 and R_3), result analysis (fault analysis) and computational time (method opening time, diagnostic time and current date).

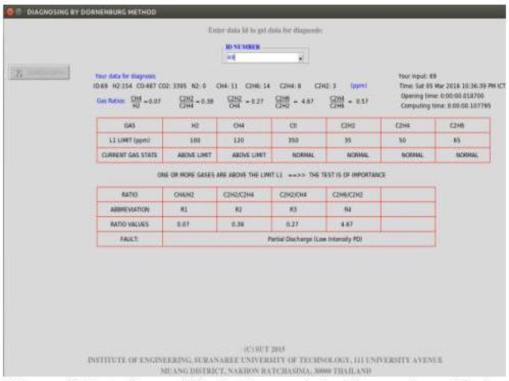


Figure 6. Interface of fault diagnosis by Dornenburg Ratio Method

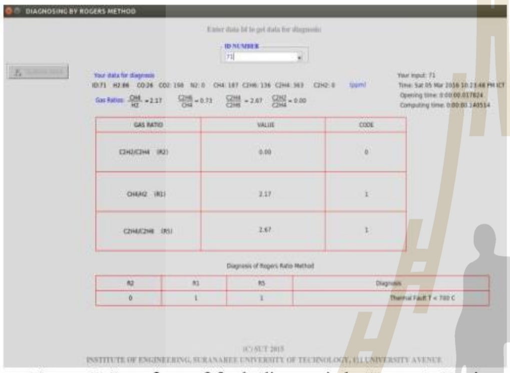


Figure 7. Interface of fault diagnosis by Roger's Ratio Method

The interface of IEC Ratio method is shown in Figure 8 which includes user request of DGA data from the database, retrieved DGA information (gases values in ppm), computations (computations of gas ratios R_1, R_2 and R_5), result analysis (fault analysis) and computational time (method opening time, diagnostic time and current date).

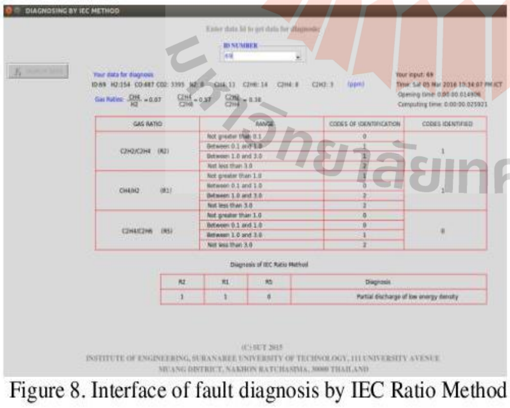


Figure 8. Interface of fault diagnosis by IEC Ratio Method

The developed user interface of the Duval Triangle method is shown in Figure 9. It includes user request of DGA data from the database, retrieved DGA information (gas values in ppm), computations ($CH_4 + C_2H_4 + C_2H_2$ and gas proportions %CH_4, %C_2H_4 and %C_2H_2), graphical representation of gas proportions, result analysis (fault analysis) and computational time (method opening time, diagnostic time and current date).

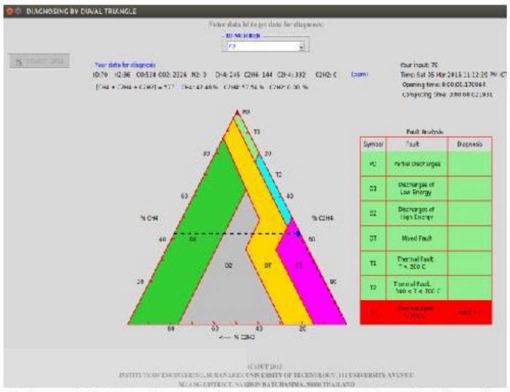


Figure 11. Interface of fault diagnosis by Duval Triangle Method

At the end of every diagnosis DGA software saves the summary of diagnosis in plain text that can also be viewed by DGA software. Summaries of different diagnoses can be combined in one file and viewed by DGA software as shown in Figure 12 and saved in separate files and rename by its diagnostic method's name and time as shown in Figure 13.

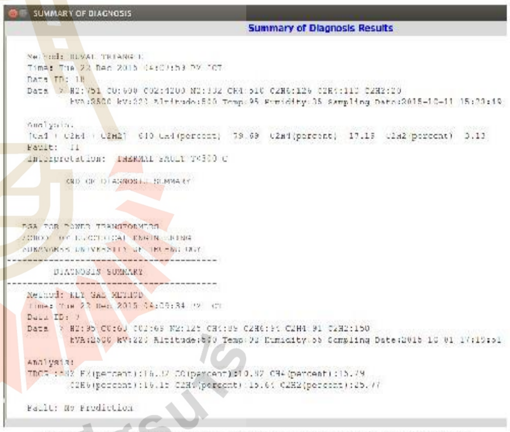


Figure 12. Summary of diagnosis saved in one file

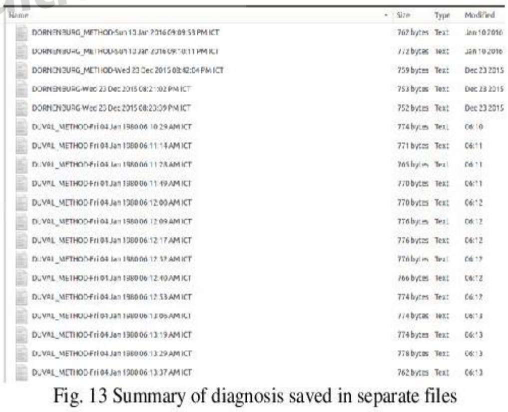


Fig. 13 Summary of diagnosis saved in separate files

4 RESULTS

Software Validation

DGA data from 50 researchers [8], [9] sampled from Oil Filled Transformers with different conditions were taken for testing the DGA software. The DGA data were first stored in the database of DGA software then retrieved and diagnosed by each diagnostic method. After diagnosis the results of predictions were used to find percentage of Prediction, P and Consistency, C. As in [10] equation (4) and (5) were used to compute P and C shown in Table 11.

$$\%Success\,Pred. = \frac{100xNumberof\,Predictions}{NumberofCases} \quad (4)$$

$$\%Consistency = \frac{100xCorrect\,Predictions}{Numberof\,Predictions} \quad (5)$$

Table 11.Summary of Five Methods from DGA Software

Method	Number of Predictions	Successful Prediction, P, %	Correct Predictions	Cons, C, %
KeyGas	3	6	2	67
Dornen	8	16	6	75
Rogers	38	76	29	76
IEC	18	36	14	77
Duval	50	100	48	96

Also equation (6) and (7) from [10] were used to compute Accuracy based on predicted cases (A_P) and based on the number of cases (A_T). The results of accuracies Tp and Tc were compared with results from other researchers as shown in Table 12 (where Tc-total cases, T_{NP}-no predictions, Tp-number of predictions, T_R-correct predictions, T_W-incorrect predictions, D-DGA software and O- other researchers).

$$Accuracybasedon\,Prediction = \frac{100xT_R}{T_P} \quad (6)$$

$$AccuracybasedonTotalCases = \frac{100xT_R}{T_C} \quad (7)$$

Table 12.Comparison of accuracies of DGA and other Researchers

	KGM		Dorn		Rog		IEC		Duval	
	D	O	D	O	D	O	D	O	D	O
T_C	50	50	50	50	50	50	50	50	50	50
T_{NP}	47	46	42	4	12	20	32	6	0	0
T_P	3	4	8	46	38	30	18	44	50	50
T_R	2	3	6	20	29	18	14	38	48	44
T_W	1	1	2	14	11	12	4	6	2	6
A_P	67	75	75	43	76	60	77	86	96	88
A_T	4	6	12	40	38	36	28	76	96	88

5 CONCLUSIONS

DGA software and database were developed successfully and connected to each other. DGA information from 50 researchers were stored in the database, retrieved and diagnosed by using five DGA methods. DGA software showed the accuracy (A_P and A_T) of 96% with the use of Duval Triangle. Also it displayed important features of every diagnostic method in a simple manner for young Electrical Engineers to learn, apply and enable them to develop improved ones. With the use of database the tedious works of inputting DGA data has been reduced, enabling them to diagnose by retrieving stored data and hence reduce diagnosing time and unnecessary errors.

References

[1] A. Saeed, N. Muhamad and A. Suleiman, " DGA Interpretation of Oil Filled Transformer Condition Diagnosis," Trans. Electrical and Electronics Materials, vol. 13, pp. 229 - 232, 2012.

[2] P. Pao-la-or, A. Isaramongkolrak and T. Kulworawanichpong, "Finite Element Analysis of Magnetic Field Distribution for 500kV Power Transmission Systems," IEEE Electrical Insulation Magazine, vol. 21, pp. 21 – 27, 2005.

[3] M. Lutz, 4Ed, "Powerful Object-Oriented Programming: Programming Python," O'Really Media, Inc; USA, 2011.

[4] R. Dyer, "MYSQL in a Nutshell: A Desktop Quick Reference," O'Really Media, Inc; USA, 2005.

[5] H. Sun, Y. Huang, and C. Huang, "A Review of Dissolved Gas Analysis in Power Transformers," Intl. Conf. Advances in Energy Engineering, Energy Procedia 14, pp. 1220 – 1225, 2012.

[6] IEEE-Std-C57.104-2008, "IEEE Guide for the Interpretation of Gases Generated in Oil-Immersed Transformers," IEEE Power & Energy Society, 2008.

[7] R. Kumar, " Different DGA Techniques for Monitoring of Transformers," Journal, Electronics and Electrical Engineering, Vol. 1, 2013.

[8] S. Ghoneim and S. Ward, " Dissolved Gas Analysis as a Diagnostic Tools for Early Detection of Transformer Faults," Advances in Electrical Engineering Systems, Vol. 1, pp. 152-156, 2012.

[9] O. Gouda, S. Saleh and S. EL-Hoshy,, " Power Transformer Incipient Faults Diagnosis Based on Dissolved Gas Analysis," Journal, vol. 17, pp. 10-16, 2016.

[10] N. A. Muhamad, B. T. Phung, T. R. Blackburn, and K. X. Lai, " Cooperative Study and Analysis of DGA Methods for Transformer Mineral Oil," IEEE Power Tech., pp. 445-50, 2012.

Praise Worthy Prize

International Review of Electrical Engineering (I.R.E.E.), Vol. 11, N. 6
ISSN 1827- 6660 *November – December 2016*

Effects of Distribution Approximations to Expanded Uncertainty of Breakdown Strength of Mineral Insulating Oil in IEC 60156

Pius V. Chombo, Thanatchai Kulworawanichpong, Boonruang Marungsri

Abstract – *Breakdown Strength test identifies the health condition of oil-immersed power transformers. The erratic and non-homogeneity behavior affects much the results of breakdown strength test. The inclusion of expanded uncertainty in test results of breakdown strength bounds the compensated test results in a defined confidence level, which is not defined in IEC 60156. This paper presents the analysis of the effects of distribution approximations used in the estimation of expanded uncertainty.*
The study involved Normal distribution (Gaussian), t-distribution (student distribution), Rectangular distribution and Triangular distribution in estimating an expanded uncertainty which incorporated Type A and Type B errors. The Expanded Uncertainty was based on a standard uncertainty multiplied by a coverage factor of k = 2, which provided a minimum confidence level of 95% in all approximated distributions. Normal distribution (Gaussian) showed the validity in the estimation of expanded uncertainty for a sufficiently large number of test results (of up to 25 tests per sample) while Student-t distribution showed invalidity for all test results. Since it is not a good practice to test insulating oil up to 25 tests, expanded uncertainty based on Normal distribution seemed to be valid for any test results less than 25. The inclusion of expanded uncertainty may predict the validity and avoid re-running of the test.

Keywords*: Breakdown Strength, Distribution Approximations, Expanded Uncertainty, IEC 60156*

Nomenclature

u_x	Expanded Uncertainty
U_x	Combined Uncertainty
k_p	Coverage Factor
p	Confidence Level
t	Student –*t* Distribution
t_p	Coverage Probability in *t*
s_x	Random Errors
b_k	Systematic Errors
v	Degree of Freedom in *t*
T	*T* - Correction Factor
a	Half Range
n	Number of Tests per sample
N	Number of samples
R	Range of test results
$\bar{X}$	Mean
$\%\frac{R}{\bar{X}}$	Percentage of Range / Mean
σ	Standard Deviation

I. Introduction

Transformer is the key element in the power system whereby the health of its insulation plays a significant role in securing its economic and technological performance. In-service transformers may found to exhibit unpredictable behaviors when they interact with other system elements which sometimes overstress insulation system of respective transformers.

An example of these behaviors is the sympathetic inrush current when a new power transformer is paralleled with the other ones [1]. Some of the techniques for monitoring and assessing the health of transformer have been discussed in [2]-[5]. Breakdown strength test is among the tests used to identify the health of an oil immersed power transformer under the guidance of IEC 60156. This standard guides the performance of breakdown strength test of mineral insulating oil mainly used in oil-filled power transformers.

An erratic, non-homogeneity behavior and uncontrollable weather conditions are primary causes of variations and hence drastically change in the breakdown values in mineral insulating oil. Currently, some studies have shown the importance of correcting and compensating measurements for minimizing the effects of variations [1]-[5]. It is clearly known that measurement errors are inevitable [6]. Despite of weather fluctuation, errors may exist due to the presence of moisture [7] and contaminating particles generated during refinery process, transportation, installation and in-service operation.

Measurement errors can be categorized as random errors and systematic errors. Random errors are due to repetition in measurement while systematic errors are from other factors including accuracy of the measuring

DOI: 10.15866/iree.v11i6.10434

instrument, weather fluctuation, etc.

Errors can cause measurement results to deviate far beyond the required values. Since deviated result may either be very small or large value, they are all termed as outliers. Chauvenet's criterion has been mostly accepted worldwide in removing outliers from the measurement results. It is distinguished from some methods like Thompson's τ method etc. due to its single shot performance. It sets a boundary and removes all outliers at once that are beyond the boundary. Thompson's τ method also removes outliers from the measurement results but it uses a multi-shot algorithm.

That means it sets a boundary and removes only one outlier at a time and repeats until all outliers are finished. It takes long time to get a required set of measurement results with no outliers. Therefore, this makes Chauvenet's criterion mostly being used among other methods. Many works have been undertaken in diminishing the effects of errors introduced during measurement. Predominant methods studied in previous suggested the following for minimizing measurement errors (i) increasing number of measurements (tests per sample) and (ii) inclusion of expanded uncertainty.

Bitch [6] discussed the reasons of estimating uncertainty as to allow the meaningful comparison of test results against reference values available in specifications or standards, and to avoid unnecessary repetition of tests for non-significant differences. The inclusion of expanded uncertainty in measured results encloses the large fraction of the distribution of values that could significantly be associated to the measure and [6], [8]-[11]. Usually, the value of expanded uncertainty is influenced by the type of distribution and coverage factor used in estimation. Coverage factors, k = 1, 2, 3 with confidence levels of 90%, 95%, and 99% respectively have some impacts in the estimation of expanded uncertainty. Unless stated otherwise or not stated, k = 2 which corresponds to 95% confidence interval can be used as a coverage factor, but k = 3 has been mostly applied in critical measurements e.g. health tests [6].

In breakdown strength test performed by IEC 60156, expanded uncertainty is not considered which can play a great role to encompass inevitable measurement errors.

In this work, expanded uncertainty of breakdown strength test based on four distributions is estimated.

The effects of Normal distribution (Gaussian), t-distribution (student distribution), Rectangular and Triangular distributions are presented. The expanded uncertainty has based on a standard uncertainty multiplied by a coverage factor of k = 2, which provided a minimum confidence level of 95% in all approximated distributions. Test results used in the estimation of expanded uncertainty are free from outliers whereby outliers were taken care by Chauvenet's criterion.

II. Methodology

In this work, the standard deviation defined in a breakdown strength test is replaced by an estimated expanded uncertainty.

The main reason is due to that the expanded uncertainty combines some compensations and corrections of errors. Since random errors are due to series of repeated tests, they were reduced by increasing the number of test results not as specified in IEC 60156 (6 test results). Also, systematic errors were reduced by applying a correction factor. Therefore, based on each distribution approximation, the estimation of expanded uncertainty was made.

Test results used in estimation involved no outliers.

II.1. Breakdown Value of Mineral Insulating Oil

The breakdown value of the mineral insulating oil is typically expressed as shown in (1) [11] where X_{best} is assumed to be mean of the test result with no outliers and an expanded uncertainty is represented by u_x after replacing the standard deviation.

Hence, an expanded uncertainty can be obtained at 95% confidence interval with k = 2 and is expressed n terms of %:

$$X_{best} \pm u_x \quad (1)$$

As discussed in the literature, the value of expanded uncertainty u_x depends on the distribution to be used.

Therefore, estimation of these four distributions is described in the next sub-sections.

II.2. Student -t Distribution

This probability distribution was invented by William Sealy Gosset on 1908 for estimating the mean of a normally distributed population in the situations where the sample size is small and population standard deviation is unknown.

The expanded uncertainty in t-distribution is given in (2) [8] where t_p is the coverage probability or level of confidence of the interval, U_x is the combined uncertainty taken as a sum of uncertainties' squares (random and systematic) as shown in Eq. (3).

The random error s_x incorporated an experimental standard deviation as shown in (4) [8], [11] while systematic error b_k covered an instrument accuracy (suggested 1%) and is given in (5):

$$u_x = t_p \times (v) \times U_x \quad (2)$$

In (1) the degree of freedom for n-observations of one sample was represented by v and is given in (6):

$$U_x = \sqrt{s_x^2 + b_k^2} \quad (3)$$

$$s_x = \frac{\sigma}{\sqrt{n}} \quad (4)$$

$$b_k = \frac{T \times Instrument\ Accuracy}{k_p} \quad (5)$$

$$v = n - 1 \tag{6}$$

The values of coverage probabilities t_p corresponding to the degree of freedom v are shown in Table I [12].

From Table I, various degrees of freedom, v are given in the first column while the rest of columns are coverage probabilities, t_p in different confidence levels, p.

Six groups of t_p relating with six confidence levels i.e. 68.27%, 90%, 95%, 95.45%, 99% and 99.73% are shown in correspondence of their degrees of freedom. For this case, t_p in 95% was used in estimation with the degree of freedom obtained from (6). As shown in Table II, the given correction factors T correspond to a number of measurements n, and confidence level required [12].

For 95% confidence level k_p=2 and a given number of tests per sample, the value of T to be used in Eq. (5) can be obtained.

II.3. Normal Distribution

The expanded uncertainty u_x, in Normal or Gaussian distribution is given in (7) [12] where k_p is the coverage factor to provide an interval having a level of confidence p close to a specified level.

Combined uncertainty U_x shown in (3) was comprised of random error s_x and systematic error b_k given in (4) and (5) respectively.

As shown in Table III, the values of k_p corresponding to the level of confidence were given for (7) while correction factors T are similar to those shown in Table II. For 95% confidence level k_p= 1.96 was used:

$$u_x = k_p \times U_x \tag{7}$$

II.4. Rectangular Distribution

In Rectangular distribution, an expanded uncertainty u_x can be given similar to (7) where k_p is the coverage factor in 95% confidence level as shown in Table III.

Combined uncertainty U_x was expressed as shown in (3). The standard uncertainty s_x was given in (8) [6], [9] where a represented a half-range equally likely between upper and lower value. The systematic error b_k, in (9) involved no correction:

$$s_x = \frac{a}{\sqrt{3}} \tag{8}$$

$$b_k = \frac{Instrument\ Accuracy}{\sqrt{3}} \tag{9}$$

II.5. Triangular Distribution

In a Triangular distribution, an expanded uncertainty u_x was the same as given in (7) except the standard uncertainty s_x. It was expressed as shown in (10) [6][13] where a is a half-range between upper and lower value.

The systematic error b_k, in (11) involved no correction:

$$s_x = \frac{a}{\sqrt{6}} \tag{10}$$

$$b_k = \frac{Instrument\ Accuracy}{\sqrt{3}} \tag{11}$$

TABLE I
VALUES OF COVERAGE PROBABILITY CORRESPONDING TO THE DEGREE OF FREEDOM [12]

v	68.27	90	95	95.45	99	99.73
1	1.84	6.31	12.71	13.97	63.66	235.80
2	1.32	2.92	4.30	4.53	9.92	19.21
3	1.20	2.35	3.18	3.31	5.84	9.22
4	1.14	2.13	2.78	2.87	4.60	6.62
5	1.11	2.02	2.57	2.65	4.03	5.51
6	1.09	1.94	2.45	2.52	3.71	4.90
7	1.08	1.89	2.36	2.43	3.50	4.53
8	1.07	1.86	2.31	2.37	3.36	4.28
9	1.06	1.83	2.26	2.32	3.25	4.09
10	1.05	1.81	2.23	2.28	3.17	3.96
15	1.03	1.75	2.13	2.18	2.95	3.59
16	1.03	1.75	2.12	2.17	2.92	3.54
17	1.03	1.74	2.11	2.16	2.90	3.51
18	1.03	1.73	2.10	2.15	2.88	3.48
19	1.03	1.73	2.09	2.14	2.86	3.45
20	1.03	1.72	2.09	2.13	2.85	3.42
25	1.02	1.71	2.06	2.11	2.79	3.33
30	1.02	1.70	2.04	2.09	2.75	3.27
35	1.01	1.70	2.03	2.07	2.72	3.23
40	1.01	1.68	2.02	2.06	2.70	3.20
45	1.01	1.68	2.01	2.06	2.69	3.18
50	1.01	1.68	2.01	2.05	2.68	3.16
100	1.005	1.660	1.984	2.025	2.626	3.077
∞	1.000	1.645	1.960	2.000	2.576	3.000

TABLE II
CORRECTION FACTORS, T [6]

n	$k_p = 1$	$k_p = 2$	$k_p = 3$
3	1.32	2.27	-
4	1.20	1.66	3.07
5	1.14	1.44	2.21
6	1.11	1.33	1.84
7	1.09	1.26	1.63
8	1.08	1.22	1.51
9	1.07	1.19	1.43
10	1.06	1.16	1.36
11	1.05	1.14	1.32
12	1.05	1.13	1.28
13	1.04	1.12	1.25
14	1.04	1.11	1.23
15	1.04	1.10	1.21
16	1.03	1.09	1.20
17	1.03	1.08	1.17
18	1.03	1.08	1.16
19	1.03	1.08	1.15
20	1.03	1.07	1.15

TABLE III
COVERAGE FACTORS [12]

Level of confidence, p	Coverage factor, k_p
68.27	1.000
90.00	1.645
95.00	1.960
95.45	2.000
99.00	2.576
99.73	3.000

II.6. Breakdown Strength Results

In order to study the effect of these four distribution approximations to the estimation of expanded uncertainty, test results should be available. To perform estimation, a large number of test results from laboratory tests must be available which is expensive. Due to unavailability of required test results, 11 sets with 3, 4, 5, 6, 7, 8, 9, 10, 15, 20 and 25 test results were generated in MATLAB software. In each set, results with the percentage of range to mean and standard deviation of 92% and 4% respectively were selected as per IEC 60156. Also, sets with mean values greater than or equal to 30 kV were chosen for best results.

The set with 6 test results that obey IEC 60156 was set in between so as to compare its performance with other sets of test results. Each set was run 15 times before performing analysis of expanded uncertainty.

II.7. Outliers

Since errors may occur due to repetition or randomness, outliers can exist within the sets. These are highest or lowest test results which can cause deviations beyond expected values. Therefore, outliers must be removed in the sets of the test result.

For removing outliers, Chauvenet's criterion was employed whereby for any data with a probability of obtaining its deviation from the mean value less than 1 - (1/2n) was considered for rejection[11].

II.8. Chauvenet's Criterion

This approach involves computation of $\bar{X}$, σ, q_i and q_n for checking each test result from its set before rejection as shown in (12) – (15) [11]:

Mean of test results, $\bar{X}$:

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{12}$$

Standard deviation of test results, σ:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2}{n-1}} \tag{13}$$

Chauvenet's parameter, q_i :

$$q_i = \frac{|X_i - \bar{X}|}{\sigma} \tag{14}$$

Chauvenet's group parameter, q_n:

$$q_n = F^{-1}(n) = F^{-1}\left\{\frac{2}{\sqrt{2\pi}}\int_0^t e^{-\frac{t}{2}}dt = 1 - \frac{1}{2n}\right\} = t \tag{15}$$

$\bar{X}$ and σ were calculated from the set of randomly generated numbers. Each test result X_i from the generated set was used to obtain the corresponding q_i as shown in (14). The values of q_n calculated for a different number of test results, n are given in Table IV [8].

Therefore, any X_i with q_i greater than q_n was considered as an outlier and was rejected from the set of test results. For the simplified outlying process, lower and upper boundaries were set from (14). The limit of which any test result X_i below or above it was considered as an outlier was defined in (16) and (17), respectively.

They are denoted as $X_{i,min}$ and $X_{i,max}$. That means, any test result, X_i with value less than $X_{i,\ min}$ or greater than $X_{i,\ max}$ was considered as an outlier and was rejected.

For the remained test results $\bar{X}$ and σ were recalculated for computation of expanded uncertainty:

$$X_{i,min} = \bar{X} - (\sigma \times q_n) \tag{16}$$

$$X_{i,max} = \bar{X} - (\sigma \times q_n) \tag{17}$$

TABLE IV
ACCEPTABLE DEVIATIONS [8]

Number of readings, n	Ratio of maximum acceptable deviation to Standard Deviation, qn
3	1.38
4	1.54
5	1.65
6	1.73
7	1.80
8	1.87
9	1.91
10	1.96
15	2.13
20	2.24
25	2.33
50	2.57
100	2.81
300	3.14
500	3.29
1000	3.48

III. Algorithm for Computation of the Expanded Uncertainty

Calculation of expanded uncertainty involved the use of valid test results (without outliers) and was computed in terms of four distribution approximations as shown below:

1. Generate 20 test results ranged:
   r = (50-35)×*rand* (1, 20) + 35
2. with test results:
   Compute R, $\bar{X}$, *% R/*$\bar{X}$ and σ
   Perform as per IEC 60156:
   Compute R, $\bar{X}$, *% R/*$\bar{X}$ and σ
3. Remove outliers from test results:
   Compute q_i and q_n
   if qi>qn:
   reject X_i
   with remained test results:
   Compute R, $\bar{X}$, *% R/*$\bar{X}$ and σ
4. Compute expanded uncertainty:
   for t – distribution:
   Compute v, b_k, s_x, U_x and u_x
   for normal distribution:

Compute b_k,s_x, U_x and u_x
for rectangular distribution:
Compute b_k, s_x, U_x and u_x
for triangular distribution:
Compute b_k, s_x, U_x and u_x

5. Plot results:
   with results *in* stage 2:
   plot X_i, $\bar{X}$ in subplot 1
   with results *in* stage 3:
   plot X_i, $\bar{X}$ in subplot 2
   with results *in* stage 4:
   plot u_x in subplot 3-4
   with results *in* (3 - 4):
   summarize in subplot 5-6
6. Finish computation
7. Generate another set of numbers?:
   with YES:
   repeat (1)
   with NO:
   END PROCESS

IV. Results

The results of valid $\bar{X}$,%$R/\bar{X}$ and σ have been tabulated in Table V. As seen in Table V, 15 results in each set of randomly generated number shave been given. From Table V, the last column gives an average of $\bar{X}$, %$R/\bar{X}$ and σ of obtained 15 results. The validity of results were compared with %$R/\bar{X}$ and σ in IEC 60156 i.e. ≤92% for %$R/\bar{X}$and ≤4% for σ. At last, valid test results from Table V were used in the estimation of expanded uncertainty.

With the valid results from Table V, expanded uncertainties based on four distribution approximations were estimated and presented in Table VI. As shown in Table VI, Normal, Student, Rectangular and Triangular represented Normal distribution, Student-t distribution, Rectangular distribution and Triangular distribution respectively. Also, results were given for each set of test results. Since each distribution approximation was estimated from 15 randomly generated sets, an average for each distribution approximation was shown in the last column of Table VI.

The computed values of $\bar{X}$, %$R/\bar{X}$, σ and expanded uncertainties based on four distributions were summarized in Fig. 1(a) for n = 3, 4, 5 and 6, Fig. 1(b) for n = 7, 8, 9, 10, and Fig. 1(c) for n= 15,20 and 25. All curves plotted in Figs. 1(a), (b) and (c) were defined in the fourth subplot of Fig. 1(c).

V. Discussion of Results

Since expanded uncertainty replaced standard deviation, the observations were limited to 4% as was accepted by IEC 60156 for standard deviation.

V.1. Normal Distribution

As depicted in Fig. 1(a) expanded uncertainty based on Normal distribution (pink curve) is higher than the standard deviation (blue curve) for n = 3 and n = 4.

For n = 5, expanded uncertainty is nearly equal to standard deviation and lower than standard deviation with n > 5. Out of Fig. 1(a), it can be observed that expanded uncertainty decreases below standard deviation when a sample is tested more than five times.

Furthermore, as illustrated in the last column of Table VI, the highest value of expanded uncertainty with Normal distribution was less than 4% (which is accepted by IEC 60156) and decreased as the number of tests per sample increased. Therefore, this indicates that expanded uncertainty cannot exceed 4% for the sufficiently large number of tests beyond n=6.

V.2. Student-t Distribution

Expanded uncertainty based on Student-*t* distribution was seen to be higher than the acceptable value of standard deviation (4%) for all test results, n (see Table VI). The smallest expanded uncertainty (12.07%) was obtained at n = 3 and was increasing with the number of test results. It was further observed that expanded uncertainty with Student-*t* distribution could not fall below 4% for any number of test results n.

Therefore, from this reason, Student-*t* distribution may not be suitable for estimation of expanded uncertainty of breakdown strength test.

V.3. Rectangular Distribution

As demonstrated in the last column of Table VI, expanded uncertainty based on Rectangular distribution showed acceptable values only for test results, n = 3, and n = 4. For test results n > 4, the value of expanded uncertainty was increasing beyond an acceptable value.

From this observation, it was seen that Rectangular distribution is valid only when estimating expanded uncertainty for test results $n \leq 4$.

V.4. Triangular Distribution

Unlike other approximations, this distribution showed acceptable values of expanded uncertainty for test results, $n \leq 8$. For test results beyond n = 8, the obtained expanded uncertainty was above the limit.

Therefore, Triangular distribution showed validity for test results $n \leq 8$.

V.5. Standard Deviation

From the last column of Table V, it can be seen that the average value of standard deviation was almost increasing as the number of test results increased.

At the maximum number of test results (n = 25), the value of standard deviation was 3.67% whereby the allowable deviation is 4%. The value of standard deviation is most likely to exceed 4% when the number of test results is increased. This value shows that standard deviation can be invalid when a certain number of test results is reached. Therefore, the behavior of expanded uncertainty with Normal distribution seemed to be better compared to standard deviation as it decreased with increased number of test results.

V.6. At n=6 (based on IEC 60156)

As indicated in the last column of Table V, the value of standard deviation at n=6 is 3.07% and was obtained after outlying test results. This level of deviation indicates that the validity of the test by using standard deviation at n=6 without outlying test results is not consistent.

Also, in the last column of Table VI at n=6 only Normal and Triangular distributions held the validity with 2.51% and 3.23% respectively after outlying test results. It means that by using Triangular distribution at n=6 the validity may not hold without outlying test results.

TABLE V
COMPUTED $\bar{X}$, $\%R/\bar{X}$ AND σ FROM SETS OF GENERATED NUMBERS

									n=3							Average
$\bar{X}$(kV)	43.0	41.1	41.7	40.5	37.5	41.2	37.5	43.4	41.9	41.7	40.5	42.7	40.1	42.1	41.8	41.57
σ(%)	0.68	3.29	3.43	3.64	0.05	3.55	0.05	1.59	3.75	2.92	1.74	1.91	3.69	2.43	0.83	2.43
$\%R/\bar{X}$(%)	3.09	15.96	15.31	15.79	3.95	17.16	3.95	6.94	17.0	13.27	8.62	8.87	16.46	11.54	3.44	11.12
								n=4								
$\bar{X}$(kV)	40.29	42.24	40.72	43.64	42.13	40.22	40.39	40.83	42.67	41.96	40.36	41.15	41.19	41.08	40.41	41.28
σ(%)	1.95	2.26	2.52	0.92	3.37	2.92	3.78	2.25	2.21	3.32	1.60	2.99	2.85	2.79	2.65	2.55
$\%R/\bar{X}$(%)	10.1	11.41	12.83	4.31	16.36	15.12	19.86	12.90	10.96	18.3	9.28	14.38	16.35	12.98	15.89	13.40
								n=5								
$\bar{X}$(kV)	40.69	42.09	40.93	40.29	40.06	40.09	41.50	42.64	41.21	41.88	40.57	40.01	40.23	40.49	40.37	41.00
σ(%)	3.11	2.94	3.34	2.91	3.66	0.89	3.47	2.95	3.04	3.43	3.38	2.59	3.35	3.38	1.87	2.84
$\%R/\bar{X}$(%)	20.31	15.18	19.66	20.04	21.63	5.4	20.20	16.19	17.28	18.77	21.57	16.85	22.43	24.01	12.71	17.14
								n=6								
$\bar{X}$(kV)	40.19	40.71	41.70	40.87	41.46	41.96	40.60	40.49	40.16	40.38	40.40	40.96	40.90	41.18	40.03	40.79
σ(%)	3.74	3.09	2.16	3.52	2.46	2.77	3.37	3.46	3.72	1.85	3.00	3.93	2.65	2.43	3.92	3.07
$\%R/\bar{X}$(%)	24.78	19.06	12.65	21.63	16.47	15.51	20.92	21.71	25.29	12.37	17.49	26.64	15.60	17.68	23.79	19.45
								n=7								
$\bar{X}$(kV)	40.60	41.38	40.62	41.26	41.19	40.46	40.07	40.49	40.58	40.08	40.12	40.57	41.89	40.09	40.83	40.67
σ(%)	3.40	1.40	2.98	2.55	3.61	2.01	3.99	2.99	3.08	2.91	3.89	2.54	2.56	2.60	2.74	2.91
$\%R/\bar{X}$(%)	20.66	9.08	18.99	20.29	24.85	14.97	26.56	18.89	20.09	18.79	28.47	18.65	14.08	20.83	20.75	19.89
								n=8								
$\bar{X}$(kV)	41.08	40.55	40.51	40.49	40.65	40.29	40.14	40.63	40.80	40.02	40.46	42.09	41.01	40.06	40.63	40.63
σ(%)	3.12	2.92	3.36	2.22	3.29	3.89	2.67	1.92	3.32	3.43	1.99	2.09	3.55	3.53	1.58	2.86
$\%R/\bar{X}$(%)	19.90	22.61	25.75	15.31	26.53	23.95	17.85	13.06	25.65	23.93	14.96	12.76	22.11	22.35	11.24	19.87
								n=9								
$\bar{X}$(kV)	41.21	40.82	40.63	40.76	40.18	40.85	40.97	40.66	40.74	40.86	40.51	40.14	40.26	40.94	40.00	40.64
σ(%)	3.86	3.21	3.88	3.93	3.30	3.34	3.19	3.09	3.45	3.62	2.93	3.17	3.85	3.32	2.59	3.38
$\%R/\bar{X}$(%)	27.39	24.92	26.11	31.71	22.07	23.37	23.07	20.43	23.04	28.82	25.99	18.94	24.86	23.74	19.25	24.24
								n=10								
$\bar{X}$(kV)	40.24	40.10	40.27	40.36	40.73	41.54	40.67	40.80	40.81	40.24	40.67	40.41	40.06	40.98	40.96	40.59
σ(%)	3.23	3.55	2.81	3.05	3.82	2.33	3.47	3.10	3.67	2.81	3.49	3.42	3.66	2.65	3.42	3.23
$\%R/\bar{X}$(%)	22.03	24.08	22.29	21.37	27.46	20.09	23.44	19.77	24.93	24.73	26.71	26.49	26.91	19.71	25.63	23.71
								n=15								
$\bar{X}$(kV)	40.55	40.60	40.32	40.70	40.03	40.22	40.26	40.64	40.39	40.33	41.04	40.12	40.13	40.49	41.13	40.46
σ(%)	3.77	3.98	3.05	3.72	3.81	3.26	3.12	2.84	3.84	3.59	2.76	3.46	3.67	2.61	3.68	3.41
$\%R/\bar{X}$(%)	24.63	32.51	22.99	30.61	34.89	26.53	28.52	27.10	32.91	32.23	20.24	26.97	35.41	20.54	33.25	28.62
								n=20								
$\bar{X}$(kV)	40.52	40.12	40.26	40.09	40.44	40.05	40.29	40.10	40.28	40.68	40.40	40.47	40.25	40.47	40.64	40.34
σ(%)	3.34	3.76	3.72	3.48	3.91	3.81	3.16	3.57	3.09	3.51	3.65	3.39	2.69	3.55	3.42	3.47
$\%R/\bar{X}$(%)	32.12	29.06	30.88	31.54	27.95	31.69	25.85	30.82	25.88	25.63	27.46	28.16	22.30	33.52	33.38	29.08
								n=25								
$\bar{X}$(kV)	40.05	40.05	40.16	40.25	40.13	40.37	40.01	40.55	40.22	40.11	40.12	40.69	40.66	40.06	40.05	40.23
σ(%)	3.66	3.68	3.93	3.51	3.42	3.53	3.65	3.88	3.64	3.58	3.95	3.48	3.81	3.57	3.68	3.67
$\%R/\bar{X}$(%)	35.70	34.45	36.12	30.77	33.01	27.70	32.25	32.97	29.19	34.87	34.89	29.79	30.19	35.27	30.84	32.54

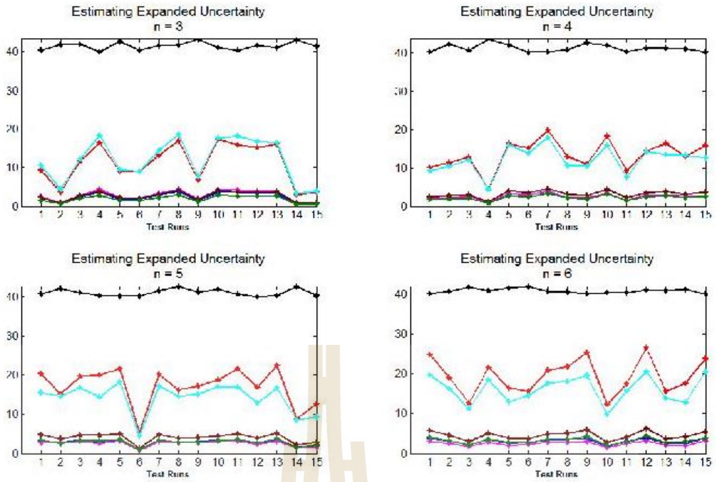


Fig. 1(a). Estimating expanded uncertainty
with n=3, 4, 5 and 6

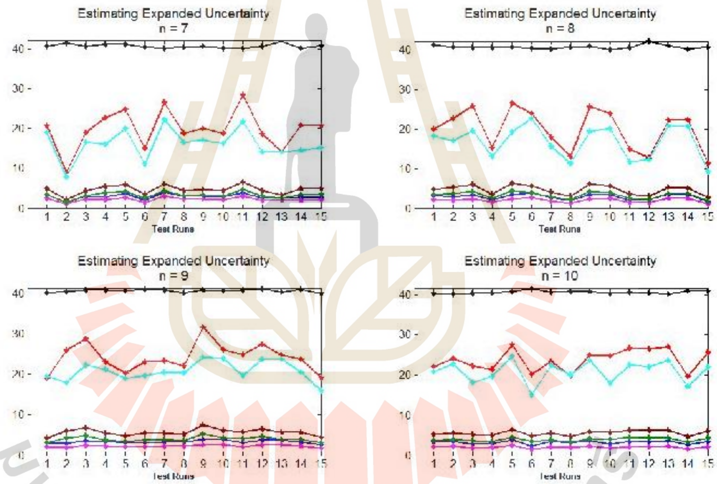


Fig. 1(b). Estimating expanded uncertainty
with n=7, 8, 9 and 10

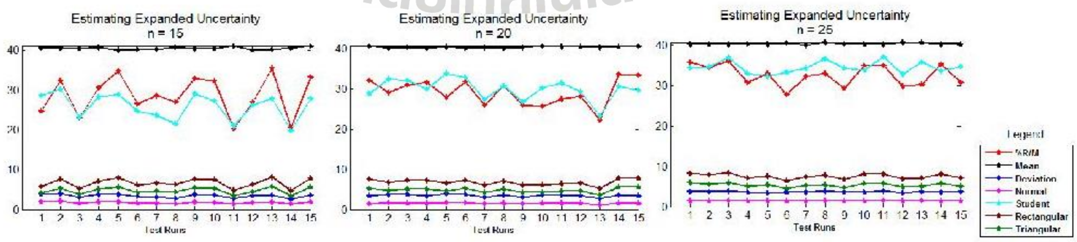


Fig. 1(c). Estimating expanded uncertainty
with n=15, 20 and 25

TABLE VI
ESTIMATED EXPANDED UNCERTAINTIES BASED ON NORMAL, STUDENT, RECTANGULAR AND TRIANGULAR DISTRIBUTIONS

Distribution								u_x								Average
Normal	2.47	1.00	2.83	4.27	2.21	2.08	3.36	4.32	1.81	4.12	4.23	3.91	3.82	0.77	0.88	2.81
Student	10.61	4.30	12.17	18.37	9.50	8.94	14.45	18.58	7.79	17.69	18.21	16.81	16.43	3.31	3.79	12.07
Rectangular	2.19	0.87	2.83	3.81	2.19	2.08	3.18	4.09	1.73	4.09	3.69	3.64	3.81	0.75	0.88	2.66
Triangular	1.55	0.61	2.00	2.69	1.55	1.47	2.25	2.89	1.22	2.89	2.61	2.57	2.69	0.53	0.62	1.88
								n=4								
Normal	1.95	2.21	2.52	0.93	3.36	2.92	3.78	2.25	2.21	3.32	1.60	2.99	2.84	2.79	2.65	2.55
Student	9.29	10.55	12.00	4.42	16.03	13.91	18.03	10.72	10.53	15.84	7.64	14.25	13.56	13.30	12.65	12.18
Rectangular	2.34	2.77	3.01	1.09	3.98	3.51	4.63	3.04	2.69	4.43	2.16	3.42	3.88	3.08	3.71	3.19
Triangular	1.66	1.96	2.13	0.78	2.81	2.48	3.27	2.15	1.91	3.14	1.53	2.42	2.75	2.18	2.62	2.25
								n=5								
Normal	2.78	2.63	2.99	2.60	3.28	0.79	3.10	2.63	2.72	3.07	3.03	2.32	2.99	1.53	1.67	2.54
Student	15.46	14.60	16.63	14.47	18.22	4.39	17.26	14.65	15.13	17.05	16.82	12.89	16.65	8.49	9.30	14.14
Rectangular	4.77	3.69	4.65	4.66	4.99	1.25	4.84	3.98	4.11	4.54	5.05	3.89	5.21	2.18	2.97	4.05
Triangular	3.37	2.61	3.29	3.29	3.54	0.88	3.43	2.82	2.91	3.21	3. 57	2.75	3.68	1.54	2.09	2.87
								n=6								
Normal	3.06	2.53	1.76	2.87	2.01	2.27	2.75	2.82	3.04	1.51	2.45	3.20	2.16	1.99	3.19	2.51
Student	19.65	16.24	11.31	18.45	12.92	14.58	17.69	18.15	19.51	9.72	15.75	20.58	13.87	12.76	20.55	16.12
Rectangular	5.76	4.48	3.04	5.10	3.94	3.78	4.90	5.07	5.87	2.88	4.08	6.29	3.68	4.21	5.50	4.57
Triangular	4.07	3.16	2.15	3.61	2.79	2.67	3.47	3.58	4.15	2.04	2.88	4.45	2.60	2.97	3.89	3.23
								n=7								
Normal	2.57	1.06	2.25	2.18	2.73	1.51	3.02	2.26	2.33	2.20	2.95	1.92	1.94	1.96	2.07	2.20
Student	18.89	7.80	16.55	16.04	20.06	11.15	22.16	16.62	17.13	16.19	21.66	14.08	14.23	14.44	15.23	16.15
Rectangular	4.84	2.17	4.46	5.41	5.91	3.49	6.14	4.41	4.71	4.35	6.59	4.36	3.41	4.82	4.89	4.67
Triangular	3.43	1.54	3.15	3.83	4.18	2.47	4.34	3.12	3.33	3.08	4.66	3.09	2.41	3.41	3.45	3.30
								n=8								
Normal	2.20	2.07	2.38	1.57	2.33	2.75	1.89	1.35	2.34	2.42	1.41	1.48	2.51	2.49	1.11	2.02
Student	18.21	17.06	19.64	12.99	19.23	22.69	15.62	11.22	19.36	20.02	11.68	12.25	20.77	20.65	9.19	16.71
Rectangular	4.72	5.29	6.02	3.58	6.22	5.57	4.13	3.07	6.04	5.53	3.49	3.10	5.24	5.17	2.63	4.66
Triangular	3.34	3.74	4.26	2.53	4.40	3.94	2.92	2.17	4.27	3.91	2.47	2.19	3.70	3.66	1.86	3.29
								n=9								
Normal	2.11	1.96	2.42	2.29	2.06	2.13	2.23	2.20	2.62	2.59	2.14	2.58	2.57	2.21	1.72	2.26
Student	19.52	18.07	22.32	21.23	19.08	19.71	20.59	20.35	24.24	23.97	19.75	23.79	23.71	20.46	15.94	20.85
Rectangular	4.39	6.08	6.80	5.42	4.79	5.46	5.51	5.12	7.47	6.13	5.87	6.52	5.73	5.61	4.44	5.69
Triangular	3.10	4.29	4.81	3.83	3.39	3.86	3.89	3.62	5.28	4.33	4.15	4.61	4.05	3.96	3.14	4.02
								n=10								
Normal	2.05	2.24	1.78	1.93	2.42	1.48	2.19	1.96	2.32	1.77	2.21	2.16	2.31	1.67	2.17	2.04
Student	20.81	22.80	18.10	19.61	24.58	15.01	22.33	19.95	23.59	18.05	22.48	22.01	23.52	17.07	22.02	20.80
Rectangular	5.12	5.58	5.18	4.98	6.46	4.82	5.50	4.66	5.87	5.74	6.27	6.18	6.22	4.66	6.06	5.55
Triangular	3.62	3.94	3.67	3.52	4.57	3.41	3.89	3.29	4.15	4.06	4.44	4.37	4.40	3.29	4.28	3.93
								n=15								
Normal	1.94	2.06	1.58	1.92	1.97	1.68	1.61	1.46	1.98	1.85	1.43	1.79	1.89	1.35	1.89	1.76
Student	28.63	30.22	23.17	28.21	28.94	24.73	23.65	21.53	29.15	27.22	20.98	26.26	27.89	19.83	27.91	25.89
Rectangular	5.77	7.62	5.35	7.19	8.06	6.16	6.63	6.36	7.67	7.51	4.79	6.25	8.20	4.80	7.89	6.68
Triangular	4.08	5.39	3.79	5.09	5.70	4.36	4.69	4.49	5.43	5.31	3.39	4.42	5.80	3.39	5.58	4.73
								n=20								
Normal	1.49	1.68	1.67	1.55	1.75	1.70	1.41	1.59	1.39	1.57	1.63	1.51	1.20	1.59	1.53	1.55
Student	28.80	32.44	32.12	29.99	33.76	32.84	27.25	30.76	26.72	30.23	31.49	29.21	23.27	30.63	29.54	29.94
Rectangular	7.51	6.73	7.17	7.30	6.53	7.32	6.01	7.14	6.02	6.02	6.40	6.58	5.18	7.83	7.83	6.77
Triangular	5.31	4.76	5.08	5.16	4.61	5.18	4.25	5.04	4.26	4.26	4.53	4.65	3.66	5.54	5.53	4.79
								n=25								
Normal	1.46	1.47	1.57	1.40	1.37	1.41	1.46	1.55	1.46	1.43	1.58	1.39	1.52	1.43	1.47	1.47
Student	34.52	34.63	37.00	33.02	32.21	33.21	34.36	36.54	34.23	33.69	37.16	32.70	35.79	33.55	34.63	34.48
Rectangular	8.26	7.97	8.37	7.15	7.64	6.46	7.45	7.71	6.78	8.08	8.08	7.00	7.09	8.16	7.13	7.56
Triangular	5.84	5.63	5.92	5.06	5.41	4.57	5.27	5.46	4.79	5.71	5.71	4.95	5.01	5.77	5.04	5.34

VI. Conclusion

The estimation of expanded uncertainty and the effect of Normal, Student-*t*, Rectangular and Triangular distributions were successfully studied.

The limit of 4% similar to standard deviation in IEC 60156 was used to analyze effects of distribution approximation. Student-*t* distribution showed invalid results for any number of test results, n. With the smallest number of test results, $n = 3$ the value of expanded was above 4% and was increased with n.

It was contemplated that expanded uncertainty based on Student-*t* distribution does not hold for any number of test results. Rectangular distribution showed validity only for $n \leq 4$. However, $n \leq 4$ does not comply with IEC 60156 which recommends 6 test results. In Triangular distribution, validity was demonstrated with test results $n \leq 8$. But this distribution became incapacitated for test results $n > 8$. With Normal distribution, the validity prevailed to hold in all test results used. Moreover, the estimated expanded uncertainty was decreasing with increase in test results n. The standard deviation was seen to increase with the number of test results n and was bounded to the limit of 4% with the outlied test results.

The validity of estimated expanded uncertainties in each distribution was considered in mineral insulating oil for test results of up to 25 without outliers.

Acknowledgements

This work was supported by the Laboratory of High Voltage Insulation Technologies of Suranaree University of Technology, Thailand.

References

[1] Sedaghat, B., Mehrvarz, H., Mashayekhi, M., Evaluation of the Sympathetic Interaction between Three 150MVA Power Transformers During Energizing the Third Transformer in Shahr-e-Kord 400/63kV Substation, (2013) *International Review on Modelling and Simulations (IREMOS)*, 6 (6), pp. 1801-1808

[2] Biçen, Y., Aras, F., İsmailoğlu, H., A Realistic Method for Remaining Service Life Prediction of Power Transformer, (2013) *International Review of Electrical Engineering (IREE)*, 8 (1), pp. 200-206.

[3] Wang, K., Liao, R., Yang, L., Yuan, L., Wu, F., Duan, L., Nonnegative Matrix Factorization Aided Principal Component Analysis for High-Resolution Partial Discharge Image Compression in Transformers, (2013) *International Review of Electrical Engineering (IREE)*, 8 (1), pp. 479-490.

[4] Taktak, M., Ville, J., Haddar, M., Gabard, G., Foucart, F., Acoustic Liners Behavior Investigation Using the Multimodal Scattering Matrix, (2015) *International Review of Aerospace Engineering (IREASE)*, 8 (6), pp. 198-208.
doi: https://doi.org/10.15866/irease.v8i6.8822

[5] Dib, S., Barkat, M., Nicolas, J., Benslama, M., Chaotic Staggered PRF and Reduced Rank STAP for Airborne Radar, (2014) *International Review of Aerospace Engineering (IREASE)*, 7 (1), pp. 25-33.

[6] Birch, K., Measurement Guide No.36: Estimating Uncertainty in Testing – An Intermediate Guide to Estimating and Reporting Uncertainty in Measurement and Testing, (2001) British Measurement and Testing Association.

[7] Hayden, J. L. R. and Eddy, W. N., Three Thousand Tests on the Dielectric Strength of Oil, (1922) *Transaction of Annual Convention of the A. I- E- E(AIEE).*

[8] JCGM 100:2008, Evaluation of Measurement Data – Guide to the Expression of Uncertainty to the Measurement, (2008) JCGM.

[9] Bell, S., Measurement Good Practice Guide No. 11: A Beginner's Guide to Uncertainty of Measurement, (1999) National Physical Laboratory.

[10] EUROLAB Technical Report 1/2006, Guide to the Evaluation of Measurement Uncertainty for Quantitative Test Results, (2006) EUROLab.

[11] Coleman, H.W. and Steele, W.G., *Experimentation, Validation and Uncertainty Analysis for Engineers*, (2009)John Wiley & Sons; Hoboken-New Jersey,

[12] Lazic', Z. R., *Design of Experiments in Chemical Engineering: A Practical Guide*, (2004) Wiley-VCH.

[13] Reddy, N., Ramana, N., Reddy, K., A Unit Commitment Solution Using Differential Evolution and Economic Dispatch Using Shuffled Complex Evolution with Principal Component Analysis, (2013) *International Review on Modelling and Simulations (IREMOS)*, 6 (3), pp. 819-833.

Authors' information

School of Electrical Engineering, Suranaree University of Technology, 111 University Avenue, Nakhon Ratchasima 30000, Thailand.

APPENDIX B

SOURCE CODES FOR DGA SOFTWARE

The following below are some of the source codes used in the development of DGA software presented in this research and its diagnostic methods used in interpretation of DGA information.

- **Codes for developing the graphical user interface.**

This section shows the source codes used to implement graphical user interface of the DGA software which includes the input form, menu items and diagnostic methods.

```
#!/usr/bin/env python
from Tkinter import *
import ttk
import shutil
import os, sys, glob, time
import tkMessageBox
import Tkinter
import math
import MySQLdb
import _mysql
from datetime import datetime
from tkMessageBox import showwarning
from Canvas import CanvasText
from decimal import ROUND_05UP, ROUND_HALF_DOWN, ROUND_FLOOR
from types import *
from datetime import datetime, timedelta
import functools
from fileinput import filename
fields1 = ["HYDROGEN  H2", "CARBON MONOXIDE CO", "CARBON DIOXIDE
            CO2", "NITROGEN N2", "METHANE CH4", "ETHANE C2H6",
```

```
"ETHYLENE C2H4", "ACETYLENE C2H2", "POWER (kVA)",  "VOLTAGE
(kV)", "OIL TEMPERATURE (C)", "ALTITUDE   (M)","HUMIDITY
(%)","TRANSFORMER LOCATION",        "TRANSFORMER NUMBER"]
entries = []
def makeform(root, fields1):
    for field in fields1:
        row = Frame(root)
        lab = Label(row, relief=RAISED, width=30, height=1, text=field)
        entt=StringVar()
        ent = Entry(row, textvariable=entt, relief=RAISED, width=32)
        row.pack(side=TOP, anchor=NW)
        lab.pack(side=LEFT)
        ent.pack(side=RIGHT, expand=NO)
        ent.delete(0, END)
        ent.insert(END, 'Input your value here')
        ent.focus()
        ent.bind('<Return>', (lambda event: fetch()))
        entries.append(entt)
    return entries
menu = Menu(root)
root.config(menu=menu)
filemenu = Menu(menu)
menu.add_cascade(label="File", menu=filemenu, underline=0)
submenu = Menu(filemenu)
submenu.add_command(label="Diagnosis Summary", command=summy)
submenu.add_command(label="Comparison Summary", command=comp)
submenu.add_command(label="Trending Summary", command=sumy)
filemenu.add_cascade(label="Open...", menu=submenu, underline=0)
filemenu.add_separator()
filemenu.add_command(label="Close App", command=syqut, underline=0)
separator = ttk.Separator()
viewmenu = Menu(menu)
menu.add_cascade(label="View", menu=viewmenu, underline=0)
viewmenu.add_command(label="Database Connection", command=dbconn)
viewmenu.add_command(label="Database Details", command=dbdetails)
viewmenu.add_command(label="Last Entry", command=gastable, underline=0)
viewmenu.add_command(label="Database Entries", command=dmanage)
commcentermenu = Menu(menu)
```

```
menu.add_cascade(label="Communication", menu=commcentermenu, underline=0)
 commcentermenu.add_command(label="Communication Center",command=ccenter)
multidgamenu = Menu(menu)
menu.add_cascade(label="Multiple DGA", menu=multidgamenu, underline=0)
multidgamenu.add_command(label="Multiple DGA", command=multipledga)
searchmenu = Menu(menu)
menu.add_cascade(label="Search", menu=searchmenu, underline=0)
searchmenu.add_command(label="Search Data", command=dsearch, underline=0)
helpmenu = Menu(menu)
menu.add_cascade(label="Help", menu=helpmenu, underline=0,)
helpmenu.add_command(label="About Application", command=about_app)
helpmenu.add_separator()
submenu = Menu(helpmenu)
submenu.add_command(label="Input Gases", command=inputgases, underline=0)
submenu.add_command(label="Save To Database",command=savetodatabase)
submenu.add_separator()
submenu.add_command(label="Diagnosing Methods", command=dmethod)
submenu.add_separator()
helpmenu.add_cascade (label="Usage", menu=submenu, underline=0)
text2 = Text (root, height=10, width=41, relief=RIDGE, padx =5, pady =2)
text2.tag_configure ('bold_italics', font= ('times', 10, 'bold', 'italic'))
text2.tag_configure ('color', foreground='darkgreen', font= ('times', 13, 'bold'))
text2.insert (END,' A Technical Quote', 'big')
quote = """
        Condition Monitoring is a strong tool\n for transformer management but an\n
        accurate diagnosis of dissolved gas-\n in-oil and interpretation makes it \n the
        strongest one...
        """
text2.insert (END, quote, 'color')
text2.config (relief=RAISED, bd=3)
text2.pack (side=BOTTOM, anchor=SW)
text2.config (cursor='hand2')
root.mainloop ()
```

- **Source codes for performing key gas method.**

This section shows the source codes used to implement key gas method. They include getting data from user, retrieving DGA information from the database, diagnosing and saving the diagnostic results.

```
def keygasmethod():
    oceann = IntVar()
    def ferrt():
        start_time = datetime.now()
        kgs=(oceann.get())
        kg=int(kgs)
        q1=can21.create_text(760,25, text="Your input: %d " % kg)
        sql=("SELECT*FROM new WHERE id= '%d'" % (kg))
        db=MySQLdb.connect("127.0.0.1", "root", "mysql125", "dissolved")
        cursor=db.cursor()
        cursor.execute(sql)
        results = cursor.fetchall()
        for col in results:
            ed = col[0]
            hydrogen = col[1]
            carbon_monoxide = col[2]
            carbon_dioxide = col[3]
            nitrogen = col[4]
            methane = col[5]
            ethane = col[6]
            ethylene = col[7]
            acetylene = col[8]
            power = col[9]
            voltage = col[10]
            oil_temperature = col[11]
            altitude = col[12]
            humidity = col[13]
            m_location = col[14]
            m_number = col[15]
        can21.create_text(115,25, text="Your data for diagnosis", fill='blue')
```

```
can21.create_text(60,45, text="ID:%s "  % ed)
can21.create_text(110,45, text="H2:%s " % hydrogen)
can21.create_text(165,45, text="CO:%s " % carbon_monoxide)
can21.create_text(225,45, text="CO2: %s " % carbon_dioxide)
can21.create_text(285,45, text="N2: %s " % nitrogen)
can21.create_text(350,45, text="CH4: %s " % methane)
can21.create_text(420,45, text="C2H6: %s " % ethane)
can21.create_text(495,45, text="C2H4: %s " % ethylene)
can21.create_text(575,45, text="C2H2: %s " % acetylene)
a=int(" %s " % hydrogen)
b=int(" %s " % carbon_monoxide)
c=int(" %s " % methane)
d=int(" %s " % ethane)
e=int(" %s " % ethylene)
f=int(" %s " % acetylene)
g1=int(" %s " % carbon_dioxide)
TDCG =  a+b+c+d+e+f
a1=(a*100)/float(TDCG), b1=(b*100)/float(TDCG),c1=(c*100)/float(TDCG)
d1=(d*100)/float(TDCG), e1=(e*100)/float(TDCG), f1=(f*100)/float(TDCG)
a2=round(a1, 2), b2=round(b1, 2), c2=round(c1, 2), d2=round(d1, 2)
e2=round(e1, 2), f2=round(f1, 2), g2=round(g1, 2)
F1="Overheating in Oil",  F2="Overheating of Cellulose",F3="Arcing in Oil
F4="Partial Discharge and Corona",  F5="Normal", F6="No Prediction"
can21.create_text(83,70, text="TDCG: %d " % TDCG)
can21.create_text(160,70, text="H2: %.2f " % a2)
can21.create_text(194,70, text="%")
can21.create_text(240,70, text="CO: %.2f " % b2)
can21.create_text(275,70, text="%")
can21.create_text(325,70, text="CH4: %.2f " % c2)
can21.create_text(365,70, text="%")
can21.create_text(420,70, text="C2H6: %.2f " % d2)
can21.create_text(463,70, text="%")
can21.create_text(520,70, text="C2H4: %.2f " % e2)
can21.create_text(563,70, text="%")
can21.create_text(620,70, text="C2H2: %.2f " % f2)
can21.create_text(660,70, text="%")
can21.create_line(180, (500-(2*a1)),220, (500-(2*a1)),width=(a1*4), fill='blue')
can21.create_line(260, (500-(2*b1)),300, (500-(2*b1)),width=(b1*4), fill='blue')
can21.create_line(340, (500-(2*c1)),380, (500-(2*c1)),width=(c1*4), fill='blue')
can21.create_line(420, (500-(2*d1)), 460,(500-(2*d1)),width=(d1*4), fill='blue')
can21.create_line(500, (500-(2*e1)), 540, (500-(2*e1)),width=(e1*4), fill='blue')
can21.create_line(580, (500-(2*f1)), 620, (500-(2*f1)), width=(f1*4), fill='blue')
can21.create_text(780, 160, text=" %s " % hydrogen)
can21.create_text(780, 200, text=" %s " % carbon_monoxide)
can21.create_text(780, 240, text=" %s " % methane)
can21.create_text(780, 280, text=" %s " % ethane)
can21.create_text(780, 320, text=" %s " % ethylene)
```

```
can21.create_text(780, 360, text=" %s " % acetylene)
can21.create_text(780, 400, text=" %s " % carbon_dioxide)
can21.create_text(780, 440, text=" %s " % nitrogen)
can21.create_text(650,44, text="(ppm)", fill='blue')
can21.create_text(880, 440, text=" N.A ")
can21.create_text(880, 360, text=" N.A ")
can21.create_text(740, 480, text=" FAULT: ")
if a > 1000:
   can21.create_text(880, 160, text="ARCING, CORONA")
else:
   can21.create_text(880, 160, text="NORMAL")
if b > 1000:
   can21.create_text(880, 200, text="SEVERE OVERHEATING")
else:
   can21.create_text(880, 200, text="NORMAL")
if c > 80:
   can21.create_text(880, 240, text="SPARKING")
else:
   can21.create_text(880, 240, text="NORMAL")
if d > 35:
   can21.create_text(880, 280, text="LOCAL OVERHEATING")
else:
   can21.create_text(880, 280, text="NORMAL")
if e > 100:
   can21.create_text(885, 320, text="SEVERE OVERHEATING")
else:
   can21.create_text(880, 320, text="NORMAL")
if g1 > 15000:
   can21.create_text(820, 400, text="SEVERE OVERHEATING")
else:
   can21.create_text(880, 400, text="NORMAL")
if a2 >= 60 and  f2 >= 30:
   can21.create_text(820, 500, text="%s " % F3)
   t=("%s "%F3)
elif a2 >= 85 and c2 >= 13:
   can21.create_text(820, 500, text="%s " % F4)
   t=("%s "%F4)
elif b2 >= 90:
   can21.create_text(820, 500, text="%s " % F2)
   t=("%s "%F2)
elif d2 >= 19 and e2 >= 63:
   can21.create_text(820, 500, text="%s " % F1)
   t=("%s "%F1)
else:
   can21.create_text(820, 506, text="%s " % F6)
   t=("%s "%F6)
cursor.close()
db.close()
```

```
lo9.configure(state=DISABLED, background='silver')
can21.create_text(935, 545, text=" *"*3, fill='green')
can21.create_text(825, 545, text="CHOOSE ANOTHER METHOD")
can21.create_text(710, 545, text=" *"*3, fill='green')
with open('.\summary.txt', mode='a+') as fl:
    lines_of_text = ["     \n\n   DGA FOR POWER TRANSFORMERS\n", "
            SCHOOL OF ELECTRICAL ENGINEERING\n", "SURANAREE
            UNIVERSITY OF TECHNOLOGY\n", "--"*20, "\n DIAGNOSIS
    SUMMARY\n", "--"*20]
    lines_of_text1 = ["\n    Method: KEY GAS METHOD"]
    lines_of_text2 = ["\n    Time: %s " % time.strftime('%c')]
    lines_of_text3 = ["\n    Data ID: %d " % (kg)]
    lines_of_text4 = ["\n Data => H2:%s "%(hydrogen), "C0:%s
            "%(carbon_monoxide), "C02:%s "%(carbon_dioxide), "N2:%s
            "%(nitrogen), "CH4:%s "%(methane), "C2H6:%s "%(ethane),
            "C2H4:%s "%(ethylene), "C2H2:%s "%(acetylene)]
    lines_of_text5 = ["\n kVA:%s "%(power), "kV:%s "%(voltage), "Temp:%s
            "%(oil_temperature), "Altitude:%s "%(altitude), "Humidity:%s
            "%(humidity), "Location:%s "%(m_location), "Number:%s
            "%(m_number)]
    lines_of_text6 = ["\n\nAnalysis:"]
    lines_of_text7 = ["\n TDCG :%d "%(TDCG), "H2(percent):%.2f "%(a2),
            "CO(percent):%.2f "%(b2), "CH4(percent):%.2f "%(c2)]
    lines_of_text8 = ["\nC2H6(percent):%.2f "%(d2), "C2H4(percent):%.2f
            "%(e2), "C2H2(percent):%.2f "%(f2)]
    lines_of_text9 = ["\n\n    Fault:", " %s "%t]
    fl.writelines(lines_of_text)
    fl.writelines(lines_of_text1)
    fl.writelines(lines_of_text2)
    fl.writelines(lines_of_text3)
    fl.writelines(lines_of_text4)
    fl.writelines(lines_of_text5)
    fl.writelines(lines_of_text6)
    fl.writelines(lines_of_text7)
    fl.writelines(lines_of_text8)
    fl.writelines(lines_of_text9)
    fl.writelines(["\n\nEND OF DIAGNOSIS SUMMARY\n", "--"*45])
    fl.close()
end_time = datetime.now()
can21.create_text(808,65,
          text=('Computing Time:{}'.format(end_time - start_time)))
```

- **Source codes for performing Dornenburg ratio method.**

This section shows the source codes used to implement Dornenburg ratio method. They include getting data from user, retrieving DGA information from the database, diagnosing and saving the diagnostic results.

```
def dornenburgmethod():
   oceann = StringVar()
   def ferrm():
      start_time = datetime.now()
      kgs=(oceann.get())
      kg=int(kgs)
      q1=can22.create_text(800,25, text="Your input: %d " % kg)
      sql=("SELECT*FROM new WHERE id= '%d'" % (kg))
      db=MySQLdb.connect("127.0.0.1", "root", "mysql125", "dissolved")
      cursor=db.cursor()
      cursor.execute(sql)
      results = cursor.fetchall()
      fiel=cursor.description
      for col in results:
         id = col[0]
         hydrogen = col[1]
         carbon_monoxide = col[2]
         carbon_dioxide = col[3]
         nitrogen = col[4]
         methane = col[5]
         ethane = col[6]
         ethylene = col[7]
         acetylene = col[8]
         power = col[9]
         voltage = col[10]
         oil_temperature = col[11]
         altitude = col[12]
         humidity = col[13]
         m_location = col[14]
         m_number = col[15]
      can22.create_text(115,25, text="Your data for diagnosis", fill='blue')
      can22.create_text(60,45, text="ID:%s "  % id)
      can22.create_text(110,45, text="H2:%s " % hydrogen)
      can22.create_text(165,45, text="CO:%s " % carbon_monoxide)
      can22.create_text(225,45, text="CO2: %s " % carbon_dioxide)
      can22.create_text(285,45, text="N2: %s " % nitrogen)
      can22.create_text(350,45, text="CH4: %s " % methane)
```

```
 can22.create_text(420,45, text="C2H6: %s " % ethane)
 can22.create_text(495,45, text="C2H4: %s " % ethylene)
 can22.create_text(575,45, text="C2H2: %s " % acetylene)
 a=int(" %s " % hydrogen), b=int(" %s " % carbon_monoxide)
c=int(" %s " % methane), d=int(" %s " % ethane)
 e=int(" %s " % ethylene), f=int(" %s " % acetylene)
 if f==0:
    d2=0
 else:
    d1=(d/float(f))
    d2=round(d1, 2)
 a1=(c/float(a)), b1=(f/float(e)), c1=(f/float(c)),  e1=(e/float(d))
 a2=round(a1, 2), b2=round(b1, 2), c2=round(c1, 2), e2=round(e1, 2)
ab=" ABOVE LIMIT ",  aba=" NORMAL ",ab1=" Thermal Decomposition "
ab2="Partial Discharge (Low Intensity PD) ", ab3="Arcing (High Intensity PD) "
 ab4=" No Prediction "
 can22.create_text(650,44, text="(ppm)", fill='blue')
 can22.create_text(83,75, text="Gas Ratios:", fill='blue')
 can22.create_text(140,70, text="CH4", fill='black')
 can22.create_line(128,77, 151, 77, width=1, fill='black')
 can22.create_text(140,85, text="H2", fill='black')
 can22.create_text(162,76, text="=", fill='black')
 can22.create_text(185,76, text="%.2f " % a2)
 can22.create_text(255,70, text="C2H2", fill='black')
 can22.create_line(240,77, 272, 77, width=1, fill='black')
 can22.create_text(255,85, text="C2H4", fill='black')
 can22.create_text(281,76, text="=", fill='black')
 can22.create_text(305,76, text="%.2f " % b2)
 can22.create_text(365,70, text="C2H2", fill='black')
 can22.create_line(350,77, 382, 77, width=1, fill='black')
 can22.create_text(365,85, text="CH4", fill='black')
 can22.create_text(395,76, text="=", fill='black')
 can22.create_text(420,76, text="%.2f " % c2)
 can22.create_text(475,70, text="C2H6", fill='black')
 can22.create_line(460,77, 490, 77, width=1, fill='black')
 can22.create_text(475,85, text="C2H2", fill='black')
 can22.create_text(505,76, text="=", fill='black')
 can22.create_text(535,76, text="%.2f " % d2)
 can22.create_text(600,70, text="C2H4", fill='black')
 can22.create_line(585,77, 615, 77, width=1, fill='black')
 can22.create_text(600,85, text="C2H6", fill='black')
 can22.create_text(630,76, text="=", fill='black')
 can22.create_text(660,76, text="%.2f " % e2)
 if a > 100:
    can22.create_text(275,185, text="%s " % ab)
 else:
```

```
    can22.create_text(275, 185, text="%s " % aba)
if b > 350:
    can22.create_text(400,185, text="%s " % ab)
else:
    can22.create_text(400, 185, text="%s " % aba)
if c > 120:
    can22.create_text(525,185, text="%s " % ab)
else:
    can22.create_text(525, 185, text="%s " % aba)
if f > 35:
    can22.create_text(650,185, text="%s " % ab)
else:
    can22.create_text(650, 185, text="%s " % aba)
if e > 50:
    can22.create_text(775,185, text="%s " % ab)
else:
    can22.create_text(775, 185, text="%s " % aba)
if d > 65:
    can22.create_text(900,185, text="%s " % ab)
else:
    can22.create_text(900, 185, text="%s " % aba)
if a > 100 or b > 350 or c > 120 or d > 65 or e > 50 or f > 35:
    can22.create_text(450,225, text="ONE OR MORE GASES ARE ABOVE
                        THE LIMIT L1   ==>>  THE TEST IS OF IMPORTANCE")
    can22.create_line(50,250, 825, 250, width=1, fill='red')
    can22.create_line(50,280, 825, 280, width=1, fill='red')
    can22.create_line(50,310, 825, 310, width=1, fill='red')
    can22.create_line(50,340, 825, 340, width=1, fill='red')
    can22.create_line(50,370, 825, 370, width=1, fill='red')
    can22.create_line(50,250, 50, 370, width=1, fill='red')
    can22.create_line(200,250, 200, 370, width=1, fill='red')
    can22.create_line(325,250, 325, 340, width=1, fill='red')
    can22.create_line(450,250, 450, 340, width=1, fill='red')
    can22.create_line(575,250, 575, 340, width=1, fill='red')
    can22.create_line(700,250, 700, 340, width=1, fill='red')
    can22.create_line(825,250, 825, 370, width=1, fill='red')
    can22.create_text(125, 265, text="RATIO")
    can22.create_text(250, 265, text="CH4/H2")
    can22.create_text(375, 265, text="C2H2/C2H4")
    can22.create_text(500, 265, text="C2H2/CH4")
    can22.create_text(625, 265, text="C2H6/C2H2")
    can22.create_text(125, 295, text="ABBREVIATION")
    can22.create_text(250, 295, text="R1")
    can22.create_text(375, 295, text="R2")
    can22.create_text(500, 295, text="R3")
    can22.create_text(625, 295, text="R4")
    can22.create_text(125, 325, text="RATIO VALUES")
```

```
    can22.create_text(250, 325, text="%s " % a2)
    can22.create_text(375, 325, text="%s " % b2)
    can22.create_text(500, 325, text="%s " % c2)
    can22.create_text(625, 325, text="%s " % d2)
    can22.create_text(125, 355, text="FAULT:")
    if a2 > 1.0 and b2 < 0.75 and c2 < 0.3 and d2 > 0.4:
        can22.create_text(550, 355, text=" %s " % ab1)
        t=("%s " % ab1)
    elif a2 < 0.1 and c2 < 0.3 and d2 > 0.4:
        can22.create_text(550, 355, text=" %s " % ab2)
        t=("%s " % ab2)
    elif a2 > 0.1 and a2 < 1.0 and b2 > 0.75 and c2 > 0.3 and d2 < 0.4:
        can22.create_text(550, 355, text=" %s " % ab3)
        t=("%s " % ab3)
    else:
        can22.create_text(550, 355, text=" %s " % ab4)
        t=("%s " % ab4)
else:
    can22.create_text(520, 335, text="VALUES ARE WITHIN THE LIMITS
            L1")
    can22.create_text(520, 355, text="NO NEED OF TEST")
    can22.create_text(250, 325, text="*"*3, fill='red')
    can22.create_text(269, 325, text="*"*3, fill='white')
    can22.create_text(295, 325, text="*"*5, fill='blue')
    can22.create_text(320, 325, text="*"*3, fill='white')
    can22.create_text(340, 325, text="*"*3, fill='red')
    can22.create_text(710, 325, text="*"*3, fill='red')
    can22.create_text(729, 325, text="*"*3, fill='white')
    can22.create_text(753, 325, text="*"*5, fill='blue')
    can22.create_text(778, 325, text="*"*3, fill='white')
    can22.create_text(798, 325, text="*"*3, fill='red')
t=("%s " % ab4)
cursor.close()
db.close()
lo7.configure(state=DISABLED, background='silver')
with open('.\summary.txt', mode='a+') as fl:
    lines_of_text = [" \n\n   DGA FOR POWER TRANSFORMERS\n", "
            SCHOOL OF ELECTRICAL ENGINEERING\n", "   SURANAREE
            UNIVERSITY OF TECHNOLOGY\n", "--"*20, "\n DIAGNOSIS
            SUMMARY\n", "--"*20]
    lines_of_text1 = ["\n    Method: DORNENBURG METHOD"]
    lines_of_text2 = ["\n    Time: %s " % time.strftime('%c')]
    lines_of_text3 = ["\n    Data ID: %d " % (kg)]
    lines_of_text4 = ["\n    Data => H2:%s "%(hydrogen), "C0:%s
            "%(carbon_monoxide), "C02:%s "%(carbon_dioxide), "N2:%s
            "%(nitrogen), "CH4:%s "%(methane), "C2H6:%s "%(ethane),
            "C2H4:%s "%(ethylene), "C2H2:%s "%(acetylene)]
```

```
 lines_of_text5 = ["\nkVA:%s "%(power), "kV:%s "%(voltage),
            "Altitude:%s "%(altitude), "Temp:%s "%(oil_temperature),
            "Humidity:%s "%(humidity), "Location:%s "%(m_location),
            "Number:%s "%(m_number)]
  lines_of_text6 = ["\n\n    Analysis:"]
  lines_of_text7 = ["\n    RATIOS:-  ", " R1=CH4/H2=%.2f "%(a2), "
            R2=C2H2/C2H4=%.2f "%(b2), " R3=C2H2/CH4=%.2f "%(c2), "
            R4=C2H6/C2H2=%.2f "%(d2), " R5=C2H4/C2H6=%.2f "%(e2)]
  lines_of_text8 = ["\n    Fault:", " %s "%t]
  fl.writelines(lines_of_text)
  fl.writelines(lines_of_text1)
  fl.writelines(lines_of_text2)
  fl.writelines(lines_of_text3)
  fl.writelines(lines_of_text4)
  fl.writelines(lines_of_text5)
  fl.writelines(lines_of_text6)
  fl.writelines(lines_of_text7)
  fl.writelines(lines_of_text8)
  fl.writelines(["\n\n END OF DIAGNOSIS SUMMARY\n", "--"*45])
  fl.close()
end_time = datetime.now()
can22.create_text(850, 85, text=('Computing time: {}'.format(end_time –
                start_time))
```

- **Source codes for performing Roger's ratio method.**

This section shows the source codes used to implement Roger's ratio method. They include getting data from user, retrieving DGA information from the database, diagnosing and saving the diagnostic results.

```
def rogersmethod():
  oceann = StringVar()
  def ferrk():
     start_time = datetime.now()
     kgs=(oceann.get())
     kg=int(kgs)
     q1=can23.create_text(800,25, text="Your input: %d " % kg)
     sql=("SELECT*FROM new WHERE id='%d'" % (kg))
     db=MySQLdb.connect("127.0.0.1", "root", "mysql125", "dissolved")
```

```
cursor=db.cursor()
cursor.execute(sql)
results = cursor.fetchall()
fiel=cursor.description
for col in results:
    id = col[0]
    hydrogen = col[1]
    carbon_monoxide = col[2]
    carbon_dioxide = col[3]
    nitrogen = col[4]
    methane = col[5]
    ethane = col[6]
    ethylene = col[7]
    acetylene = col[8]
    power = col[9]
    voltage = col[10]
    oil_temperature = col[11]
    altitude = col[12]
    humidity = col[13]
    m_location = col[14]
    m_number = col[15]
can23.create_text(115,25, text="Your data for diagnosis", fill='blue')
can23.create_text(60,45, text="ID:%s "  % id)
can23.create_text(110,45, text="H2:%s " % hydrogen)
can23.create_text(165,45, text="CO:%s " % carbon_monoxide)
can23.create_text(225,45, text="CO2: %s " % carbon_dioxide)
can23.create_text(285,45, text="N2: %s " % nitrogen)
can23.create_text(350,45, text="CH4: %s " % methane)
can23.create_text(420,45, text="C2H6: %s " % ethane)
can23.create_text(495,45, text="C2H4: %s " % ethylene)
can23.create_text(575,45, text="C2H2: %s " % acetylene)
a=int(" %s " % hydrogen)
b=int(" %s " % carbon_monoxide)
c=int(" %s " % methane)
d=int(" %s " % ethane)
e=int(" %s " % ethylene)
f=int(" %s " % acetylene)
a1=(c/float(a)), b1=(d/float(c)), c1=(e/float(d)), d1=(f/float(e))
a2=round(a1, 2), b2=round(b1, 2), c2=round(c1, 2), d2=round(d1, 2)
ba1 = " Normal deterioration ", ba2 = " Low Temperature Thermal Overloading "
ba3 = " Thermal Fault T < 700 C ", ba4 = " Thermal Fault T > 700 C "
ba5 = " Partial Discharge ", ba6 = " Arcing-High Energy "
ba7 = " No Prediction "
can23.create_text(650,44, text="(ppm)", fill='blue')
can23.create_text(83,75, text="Gas Ratios:", fill='blue')
can23.create_text(140,70, text="CH4", fill='black')
```

```
can23.create_line(123,77, 156, 77, width=1, fill='black')
can23.create_text(140,85, text="H2", fill='black')
can23.create_text(167,76, text="=", fill='black')
can23.create_text(190,76, text="%.2f " % a2)
can23.create_text(255,70, text="C2H6", fill='black')
can23.create_line(240,77, 272, 77, width=1, fill='black')
can23.create_text(259,85, text="CH4", fill='black')
can23.create_text(281,76, text="=", fill='black')
can23.create_text(305,76, text="%.2f " % b2)
can23.create_text(365,70, text="C2H4", fill='black')
can23.create_line(350,77, 382, 77, width=1, fill='black')
can23.create_text(365,85, text="C2H6", fill='black')
can23.create_text(395,76, text="=", fill='black')
can23.create_text(420,76, text="%.2f " % c2)
can23.create_text(470,70, text="C2H2", fill='black')
can23.create_line(455,77, 485, 77, width=1, fill='black')
can23.create_text(470,85, text="C2H4", fill='black')
can23.create_text(495,76, text="=", fill='black')
can23.create_text(520,76, text="%.2f " % d2)
if a2 <= 0.1:
   can23.create_text(440, 265, text="%.2f " % a2)
   can23.create_text(670, 265, text=" 5 ")
   can23.create_text(280, 475, text=" 5 ")
if a2 > 0.1 and a2 < 1.0:
   can23.create_text(440, 265, text="%.2f " % a2)
   can23.create_text(670, 265, text=" 0 ")
   can23.create_text(280, 475, text=" 0 ")
if a2 >= 1.0 and a2 < 3.0:
   can23.create_text(440, 265, text="%.2f " % a2)
   can23.create_text(670, 265, text=" 1 ")
   can23.create_text(280, 475, text=" 1 ")
if a2 >= 3.0:
   can23.create_text(440, 265, text="%.2f " % a2)
   can23.create_text(670, 265, text=" 2 ")
   can23.create_text(280, 475, text=" 2 ")
if c2 < 1.0:
   can23.create_text(440, 337, text="%.2f " % c2)
   can23.create_text(670, 337, text=" 0 ")
   can23.create_text(430, 475, text=" 0 ")
if c2 > 1.0 and c2 < 3.0:
   can23.create_text(440, 337, text="%.2f " % c2)
   can23.create_text(670, 337, text=" 1 ")
   can23.create_text(430, 475, text=" 1 ")
if c2 > 3.0:
   can23.create_text(440, 337, text="%.2f " % c2)
   can23.create_text(670, 337, text=" 2 ")
   can23.create_text(430, 475, text=" 2 ")
```

```
if d2 < 0.5:
    can23.create_text(440, 180, text="%.2f " % d2)
    can23.create_text(670, 180, text=" 0 ")
    can23.create_text(130, 475, text=" 0 ")
if d2 > 0.5 and d2 < 3.0:
    can23.create_text(440, 180, text="%.2f " % d2)
    can23.create_text(670, 180, text=" 1 ")
    can23.create_text(130, 475, text=" 1 ")
if d2 > 3.0:
    can23.create_text(440, 180, text="%.2f " % d2)
    can23.create_text(670, 180, text=" 2 ")
    can23.create_text(130, 475, text=" 2 ")
if d2 < 0.1 and a2 > 0.1 and a2 < 1.0 and c2 < 1.0:
    can23.create_text(800, 475, text="%s " % ba1)
    t=("%s " % ba1)
elif d2 < 0.1 and a2 > 0.1 and a2 < 1.0 and c2 > 1.0 and c2 < 3.0:
    can23.create_text(800, 475, text="%s " % ba2)
    t=("%s " % ba2)
elif d2 < 0.1 and a2 > 1.0 and c2 > 1.0 and c2 < 3.0:
    can23.create_text(800, 475, text="%s " % ba3)
    t=("%s " % ba3)
elif d2 < 0.1 and a2 > 1.0 and c2 > 3.0:
    can23.create_text(800, 475, text="%s " % ba4)
    t=("%s " % ba4)
elif d2 < 0.1 and a2 < 0.1 and c2 < 1.0:
    can23.create_text(800, 475, text="%s " % ba5)
    t=("%s " % ba5)
elif d2 > 0.1 and d2 < 3.0 and a2 > 0.1 and a2 < 1.0 and c2 > 3.0:
    can23.create_text(800, 475, text="%s " % ba6)
    t=("%s " % ba6)
else:
    can23.create_text(800, 475, text="%s " % ba7)
    t=("%s " % ba7)
cursor.close()
db.close()
lo5.configure(state=DISABLED, background='silver')
with open('.\summary.txt', mode='a+') as fl:
        lines_of_text = ["  \n\n   DGA FOR POWER TRANSFORMERS\n", "
                        SCHOOL OF ELECTRICAL ENGINEERING\n", "
                        SURANAREE UNIVERSITY OF TECHNOLOGY\n", "--
                        "*20, "\n DIAGNOSIS SUMMARY\n", "--"*20]
    lines_of_text1 = ["\n    Method: ROGERS METHOD"]
    lines_of_text2 = ["\n    Time: %s " % time.strftime('%c')]
    lines_of_text3 = ["\n    Data ID: %d " % (kg)]
    lines_of_text4 = ["\n    Data => H2:%s "%(hydrogen), "C0:%s
              "%(carbon_monoxide), "C02:%s "%(carbon_dioxide), "N2:%s
```

```
            "%(nitrogen), "CH4:%s "%(methane), "C2H6:%s "%(ethane),
                "C2H4:%s "%(ethylene), "C2H2:%s "%(acetylene)]
    lines_of_text5 = ["\n        kVA:%s "%(power), "kV:%s "%(voltage),
    "Altitude:%s "%(altitude), "Temp:%s "%(oil_temperature), "Humidity:%s
    "%(humidity), "Location:%s "%(m_location), "Number:%s "%(m_number)]
    lines_of_text6 = ["\n\n    Analysis:"]
    lines_of_text7 = ["\n    RATIOS:-  ", " R1=CH4/H2=%.2f "%(a2), "
            R4=C2H6/CH4=%.2f "%(b2), " R5=C2H4/C2H6=%.2f "%(c2), "
            R2=C2H2/C2H4=%.2f "%(d2)]
    lines_of_text8 = ["\n    Fault:", " %s "%t]
    fl.writelines(lines_of_text)
    fl.writelines(lines_of_text1)
    fl.writelines(lines_of_text2)
    fl.writelines(lines_of_text3)
    fl.writelines(lines_of_text4)
    fl.writelines(lines_of_text5)
    fl.writelines(lines_of_text6)
    fl.writelines(lines_of_text7)
    fl.writelines(lines_of_text8)
    fl.writelines(["\n\n END OF DIAGNOSIS SUMMARY\n", "--"*45])
    fl.close()
    end_time = datetime.now()
```

- **Source codes for performing IEC ratio method.**

This section shows the source codes used to implement IEC ratio method. They include getting data from user, retrieving DGA information from the database, diagnosing and saving the diagnostic results.

```
def iecmethod():
    oceann = StringVar()
  def ferrmi():
    start_time = datetime.now()
    kgs=(oceann.get())
    kg=int(kgs)
    q1=can24.create_text(800,25, text="Your input: %d " % kg)
    sql=("SELECT*FROM new WHERE id='%d'" % (kg))
    db=MySQLdb.connect("127.0.0.1", "root", "mysql125", "dissolved")
    cursor=db.cursor()
    cursor.execute(sql)
```

```
results = cursor.fetchall()
fiel=cursor.description
for col in results:
    id = col[0]
    hydrogen = col[1]
    carbon_monoxide = col[2]
    carbon_dioxide = col[3]
    nitrogen = col[4]
    methane = col[5]
    ethane = col[6]
    ethylene = col[7]
    acetylene = col[8]
    power = col[9]
    voltage = col[10]
    oil_temperature = col[11]
    altitude = col[12]
    humidity = col[13]
    m_location = col[14]
    m_number = col[15]
can24.create_text(115,25, text="Your data for diagnosis", fill='blue')
can24.create_text(60,45, text="ID:%s "  % id)
can24.create_text(110,45, text="H2:%s " % hydrogen)
can24.create_text(165,45, text="CO:%s " % carbon_monoxide)
can24.create_text(225,45, text="CO2: %s " % carbon_dioxide)
can24.create_text(285,45, text="N2: %s " % nitrogen)
can24.create_text(350,45, text="CH4: %s " % methane)
can24.create_text(420,45, text="C2H6: %s " % ethane)
can24.create_text(495,45, text="C2H4: %s " % ethylene)
can24.create_text(575,45, text="C2H2: %s " % acetylene)
a=int(" %s " % hydrogen)
c=int(" %s " % methane)
d=int(" %s " % ethane)
e=int(" %s " % ethylene)
f=int(" %s " % acetylene)
a1=(c/float(a)),  c1=(e/float(d)), d1=(f/float(e))
a2=round(a1, 2), c2=round(c1, 2), d2=round(d1, 2)
baa = " Normal aging (No fault)"
baa1 = " Partial discharge of low energy density "
baa2 = " Partial discharge of high energy density "
baa3 = " Discharge of low energy "
baa4 = " Discharge of high energy "
baa5 = " Thermal fault of low temperature < 150 C "
baa6 = " Thermal fault of low temperature range 150 - 300 C "
baa7 = " Thermal fault of medium temperature range 300 - 700 C "
baa8 = " Thermal fault of high temperature > 700 C "
baa9 = " No prediction "
can24.create_text(650,44, text="(ppm)", fill='blue')
```

```
can24.create_text(83,75, text="Gas Ratios:", fill='blue')
can24.create_text(140,70, text="CH4", fill='black')
can24.create_line(123,77, 156, 77, width=1, fill='black')
can24.create_text(140,85, text="H2", fill='black')
can24.create_text(167,76, text="=", fill='black')
can24.create_text(190,76, text="%.2f " % a2)
can24.create_text(255,70, text="C2H4", fill='black')
can24.create_line(240,77, 272, 77, width=1, fill='black')
can24.create_text(259,85, text="C2H6", fill='black')
can24.create_text(281,76, text="=", fill='black')
can24.create_text(305,76, text="%.2f " % c2)
can24.create_text(365,70, text="C2H2", fill='black')
can24.create_line(350,77, 382, 77, width=1, fill='black')
can24.create_text(365,85, text="C2H4", fill='black')
can24.create_text(395,76, text="=", fill='black')
can24.create_text(420,76, text="%.2f " % d2)
if d2 < 0.1:
   can24.create_text(850, 180, text=" 0 ")
   can24.create_text(250, 475, text=" 0 ")
if d2 > 0.1 and d2 < 1.0:
   can24.create_text(850, 180, text=" 1 ")
   can24.create_text(250, 475, text=" 1 ")
if d2 >= 1.0 and d2 < 3.0:
   can24.create_text(850, 180, text=" 1 ")
   can24.create_text(250, 475, text=" 1 ")
if d2 > 3.0:
   can24.create_text(850, 180, text=" 2 ")
   can24.create_text(250, 475, text=" 2 ")
if a2 < 0.1:
   can24.create_text(850, 260, text=" 1 ")
   can24.create_text(350, 475, text=" 1 ")
if a2 > 0.1 and a2 < 1.0:
   can24.create_text(850, 260, text=" 0 ")
   can24.create_text(350, 475, text=" 0 ")
if a2 >= 1.0 and a2 < 3.0:
   can24.create_text(850, 260, text=" 2 ")
   can24.create_text(350, 475, text=" 2 ")
if a2 > 3.0:
   can24.create_text(850, 260, text=" 2")
   can24.create_text(350, 475, text=" 2 ")
if c2 < 0.1:
   can24.create_text(850, 342, text=" 0 ")
   can24.create_text(450, 475, text=" 0 ")
if c2 > 0.1 and c2 < 1.0:
   can24.create_text(850, 342, text=" 0 ")
   can24.create_text(450, 475, text=" 0 ")
if c2 >= 1.0 and c2 < 3.0:
```

```
        can24.create_text(850, 342, text=" 1 ")
        can24.create_text(450, 475, text=" 1 ")
    if c2 > 3.0:
        can24.create_text(850, 342, text=" 2 ")
        can24.create_text(450, 475, text=" 2 ")
    if d2 <= 0.1 and a2 >= 0.1 and a2 <= 1.0 and c2 <= 0.1:
        can24.create_text(770, 475, text="%s " % baa)
        t=("%s " % baa)
    elif d2 >= 0 and a2 <= 0.1 and c2 <= 1.0:
        can24.create_text(770, 475, text="%s " % baa1)
        t=("%s " % baa1)
    elif d2 >= 0.1 and d2 <= 3.0 and a2 <= 0.1 and c2 <= 1.0:
can24.create_text(770, 475, text="%s " % baa2)
        t=("%s " % baa2)
   elif d2 >= 0.1 and d2 <= 3.0 and a2 >= 0.1 and a2 <= 1.0 and c2 >= 1.0 and
   c2 < = 3.0:
        can24.create_text(770, 475, text="%s " % baa3)
        t=("%s " % baa3)
    elif d2 >= 0.1 and d2 <= 3.0 and a2 >= 0.1 and a2 <= 1.0 and c2 >= 3.0:
        can24.create_text(770, 475, text="%s " % baa3)
        t=("%s " % baa3)
    elif d2 >= 3.0 and a2 >= 0.1 and a2 <= 1.0 and c2 >= 1.0 and c2 <= 3.0:
        can24.create_text(770, 475, text="%s " % baa3)
        t=("%s " % baa3)
    elif d2 >= 3.0 and a2 >= 0.1 and a2 <= 1.0 and c2 >= 3.0:
        can24.create_text(770, 475, text="%s " % baa3)
        t=("%s " % baa3)
    elif d2 >= 0.1 and d2 <= 3.0 and a2 >= 0.1 and a2 <= 1.0 and c2 >= 3.0:
        can24.create_text(770, 475, text="%s " % baa4)
        t=("%s " % baa4)
    elif d2 <= 0.1 and a2 >= 0.1 and a2 <= 1.0 and c2 >= 1.0 and c2 <= 3.0:
        can24.create_text(770, 475, text="%s " % baa5)
        t=("%s " % baa5)
    elif d2 <= 0.1 and a2 >= 3.0 and c2 <= 1.0:
        can24.create_text(770, 475, text="%s " % baa6)
        t=("%s " % baa6)
    elif d2 <= 0.1 and a2 >= 3.0 and c2 >= 1.0 and c2 <= 3.0:
        can24.create_text(770, 475, text="%s " % baa7)
        t=("%s " % baa7)
    elif d2 <= 0.1 and a2 >= 3.0 and c2 >= 3.0:
        can24.create_text(770, 475, text="%s " % baa8)
        t=("%s " % baa8)
    else:
        can24.create_text(770, 475, text="%s " % baa9)
        t=("%s " % baa9)
    cursor.close()
    db.close()
```

```
lo3.configure(state=DISABLED, background='silver')
with open('.\summary.txt', mode='a+') as fl:
    lines_of_text = ["     \n\n   DGA FOR POWER TRANSFORMERS\n", "
            SCHOOL OF ELECTRICAL ENGINEERING\n", "   SURANAREE
            UNIVERSITY OF TECHNOLOGY\n", "--"*20, "\n  DIAGNOSIS
            SUMMARY\n", "--"*20]
    lines_of_text1 = ["\n    Method: IEC METHOD"]
    lines_of_text2 = ["\n    Time: %s " % time.strftime('%c')]
    lines_of_text3 = ["\n    Data ID: %d " % (kg)]
    lines_of_text4 = ["\n    Data => H2:%s "%(hydrogen), "C0:%s
            "%(carbon_monoxide), "C02:%s "%(carbon_dioxide), "N2:%s
            "%(nitrogen), "CH4:%s "%(methane), "C2H6:%s "%(ethane),
            "C2H4:%s "%(ethylene), "C2H2:%s "%(acetylene)]
    lines_of_text5 = ["\n           kVA:%s "%(power), "kV:%s "%(voltage),
            "Altitude:%s "%(altitude), "Temp:%s "%(oil_temperature),
            "Humidity:%s "%(humidity), "Location:%s "%(m_location),
            "Number:%s "%(m_number)]
    lines_of_text6 = ["\n\n    Analysis:"]
    lines_of_text7 = ["\n    RATIOS:-  ", " R2=C2H2/C2H4=%.2f "%(d2), "
            R1=CH4/H2=%.2f "%(a2), " R5=C2H4/C2H6=%.2f "%(c2)]
    lines_of_text8 = ["\n    Fault:", " %s"%t]
    fl.writelines(lines_of_text)
    fl.writelines(lines_of_text1)
    fl.writelines(lines_of_text2)
    fl.writelines(lines_of_text3)
    fl.writelines(lines_of_text4)
    fl.writelines(lines_of_text5)
    fl.writelines(lines_of_text6)
    fl.writelines(lines_of_text7)
    fl.writelines(lines_of_text8)
    fl.writelines(["\n\n END OF DIAGNOSIS SUMMARY\n", "--"*45])
    fl.close()
end_time = datetime.now()
can24.create_text(850,85, text=('Computing time: {}'.format(end_time –
            start_time)))
```

- **Source codes for performing Duval triangle method.**

This section shows the source codes used to implement Duval triangle method. They include getting data from user, retrieving DGA information from the database, diagnosing and saving the diagnostic results.

```
def duvaltriangle():
  start_time = datetime.now()
  oceann=StringVar()
  def fermmd():
     start_time = datetime.now()
     kgs=(oceann.get())
     kg=int(kgs)
     q1=can25.create_text(800,25, text="Your input: %d " % kg)
     sql=("SELECT*FROM new WHERE id='%d'" % (kg))
     db=MySQLdb.connect("127.0.0.1", "root", "mysql125", "dissolved")
     cursor=db.cursor()
     cursor.execute(sql)
     results = cursor.fetchall()
     fiel=cursor.description
     for col in results:
        id = col[0]
        hydrogen = col[1]
        carbon_monoxide = col[2]
        carbon_dioxide = col[3]
        nitrogen = col[4]
        methane = col[5]
        ethane = col[6]
        ethylene = col[7]
        acetylene = col[8]
        power = col[9]
        voltage = col[10]
        oil_temperature = col[11]
        altitude = col[12]
        humidity = col[13]
        m_location = col[14]
        m_number = col[15]
     can25.create_text(115,25, text="Your data for diagnosis", fill='blue')
     can25.create_text(650,44, text="(ppm)", fill='blue')
     can25.create_text(60,45, text="ID:%s "  % id)
     can25.create_text(110,45, text="H2:%s " % hydrogen)
     can25.create_text(165,45, text="CO:%s " % carbon_monoxide)
     can25.create_text(225,45, text="CO2: %s " % carbon_dioxide)
```

```
can25.create_text(285,45, text="N2: %s " % nitrogen)
can25.create_text(350,45, text="CH4: %s " % methane)
can25.create_text(420,45, text="C2H6: %s " % ethane)
can25.create_text(495,45, text="C2H4: %s " % ethylene)
can25.create_text(575,45, text="C2H2: %s " % acetylene)
a=int(" %s " % hydrogen), b=int(" %s " % carbon_monoxide)
c=int(" %s " % methane), d=int(" %s " % ethane),e=int(" %s " % ethylene)
f=int(" %s " % acetylene)
g1=c+e+f
pc=(c*100)/float(g1), pe=(e*100)/float(g1), pf=(f*100)/float(g1)
pc2 = round(pc, 2), pe2 = round(pe, 2), pf2 = round(pf, 2)
can25.create_text(140,70, text="[CH4 + C2H4 + C2H2] = %d " % g1)
can25.create_text(280,70, text="CH4: %.2f " % pc2)
can25.create_text(320,70, text="%")
can25.create_text(380,70, text="C2H4: %.2f " % pe2)
can25.create_text(425,70, text="%")
can25.create_text(480,70, text="C2H2: %.2f " % pf2)
can25.create_text(525,70, text="%")
pc3 = (pc2 * 2.5), pc32= round(pc3, 0), pc31= (pc2 * 5 * 0.866)
pc33= round(pc31, 0), pe3 = (pe2 * 2.5), pe31= round(pe3, 0)
pe32= (pe2 * 5 * 0.866),  pe33= round(pe32, 0), pf3 = (pf2 * 5)
pf31 = round(pf3, 0),  pf32 = (500 - pf31)/2
pf33 = round(pf32, 0), pf34 = (1.732 * pf32), pf35 = round(pf34, 0)
m1 = (550-pc33-(550-pc33))/(620-pc32-(120+pc32))
m2 = (550-(112+pe33))/((120+2*pe31)-(368+pe31))
m3 = ((550-pf35)-550)/((620-pf31-pf33)-(620-pf31))
xp = ((112+pe33)-(550 - pc33) + (m1*(120+pc32))-(m2*(368+pe31)))/(m1-m2)
yp = (m1*(xp-(120+pc32)) + (550 - pc33))
can25.create_line((120+pc32),(550 - pc33), xp, yp, width=2, fill='black')
can25.create_line((368+pe31),(112+pe33), xp, yp, width=2, fill='black')
can25.create_line((620-pf31),(550), xp, yp, width=2, fill='black', dash=(8, 8))
can25.create_oval((xp-5),(yp-5), (xp+5), (yp+5), fill='blue')
d = round(math.sqrt((235-120)**2 + (550-550)**2), 0)
e = round(math.sqrt((xp-(120+pc32))**2 + (yp-(550 - pc33))**2), 0)
f = round(math.sqrt((475-120)**2 + (550-550)**2), 0)
g = round(math.sqrt((375-365)**2 + (126-126)**2), 0)
if (e <= d and yp >= 173 and yp >= 225 and yp >= 277 and yp <= 550):
    t="D1"
    tr="DISCHARGE OF LOW ENERGY"
    can25.create_text(900,275, text="Fault !!!")
    asa=can25.create_line(650,225, 950, 225, width=50, fill='lightgreen')
    ad=can25.create_line(650,275, 950, 275, width=50, fill='red')
    ac=can25.create_line(650,325, 950, 325, width=50, fill='lightgreen')
    ae=can25.create_line(650,375, 950, 375, width=50, fill='lightgreen')
    af=can25.create_line(650,425, 950, 425, width=50, fill='lightgreen')
    ag=can25.create_line(650,475, 950, 475, width=50, fill='lightgreen')
    ah=can25.create_line(650,525, 950, 525, width=50, fill='lightgreen')
```

```
    can25.tag_lower(asa), can25.tag_lower(ad), can25.tag_lower(ac)
    can25.tag_lower(ae), can25.tag_lower(af), can25.tag_lower(ag)
  can25.tag_lower(ah)
elif (xp >= d and xp <= f and yp >=277  and yp <=550):
    t="D2"
    tr="DISCHARGE OF HIGH ENERGY"
    can25.create_text(900,325, text="Fault !!!")
    asa=can25.create_line(650,225, 950, 225, width=50, fill='lightgreen')
    ad=can25.create_line(650,275, 950, 275, width=50, fill='lightgreen')
    ac=can25.create_line(650,325, 950, 325, width=50, fill='red')
    ae=can25.create_line(650,375, 950, 375, width=50, fill='lightgreen')
    af=can25.create_line(650,425, 950, 425, width=50, fill='lightgreen')
    ag=can25.create_line(650,475, 950, 475, width=50, fill='lightgreen')
    ah=can25.create_line(650,525, 950, 525, width=50, fill='lightgreen')
    can25.tag_lower(asa), can25.tag_lower(ad), can25.tag_lower(ac)
    can25.tag_lower(ae), can25.tag_lower(af), can25.tag_lower(ag)
    can25.tag_lower(ah)
elif (xp <= g and yp <= 126):
    t="PD"
    tr="PARTIAL DISCHARGE"
    can25.create_text(900,225, text="Fault !!!")
    asa=can25.create_line(650,225, 950, 225, width=50, fill='red')
    ad=can25.create_line(650,275, 950, 275, width=50, fill='lightgreen')
    ac=can25.create_line(650,325, 950, 325, width=50, fill='lightgreen')
    ae=can25.create_line(650,375, 950, 375, width=50, fill='lightgreen')
    af=can25.create_line(650,425, 950, 425, width=50, fill='lightgreen')
    ag=can25.create_line(650,475, 950, 475, width=50, fill='lightgreen')
    ah=can25.create_line(650,525, 950, 525, width=50, fill='lightgreen')
    can25.tag_lower(asa), can25.tag_lower(ad), can25.tag_lower(ac)
    can25.tag_lower(ae), can25.tag_lower(af), can25.tag_lower(ag)
    can25.tag_lower(ah)
elif (xp >= 360 and xp <= 420 and yp <= 410 and yp >= 126):
    t="T1"
    tr="THERMAL FAULT T<300 C"
    can25.create_text(900,425, text="Fault !!!")
    asa=can25.create_line(650,225, 950, 225, width=50, fill='lightgreen')
    ad=can25.create_line(650,275, 950, 275, width=50, fill='lightgreen')
    ac=can25.create_line(650,325, 950, 325, width=50, fill='lightgreen')
    ae=can25.create_line(650,375, 950, 375, width=50, fill='lightgreen')
    af=can25.create_line(650,425, 950, 425, width=50, fill='red')
    ag=can25.create_line(650,475, 950, 475, width=50, fill='lightgreen')
    ah=can25.create_line(650,525, 950, 525, width=50, fill='lightgreen')
    can25.tag_lower(asa), can25.tag_lower(ad), can25.tag_lower(ac)
    can25.tag_lower(ae), can25.tag_lower(af), can25.tag_lower(ag)
    can25.tag_lower(ah)
elif (xp >= 420 and xp <= 495 and yp <= 353 and yp >= 204):
    t="T2"
```

```
        tr="THERMAL FAULT 300<T<700 C"
        can25.create_text(900,475, text="Fault !!!")
        asa=can25.create_line(650,225, 950, 225, width=50, fill='lightgreen')
        ad=can25.create_line(650,275, 950, 275, width=50, fill='lightgreen')
        ac=can25.create_line(650,325, 950, 325, width=50, fill='lightgreen')
        ae=can25.create_line(650,375, 950, 375, width=50, fill='lightgreen')
        af=can25.create_line(650,425, 950, 425, width=50, fill='lightgreen')
        ag=can25.create_line(650,475, 950, 475, width=50, fill='red')
        ah=can25.create_line(650,525, 950, 525, width=50, fill='lightgreen')
        can25.tag_lower(asa), can25.tag_lower(ad), can25.tag_lower(ac)
        can25.tag_lower(ae), can25.tag_lower(af), can25.tag_lower(ag)
      can25.tag_lower(ah)
    elif (xp >= 462 and xp <= 620 and yp <= 550 and yp >= 334):
        t="T3"
        tr="THERMAL FAULT T>700 C"
        can25.create_text(900,525, text="Fault !!!")
        asa=can25.create_line(650,225, 950, 225, width=50, fill='lightgreen')
        ad=can25.create_line(650,275, 950, 275, width=50, fill='lightgreen')
        ac=can25.create_line(650,325, 950, 325, width=50, fill='lightgreen')
        ae=can25.create_line(650,375, 950, 375, width=50, fill='lightgreen')
        af=can25.create_line(650,425, 950, 425, width=50, fill='lightgreen')
        ag=can25.create_line(650,475, 950, 475, width=50, fill='lightgreen')
        ah=can25.create_line(650,525, 950, 525, width=50, fill='red')
        can25.tag_lower(asa), can25.tag_lower(ad), can25.tag_lower(ac)
        can25.tag_lower(ae), can25.tag_lower(af), can25.tag_lower(ag)
        can25.tag_lower(ah)
    else:
        can25.create_text(900,375, text="Fault !!!")
        t="DT"
        tr="MIXED FAULT"
        asa=can25.create_line(650,225, 950, 225, width=50, fill='lightgreen')
        ad=can25.create_line(650,275, 950, 275, width=50, fill='lightgreen')
        ac=can25.create_line(650,325, 950, 325, width=50, fill='lightgreen')
        ae=can25.create_line(650,375, 950, 375, width=50, fill='red')
        af=can25.create_line(650,425, 950, 425, width=50, fill='lightgreen')
        ag=can25.create_line(650,475, 950, 475, width=50, fill='lightgreen')
        ah=can25.create_line(650,525, 950, 525, width=50, fill='lightgreen')
        can25.tag_lower(asa), can25.tag_lower(ad), can25.tag_lower(ac)
        can25.tag_lower(ae), can25.tag_lower(af), can25.tag_lower(ag)
        can25.tag_lower(ah)
    with open('.\summary.txt', mode='a+') as fl:
        lines_of_text = ["     \n\n   DGA FOR POWER TRANSFORMERS\n", "
                SCHOOL OF ELECTRICAL ENGINEERING\n", "   SURANAREE
                UNIVERSITY OF TECHNOLOGY\n", "--"*20, "\n DIAGNOSIS
                SUMMARY\n", "--"*20]
        lines_of_text1 = ["\n    Method: DUVAL TRIANGLE"]
        lines_of_text2 = ["\n    Time: %s " % time.strftime('%c')]
```

```
    lines_of_text3 = ["\n    Data ID: %d " % (kg)]
    lines_of_text4 = ["\n    Data => H2:%s "%(hydrogen), "C0:%s
                "%(carbon_monoxide), "C02:%s "%(carbon_dioxide), "N2:%s
                "%(nitrogen), "CH4:%s "%(methane), "C2H6:%s "%(ethane),
                "C2H4:%s "%(ethylene), "C2H2:%s "%(acetylene)]
    lines_of_text5 = ["\n kVA:%s "%(power), "kV:%s "%(voltage), "Altitude:%s
                        "%(altitude), "Temp:%s "%(oil_temperature), "Humidity:%s
                        "%(humidity), "Location:%s "%(m_location), "Number:%s
                        "%(m_number)]
    lines_of_text6 = ["\n\n    Analysis:"]
    lines_of_text7 = ["\n    [CH4 + C2H4 + C2H2] =%d" %(g1), " CH4(percent)
                        =%.2f "%(pc2), " C2H4(percent) =%.2f "%(pe2), "
                        C2H2(percent) =%.2f "%(pf2)]
    lines_of_text8 = ["\n    Fault:", "  %s "%t]
    lines_of_text9 = ["\n    Interpretation:", "  %s "%(tr)]
    fl.writelines(lines_of_text)
    fl.writelines(lines_of_text1)
    fl.writelines(lines_of_text2)
    fl.writelines(lines_of_text3)
    fl.writelines(lines_of_text4)
    fl.writelines(lines_of_text5)
    fl.writelines(lines_of_text6)
    fl.writelines(lines_of_text7)
    fl.writelines(lines_of_text8)
    fl.writelines(lines_of_text9)
    fl.writelines(["\n\n END OF DIAGNOSIS SUMMARY\n", "--"*45])
    fl.close()
end_time = datetime.now()
can25.create_text(850, 85,
                    text=('Computing time: {}'.format(end_time -start_time)))
cursor.close()
db.close()
```

APPENDIX C

ESTABLISHMENT OF REMAINED LIFE MODEL

This section shows the results of breakdown strength test done from twenty-eight samples of mineral insulating oil available in Suranaree University of Technology, Nakhon Ratchasima, Thailand.

C.1 Analysis of breakdown strength

Valid test results of breakdown strength with the percentage of range to mean less than or equal to 92 % and standard deviation less than 4% are given in Table C1. Obtained values of breakdown strength were seen to range between 20 kV to 60 kV. Mean values of valid test results were plotted against individual dissolved gas (see section 6.1) for investigating the existing relationship between them.

Table C.1 Results of breakdown strength test

No	**Test results**						**R**	$\overline{X}$	$\%\frac{R}{\overline{X}}$	**σ**
1	32.4	32.4	30.8	29.1	27.5	27.2	5.20	29.90	17.39	2.32
2	28.1	37.4	35.3	31.4	25.4	24.5	12.9	30.35	42.50	5.27
3	30.1	37.4	40.2	38.1	36.3	36.2	8.10	36.72	22.06	2.69
4	41.4	37.8	37.3	32.6	30.7	33.0	6.10	35.47	17.20	4.03
5	30.6	33.2	31.8	31.3	29.1	27.1	6.10	30.52	19.99	2.15
6	25.8	20.8	22.6	22.4	21.9	18.5	7.30	22.00	33.18	2.39
7	24.1	24.6	23.7	19.3	17.7	15.0	9.60	20.73	46.30	3.98
8	46.4	45.1	34.0	28.0	26.9	24.8	21.6	34.20	63.16	9.46
9	48.5	44.7	40.4	41.7	40.1	39.7	8.80	42.52	20.70	3.45
10	31.7	31.4	31.3	28.8	28.5	28.2	8.70	33.32	26.11	3.25
11	31.8	31.0	29.3	31.2	34.2	28.9	5.30	31.07	17.06	1.91
12	40.6	55.5	56.5	54.5	57.0	57.5	5.50	56.83	9.68	1.89
13	56.2	51.5	60.7	57.9	60	57.5	9.20	57.30	16.06	3.29
14	37.7	40.9	33.9	39.7	43.2	38.4	9.30	38.97	23.87	3.16
15	42.8	48.2	42.4	47.8	41.9	32.9	15.3	42.67	35.86	5.53

Table C.1 Results of breakdown strength test (Continued)

16	33.1	32.0	34.6	31.9	33.9	31.0	3.60	32.75	10.99	1.36
17	45.7	49.2	49.8	47.4	47.2	53.4	7.70	48.78	15.78	2.70
18	43.8	42.0	44.0	43.4	43.0	51.7	9.70	44.65	21.72	3.53
19	36.3	30.9	30.6	29.3	28.0	28.0	8.30	30.52	27.20	3.09
20	34.0	29.3	28.7	26.0	25.0	32.0	9.00	29.17	30.86	3.44
21	60.0	57.0	56.4	46.7	54.5	60.0	13.3	55.77	23.85	4.93
22	35.4	29.6	34.2	32.1	39.2	27.9	11.3	33.07	34.17	4.10
23	28.4	32.7	27.3	28.8	27.4	27.0	5.70	28.60	19.93	2.13
24	53.4	53.0	59.0	60.0	58.5	60.0	7.00	57.32	12.21	3.24
25	43.3	42.5	43.9	45.8	41.2	38.1	7.70	42.47	18.13	2.63
26	27.4	28.8	24.1	24.9	24.5	21.6	7.20	25.22	28.55	2.55
27	45.0	40.1	45.2	47.0	42.0	38.1	8.90	42.90	20.75	3.41
28	39.0	37.2	37.3	38.9	45.5	34.8	10.7	38.78	27.59	3.63

C.2 Relationship between DGA and Breakdown strength

Figure C.1 shows the relationship between individual dissolved gasses and mean values of breakdown strength of tested samples. From Figure C.1 it was observed that with zero ppm of H_2, CH_4, C_2H_6, C_2H_4, and C_2H_2 both lower and higher values of breakdown strength were observed. Similar values of breakdown strength were also obtained in high levels of these gasses.

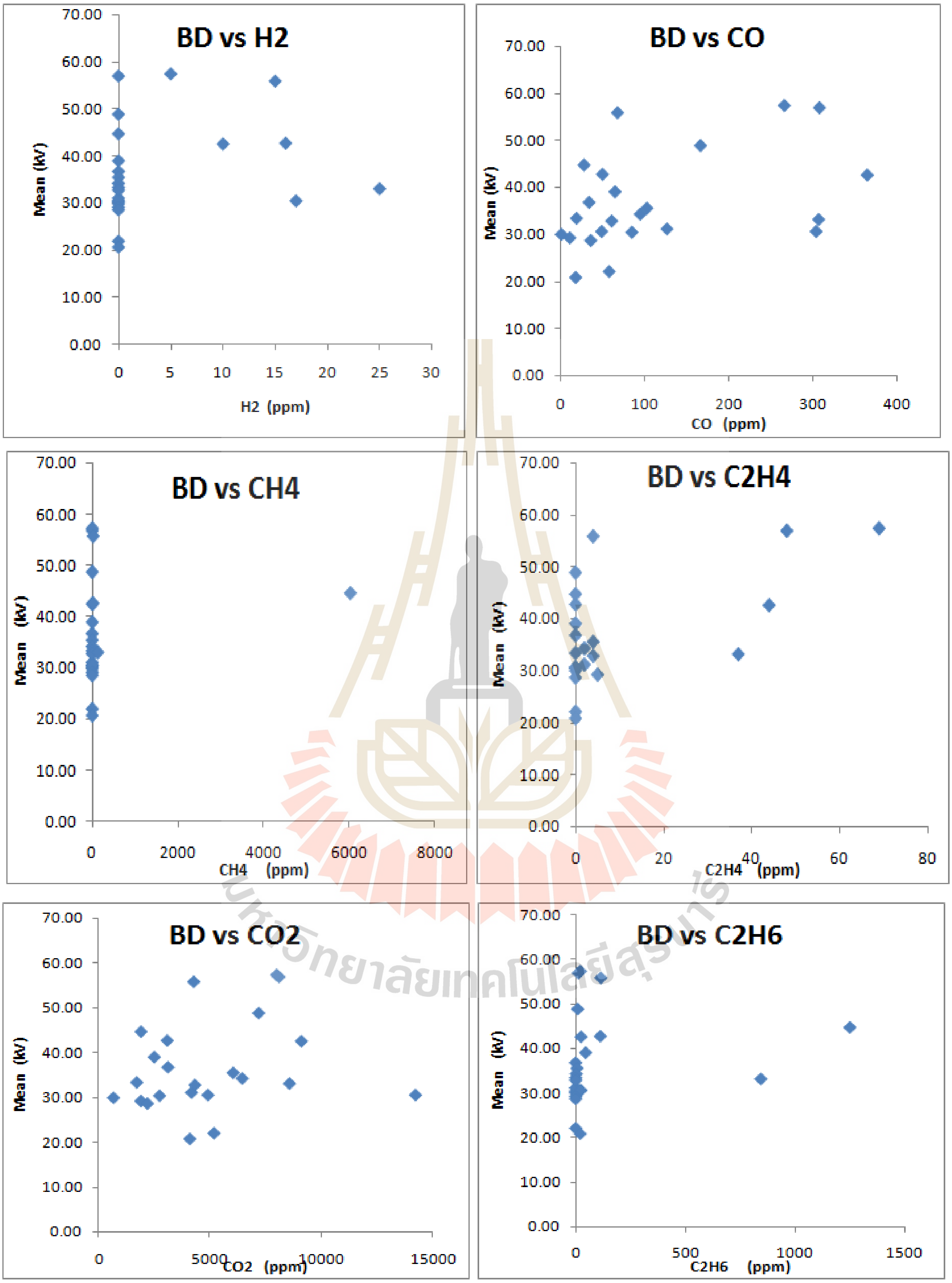


Figure C.1 Relationship between dissolved gasses and breakdown strength

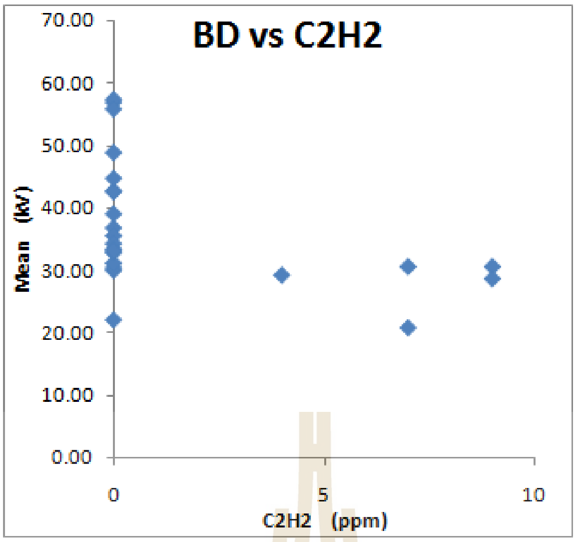


Figure C.1 Relationship between dissolved gasses and breakdown strength (Continued)

For CO and CO_2, the lower and higher values showed no clear relationships with the increase or decrease of dissolved gasses. This was quite complicated when interpreting the quality of insulating with dissolved gasses.

C.3 Estimation of transformer's remained life

Estimating transformer's remained life need to define an existing relationship between two parameters. A strong ambiguity existed between H_2, CH_4, C_2H_6, C_2H_4 and C_2H_2 and breakdown strength values that gave unclear interpretations of the existing relationships. CO and CO_2 showed little ambiguity although failure occurred in interpreting the existing relationships. Since estimation of transformers remained life depended on the existing relationships, the models were not successfully established. Mineral oil served over the long service life was mostly recommended in future research.

APPENDIX D

LIST OF SAMPLES

The following is the list of samples of mineral insulating oil as shown in Table D.1. They have been listed in the order of their testing. These samples were collected from twenty-eight in-service transformers available in Suranaree University of Technology, Nakhon Ratchasima, Thailand.

Table D.1 List of samples of mineral insulating oil

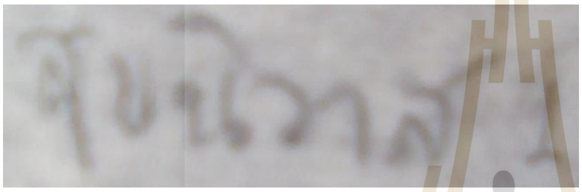
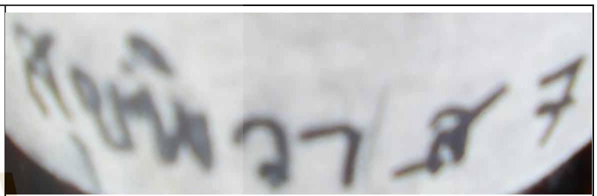
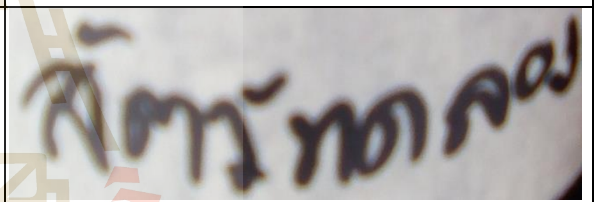
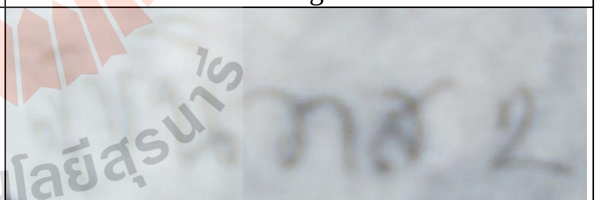
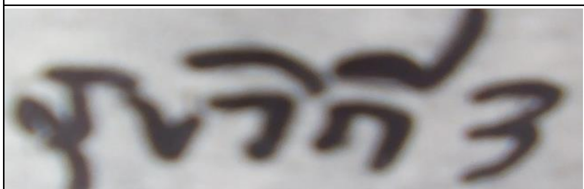
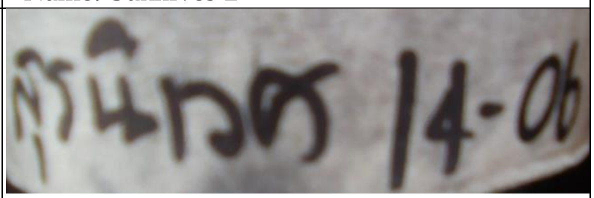
 SAMPLE NO. 1	 SAMPLE NO. 5
Name: Suknives 1	Name: Suknives 7
 SAMPLE NO. 2	 SAMPLE NO. 6
Name: Sukkhawithi 2	Name: Animals Building
 SAMPLE NO. 3	 SAMPLE NO. 7
Name: Suknives 3	Name: Suknives 2
 SAMPLE NO. 4	 SAMPLE NO. 8
Name: Sukkhawithi 3	Name: Suranives 14

Table D.1 List of samples of mineral insulating oil (Continued)

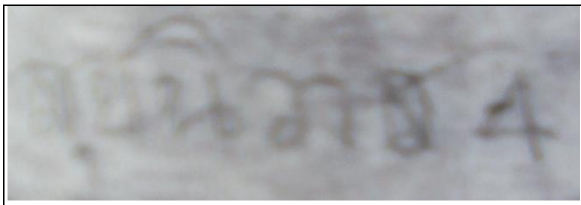
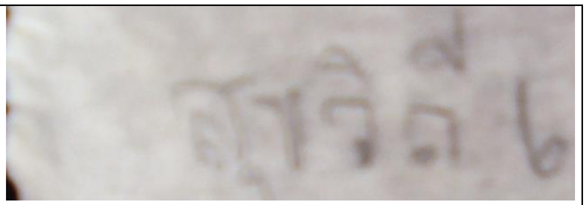
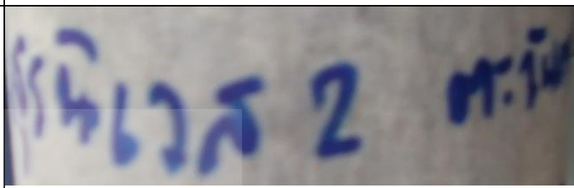
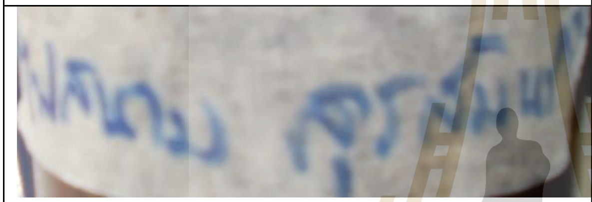
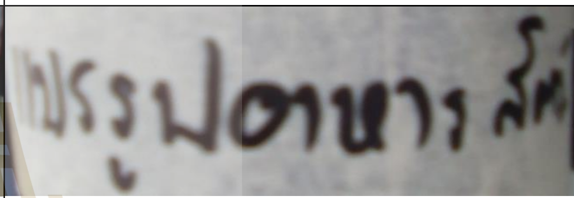
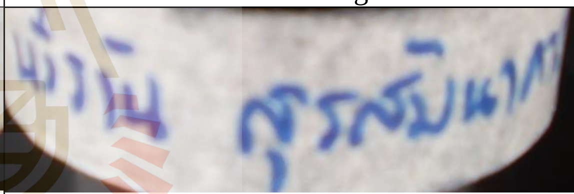
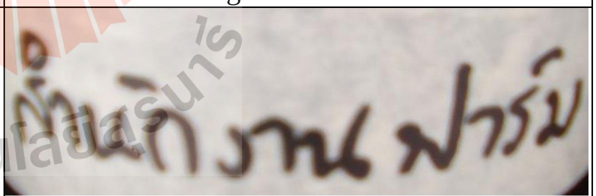
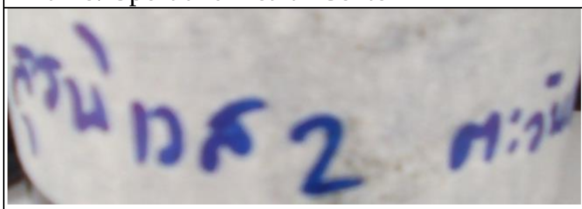
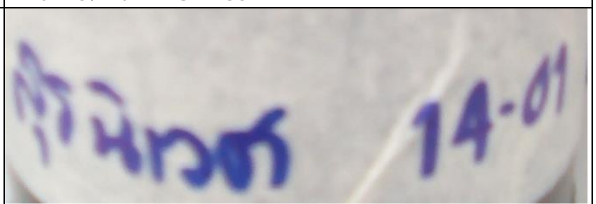
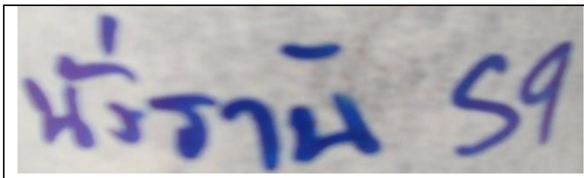
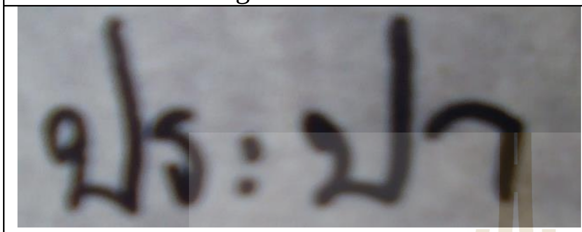
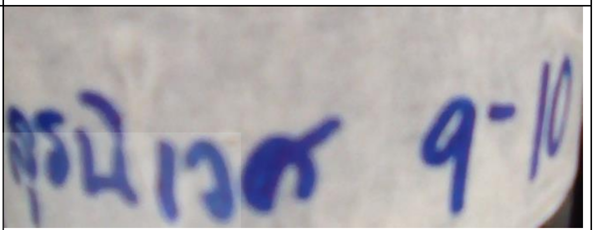
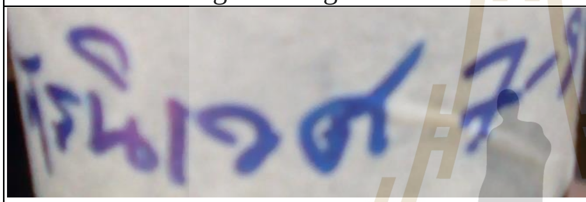
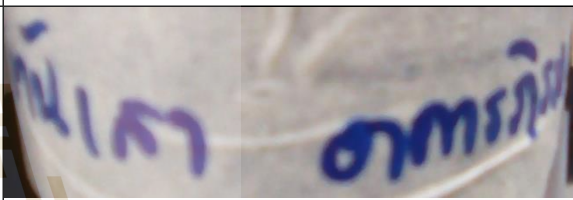
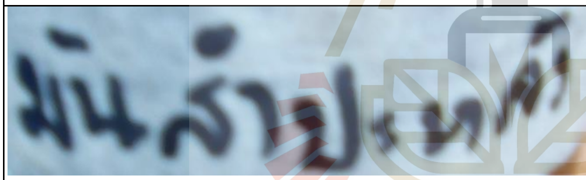
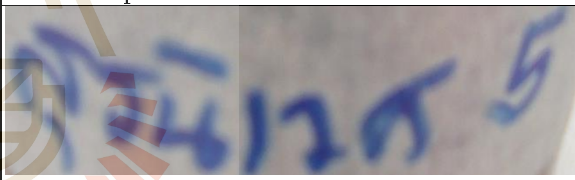
 SAMPLE NO. 9	 SAMPLE NO. 10
Name: Suknives 4	Name: Sukkhawithi 6
 SAMPLE NO. 11	 SAMPLE NO. 12
Name: Suknives 8	Name: Suranives 2
 SAMPLE NO. 13	 SAMPLE NO. 14
Name: Lawn at Surasamanakhan Hotel	Name: Animal Feed Building
 SAMPLE NO. 15	 SAMPLE NO. 16
Name: Football Field	Name: Scaffolding Surasamanakhan Hotel
 SAMPLE NO. 17	 SAMPLE NO. 18
Name: Sport and Health Center	Name: Farm Office
 SAMPLE NO. 19	 SAMPLE NO. 20
Name: Suranives 2	Name: Suranives 14

Table D.1 List of samples of mineral insulating oil (Continued)

 SAMPLE NO. 21	 SAMPLE NO. 22
Name: Scaffolding Suranives 9	Name: Suranives 11-12
 SAMPLE NO. 23	 SAMPLE NO. 24
Name: Plumbing Building	Name: Suranives 9-10
 SAMPLE NO. 25	 SAMPLE NO. 26
Name: Suranives 7-8	Name: Sport and Health Center
 SAMPLE NO. 27	 SAMPLE NO. 28
Name: Cassava Research Building	Name: Suranives 5

BIOGRAPHY

Mr. Pius Victor was born on Wednesday, 15 October 1986 in Dar es Salaam Region, Tanzania. He received his Bachelor degree in Electrical Engineering from Dar es Salaam Institute of Technology (DIT) in 2013. He then continued his Master Degree in the school of Electrical Engineering, Institute of Engineering, Suranaree University of Technology, Nakhon Ratchasima, Thailand (March 2015-Feb, 2017). His fields of interest include High Voltage systems design and monitoring, laboratory, and field system programming. While he was pursuing his Master's degree study, he presented three international conference papers entitled; "*Implementation of Computer Based Software for Oil-Immersed power transformer Conditions Monitoring via Dissolved Gas Analysis Results*" in the IEEJ P&ES-IEEE PES Thailand Joint Symposium on Advanced Technology in Power Systems, Bangkok, Thailand, "*Training of Electrical Engineers for Transformer Condition Based Maintenance through Computer Based DGA*" in the International Conference for Electrical Engineers, Okinawa, Japan and " *A Study of Condition Monitoring of Oil-Immersed power transformer via Results of Dissolved Gas Analysis and Breakdown Strength*" in the IEEJ P&ES-IEEE PES Thailand Joint Symposium on Advanced Technology in Power Systems, Bangkok, Thailand. Also, he published one international journal paper entitled of "*Effects of Distribution Approximations to Expanded Uncertainty of Breakdown Strength of Mineral Insulating Oil in IEC 60156*" in International Review of Electrical Engineering (I.R.E.E).