SAKARYA UNIVERSITY OF APPLIED SCIENCES
INSTITUTE OF GRADUATE EDUCATION


# A POWER LINE FAULT DETECTION SYSTEM DESIGN BASED ON A HYBRID BILSTM NETWORK

**M.Sc THESIS**


**Mohammed I. M. EISHORAFA**

| | | |
|---|---|---|
| **Department** | **:** | **ELECTRICAL AND ELECTRONICS ENGINEERING** |
| **Supervisor** | **:** | **Prof. Dr. İhsan PEHLİVAN** |



**May 2023**

**SAKARYA UNIVERSITY OF APPLIED SCIENCES**
**GRADUATE EDUCATION INSTITUTE**

# A POWER LINE FAULT DETECTION SYSTEM DESIGN BASED ON A HYBRID BILSTM NETWORK

**M.Sc THESIS**

**Mohammed I. M. EISHORAFA**

**Department : ELECTRICAL AND ELECTRONICS ENGINEERING**

**This thesis was accepted by the commitee given below on 17/05/2023 by unanimously of votes in Institute of Graduate Education, Sakarya University of Applied Sciences.**

| COMMITTEE | STATUS |
|---|---|
| Head of Committee: Prof. Dr. İhsan PEHLİVAN | SUCCESFULL |
| Member: Dr. Öğr. Üyesi Ali Furkan KAMANLI | SUCCESFULL |
| Member: Doç. Dr. Ünal ÇAVUŞOĞLU | SUCCESFULL |

# DECLARATION STATEMENT

I confirm that all of the data presented in this thesis were collected by me in compliance with academic regulations. The results and any visual or written information are presented according to ethical and academic standards. The data has not been manipulated, and any references or use of the works of others are properly cited and in accordance with scientific norms. I affirm that this data has not been used in any other thesis work. Additionally, I have obtained official approval from an ethics committee. I am aware of the potential legal consequences that may arise if any part of this statement is found to be untrue

Mohammed I. M. EISHORAFA

17/05/2023

## ACKNOWLEDGEMENT

To Allah, we firstly admit all our praise and thanks for his gracing and giving which enabled us to finish this work as perfect as it is. We also send our pleasing and peace upon his prophet Mohammed the mercy and teacher for the mankind.

Our thanks continued to our parents for their effort and care, to our lecturers for their support and instructions, to friends and all those who helped us to end this work. We would like to extend our sincerest gratitude to our institution, as well as to all the doctors and teachers who have supported us throughout our academic journey. We would also like to express our heartfelt appreciation to Prof. Dr. İhsan PEHLİVAN for his exceptional guidance and mentorship.

# CONTENTS

## ABBREVIATIONS

| | |
|---|---|
| AC | : Alternating Current |
| ADC | : Analog to Digital Converter |
| AI | : Artificial Intelligence |
| ANN | : Artificial Neural Network |
| BiLSTM | : Bidirectional Long Short-Term Memory |
| CNN | : Convolutional Neural Network |
| CPU | : Central Processing Unit |
| DC | : Direct Current |
| DNA | : Deoxyribonucleic Acid |
| ECG | : Electrocardiography |
| EEG | : Electroencephalography |
| FN | : False Negative |
| FP | : False Positiv |
| GPS | : Global Positioning System |
| GPU | : Graphics Processing Unit |
| GIS | : Gas Insulated Substation |
| HIF | : High-Impedance Faults |
| MCC | : Matthews Correlation Coefficient |
| ML | : Machine Learning |
| MLP | : Multi-layer Perceptrons |
| NLP | : Natural Language Processing |
| NO | : Nitric Oxide |
| O3 | : Ozone |
| PD | : Partial Discharge |
| PMSM | : Permanent-Magnet Synchronous Machine |
| PVC | : Polyvinyl Chloride |
| RNN | : Recurrent Neural Network |

| | |
|---|---|
| TF | : TensorFlow |
| TN | : True Negative |
| TP | : True Positive |

## SYMBOLS

| | |
|---|---|
| D | : Katz fractal dimension |
| M | : Mean |
| $S_n$ | : Standard deviation |
| $S_{Top}$ | : Standard deviation top |
| $S_{Bottom}$ | : Standard deviation bottom |
| P | : Petrosian fractal dimension |
| $X_{Normalized}$ | : Min-Max scaling |
| Y | : Embedded matrix |
| Z | : Standard scaling |

# LIST OF TABLES

## LIST OF FIGURES

## A POWER LINE FAULT DETECTION SYSTEM DESIGN BASED ON A HYBRID BILSTM NETWORK


## ABSTRACT

High and medium-voltage transmission lines extend for many miles to provide cities with electricity to run the facilities and homes. Because of the wide range of distance, manual checking of power lines for damage gets infeasible and extravagant. In addition, The occurrence of partial discharge (PD), which is caused by a malfunction in the supporting and installing insulators of power lines, can make the power lines more vulnerable to malfunction. These lines may also be harmed by solid objects colliding with them. Leaving them unrepaired, PD can damage equipment to the point that it stops functioning completely. The challenge in this thesis is to detect partial discharges so that repairs can be made before any permanent damage occurs. Bidirectional Long short-term memory (BiLSTM) deep neural network approach has been proposed to detect the PD. In this thesis, BiLSTM neural network was designed to provide a pre-trained model that can be used to detect the practical discharge phenomenon in electric power transmission lines in Sakarya or Turkey in general by using a transfer learning approach for the pretrained AI model which was trained based on a dataset that is in form of signals acquired from these power lines with a new meter designed at the ENET Centre at VSB-Technical University of Ostrava. The dataset used to train the neural network model was sourced from an environment outside of Turkey. Designing effective classifiers depending on this data will make continuously monitoring power lines and detecting faults possible. To attain an efficient classifier, statistical techniques related to central tendency, statistical dispersion and also complex features extraction techniques related to entropy and fractality were utilized to preprocess the dataset because the signal data comes from the real environment, not a lab, and they contain a lot of background noise. The significant difference in the spectrum of noise and quality of PDs poses a challenge for achieving accurate and reliable classification, which is still an ongoing issue.

Finally, in this research, the idea of merging two systems or two things was followed, as each of them is characterized by a specific thing. To take advantage of or combine the characteristics of the two systems and obtain a system that is superior to each of them, two neural networks were designed and named Model_160 and Model_320, each of which receives different data inputs, and then a hybrid model was designed by linking the two models, and a higher accuracy (0.975) and MCC (0.79) were obtained from it. Moreover, this hybrid model outperforms Kaggle winner and other models that used the same data. Thus, the idea of merging achieved the desired goal.



Keywords: BiLSTM, PD, AI, MCC, Deep learning, Neural network.

# HİBRİT BİLSTM AĞI ESASLI BİR GÜÇ HATTI HATA TESPİT SİSTEMİ TASARIMI

## ÖZET

Yüksek ve orta gerilim iletim hatları, şehirlere tesisleri ve evleri çalıştırmak için elektrik sağlamak üzere kilometrelerce uzanır. Geniş mesafe aralığından dolayı, elektrik hatlarının hasara karşı manuel olarak kontrol edilmesi olanaksız ve abartılı hale gelir. Ayrıca, elektrik hatlarının destekleyici ve montaj izolatörlerindeki bir arızadan kaynaklanan kısmi boşalma (PD) oluşumu, elektrik hatlarını arızaya karşı daha savunmasız hale getirebilir. Bu hatlar, katı cisimlerin çarpmasıyla da zarar görebilir. Onları tamir etmeden bırakan PD, ekipmana tamamen çalışmayı durduracak kadar zarar verebilir. Bu tezdeki zorluk, herhangi bir kalıcı hasar oluşmadan önce onarımların yapılabilmesi için kısmi deşarjları tespit etmektir. PD'yi saptamak için çift yönlü Uzun kısa süreli bellek (BiLSTM) derin sinir ağı yaklaşımı önerilmiştir. Bu tezde, BiLSTM sinir ağı, önceden eğitilmiş AI modeli için bir transfer öğrenme yaklaşımı kullanarak Sakarya veya Türkiye genelindeki elektrik iletim hatlarındaki pratik deşarj olayını tespit etmek için kullanılabilecek önceden eğitilmiş bir model sağlamak üzere tasarlanmıştır. VSB-Technical University of Ostrava'daki ENET Merkezinde tasarlanan yeni bir sayaç ile bu güç hatlarından elde edilen sinyaller biçimindeki bir veri kümesine dayalı olarak eğitilmiştir. Sinir ağı modelini eğitmek için kullanılan veri seti, Türkiye dışındaki bir ortamdan alınmıştır. Bu verilere bağlı olarak etkili sınıflandırıcılar tasarlamak, enerji hatlarının sürekli olarak izlenmesini ve arızaların tespit edilmesini mümkün kılacaktır. Etkili bir sınıflandırıcı elde etmek için, merkezi eğilim, istatistiksel dağılım ile ilgili istatistiksel teknikler ve ayrıca entropi ve fraktalite ile ilgili karmaşık öznitelik çıkarma teknikleri veri setini ön işlemek için kullanıldı çünkü sinyal verileri bir laboratuvardan değil gerçek ortamdan geliyor ve bunlar bir çok fazla arka plan gürültüsü. Gürültü spektrumundaki önemli fark ve PD'lerin kalitesi, hala devam eden bir sorun olan doğru ve güvenilir sınıflandırma elde etmek için bir zorluk teşkil etmektedir.Son olarak, bu araştırmada, her biri belirli bir şeyle karakterize edildiğinden, iki sistemi veya iki şeyi birleştirme fikri izlendi. İki sistemin özelliklerinden yararlanmak veya özelliklerini birleştirmek ve her birinden üstün bir sistem elde etmek için Model_160 ve Model_320 olarak adlandırılan ve her biri farklı veri girdileri alan iki sinir ağı tasarlanmış ve daha sonra hibrit bir model tasarlanmıştır. iki modeli birbirine bağlayarak daha yüksek bir doğruluk (0.975) ve MCC (0.79) elde edildi. Üstelik bu hibrit model, Kaggle kazananı ve aynı verileri kullanan diğer modellerden daha iyi performans gösteriyor. Böylece birleştirme fikri istenilen amaca ulaşmış oldu.

Anahtar Kelimeler: BiLSTM, PD, AI, Derin öğrenme, Sinir ağı.

# CHAPTER 1. INTRODUCTION

Monitoring electrical connections is very important to maintain the supply of energy to homes, establishments, and institutions especially hospitals when any possible malfunction is detected. In addition to this, any defect in power transmission lines can lead not only to power outages but also leads to fire and breakdown in high or medium voltage installations, stations Power distribution, or also in electrical transformers. Although there are several factors that can contribute to the breakdown of high or medium-power transmission lines, this study will focus on the issue of partial discharge. By and large, Partial discharge (PD) is a result of concentrations of electrical stress has been inside the insulation or on the surface[1].

The primary cause of failure in high voltage equipment is typically attributed to insulator breakdown in transmission lines, whether they are overhead or underground. PD occurs at about 84% and 89% in the transformer and insulating cables respectively. PD can take place in different mediums such as gaseous, liquid, or solid insulating. This phenomenon occurs when an appropriate electrical stress amount exits in insulating parts [2]. Moreover, the breakdown of the insulators isn't immediately happened by PD but it remains the main factor that ultimately results in the failure of electrical equipment [3].

Partial discharge gives a hint or sign that the insulating materials are not in good condition or also indicate the onset of potential equipment failure. In addition to this, the deterioration of the insulating materials may further reach a stage that is unable to withstand electrical stress which leads to flashing and the occurrence of final failure. These are often sudden and catastrophic in high voltage and sometimes 'medium' voltage situations, resulting in major damage and interruptions in supply. As a result, this phenomenon created a challenge for electrical engineers, in particular, to work on detecting it by conducting the necessary tests without cutting off the power. That is online monitoring for electrical connections. Following the examination results, appropriate maintenance measures are taken, such as replacing insulators or other electrical components that are at risk of failure, to prevent potential issues in the future.

Partial Discharge (PD) is divided into three types:

1-Internal Partial Discharge, occurring inside insulation.

2- Surface Partial Discharge, tracking across insulation.

3- Corona Partial Discharge [4].

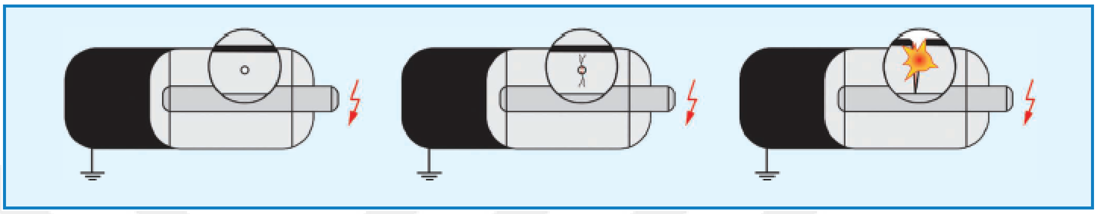


Figure 1.1 : Partial discharge (PD) is a result of concentrations of electrical stress has been inside the insulation or on the surface [1].

Artificial intelligence (AI) encompasses a broad array of domains and involves a diverse range of terms and skills like learning, reasoning, and self-correction that aim to design or make the computer act like a human ability to learn and infer useful and required knowledge from data to solve the problem, for instance. [5]. Artificial intelligence is not a new thing but many years ago, it was founded. In 1956, It was suggested as an academic field of study and in the years since has faced a lot of hopefulness stages [6]. When it comes to repetitive tasks, classifying large amounts of data or predicting the pattern of data flow in the future based on previous data based on it, such as classifying spam messages, classifying some diseases such as cancer, or predicting stock price, here artificial intelligence can perform tasks better than humans.

AI techniques can deal, process with data and also predict in a manner that is faster and more precise than humans. As a consequence, AI technologies, such as Machine Learning (ML) which emulates human intelligence, and deep learning neural networks, are being progressively and rapidly improved and developed. Moreover, ML uses computational algorithms to make the computer learn and understand the surrounding environment [7]. It is unfeasible to develop traditional algorithms to perform tasks such as email filtering, computer vision and speech and face recognition, so machine learning algorithms are used for such situations. As machine learning algorithms have a general shape and structure, they generally depend on a previous set of data about the problem or application that you want to solve for study, but traditional algorithms differ completely from one problem or application to another [8]. Deep learning has the advantage that it allows dealing with large data, whether it is raw or preprocessed and extracting the useful knowledge for detection or classification purposes. The deep learning approach follows the same structure of perceptron but relay on more deep layers that's why it is called deep learning. [9]. As shown in figure 2.1, the area of intersection between many different sectors is the thesis work area.

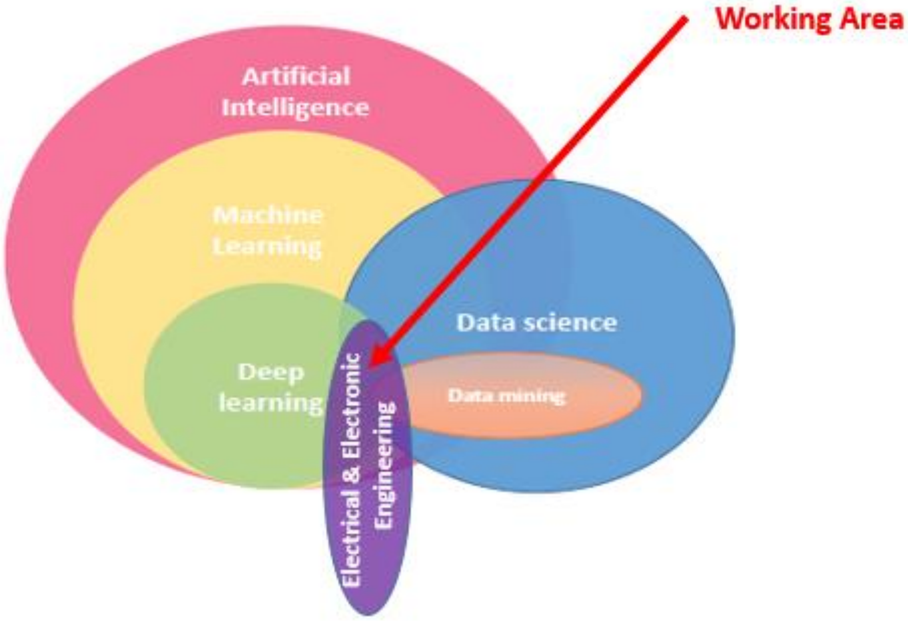



Figure 1.2 : Illustrates the interconnection between different concepts in the field of artificial intelligence, including machine learning (ML), deep learning, data science, and data mining [10].

Over the past few years, deep neural networks have gained widespread recognition and achieved remarkable success in pattern recognition [11]. There is a type of neural network known as Multilayer Percerptrons and it is not more than nonlinear regression but it is used a lot because it is considered a good choice for many problems [12]. While there are several varieties of deep neural networks, they are not equally well-known or widely used. There are four types which got a great reputation and are frequently used Convolutional Neural Networks (CNN) (widespread for dealing with images), Recurrent Neural Networks (RNN)(popular with dealing with time series data) and Long short-term memory (LSTM)(popular with dealing with time-series data but has memory ), Multi-Layer Perceptrons (MLP)[13]. Deep neural networks are used in many medical fields such as diabetes diagnosis, engineering in various sectors and even in agriculture[14], deep neural networks are used for Classification purposes in Android, IOS, and malware applications [15].

Accordingly, The phenomenon of partial discharge can be effectively monitored online without interrupting power supply, by leveraging artificial intelligence (AI) and deep learning techniques. This gives an advantage over other solutions to detect when this phenomenon occurs. The implementation of this thesis involved four key stages. Firstly, a suitable dataset was acquired for training and testing the deep neural network. Next, the dataset underwent a pre-processing stage to ensure its suitability for training. The artificial model was then designed and trained to enable accurate classification. Finally, the model was tested using data from Kaggle that underwent the same pre-processing stages as the training dataset.

## 1.1. Thesis Objective

Based on what was mentioned above, this problem is big and can be a difficult job to manually monitor it periodically using sensors and specialized devices. So the science of artificial intelligence will be used to contribute to solving this dilemma and discovering the phenomenon of PD in the supply lines of high or medium voltage for the three phases by designing an AI model to predict and classify if the phenomenon was formed or not. Accordingly, the appropriate maintenance procedure will be taken to address the problem without loss of energy supply. This means that the process of monitoring the power lines will be online, by taking readings from each phase, and then this data will be subjected to processing operations, and then it will be given to the artificial intelligence system.

## 1.2. Literature Review

The interest in measuring and locating the phenomenon of PD was the focus of local and global interest. This also includes organizations interested in this field, electrical and electronic departments in universities, electricity distribution companies, since at least the early 1940s [16-19]. Moreover, the military was interested in detecting this phenomenon. Early studies have indicated that size, cavity shape, temperature, and electrical pressure can affect the partial discharge phenomenon [20]. PD is the main reason for insulator damage at all times but does not drive to instant collapse. The insulators must be in good condition in DC and AC voltage applications, checking the condition of the insulators can be done through multiple approaches, but the detection of this phenomenon has emerged as the most popular means of gauging the state of the insulators [21]. Plant industrialists can use specific sensors to find partial discharges in transformers. However, by using electrical measurements they cannot discover the exact place of the phenomenon. To reduce the cost of equipment failures, early detection of the phenomenon is key to preventing full insulator failure [22]. Because of electromagnetic interference which can be happened or noise of people, cars, and even birds, the detection of the PD became worthless by using electrical measurements, especially, on transformers while they are in service. Therefore, using chemical approaches to detect the PD works well by detecting the gases which are combusted [23]. Insulation equipment is collapsed as a result of two things which are chemical dissolution and bombing nitrogen ions [24]. Resorting to the science of artificial intelligence in all its sections to solve engineering problems and others is a sound and effective choice, and this science has been used in discovering electrical faults of all kinds and has shown great results.This leads to the dispensation of chemical detection methods. Akbal et al., employed ANN to predict the sheath current. Furthermore, this type of current must be specified before installing a new cable line underground to avoid cable faults could be occurred [25]. An expert system was devised by Yaw et al. to detect short-circuits in the stator windings of a permanent-magnet synchronous machine (PMSM) [26]. Using CNN was proposed to solve power problems such as transmission lines fault detection by Wang et al. [27]. Faults on overhead power lines caused serious accidents and affect the power grids negatively, for example, the breakdown of the insulators which support the power lines and which leads to a drop-down of these lines on the trees or ground that type of fault is called High-impedance faults (HIFs). That implies vast wildfires can happen. [28]. Online Monitoring artificial intelligence system was proposed to detect HIF accurately and fast [29]. Tehrani, et al. developed an AI model to extract the feature and frequency information by integrating a

Long-Short Term Memory architecture with an attention layer, and one-dimensional CNN [30].

A deep learning model was created in this thesis to anticipate PD without having to cut the power. That model has 4 inputs and one output, they represent the processed three-phase voltage signal, and the prediction of the model, respectively. Two working areas are visible in figure 1.3, firstly, the Kaggle working area is responsible to provide training and testing datasets that have been acquired from power lines with a new meter designed at the ENET Centre at VŠB. This means that getting data is Kaggle's responsibility. Secondly, the thesis's working area is to use this data to train an AI model after pre-processing data.

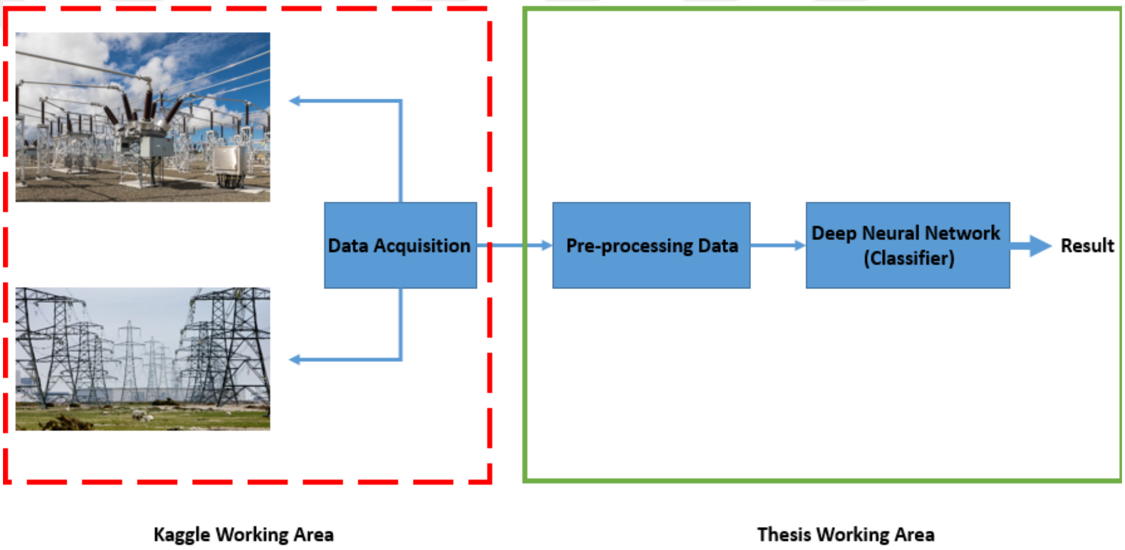



Figure 1.3 : Overall detection of the partial discharge system.

Table 1.1 shows a comparison of the performance and accuracy of several different partial discharge detection methods using the same data set. This table shows different methods using artificial intelligence techniques, not other traditional methods, which will be compared with the thesis's proposed model in the second chapter. As shown, the proposed model in this thesis, which was designed to rely on the idea of hybridization or merging of two different models, outperforms all other methods and even Kagger winner in terms of accuracy and MCC.

Table 1.1. Performance comparison between different AI methods [30].

| Method | MCC | Accuracy |
|---|---|---|
| **Kaggle Winner** | 0.450 | 0.922 |
| **LSTM-Attenstion** | 0.3870 | 0.913 |
| **DNN with Peak features** | 0.258 | 0.873 |
| **CNN with frequency features** | 0.293 | 0.917 |
| **Frequency-based Multi Task learning With Attention Mechanism** | 0.433 | 0.961 |
| **Thesis's proposed model** | **0.79** | **0.975** |

# CHAPTER 2. GENERAL INFORMATION

## 2.1. Partial Discharge Phenomenon

Partial Discharge (PD) can be occurred inside the solid insulator (existence void helps the formation of PD) or on its surface. Furthermore, PD is created when electrical stress develops in the gap between two conducting points. The discharge could be in a gas bubble in a liquid insulator.

### 2.1.1. Modeling of defects present inside a dielectric

Figure 2.1 presents a simple and general model for the phenomenon of partial discharge. Where defects can be modeled with a capacitive insulator. That is, each defect inside the insulator can be related a separate capacitance. If the electric field across a given capacitor creates a voltage higher than that the capacitor can withstand, partial discharge occurs leading to a current pulse that travels along with the capacitor.

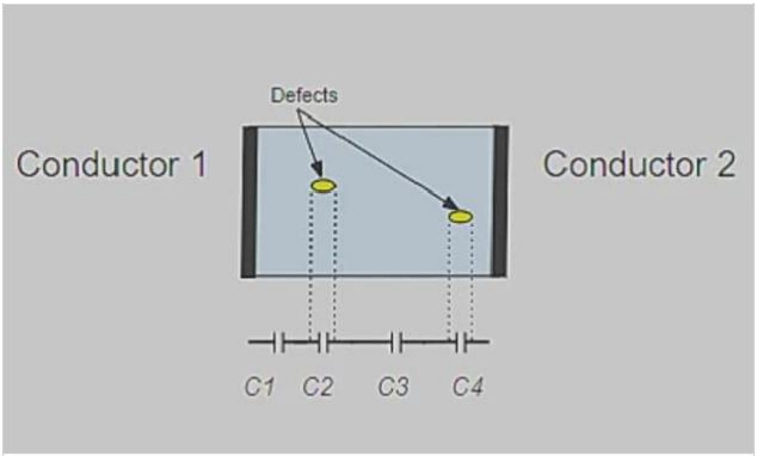



Figure 2.1 : Scheme of electrical model of defects present inside a dielectric [30].

Defects in different types of insulators are the major factor behind this phenomenon. Defects cause a no-uniform dielectric strength and in the parts where it becomes lower than the local field applied, the discharge can occur. One reason for this is that it does not affect the entirety of the dielectric material. This effect can occur on high and medium voltage lines, but the likelihood of its occurrence is greater at high voltages. It also can occur in liquid insulation because of the presence of gas bubbles or in solid insulation for example inside the dielectric of a cable.

### 2.1.2. Types of partial discharge

The partial discharge phenomenon is classified into three main groups, and each type has specific characteristics and location.

#### 2.1.2.1 Corona discharge

Figure 2.2 illustrates that this type of partial discharge is widespread and can be formed in the sharp surface of the conductor and is caused by direct discharge air emitted from the surface and causes acoustic and radio frequency emissions.

Figure 2.2 : Formation of corona discharge [31].

Corona discharge is common to be in high voltage applications like cathode ray tube TV and X-ray machines but it is not desired because it is viewed one of the energy waste ways and also an unwanted load on the circuit. In addition to this, corona with air can produce many contaminated gases for human-like ozone ($O_3$) and nitric oxide

(NO). These gases can be harmful to the near material and equipment and cause damage to them [31].

#### 2.1.2.2 Surface discharge

As depicted in Figure 2.3, this sort of discharge takes place along the surface as it is called and produces a crackling and ozone smell. [32].

Figure 2.3 : Surface discharge Tracking [33].

#### 2.1.2.3 Void (internal) discharge

Void discharge typically keeps expanding till the insulation completely failed and Void discharge is also extremely subversive to insulation. When the failure occurs, It is not easy to reveal PD before the failure takes place because no indicator like sound or smell or even hint to point out the problem. That implies, it is a silent defect and this is a large problem and challenge for engineers. Moreover, this type mostly happened in solid insulators and gas-insulated substation (GIS) junction insulation, for instance [4].

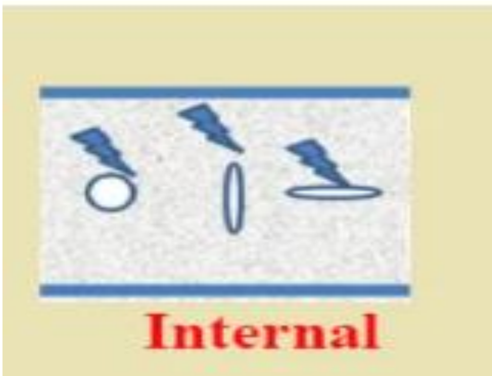



Figure 2.4 : Internal discharge .

### 2.1.3. Detection methods of partial discharges

Generally, partial discharge is responsible for the final breakdown and dissolution of the insulators. That's why in the factory testing or monitoring in-service of many types of high voltage equipment is a criterion and necessary procedure because in cases of insulation failure is about to occur, it is essential to take prompt action. To avoid disasters or service interruption and that makes these types of procedures are mandatory. In addition to this, high and medium voltage equipment is very expensive and damage to it leads to disasters such as fires or maybe can also lead to loss of service. Therefore, there are many techniques for detecting this phenomenon to improve the reliability of the system. PD may be detected by many techniques like thermography, electrical detection, chemical reactions, and acoustic emission. The benefits and drawbacks of various PD tracking techniques are outlined in table 2.1.

Table 2.1: Evaluating various methods and approaches [2].

| Approach | Sensitivity | Applicable on site | Real time | Accuracy |
|---|---|---|---|---|
| **Electrical** | ✓ | ✗ | ✓ | ✓ |
| **Chemical** | ✓ | ✓ | ✓ | ✗ |
| **Acoustic** | ✗ | ✓ | ✓ | ✗ |
| **Thermography** | ✓ | **Just surface PD** | ✓ | ✓ |
| **Proposed Model** | ✓ | ✓ | ✓ | ✓ |

### 2.2. Insulators Used in Transmission (Overhead) Lines

Insulators are materials that restrict the flow of electric current. Insulators are important in fixing overhead power lines, and they must be made from non-conductive materials. There are many materials such as PVC, polymer, and glass that are suitable for the manufacture of these insulators and other electrical equipment, but Porcelain is the most commonly employed material in industry applications. Moreover, with the advancement of research in material science and physics, new materials and compositions were invented that offer improved insulation and environmental resistance properties, such as glass stone. [34]. There is great interest in the insulator industry because it provides support for voltage lines in general and prevents current from leaking to the ground through the towers and if it is not isolated, disasters, fires, and electric shocks will arise for individuals.

### 2.2.1. Properties of overhead line insulators

As mentioned above the purpose of the insulators and their importance in fixing, supporting overhead power transmission lines and also prevent current leakage to earth. However, the insulators need to possess particular characteristics in order to fulfill this requirement. In fact, there are many properties which are considered in the insulators such as withstanding environmental factors and high resistance for electricity to prevent leakage and electrical shocks. Furthermore, insulators that support overhead power lines are exposed to wind and the weight of cables, that's why high mechanical strength insulators are made [35].

### 2.2.2. Types of insulators

Above, the properties and materials used to manufacture them have been discussed but to ensure optimal performance of power distribution and transmission systems, various configurations or categories of insulators are employed based on their location and application..

#### 2.2.2.1. Pin insulator

This class of insulator consists of many parts based on the required purpose but even though this type is the oldest, it is suitable to be used with overhead power lines which transfer up to 30 KV.

Figure 2.5 : Pin insulator [36].

### 2.2.2.2. Suspension insulator

A suspension insulator is composed of multiple disc insulators that are linked together to support power lines and gives the flexibility to withstand wind and dynamic loads. In general, this type of insulator is frequently utilized for overhead power lines ranging from 11 kV up to 765 kV

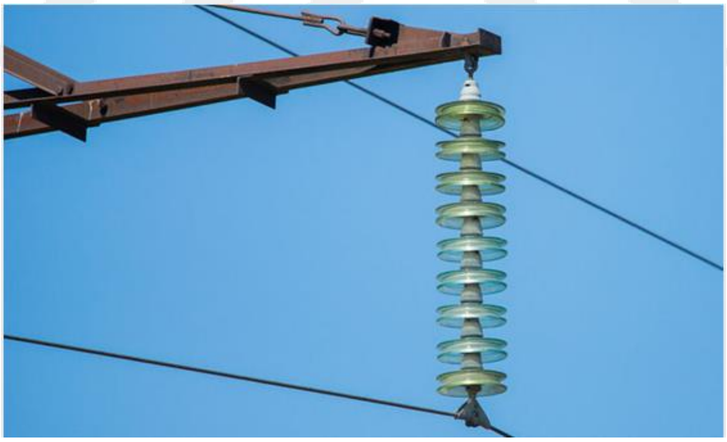


Figure 2.6 : Suspension insulator [37].

### 2.2.2.3. Strain insulator

This insulator is similar to the suspension type in many respects and is used in power lines that work at 33kV like a pin insulator. In addition to this, this type of insulator is designed to withstand high loads and is suitable for insulation purposes.

Figure 2.7 : Starin insulator [37].

### 2.2.2.4. Stay insulator

This type of insulator is simple and used for low voltage purposes and prevents the power line from falling to the ground by supporting the tension wire in case the insulator fails. It is made of porcelain as illustrated in Figure 2.8.

Figure 2.8 : Stay insulator [37].

### 2.2.2.5. Shackle insulator

In general, this type is similar to a pin insulator but this can be used in different configuration horizontal, bend ,vertical, and circular, it also works at voltage up to 33 KV like the pin type. Moreover, the Shackle insulator is relatively small size and is used to support overhead power transmission lines.

Figure 2.9 : Shackle insulator [37].

## 2.3. Micro-Phasor Measurement units(µPMUs)

In this thesis, the phenomenon of partial discharge will be discovered by diagnosing the voltage signals passing through the power lines, but there is a need for a tool or device to read the voltage signal and convert it into a digital form and send it to the processing and monitoring center and given it to the artificial intelligence model that has been designed to determine if it is The line is faulty or not. Pharos measurement units (PMUs) device is used to do this mission and it has the ability to read the amplitude and phase of voltage signal and beyond then send it. Moreover, it has GPS to provide the location coordinates because it can read data from different locations and do synchronization between them. [38].

The µPMU's functional block diagram is presented in Figure 2.10. Its high processing speed allows for accurate time stamping and phasor estimation of input signals. The anti-aliasing filter functions to eliminate signal components that have frequencies equal to or greater than half the Nyquist rate. To convert analog AC waveforms, such as voltage and current phasors, into digital signals, an analog-to-digital converter is used. The GPS signal is transformed into high-speed timing pulses for waveform sampling through the use of a phase-locked oscillator. The phasor microprocessor carries out phasor calculations, and the resulting phasors are combined to generate measurements in the positive sequence. The phasors are then time stamped with the clock's timing information and GPS receiver's second of century count. Finally, a modem is used to transmit the phasor value to a data center.

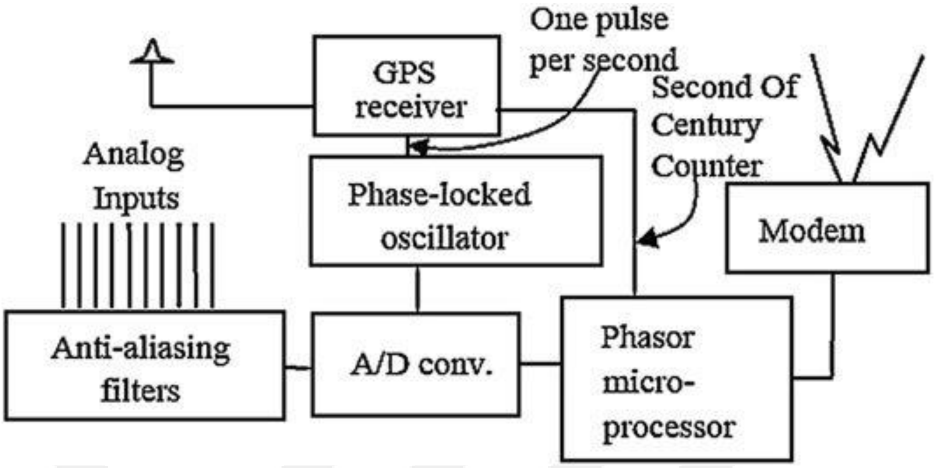



Figure 2.10 : Block diagram of the μPMU.

## 2.4. Artificial Intelligence

Many algorithms which mimic human logic are developed and used when they make logical deductions and solve puzzles in the 1980s and 1990s [41]. Economics and probability studies helped the researchers to improve methods that make the artificial intelligence able to deal with uncertain and missing data [42].

Artificial intelligence has been defined in numerous ways, all of which concur that it involves imbuing machines with the cognitive abilities of creatures like humans. Essentially, this involves teaching machines and computers to learn and carry out tasks based on what they have learned. While artificial intelligence systems cannot perfectly replicate human behavior across a wide range of tasks, they are capable of performing specific human tasks. B.J. Copeland has described the science of intelligence as the ability of a machine or robot to act and make decisions in a manner that is similar to humans [43].

As shown in figure 2.11, the perceptron is defined by using six specific mathematical representations that elaborate learning mechanism. These are representations are input, weight,bias, summation and non-linear activation function to add the ability to deal with nonlinear problems [44].

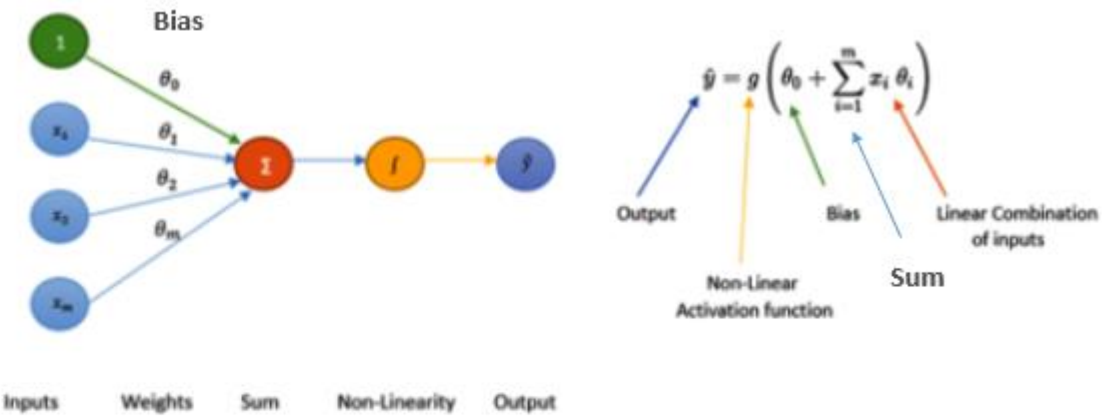



Figure 2.11 : Building a perceptron.

There are many branches of AI like pattern recognition, inference, learning from experience, planning, heuristics and genetic programming [45].

### 2.4.1. Machine learning

Literally, machine learning is not artificial intelligence, because machine learning is a way to train a machine and imitate human intelligence which leads to the achievement of artificial intelligence. ML is a subfield of AI, and there exists a close relationship between the two. Also, machine learning is algorithms based on data and not on clear programming to make the machine aware of most of the conditions and it has not been trained in[43]. Machine learning is a fast-developing field and plays a crucial role in data mining, relying on statistical methods for achieving its goals. It focuses on how to set up computers that improve automatically through experience depending on the dataset using it during learning[46]. It can serve many sectors in life such as marketing, manufacturing, education, and health care. Machine learning works by moving through three stages or parts [47]:

- **A Decision Process:** In this stage machine learning algorithm will infer the pattern of data to predict the unseen data or for classification. In general, the data can be labeled or not and both.

- **An Error Function:** It acts as a tool for assessing the reliability and performance of a model.

- **An Model Optimization Process:** Optimization algorithm determines how the network parameters are updated by looking at the change in the loss function. The weights are adjusted to minimize the difference between the predicted output and the actual output.

The category of machine learning can be determined based on human intervention on the data by offering rewards and the type of data if it is labeled or unlabeled. There are four categories[48]:

- **Supervised learning:** It is type of machine learning deals with labeled data to classify and prediction and each datapoint has a label or target.

- **Unsupervised learning:** This type of machine learning involves using unlabeled data and it clusters the data by disclosing the hidden pattern because it doesn't have a target or label.

- **Semi-supervised learning:** It can handle both labeled data or unlabeled for classification or regression purposes. It is also useful with not enough labeled data.

- **Reinforcement learning:** This approach interacts with the environment and also uses a trial-and-error techniques to learn the model by giving a reward when the model works correctly or else it will be punished.

### 2.4.2. Deep learning

The first theorization of deep learning was in the 1980s but from 2006, it has become a new and more prevalent trend than machine learning in the AI community because as noted in [49], is capable of processing vast quantities of data and can extract features from raw data using multiple layers of processing [50].

The deep neural networks can deal with and learn from a dataset without human intervention as shown in figure 2.12. That means it does not need human knowledge to convert primary data to useful formate because it consists of multilayers. The deep learning model can be better if huge training data is provided.

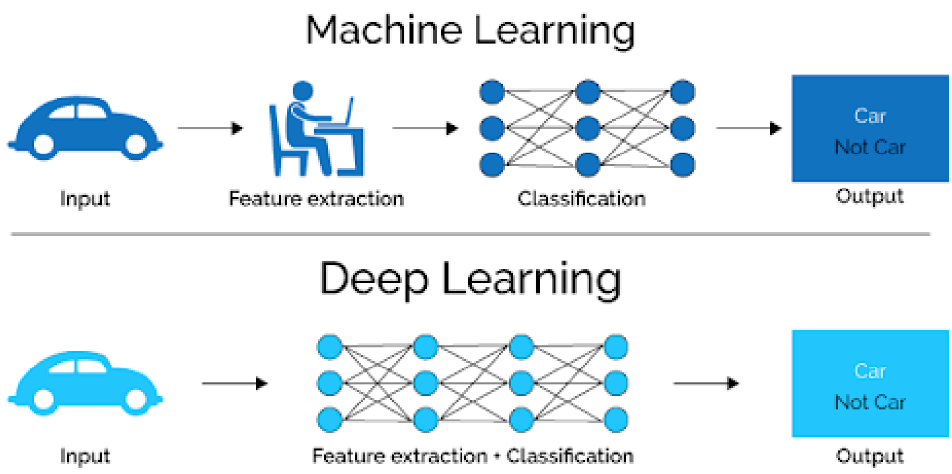



Figure 2.12 : Illustrates differences between machine learning and deep learning [51].

Deep learning has dramatic achievements in many sectors such as object detection and speech recognition [52]. Since the neural network hanels a vast quantities of data, the use of the GPU will give high performance because the GPU provides parallel processing[53]. For example, in the project of the driverless car using GPU in training the AI model will give great results because input data consists of millions of videos and images. In general, the structure of the deep network does not differ from the traditional one, except only in the number of layers. Deep networks, as demonstrated in

Figure 2.13, are defined by their multitude of layers, allowing them to effectively handle vast amounts of data and learn more effectively than traditional networks

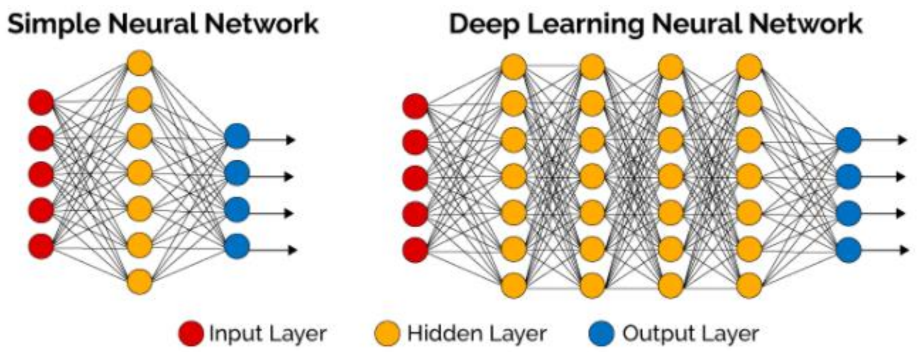



Figure 2.13 : Traditional neural network vs deep neural network [54].

Deep learning neural network has many types:

- **Artificial Neural Networks (ANN):** It is constructed from many weighted linked perceptrons. It basically imitates the human brain with multiple layers of massively parallel computational models. The neuron performs a collection of inputs and generation of output. The ANNs have been used in complex nonlinear problems [55].

- **Convolutional Neural Networks (CNN**): The name came from the convolution computation that is is utilized to extract features from images. It can take in an input image and extract the features by itself using convolution computations and beyond then adjust the learnable weights and biases to classify images by effectively distinguishing between distinct categories. CNN can handle and preprocess images on its own. In addition, CNN is faster than other classification algorithms. [56].

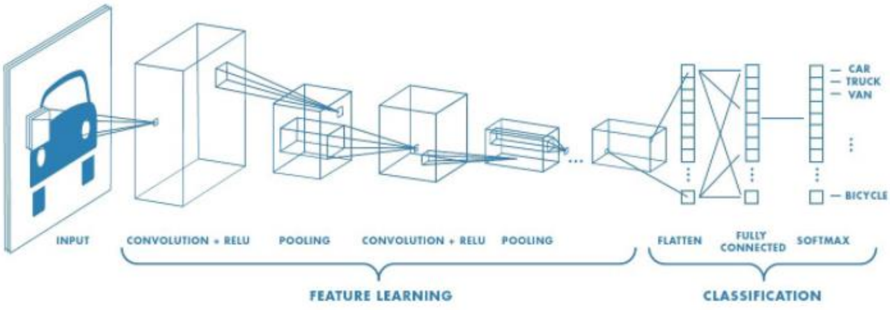



Figure 2.14 : Convolution neural networks (CNN) [56].

- **Recurrent Neural Networks (RNN)**: It is most commonly used with time series data and not images like CNN is. RNN is used in many useful projects and can solve many temporal problems such as language translation, natural language processing (NLP), voice and speech recognition [57]. Moreover, RNN differs from CNN and even ANN in that it has a memory for saving the information of the previous inputs and using it with the current inputs in its computations. This implies that the RNN's input comprises both the current input and past samples, and that the output of the recurrent neural networks influences preceding elements in the sequence but in ANN and CNN it does not. In other words, the RNN uses the previous states in the decision-making. The effect of previous states vanishes over time.

The RNN has Variant architectures such as:

- **Long short-term memory (LSTM):** As previously stated, the vanishing gradient affects the performance of RNN so, in 1997, Hochreiter and Schmidhuber proposed a new type of neural network to solve this problem [58]. He invented the LSTM and one of the advantages of it, is insensitivity to gap length over RNN[59].In other words, LSTM aims to maintain the effect of previous data using the previous states and a carry line.

  In this thesis, another type of LSTM called BiLSTM has been used in building the classifier model.

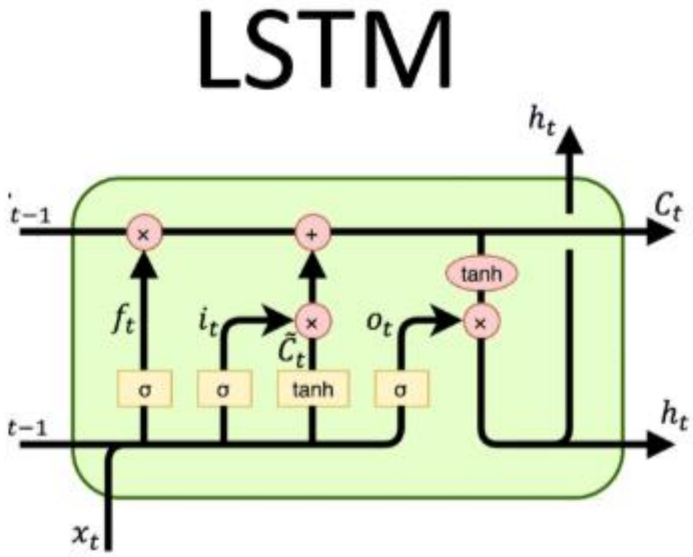



Figure 2.15 : LSTM network structure [60].

### 2.4.3. Trasnsfer learning

It is a recycling operation for previously trained models. By implementing certain modifications, it can also be applied to address a related problem in the field of machine learning (ML) [61]. The utilization of Electroencephalography (EEG) signals may be applicable in resolving issues linked to the recognition of Electrocardiography (ECG) signals. Figure 2.15 illustrates that through transfer learning, we can utilize pre-existing models trained on vast datasets, and subsequently modify them to function as the starting point for another task, if needed to fit this task and this provides a shortcut to save time or get better performance. Furthermore, a prevalent application of this technique is to leverage an AI model that has been trained on a massive dataset, and whose acquired knowledge is preserved for future use. The preserved network can be employed to address a comparable task, with necessary modifications if required.

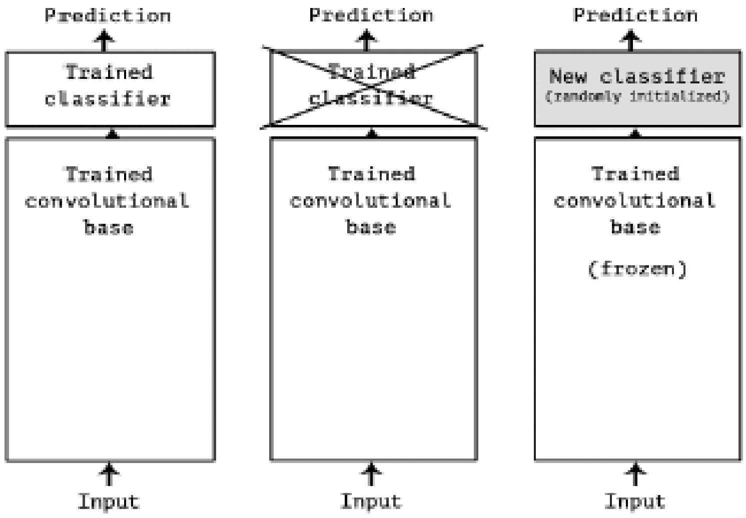



Figure 2.16 : Using a pretrained model to another task.

### 2.5. Python Language Programming

Guido van Rossum developed Python to replace the ABC Programming language in the 1980s [63Python is a popular choice in the field of AI due to its suitability for both small and large projects, and its status as a high-level language. It was created to be highly extensible, with the capability of incorporating modules as needed, making it a versatile tool for various applications.

It has a diverse range of applications in various industries, including but not limited to web development as a back-end, mathematics, robotics, system scripting, and artificial intelligence. In addition to this, Python is distinguished from other languages by [64]:

- Python is a cross-platform language that can run on multiple operating systems, including Linux, Windows, Raspberry Pi, and Mac.Python can be treated as an object-oriented program.

- Python gives programmers the ability to write their code with fewer lines than other languages.

- Python has a simple syntax that makes programming lines easy to write and read.

Consequently,Python offers many powerful built-in libraries. For example, matplotlib ,NumPy, SciPy,nltk, SimpleAI are embedded in Python for general purpose but for AI and deep learning there is TensorFlow and Keras libraries that are developed by Google [65].

### 2.6. TenserFlow and Keras Library

The Apache License 2.0 was adopted for the release of TensorFlow (TF) in 2015 but in 2019 Google released another version of TF. It is a library for building up and describing machine learning and artificial intelligence models by providing simple built-In functions. Particularly, the TF library provides powerful tools and functions for designing and training deep neural networks of different types. The Google Brain team created it for their own internal use in both research and production purposes. [66]. TF provides the ability to train models on the CPU and GPU and as a result, the GPU training process can be much faster. Moreover, Although it employs multiple programming languages like C++, Javascript, and Python, it is primarily based on Python. [67].

Representing data in an array is useful to make processing data easy. So Python has many powerful libraries which can represent data in form of an array (Tensor) such as Numpy. Tensor is an n-d array and it is helpful to represent data as shown in figure 2.15.

Keras is a library similar to TF and both focus on using Python to build ML and deep neural networks, but Keras is superior in simplicity, easier to learn, and better for small data sets. It is also powered by Google and there is excellent technical support and solutions to problems that users might encounter.

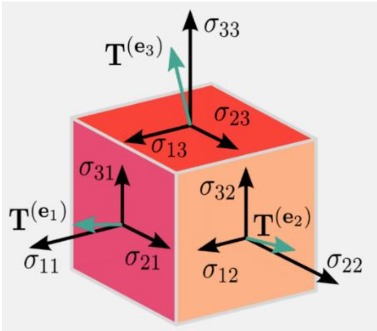



Figure 2.17 : Tensor [68].

## 2.7. Kaggle

This platform or community offers machine learning professionals access to a variety of datasets and enables them to publish their own datasets. Furthermore, it presents an array of artificial intelligence competitions, with the goal of promoting machine learning practice among students and researchers. Such competitions involve tasks that include prediction and classification, encouraging participants to develop effective models and solve data science challenges available on the platform.

Kaggle started its service by providing machine learning workshops, and open-source AI code in 2010. It also provides the required dataset for challenges announced on its online platform with prizes for winning competitors [69].

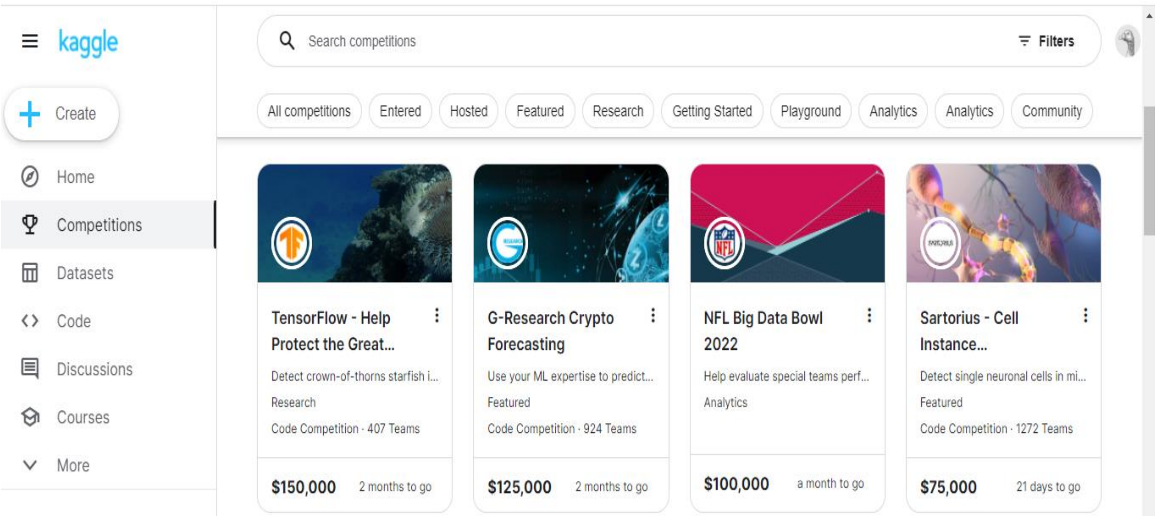



Figure 2.18 : Kaggle website for competitions [70].

# CHAPTER 3. DATA PROCESSING AND FEATURE ENGINEERING, EXTRACTION

## 3.1. Introduction

As mentioned earlier in the problem statement, which is the detection of faults in the power transmission lines through the binary classification because there are two cases faulty case or not, if the phenomenon of partial discharge is present in phase that means faulty or class-1, or not it is mentioned as class-0. It is necessary to put the dataset in the form to be understandable before designing the AI model by preprocessing the raw data and extracting the features from them because the use of raw data as it is and its effect on the neural network in many cases leads to the inability of the model to learn.

## 3.2. Pre-Processing Dataset And Feature Extraction

Firstly, The training and testing datasets used in this study were sourced from the Kaggle competition website which gives the competitors the ability or even anyone to publish and find trained AI models and even datasets for free. Moreover, Kaggle is a subsidiary of Google LLC, which is an online community comprising of data scientists and practitioners of ML. The signal data comes from the real environment, not a lab, and they contain a lot of background noise. These signals are measured by the patented device of ENET with a lower sampling rate (cost efficiency purpose).In addition to this, metering devices were deployed in more than 20 different locations(8,712 phases). This implies that the spectrum of noise and quality of PDs can vary significantly from one another, that the correct and robust classification is still an ongoing problem.

In this thesis, three deep learning neural networks were designed to build a robust classifier and all of them were built by using BiLSTM. Each model has a structure consisting of two inputs and one output. which decides, if the power line is faulty or normal. After that, all three models have been trained, compared and selected the optimum model. It is the classifier which can depend on it to distinguish or detect the faulty line.

### 3.2.1. Data description

Each training and testing dataset signals have the same properties of sample time, several samples, and grid operating frequency. The three-phase voltage signal is sampled every 25 nanoseconds, which is a great sampling time for a high-quality ADC signal. In addition, the electrical system operates at 50 Hz, This means that each phase has been digitized with a total of 800,000 samples. In short, the sample number covers a complete voltage signal.

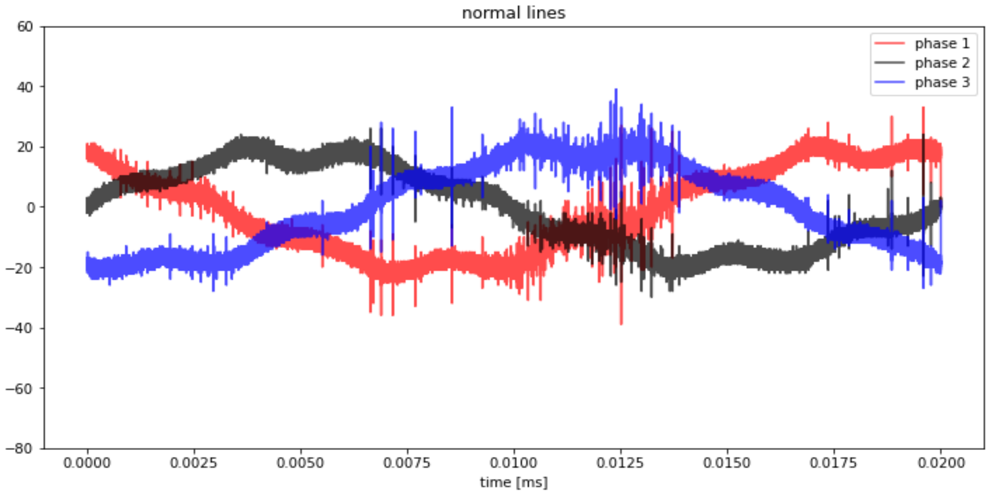



Figure 3.1 : Normal 3 phase power lines.

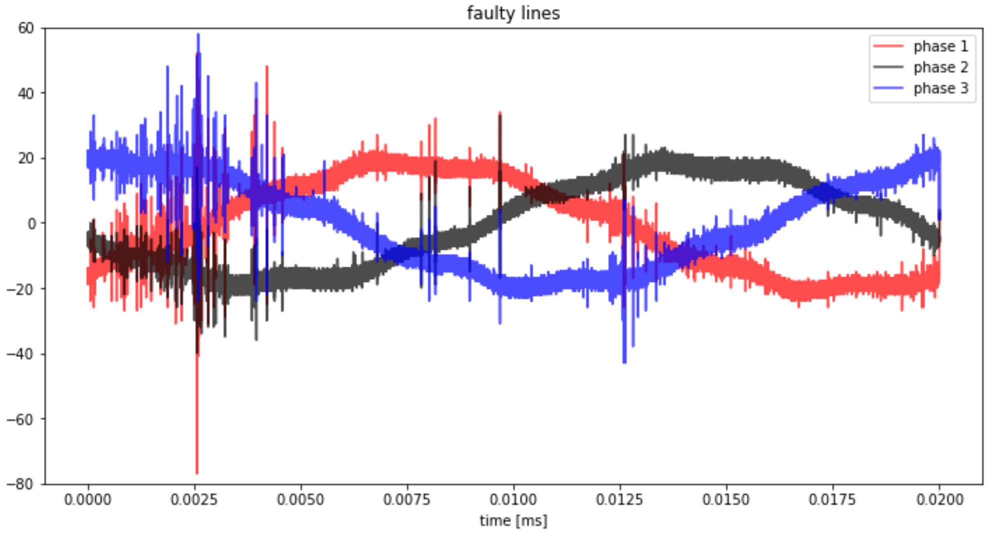



Figure 3.2 : Faulty 3 phase power lines.

As depicted in figures 3.1 and 3.2, the raw signals for the unproblematic (class-0) and problematic (class-1) signals are almost identical, which makes this problem particularly challenging.

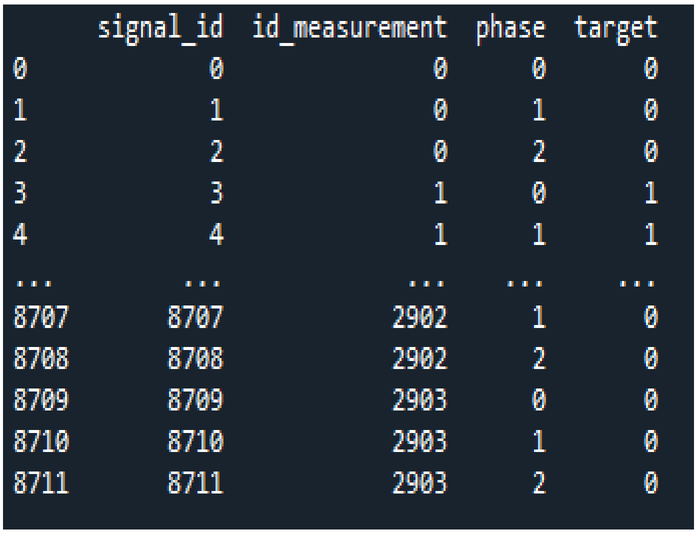
The datasets provided include training and testing datasets, each of which consists of two files: metadata, which provides additional information about the data, and parquet, which contains the signal data values organized into column groups. Each column contains one signal and each signal was converted by ADC to 800,000 samples and was stored in an integer 8-bit data type in range [-128,127], the signal data values are stored in a pyarrow.parquet file, as illustrated in figure 3.3.

| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 16 | 0 | -18 | -15 | -5 | 18 |
| 1 | 16 | 0 | -18 | -15 | -5 | 18 |
| 2 | 16 | 0 | -18 | -15 | -5 | 18 |
| 3 | 16 | 0 | -18 | -15 | -5 | 18 |
| 4 | 16 | 0 | -18 | -15 | -5 | 18 |
| ... | .. | .. | .. | .. | .. | .. |
| 799995 | 0 | 0 | 0 | 0 | 0 | 0 |
| 799996 | 0 | 0 | 0 | 0 | 0 | 0 |
| 799997 | 0 | 0 | 0 | 0 | 0 | 0 |
| 799998 | 0 | 0 | 0 | 0 | 0 | 0 |
| 799999 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 3.3 : One phase signal “pyarrow.parquet” file representation.

As shown in figure 3.4, metadata file provides many parameters for signals as follow :

- **id_measurement**: it is an unique ID for each 3 phases signal which are recorded at the same time.
- **signal_id**: it is an unique number that is given for each measurement as an ID. That means each signal has own ID number.
- **phase**: indicates the order of phases at each  simultaneous measurements stage.
- **target**: 0 for normal and 1 for faulty power line.



| | signal_id | id_measurement | phase | target |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 2 | 2 | 0 | 2 | 0 |
| 3 | 3 | 1 | 0 | 1 |
| 4 | 4 | 1 | 1 | 1 |
| ... | ... | ... | ... | ... |
| 8707 | 8707 | 2902 | 1 | 0 |
| 8708 | 8708 | 2902 | 2 | 0 |
| 8709 | 8709 | 2903 | 0 | 0 |
| 8710 | 8710 | 2903 | 1 | 0 |
| 8711 | 8711 | 2903 | 2 | 0 |

Figure 3.4 : “Metadata” excel file representation.

The training dataset and testing dataset sizes are 8,712 and 29,048, respectively. After observing the distribution of the classes that out of the 8,712 phase signals (in different 20 locations), only 525 signals are class 1, representing only 6% of the total data (imbalance data).

As illustrated in figure 3.1 and 3.2 the raw (faulty and normal) signals, it was difficult to notice a difference between them. In addition, as depicted in figure 3.5, every Electrical signal is primarily composed of Sinusoidal Waves (in various frequencies).like the dark black line in the figure above.

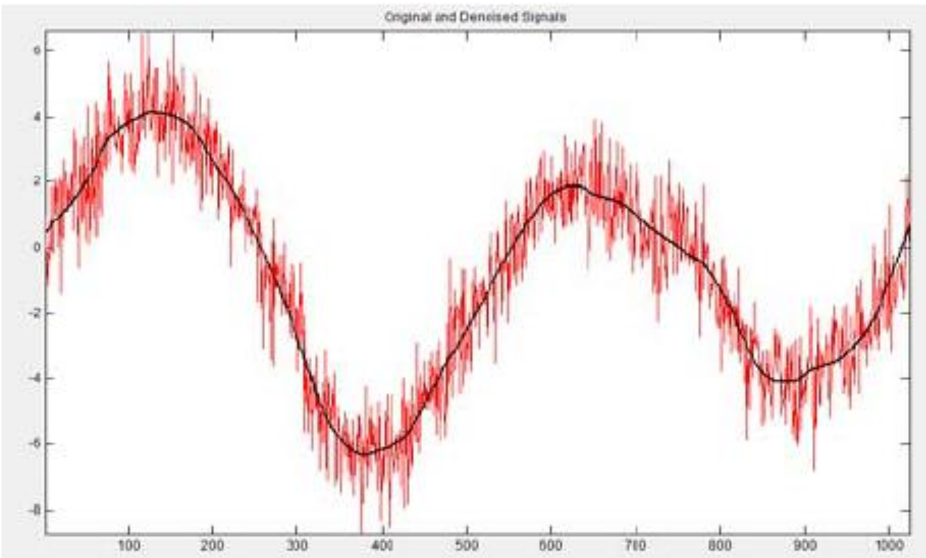



Figure 3.5 : Noisy signal and its sinusoidal component.

That means that the common characteristic between them that they are both Sinusoidal waves with a frequency of 50 Hz, and that noise is a different thing as a result of the interfering waves resulting from the phenomenon, and therefore, firstly, the idea that these Sinusoidal Waves should have removed because when considering visualization of the bad signal the key point deciding whether or not the wiring is partially damaged is more likely the variance around these Sinusoidal Signals. There are many techniques or algorithms to remove noise from signals like Wavelet Denoising, Fourier Denoising, and Moving Average Denoising.

In this thesis, wavelet Denoising was used to remove sinusoidal waves from the signal.

### 3.2.2. Wavelet denoising algorithm

Simplicity, Wavelet-based denoising is the hybrid between representing data in both time and frequency domains. It is more powerful than Fourier transformation because Fourier just has the ability to represent signals in the frequency domain. In 1997, Safavi et. al, were interested in wavelets applications such as systems analysis and fault detection in their research [71]. Wavelet thresholding has been used in signal and image processing since the advent of wavelet transforms. The technique involves decomposing the noisy data into an orthogonal wavelet basis, then suppressing the wavelet coefficients that are smaller than a certain amplitude by using soft or hard thresholding. The data is then transformed back into the original domain. This non-linear thresholding estimator can be applied on an orthogonal basis, such as Fourier or cosineAccording to Donoho and Johnstone, this technique exhibits performance comparable to an ideal coefficient selection and attenuation [72].

As shown in figure 3.6 and 3.7, faulty and normal signals were denosied by using Wavelet Denoising method. Denoised Signals can be seen slightly different.

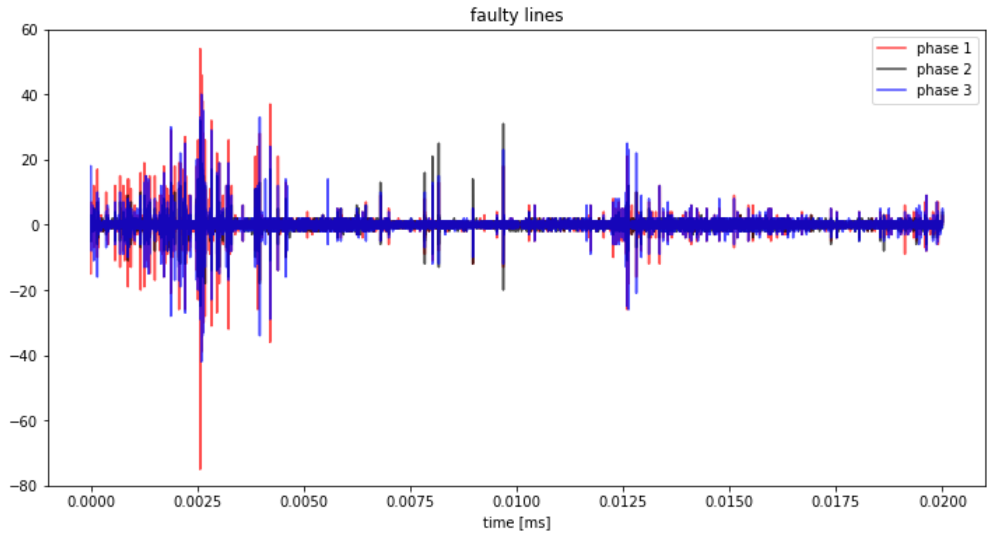



Figure 3.6 : Class-1 denoised 3 phase power lines.

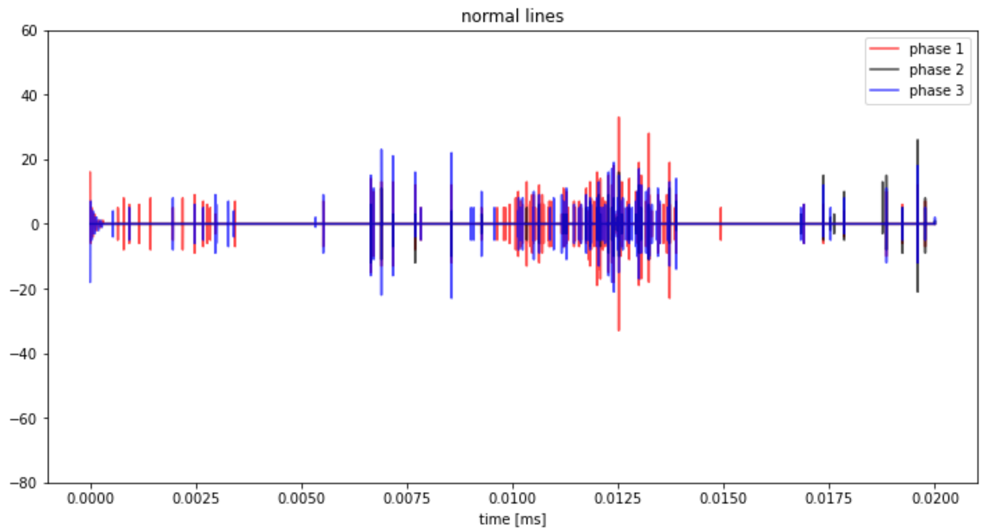



Figure 3.7 : Class-0 denoised 3 phase power lines.

### 3.2.3. Feature engineering and extraction

Sometimes a machine learning model needs preprocessed input data by feature extraction as this would help the model to learn faster and better. This leads to the process of feature engineering, where features are extracted from raw data before they are fed into a machine learning model. It is one of the vital steps in the predictive modeling process, it is the process of transforming arbitrary data into a form that is understandable in supervised learning models.

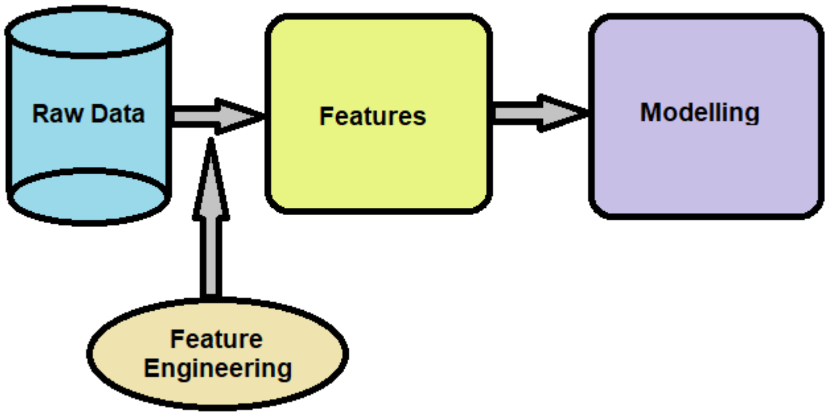



Figure 3.8 : Feature extraction process block diagram.

Feature engineering is very important step during developing artificial intelligence model because if the features are extracted from raw data correctly as possible that makes the model to learn easily and predicte unseen data accuratly.

In this thesis, the training dataset was described and original dataset was processed by using wavelet denoising algorithm. Therefore, three type of training dataset could be used as follow :

- **Original or raw dataset**: It is a raw dataset which is taken from the Kaggle competition website and this data was captured by ENET company at VSB-TU Ostrava as mentioned before and the shape of data is (8,712 X 800,000) and (29,048 X 800,000) for training and testing sets respectively.
- **Denoised dataset**: The raw dataset was subjected to a wavelet denoising algorithm to eliminate sinusoidal components from the original signal. The resulting denoised dataset retains the same shape as the raw data, with dimensions of (8,712 X 800,000) for the training and (29,048 X 800,000) for testing sets. This type of dataset was used in this thesis.
- **Combined dataset**: It is concatenating or combining original and processed data to compromise the properties and information between data before and after processing.

### 3.2.4. Converting signal to vector

As illustrated each phase has a sample size of 800,000 and these data range between -128 to 127 because it is an 8int data type. Furthermore, neither the original nor the processed data are suitable inputs for the ML model, and also it is not great modeling to design a model with 800,000 inputs and also is not correct to train the model with each phase separately that means there will be three models and each model with 800,000 inputs as depicted in figure 3.9. Therefore, the data should be converted from signal shape to vector shape and extract the features.

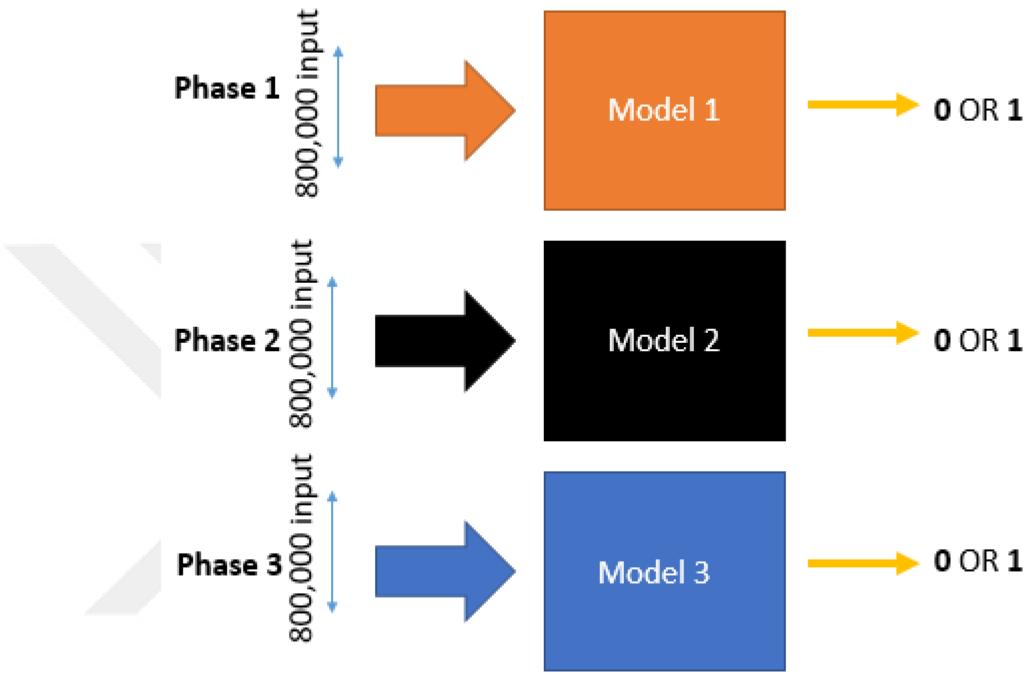



Figure 3.9 : Model design deals with raw data.

The process of extracting features from the signals involved several stages, as depicted in figure 3.10.


Figure 3.10 : Features extraction block diagram.

Initially, the feature time step was set at 160, resulting in 160 windows, as depicted in figure 3.11. As previously mentioned in the dataset description, each phase has a sample size of 800,000 samples.

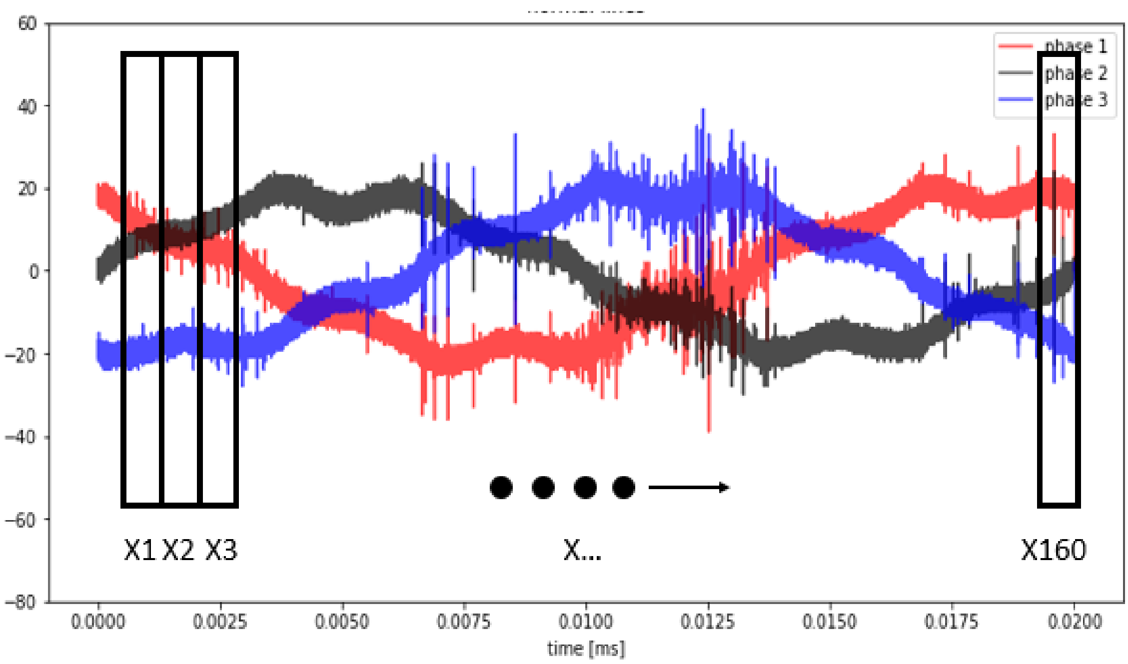



Figure 3.11 : Dividing 3 phase signals to 160 windows.

So the window size is 800000/160=5000 voltage values as shown in the figure 3.12.

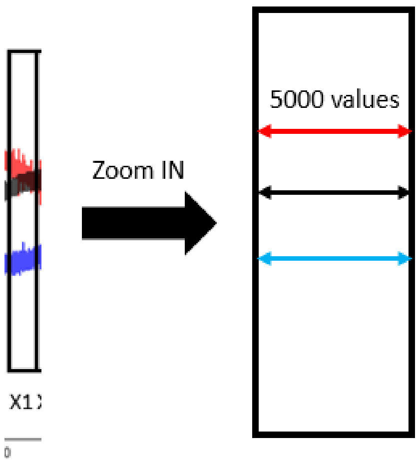



Figure 3.12 : Zoomed in window.

### 3.2.4.1. Vectorization stages

The objective of this procedure is to enable an AI model or algorithm to process a group of values simultaneously instead of individually. As depicted in Figure 3.13, the vectorization stages start by taking input data as signals and finish by producing two output data as vectors. These vectors represent the simple and and complex feature extracted data .

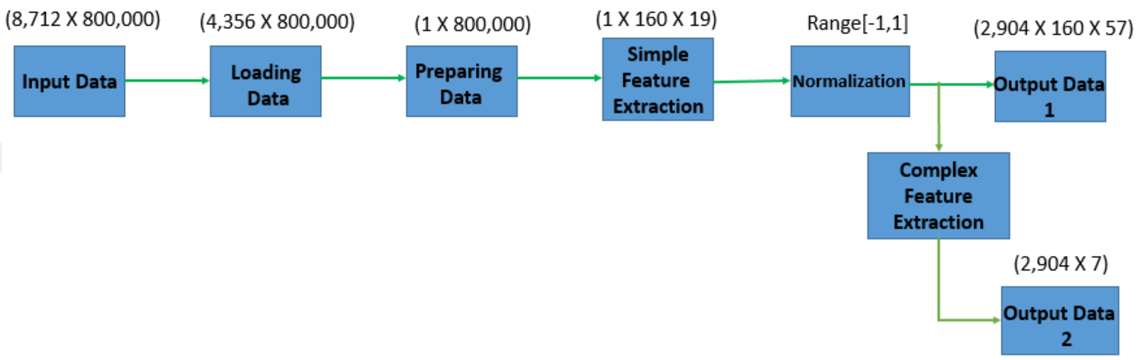



Figure 3.13 : Vectorization stages.

**Input data**: The raw signal data with shape (8,712 X 800,000) with range (-128,127) and also 800,000 represent one cycle at 50Hz.

**Loading data:** Since the signal size (8712, 800,000) is larger than the RAM that the Kaggle Kernel has, this function will perform a simple task. By dividing the input signal into two intervals (4,356 X 800,000) and passing it to the prep_data function.

**Preparing data:** At this stage, each phase is sent separately to the feature extraction stage that indicates a single input of size (1 X 800,000) will be forwarded to the following step for extracting the relevant features.

**Simple extracting features:** At this stage, the features are extracted separately from each phase during the processing. As stated previously, the number of windows is 160, which means the features will be extracted from 5000 voltage values, and 19 features will be extracted from each phase. So after that, the shape of the vector will be (1 X 160 X 19) for each phase but for every 3 phases, the vector shape will be 3X(1,160,19)=(1 X 160 X 57). Extracted features are as follows:

- **Mean:** The mean, or average, is a central measure of tendency for a finite group of numbers. It is calculated as the sum of all the values divided by the total number of values in the set.

$$M = \sum_{i=1}^{n} x_i = \frac{x1+x2+\cdots+x_n}{n} \tag{3.1}$$

- **Standard deviation:** This is a definition of variance, which quantifies the spread of a dataset by computing the average of the squared differences between each value and the mean of the set, you can obtain the variance..

$$Sn = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - mean)^2} \tag{3.2}$$

- **Standard deviation top:** It is standard deviation plus the mean.

$$S_{Top} = S_n + mean \tag{3.3}$$

- **Standard deviation bottom:** It's the standard deviation minus the mean.

$$S_{Bottom} = S_n - mean \tag{3.4}$$

- **Percentile[ 0 1$^{th}$ 25$^{th}$ 50$^{th}$ 75$^{th}$ 99$^{th}$ 100$^{th}$ ]:** A percentile score, also known as k-th percentile or centile, is a numerical value that represents the percentage of scores in a distribution that falls below that value. It is the score at or below which a certain percentage of the data set falls. Percentiles are valuable because they provide insight into how a particular value compares to the other values in the data set. The percentile could be calculated as in equation 3.5 and 3.6. where Y is the total no. of data, M is the rank of data .

$$\textbf{Percentile rank} = \left[\frac{M}{Y+1}\right]\textbf{x100 if y is odd} \tag{3.5}$$

$$\textbf{Percentile rank} = \left[\frac{M}{Y}\right]\textbf{x100} \tag{3.6}$$

- **Relative percentile:** It is the percentile subtract mean.

$$\boldsymbol{Relative\ percentile = percentile\ - mean} \tag{3.7}$$

- **Max range:** This is the amplitude of the chunk.

$$\boldsymbol{Max\ range = percentile\ 100th - percentile\ 0} \tag{3.8}$$

**Normalizing data:** After converting each phase from (1 X 800,000) as a signal to (1 X 160 X 57) as a vector but the voltage values could be in the range [-128,127] because 8int data type. Hence, this vector should be normalized. Normalization is a data preparation technique that is frequently used in machine learning and means rescales the values into a range of [0, 1] but because the dataset has negative values have been rescaled into a range of [-1, 1].

**Normalization techniques in Machine Learning:**

The most widely used types of normalization in machine learning are:

- **Min-Max Scaling**: Subtracting the minimum value from each column's highest value and dividing by the range. The minimum and maximum values for each newly added column are -1 and 1, respectively.

$$\boldsymbol{X_{normalized}} = \frac{(\boldsymbol{X - X_{min}})}{\boldsymbol{X_{max} - X_{min}}} \tag{3.9}$$

- **Standardization Scaling** :The term "standardization" refers to the process of centering a variable at zero and standardizing the variance at one. Subtracting the mean of each observation and then dividing by the standard deviation is the procedure.

$$\boldsymbol{Z} = \frac{\boldsymbol{X} - \boldsymbol{\mu}}{\boldsymbol{\sigma}} \tag{3.10}$$

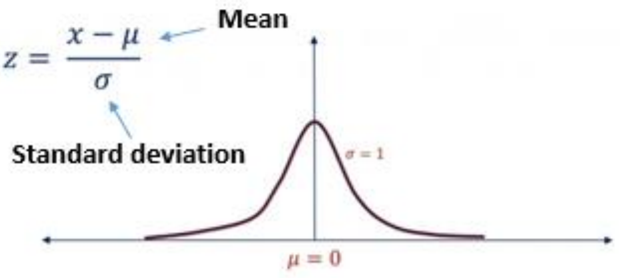



Figure 3.14 : Standardization.

The original one-phase data were normalized and standardized as shown in figure 3.15, While normalized data values are within the range of -1 to 1, standardized data values fall within the range of -3 to 3.

Min-Max scaling technique was proposed to normalize data in this thesis because It is known that the range of the original data falls within the range of -128 to 127 and the result of normalizing is putting data in the range [-1,1].

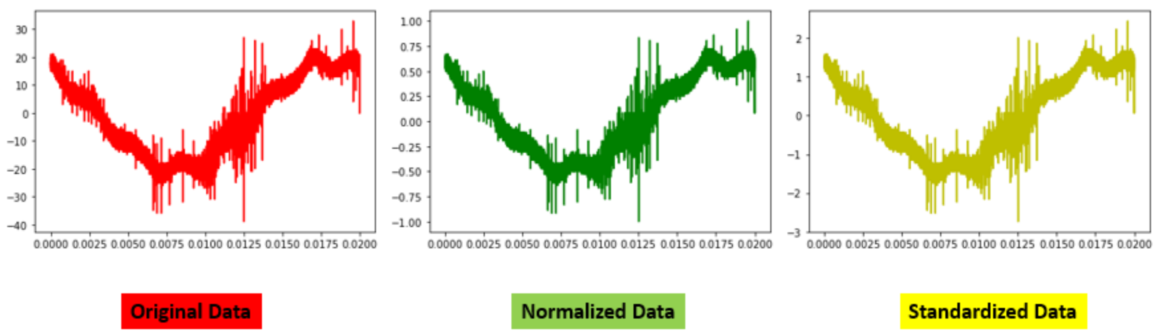



Figure 3.15 : Standardization vs normalization.

In general, data normalization speeds up learning and provides better prediction, leading to faster convergence and generally more efficient neural network training. Consequently, The Min-Max scaling technique was utilized in this thesis to normalize the datasets.

**Output data 1**: It is the result of all consecutive previous stages and the final vector out shape is (2,904 X 160 X 57) and the value range falls within [-1,1]. In addition to this, in this stage data is convenient and ready to be as input to a deep neural network model.


Figure 3.16 : Faulty power line classifier block diagram.

### 3.3. Extracting More Complex Features

In the feature extraction stage, 19 features were extracted from 5000 voltage values, i.e. from each window, such as the mean, modular deviation, and others. This type of characteristic belongs to the measures of central tendency and statistical dispersion. Whereas measures of central tendency are numerical measures used to measure the concentration or aggregation of data. Also, these scales are used to describe the data set and also to compare the different sets of data. The mean is considered one of the most important measures among them. The statistical dispersion can be measured by several different statistics, such as standard deviation, variance, and range. These are statistical term that depicts the distribution size of values.

However, there is another category of feature related to the complexity of time series that has been extracted such as entropy.

#### 3.3.1. Complexity of time series data

A time series refers to a collection of recorded measurements for one or more variables, organized based on their respective timestamps. The most common form of a time series consists of a sequence of evenly spaced measurements taken over time, resulting in a sequence of discrete-time data. Various examples of time series data exist, such as population, temperature, and the daily closing values of gold, silver, and oil.

A run chart is much used to plot the time series data. In addition, the time series is helpful in many sectors or applications such as finance and economy to predict the price of the stock, earthquake, signal processing, and weather forecasting.[74].

Complexity is one of the tools of time series analysis. And that there is great interest in measuring the complexity of many signals such as bird song, ECG, a protein sequence, or DNA. The evaluation of complexity in time series data can be categorized into three main categories: fractality (mono- or multi-fractality) for assessing self-similarity (long-term persistence or system memory), methods based on nonlinear dynamics (diagram descriptions or attractor invariants) for analyzing entropy (dynamical or structural entropy) and attractor properties in phase-space, for gauging the disordered state of a nonlinear system. In the case of discrete systems, Shannon's entropy has been employed to quantify their complexity [75]. Although the concept of entropy was originally developed in thermodynamics, it has been widely applied in various fields, with its meaning being adapted to the specific application area and providing tools for numerous applications [76]

### 3.3.2. Fractal dimensions

Fractals refer to highly complex patterns that exhibit self-similarity across different scales. The Fractal dimension serves as a metric for quantifying the degree of "complexity" inherent in a self-similar figure. The terms fractal dimension and fractal were coined by Mandelbrot in 1975[77]. Moreover, Humans are fractal. their circulatory system, lungs, and brains are like trees.

As shown in figure 3.17, the fractal dimensions for some regular shapes.

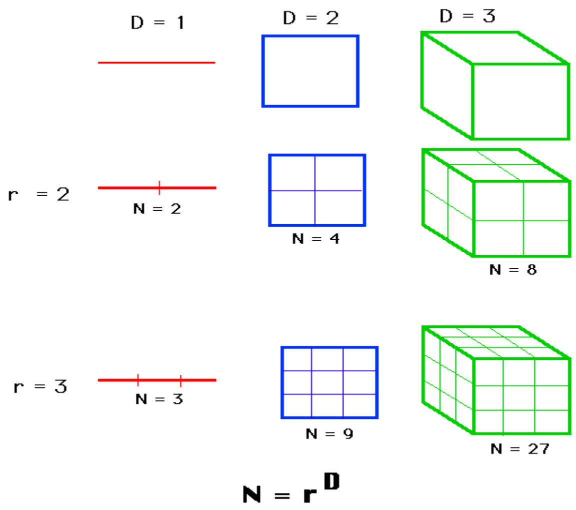



Figure 3.17 : Fractal dimensions for regular shapes.

A fractal is characterized as a point set that is self-similar, meaning that it is composed of components that closely resemble the original set of points. Fractal Dimension represents a method of gauging the complexity of a fractal.

There are also regular shapes such as rectangles and squares, but there are many irregular shapes and most of the shapes in nature are irregular. As shown in the figure 3.18, some examples of irregular shapes and their fractal dimension.

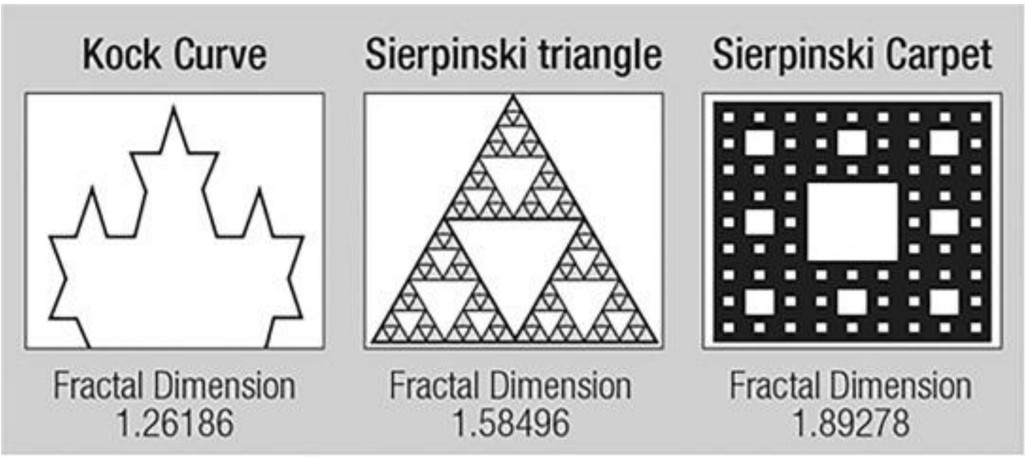



Figure 3.18 : Fractal dimensions for irregular shapes [78].

To scale a fractal dimensional shape by a scaling magnification factor, number of self similar pieces of the original shape needed will be given by equation 3.11:

$$Fractal\ dimension = \frac{\log(number\ of\ self\ similar\ pieces)}{\log(magnification\ factor)} \tag{3.11}$$

### 3.3.3. Entropy

This quantity represents the amount of a system's thermal energy that cannot be utilized for mechanical work, and is commonly understood as a measure of the system's disorder or randomness in thermodynamics. As shown in figure 3.19, the relationship between entropy and disorderliness is direct.

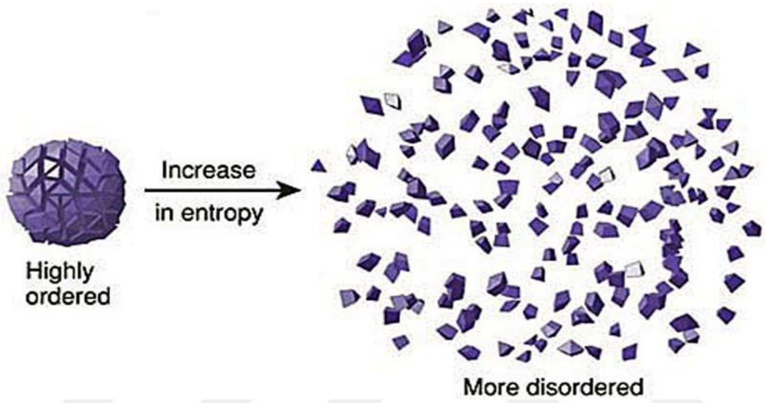



Figure 3.19 : the relationship between entropy and disorderliness.

Entropy is a critical notion within the realm of machine learning and data science, and it has generated significant interest, as well as sparked numerous debates. In machine learning, entropy is used as a metric to gauge the level of randomness in the information being processed. Higher levels of entropy correspond to a greater difficulty in drawing meaningful conclusions from the data. As such, entropy is a valuable tool in machine learning for a variety of applications, including feature selection, decision tree construction, and classification model fitting, among others [79].

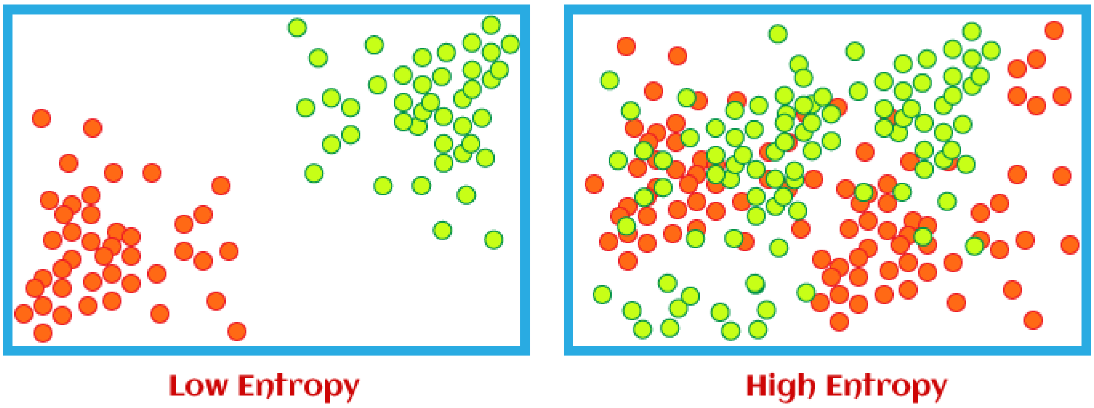



Figure 3.20 : Low vs high entropy.

### 3.3.4. Extra feature extraction

- **Petrosian fractal dimension:** Petrosian fractal dimension for a given time-series, x, is as follows:

$$P = \frac{\log_{10}(N)}{\log_{10}(N)+\log_{10}(\frac{N}{N+0.4N_{\delta}})} \tag{3.12}$$

This equation defines N as the the time series length, while Nδ represents the count of sign changes present in the derivative of the signal. [80].

- **Katz Fractal Dimension:** The Katz fractal dimension estimation method involves several steps. Firstly, the total Euclidean distances between consecutive points in the sample are added up and then divided by the number of points to obtain the average distance denoted by 'a'. Similarly, the sum of all these distances is represented by 'L'. Secondly, the maximum distance between the first point and any other point in the sample is determined and denoted by 'd'. Finally, the Katz sample fractal dimension is defined as D and can be calculated using the previously determined values of L, a, and d.

$$D = \frac{\log(^{L}/_{a})}{\log(^{d}/_{a})} = \frac{\log(n)}{\log(n)+\log(^{d}/_{L})} \tag{3.13}$$

- **Higuchi Fractal Dimension:** The computation of the fractal dimension for a time series is done through a direct method in the time domain using various mathematical equations, making the calculation process quite complex. That's why, the predefined package was used to calculate.

- **Permutation Entropy**: This method involves reconstructing the phase space of signal or a time series obtained from a physical system, taking into account its non-linear behavior, in order to describe its complexity. The calculation of this type of fractal dimensions is required a lot of mathematical equations. That's why, the predefined package was used to calculate.

- **Singular Value Decomposition entropy (SVD entropy):** The dimensionality of a dataset is quantified by the number of eigenvectors required to adequately explain it. In essence, It signifies the count of dimensions or characteristics present in the data. The SVD entropy of a signal x is defined as:

$$\boldsymbol{H = -\sum_{i=1}^{M} \bar{\sigma}_i \, log_2(\bar{\sigma}_i)} \tag{3.14}$$

The process of SVD involves breaking down a matrix A into three separate matrices, namely U, S, and V. The matrix S is represented as a diagonal matrix containing the singular values, as depicted in Figure 3.21.

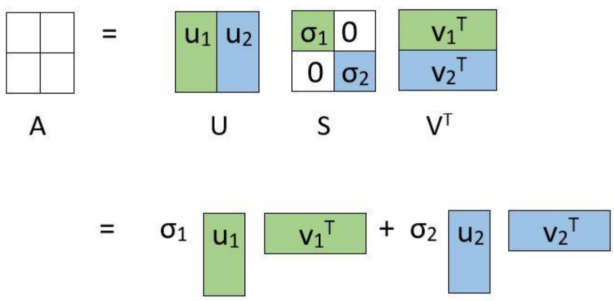



Figure 3.21 : the decomposition of a matrix A into three matrices, namely U, S, and V, as a part of the Singular Value Decomposition (SVD) process.

Here, 'M' represents the count of singular values present in the embedded matrix 'Y', and $\sigma 1, \sigma 2, \ldots, \sigma M$ signify the normalized singular values of 'Y'.

To create the embedded matrix 'Y', as follows:

$$y(i) = [x_i, x_{i+\text{delay}}, \ldots, x_{i+(\text{order}-1)*\text{delay}}]$$
$$Y = [y(1), y(2), \ldots, y(N - (\text{order} - 1)) * \text{delay})]^T \tag{3.15}$$

- **Approximate Entropy (ApEn):** The purpose of this method is to gauge the level of regularity and randomness in fluctuations observed in time-series data. It involves numerous mathematical equations, which can be quite complex. Consequently, a pre-existing package was employed to perform the required entropy calculations.

- **Sample Entropy (sampEn)**: It is a complexity measure that is derived from the approximate entropy (ApEn). This method is commonly used to assess the complexity of physiological time-series signals, with the goal of identifying abnormal states associated with various diseases. Unlike ApEn, SampEn does not take self-similar patterns into account. Calculating this type of entropy involves a significant number of mathematical equations, which is why a pre-defined package was utilized for its computation.

### 3.4. Extracting Extra Complex Features

As mentioned section 3.2, the dataset was converted from signal to vector form to be in shape (2,904 X 160 X 57). That means the dataset was divided into 160 windows and each window has 5000 voltage values. Moreover, from each phase single 19 features were extracted, and for 3 phases 19 X 3=57, but these features are related to central tendency and statistical dispersion.

**Complex extracting features:** As depicted in Figure 3.14, once more intricate characteristics that are linked to fractal dimension and entropy in time series were extracted, The original 57 features were used to derive an additional 7 features for every phase.

**Output 2:** the final result from the complex extraction stages is the feature data is in shape (2,904 X 7).

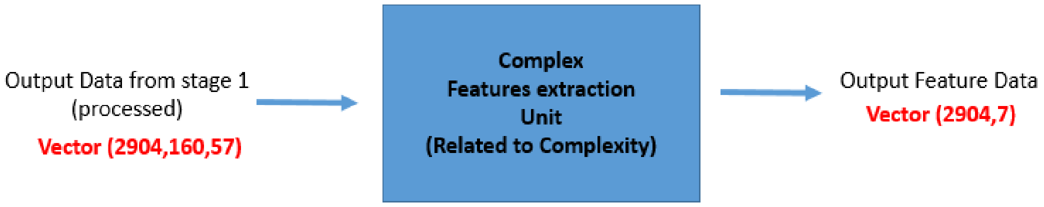



Figure 3.22 : Complex feature extraction block diagram.

# CHAPTER 4. AI MODEL DESIGN AND TRAINING

## 4.1. Introduction

In the previous chapter, the raw datasets were pre-processed and features extracted. In feature extraction stage there are two ways. The first one is divided dataset into 160 windows and features extracted from each 5000 voltage values to produce vector in shape (2904, 160, 57) and the second one is divided dataset into 320 windows and feature extracted from 2500 voltage values to produce vector in shape (2904, 320, 57).And then, extra 7 features extracted related to complexity from the 57 features which extracted before. The result of that, there are four input data as shown in figure 4.1.

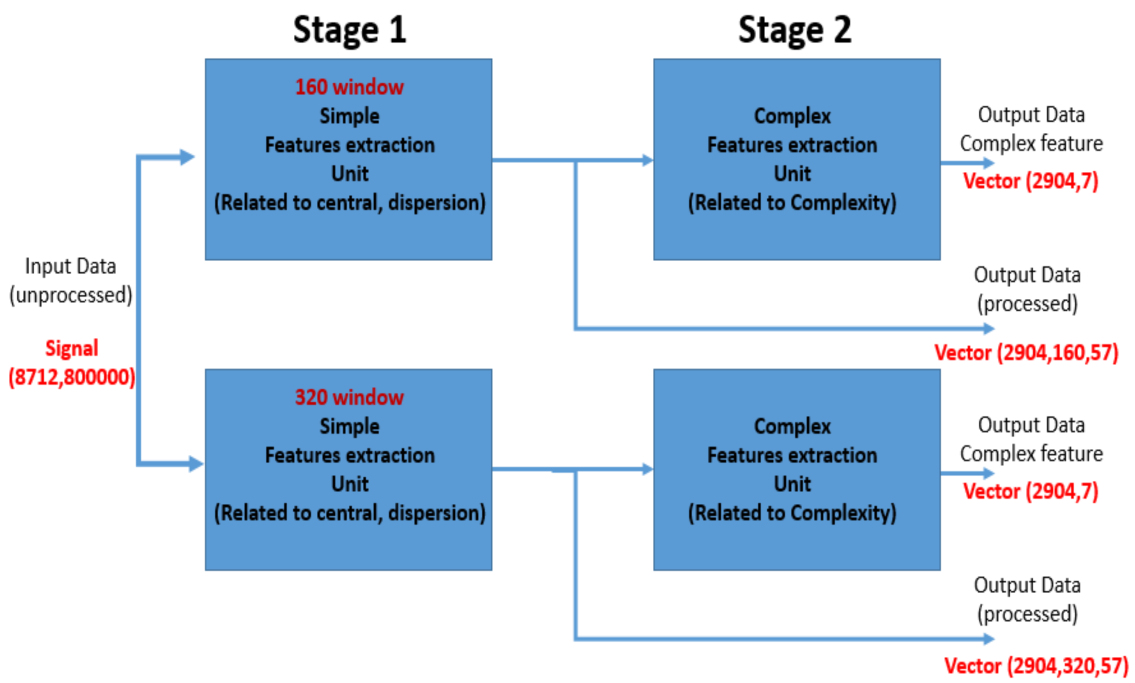



Figure 4.1 : Overall preprocessing dataset block diagram.

In this chapter, the AI model used in binary classification will be discussed and several deep neural networks will be designed and trained after comparing different neural network architecture designs, and the best performance and accuracy of the AI model will be used as a classifier for existence detection from partial discharge phenomenon.

## 4.2. AI Model Structure

In general and mainly, recurrent neural network RNN type was proposed in designing AI model but particularly, Bidirectional Long short-term memory (BiLSTM) preferred to RNN because vanishing gradients was a series problem and which makes the network not learn significantly. In addition to this, the BiLSTM type of recurrent neural network provides better predictions than regular LSTM-based models. Moreover, there are other types of neural networks that were used to build and design AI models.

### 4.2.1. Layer types

The layers in an AI model can be thought of as the structure of the model. There can be different types of layers that can be used in models. All these different layers have their significance based on their features.

1. **Bidirectional LSTM layer :** In short, it consists of two LSTMs in the forward and the backward direction and also can handle sequence time series data. This in turn gives the strong advantage of increasing the amount of training data because it is processed from both directions. Thus, it will produce a strong model and work better than the LSTM, but takes more time during training.

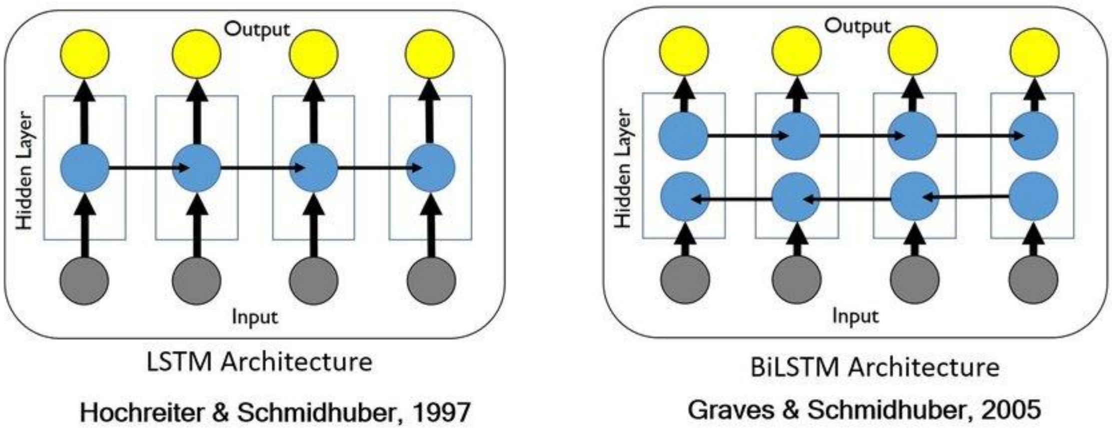



Figure 4.2 : LSTM vs BiLSTM architecture [81].

2. **Attention layer :** It is a layer that offers a great advantage in that it can give some information more important than others or even ignore it completely. In general, it simulates cognitive attention.

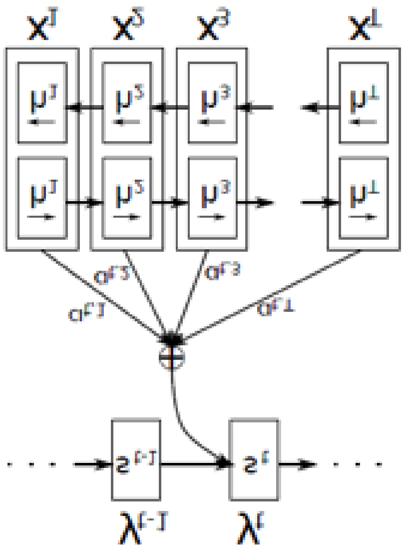


Figure 4.3 : Attention layer architecture [82].

3. **Dense layer :** It is the simplest type of layer that is used in building an artificial intelligence model. It is also used in most networks because it is considered a link between the previous layers and the output, and its function is to change the dimensions of the outputs according to the purpose.

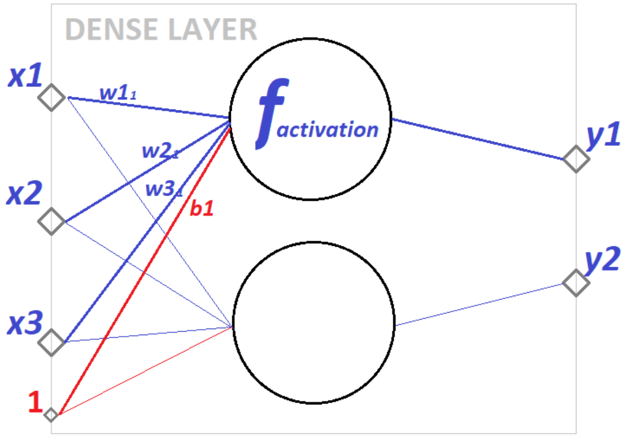



Figure 4.4 : Dense layer architecture [83].

### 4.2.2. Keras model structure types

Neural networks are very complex in architecture and computationally expensive. Since the Keras library was used to build the model. In addition to this, Keras offers three ways to build neural networks. They are as follows: Sequential, Functional API, and Model subclassing.

#### 4.2.2.1. Sequential API

This allows models to be created in successive layers. This approach is limited in that it does not provide the option to create a model with multiple inputs or outputs, nor does it allow for layer overlap. This approach is used in many with many applications because it is simple.


Figure 4.5 : Sequential API.

### 4.2.2.2. Functional API

The functional API provides greater flexibility and potency than the sequential API, as it can incorporate multiple inputs and outputs and allows for layer branching and sharing.

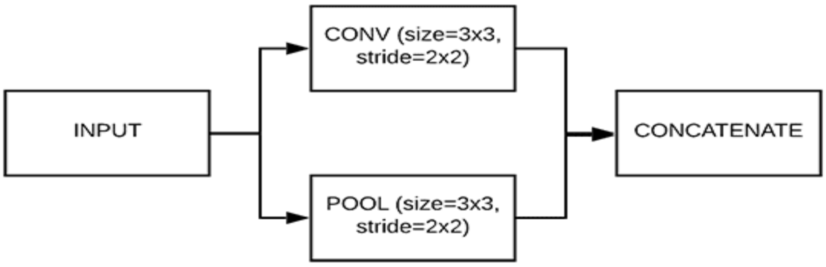



Figure 4.6 : Functional API.

### 4.2.2.3. Model subclassing

Simply put, this approach enables developers to create a customized model architecture, but it is more complex than previous methods. An example of this is the Attention layer, which is a subclass of the model.

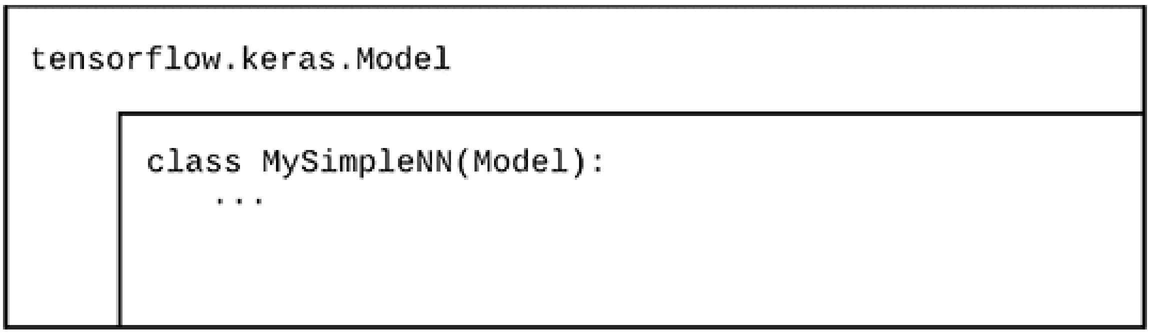



Figure 4.7 : Model subclassing.

## 4.3. Deep Nerual Network Design

The AI model was developed and built using the Functional API because a high-performance partial discharge classifier model requires a multi-input approach to balance the extracted features that relate to central tendency with features that relate to data complexity and entropy. Additionally, two types of data were produced during the pre-processing and feature extraction stages. Consequently, three functional API models were developed, trained under identical conditions, and compared. The most accurate model was selected as the final classifier to be relied upon. As monitored in previous chapters the training dataset is little instructional data and is imbalanced. When the distribution of the classes is observed. Whereas found that out of the 8712 signals, only 525 signals are class 1, representing only 6% of the total data (imbalance data) and that is expected because partial discharge is not a normal thing to be as shown in figure 4.8.

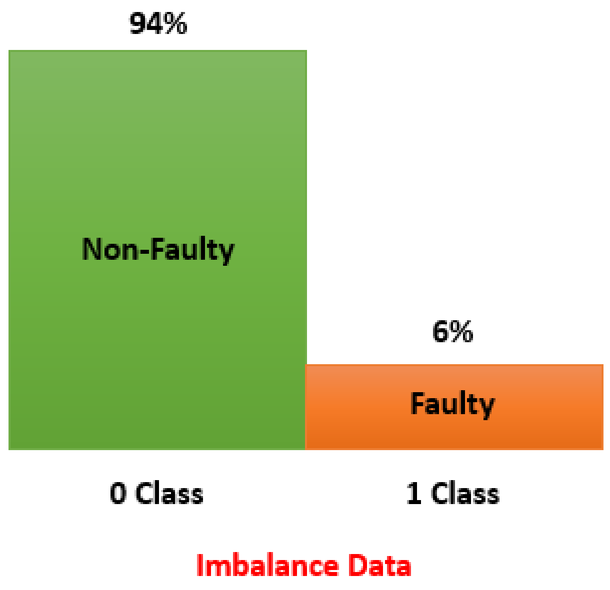



Figure 4.8 : Bar chart input data distribution.

In order to override these problems stratified K-Folds technique was used. Since both input data were divided into K-fold times. So the AI model was trained K-times and K is set 5 and at each K-fold the model has trained 30 epochs and saved the best models of each fold to load them together. In order to determine which model is the best, a simple solution is to calculate the Matthew Correlation Coefficient( MCC) value at every instructional cycle (Epoch) and record the model update with the highest MCC value. All three Deep learning network models were trained under the same conditions as follows:

**EPOCH:** Means how many time the model have been trained. At 30 epochs all models have been trained.

**Optimizer:** It is an algorithm responsible for adjusting the weights of connections between layers and cells until obtaining acceptable accuracy, and it also helps in improving model performance by reducing loss. Adam optimizer was used at a learning rate equal to 0.001.

### 4.3.1. K-fold validation

K-fold approach provides the ability to use the dataset in training and uses an amount from it for verification purposes. The great importance of this approach appears with a small dataset, as following this extends the data theoretically. For example, if the dataset is divided into five folds as followed in this thesis, that means the accuracy is calculated five times and the overall accuracy is the average.

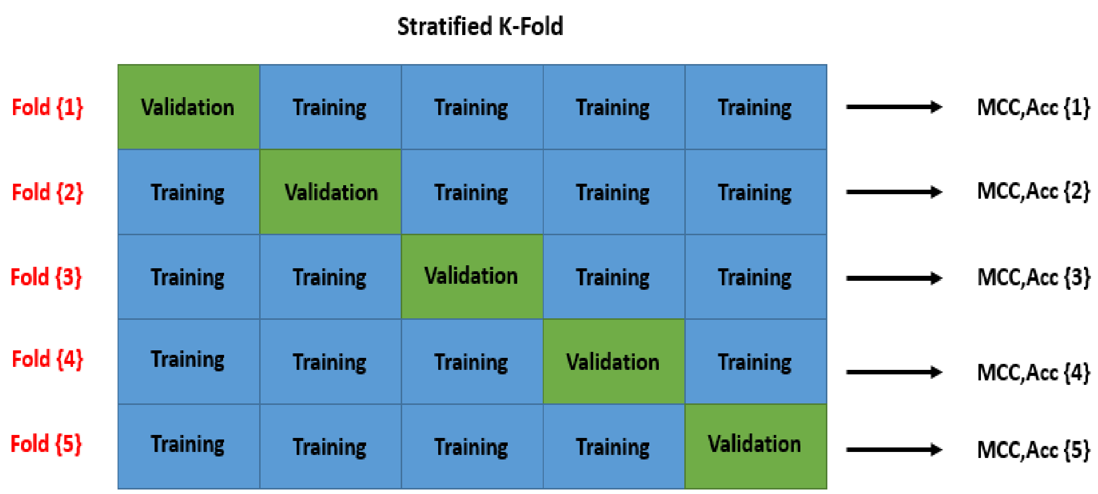



Figure 4.9 : Stratified training data in K-fold.

### 4.3.2. Matthews correlation coefficient (MCC)

Whereas, accuracy reflects model performance and predictability. There are many statistical tools that can measure performance in the field of deep learning such as accuracy, F1 score, and recall but the most common one is accuracy. Accuracy doesn't work well every time, especially when dealing with unbalanced data and in this thesis the data in there is unbalanced, so depending on the accuracy in measuring how the model works would be wrong. In 1975, a new tool proposed by Brian W. Matthews is called the Matthews Correlation Coefficient (MCC) for evaluating performance. Moreover, MCC is not sensitive to unbalanced data because it takes into account four parameters as follows: true negatives (TN), true positives (TP), false negatives (FN), and false positives (FP).

$$\boldsymbol{MCC} = \frac{\boldsymbol{TN\ X\ TP - FP\ X\ FN}}{\sqrt{(\boldsymbol{TN+FN})(\boldsymbol{FP+TP})(\boldsymbol{TN+FP})(\boldsymbol{FN+TP})}} \tag{4.1}$$

The closer the value is to one, the more reliable the model works, but if the value is zero, this means that the model does not learn and gives random results, nothing more. In addition, if the value approaches one from the negative side, this also means that the system is working fine, but in reverse, and this can be fixed by reversing the output.

### 4.3.3. Activation function

It is responsible for activating some neurons, and disabling others as required. There are many types of activation functions and the preference of one over the other during the design of the neural network depends on the purpose of the neural network. Examples of the types of functions are ReLU, liner, sigmoid, etc. and each type of them has specific characteristics. As shown in figure 4.10, in ReLU, if the input is positive, it will come out as it is, but if it is negative, the output will become zero and thus the model will learn better and faster because it overcomes the vanishing gradient.

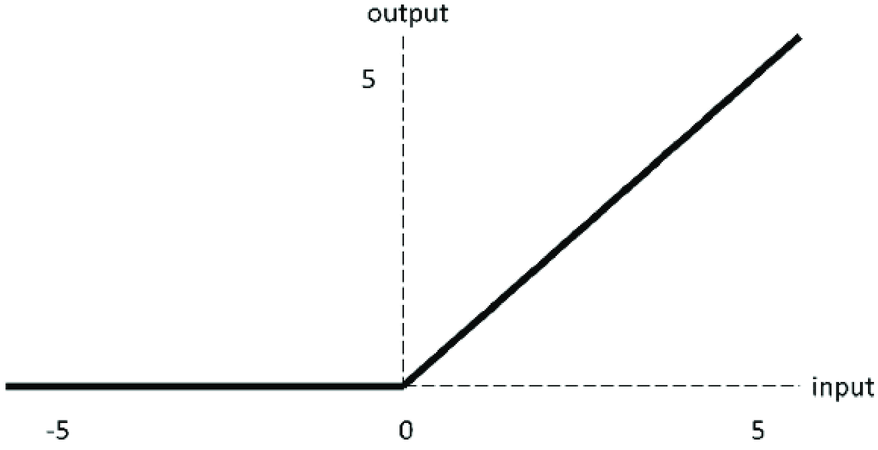



Figure 4.10 : ReLU activation function.

Figure 4.11 demonstrates that the sigmoid function is commonly utilized in the output layer since it produces values between 0 and 1, which is ideal for determining output probabilities. Additionally, the sigmoid function is well-suited for binary classifications and regression tasks.

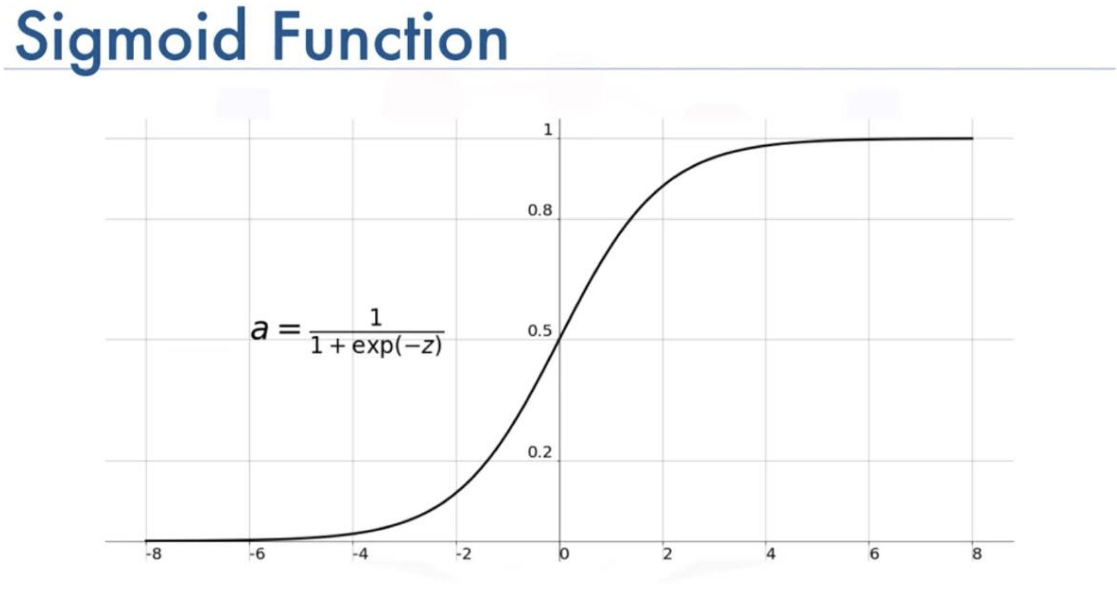



Figure 4.11 : Sigmoid activation function.

### 4.4. Deep Neural Networks Training and Validation

The three models will undergo a comparison process based on several factors such as accuracy, training time, and other considerations, with the best model ultimately being selected. Given that the input data consists of time-series data, BiLSTM is an excellent choice for the model due to its numerous properties, as mentioned previously.

#### 4.4.1. Model_160

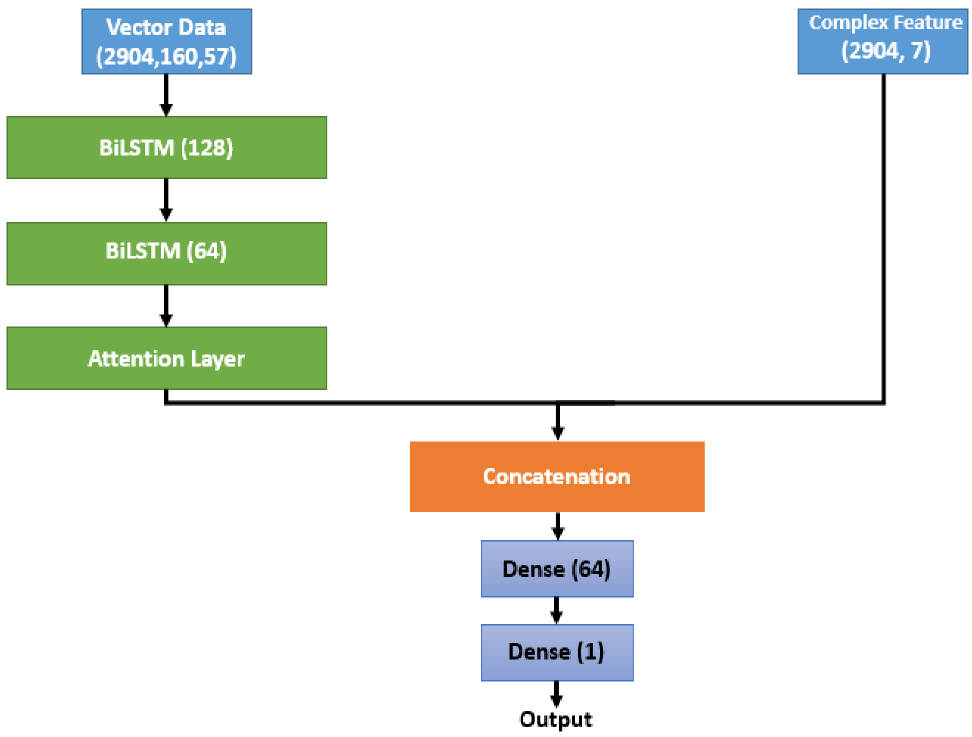



Figure 4.12 : Model_160.

As shown in figure 4.12, the model_160 which deals with data in shape(2904,160,57) and (2904,7) consists of many layers as follows:

BiLSTM layer with 128 cells which receives input data vector (2904,128,57), then

BiLSTM layer with 64 cells which receive the output of previous BiLSTM(128) layer, then

The attention layer receives the output of the previous BiLSTM(64) layer, after that

Concatenation layer which connects the output of the attention layer with the complex feature data in shape(2904,7), then

The model includes a dense layer that has 64 cells and uses ReLu as an activation function. This layer accepts a sequence of vectors as input and converts it into a single vector output. Additionally, there is a dense layer with one cell that uses sigmoid as an activation function. This layer serves as the output layer and predicts whether the input belongs to class 0 or 1. According to the model summary in Figure 4.13, the total number of parameters that can be trained is 363,873.

| Layer (type) | Output Shape | Param # | Connected to |
|---|---|---|---|
| input_1 (InputLayer) | [(None, 160, 57)] | 0 | [] |
| bidirectional (Bidirectional) | (None, 160, 256) | 190464 | ['input_1[0][0]'] |
| bidirectional_1 (Bidirectional ) | (None, 160, 128) | 164352 | ['bidirectional[0][0]'] |
| attention (Attention) | (None, 128) | 288 | ['bidirectional_1[0][0]'] |
| input_2 (InputLayer) | [(None, 7)] | 0 | [] |
| concatenate (Concatenate) | (None, 135) | 0 | ['attention[0][0]', 'input_2[0][0]'] |
| dense (Dense) | (None, 64) | 8704 | ['concatenate[0][0]'] |
| dense_1 (Dense) | (None, 1) | 65 | ['dense[0][0]'] |

Total params: 363,873
Trainable params: 363,873
Non-trainable params: 0

Figure 4.13 : Model_160 summary.

#### 4.4.1.1. Training and validation model_160

The model_160 has been trained in each fold 30 times and during the training the MCC was continually being checked and at best MCC the model weights was saved in .h file.

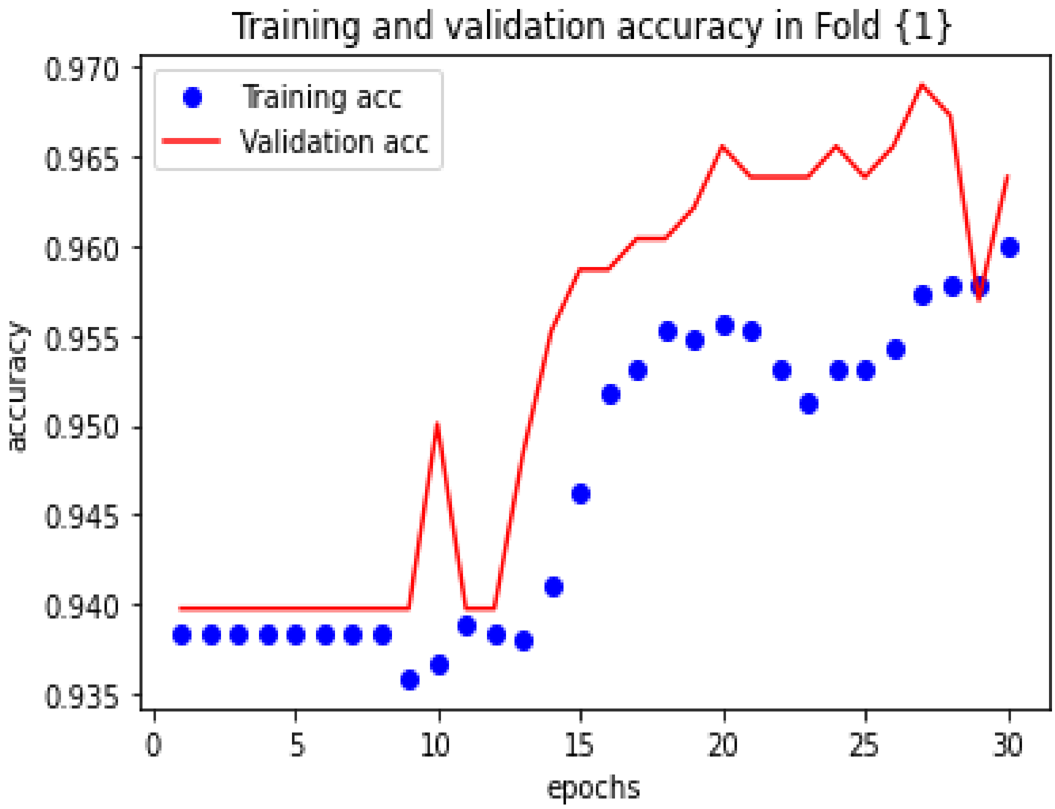



Figure 4.14 : The model_160 training and validation accuracy in fold 1.

Figure 4.14 presents a chart displaying the training_accuracy and validation_accuracy during the training of fold 1. The model attained its highest training_accuracy of 0.96 at the 30th epoch, while the maximum validation accuracy of 0.97 was observed at the 26th epoch. Additionally, the MCC for this fold was 0.56. The fact that the training_accuracy and validation_accuracy values are not close points out that the system doesn’t fits the data well in this fold.

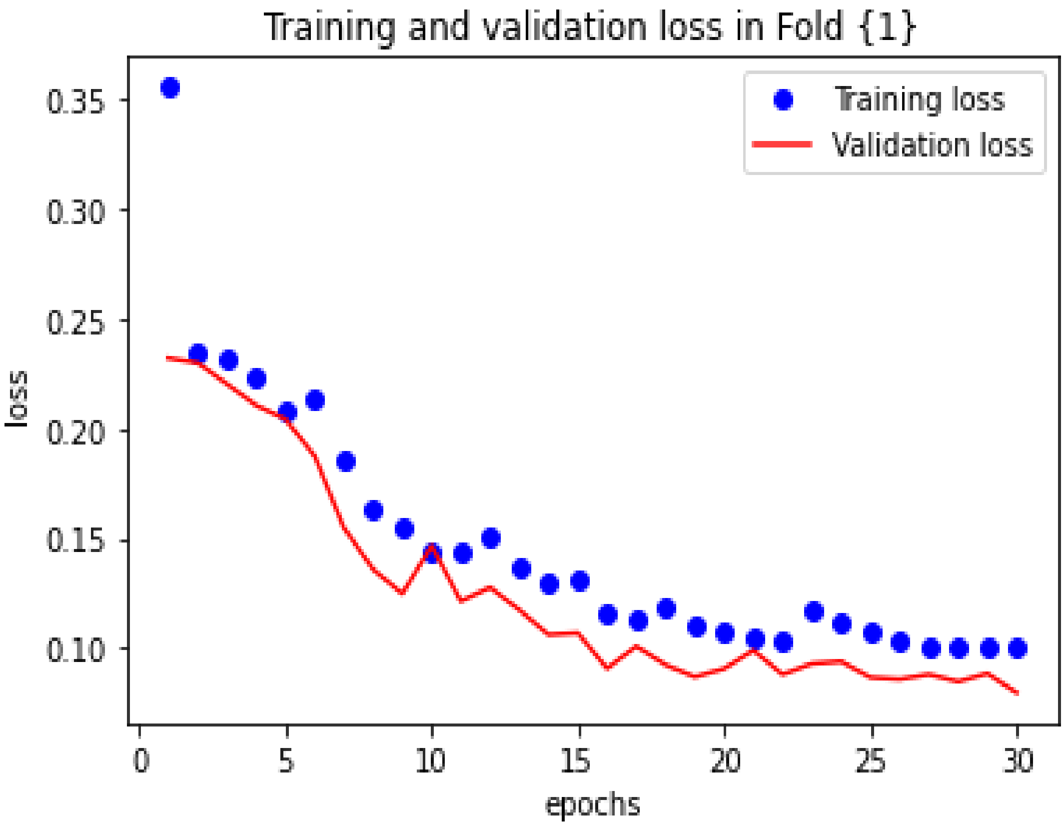



Figure 4.15 : The model_160 training and validation loss in fold 1.

The chart presented in figure 4.15 shows the loss_ validation and loss_training values the model during fold 1's training process. At the 30th epoch, the model attained its highest training loss of 0.11, while the maximum validation loss of 0.09 was also observed at the same epoch. Additionally, the small variation between the loss_training and loss_validation values.

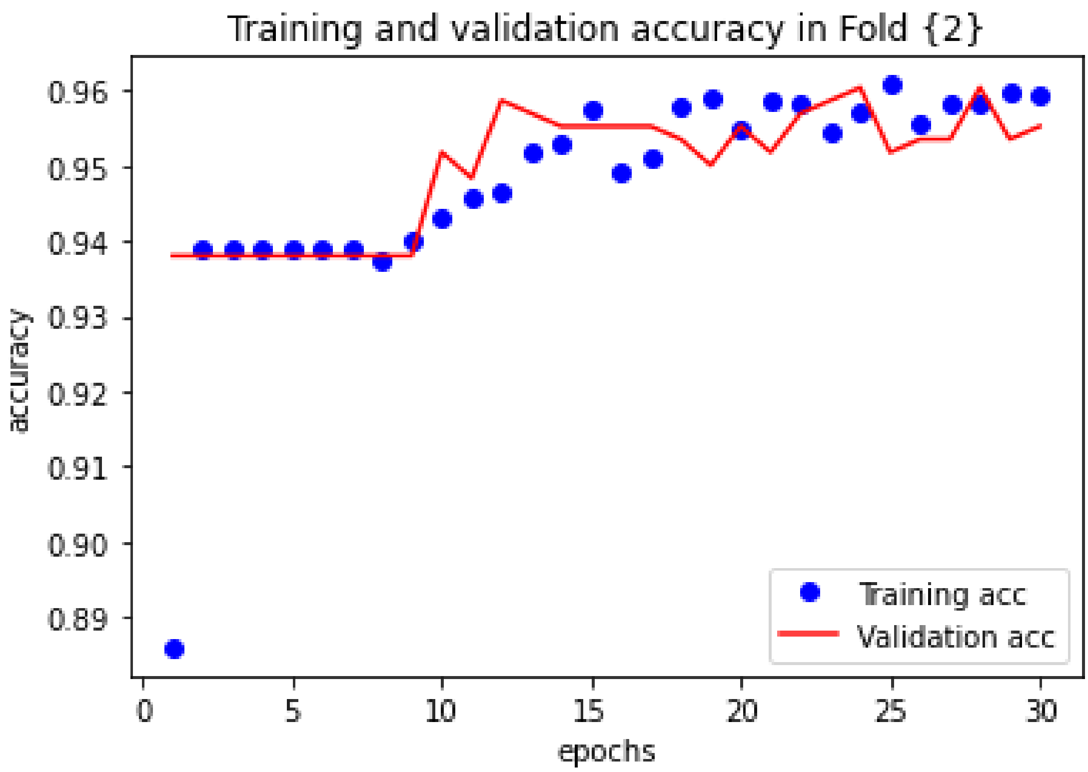



Figure 4.16 : The model_160 training and validation accuracy in fold 2.

Figure 4.16 illustrates the accuracy(ACC) for both the training and validation of the system during fold 2's training process. The model achieved its highest training accuracy of 0.96 at the 30th epoch, and the maximum validation accuracy of 0.96 was also observed at the same epoch. Furthermore, the MCC for this fold was 0.51. The training ACC and validation ACC values are close, not like previous fold.

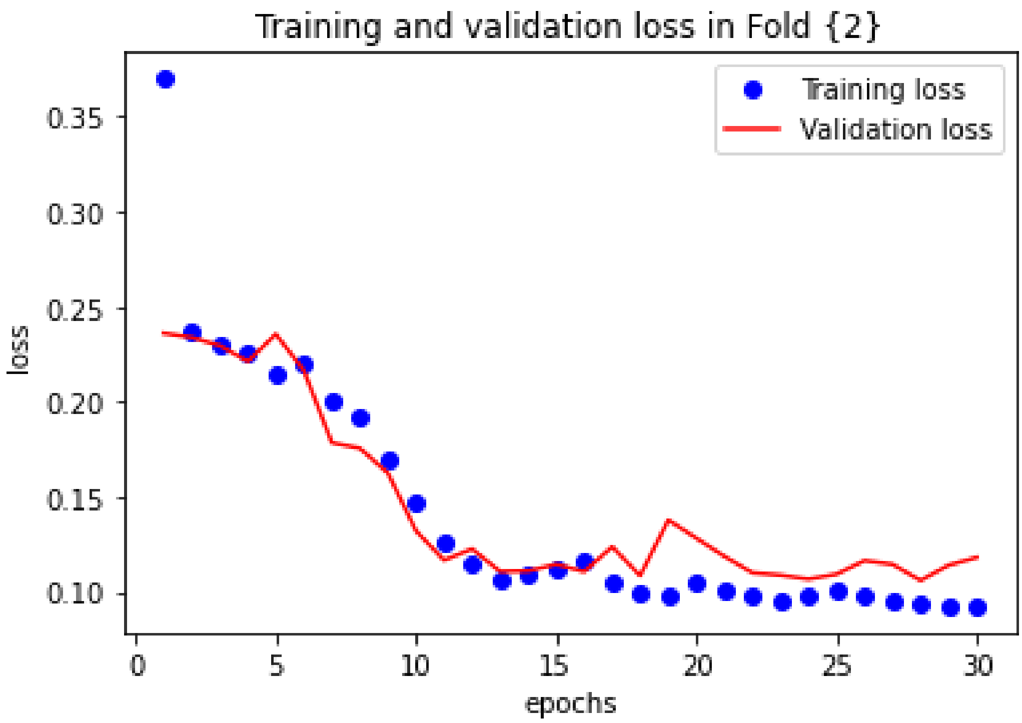



Figure 4.17 : The model_160 training and validation loss in fold 2.

Figure 4.17 shows the loss_training and loss_validation during the training of fold 2. The model attained its highest training loss of 0.09 at the 30th epoch, whereas the maximum validation loss of 0.12 was observed at the 28th epoch. Additionally, the small discrepancy between the loss_validation and loss_training values as pervious.

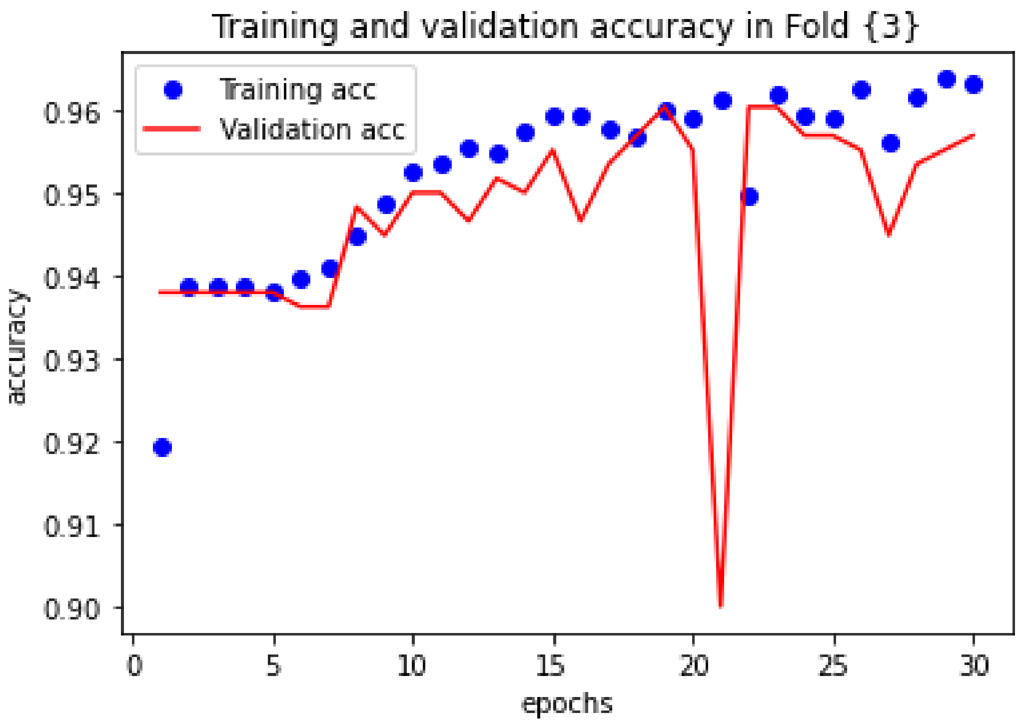



Figure 4.18 : The model_160 training and validation accuracy in fold 3.

The chart presented in figure 4.18 depicts the ACC_train and ACC_val values of the AI system through the process of fold 3. The model achieved its highest training accuracy of 0.965 at the 30th epoch and its maximum validation accuracy of 0.955 at the 24th epoch. At the 22nd epoch, there was a drop in the validation accuracy to 0.9, while the training accuracy remained at 0.96. However, the model's performance subsequently improved. The MCC for this fold was 0.49. Overall, the close proximity of the training and validation ACC values points out that the system is well-suited as pervious.

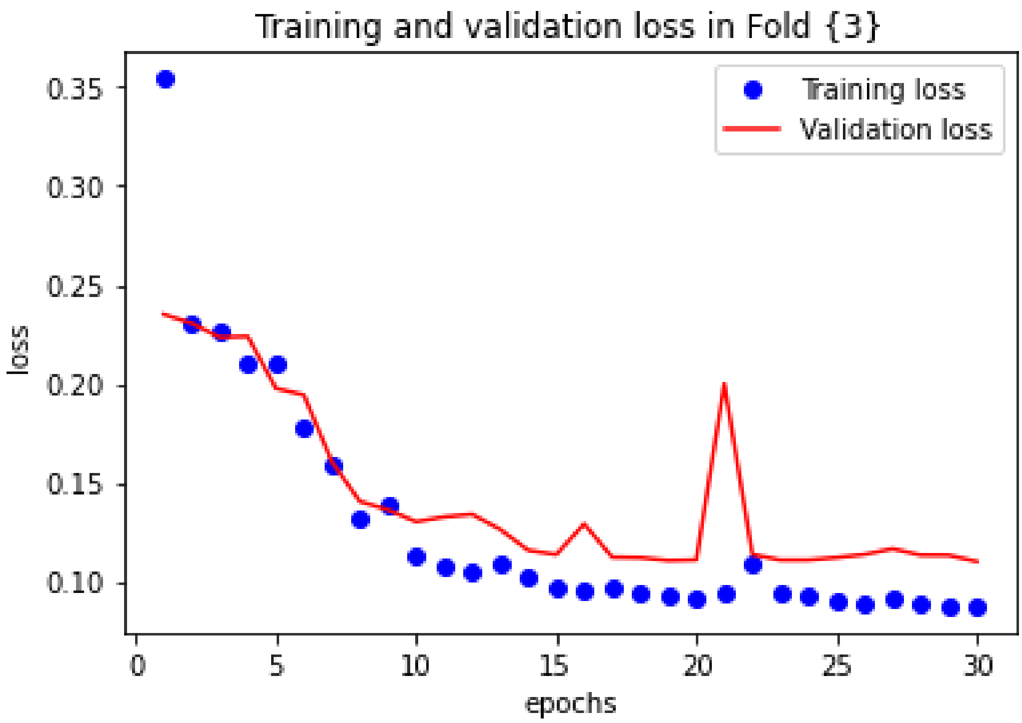



Figure 4.19 : The model_160 training and validation loss in fold 3.

The chart shown in figure 4.19 depicts training and validation loss values of the model during fold 3. At the 30th epoch, the maximum training loss was 0.09, while the maximum validation loss was observed at the 28th epoch and reached 0.12. Additionally, the small variation between the loss-training and loss-validation values points out that the model is as in fold 2 .

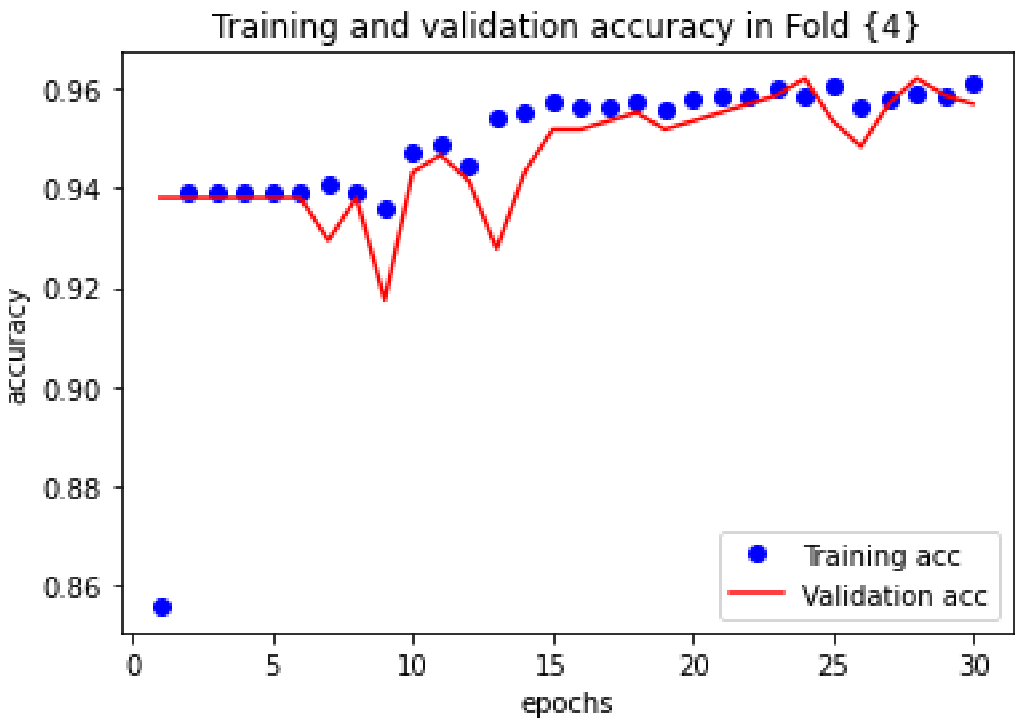



Figure 4.20 : The model_160 training and validation accuracy in fold 4.

The chart in figure 4.20 depicts the ACC of the system throughout the training in fold 4. The highest training ACC was achieved at the 30th epoch, reaching 0.965, while the maximum validation accuracy of 0.955 was attained at the 24th epoch. However, at the 22nd epoch, the validation accuracy decreased to 0.9, while the training accuracy remained at 0.96, after which the model's performance improved. Additionally, the MCC for this fold was 0.49. Overall, the validation and training accuracies were similar.

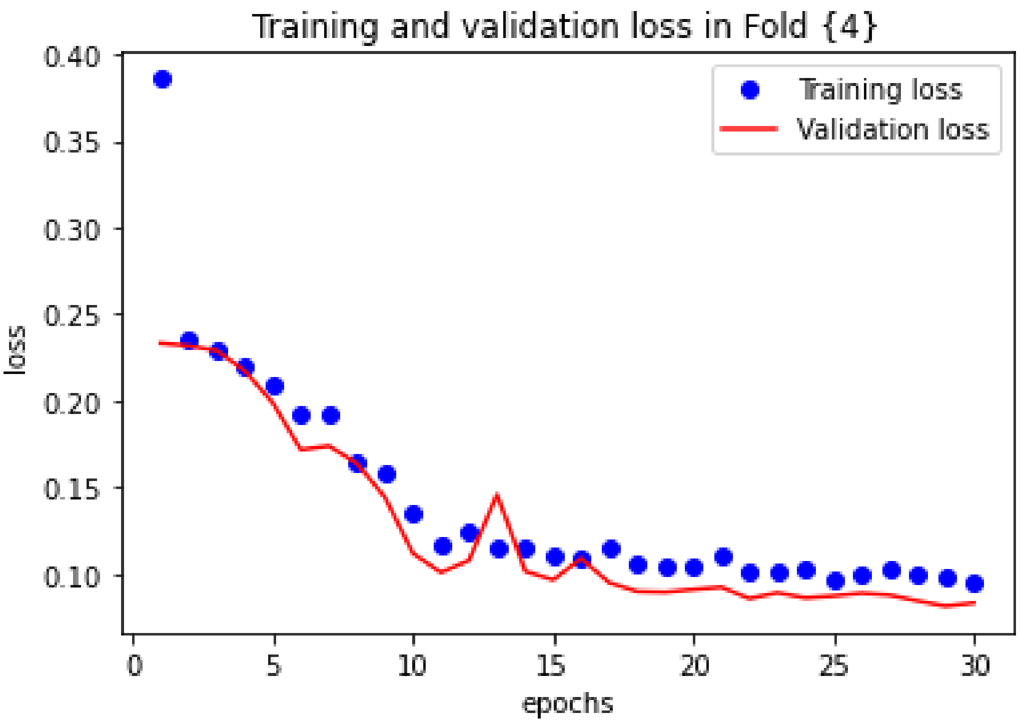



Figure 4.21 : The model_160 training and validation loss in fold 4.

The chart displayed in figure 4.21 exhibits the loss_validation and loss_training through the training in fold 4. At the 30th epoch, the model's maximum training loss reached 0.09, while the maximum validation accuracy was 0.12 at the 28th epoch. Furthermore, the loss-training and loss-validation values are similar as in fold 3.

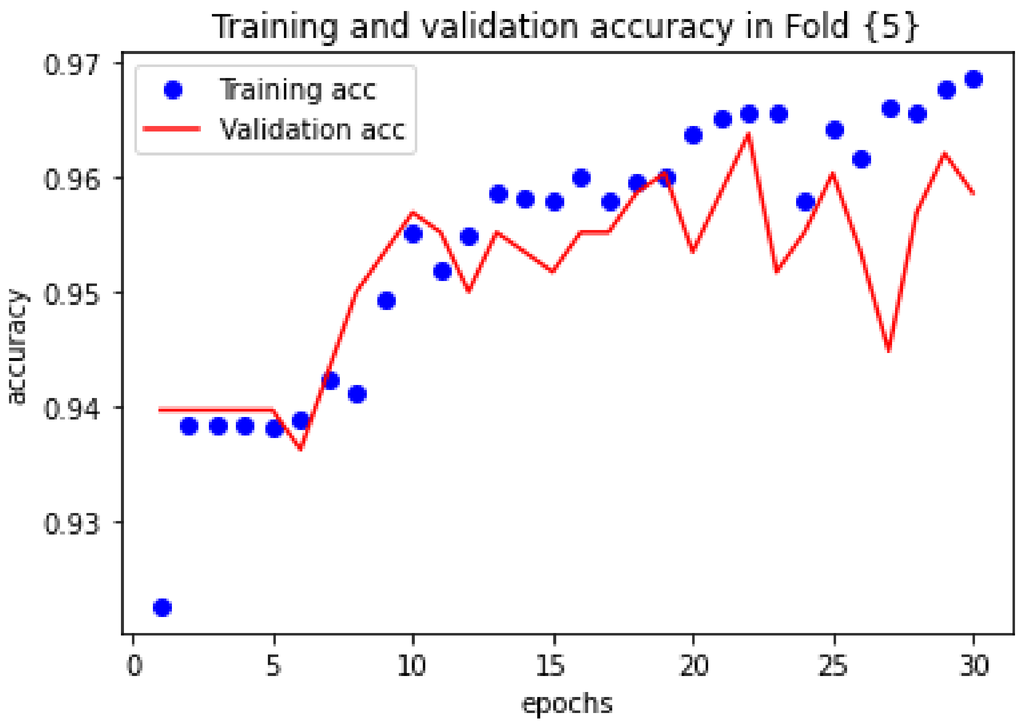



Figure 4.22 : The model_160 training and validation accuracy in fold 5.

The chart displayed in Figure 4.22 shows the accuracy for validation and training in fold 3. The model's highest training accuracy was 0.965 at the 30th epoch, while the highest validation ACC was 0.955 at the 24th epoch. At the 22nd epoch, the validation accuracy dropped to 0.9, but the training accuracy remained at 0.96, and the model subsequently improved. Additionally, the MCC in this fold is 0.49. Overall, the model's training and validation ACC are close.

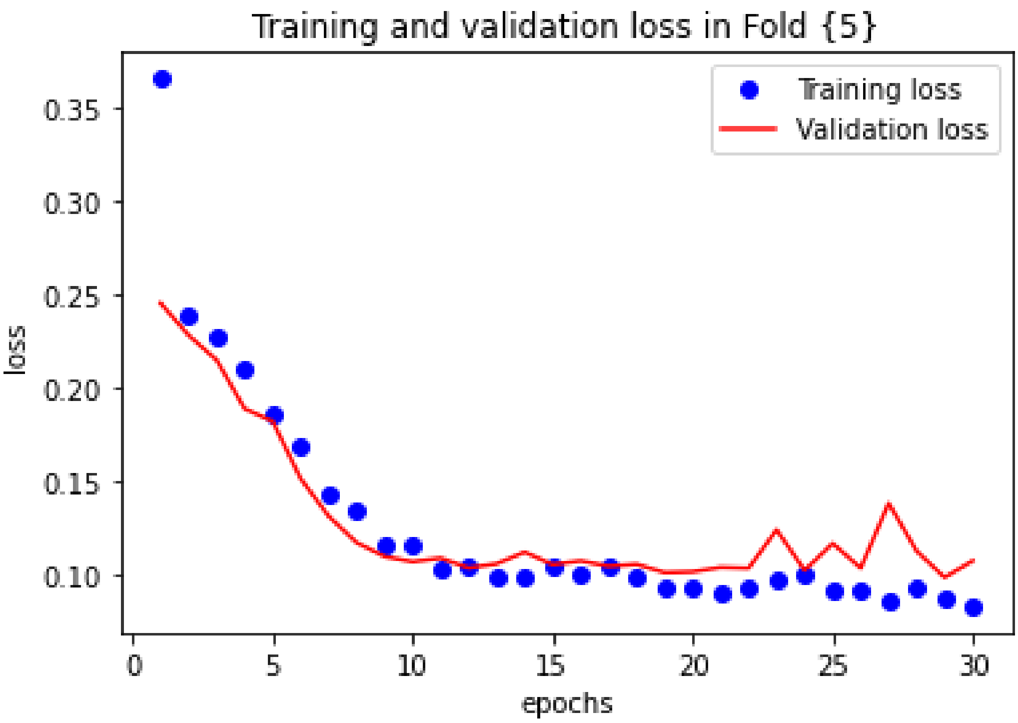



Figure 4.23 : The model_160 training and validation loss in fold 5.

The chart in figure 4.23 depicts the loss_ training and loss_validation while training the model in fold 5. At the 30th epoch, the maximum training loss for the model was 0.09, but the maximum validation accuracy was 0.12 at the 28th epoch. Additionally, the validation and training loss are closely matched.

#### 4.4.1.2. Testing model_160

Upon completing the training of model_160, the weights were saved for each fold. This resulted in the production of five files in the .h extension, which can be used for testing the AI system. The testing of the model_160 produced 1023 faulty lines. The bar chart, as depicted in figure 4.24, illustrates the distribution of the classifier's output.

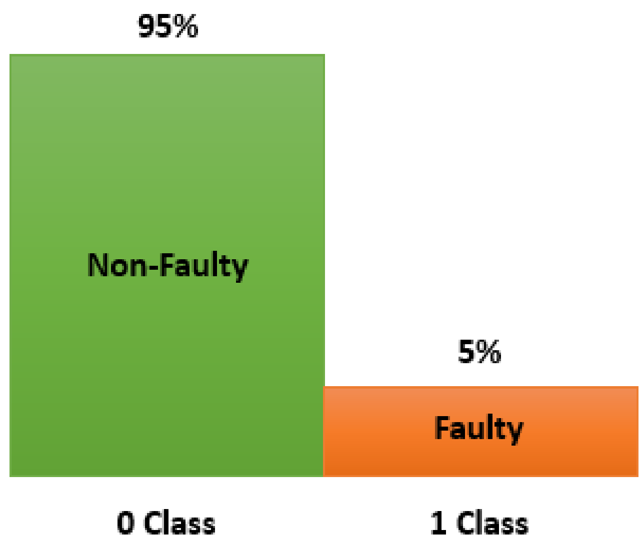



Figure 4.24 : The bar char of the output distribution of the model_160 .

### 4.4.2. Model_320

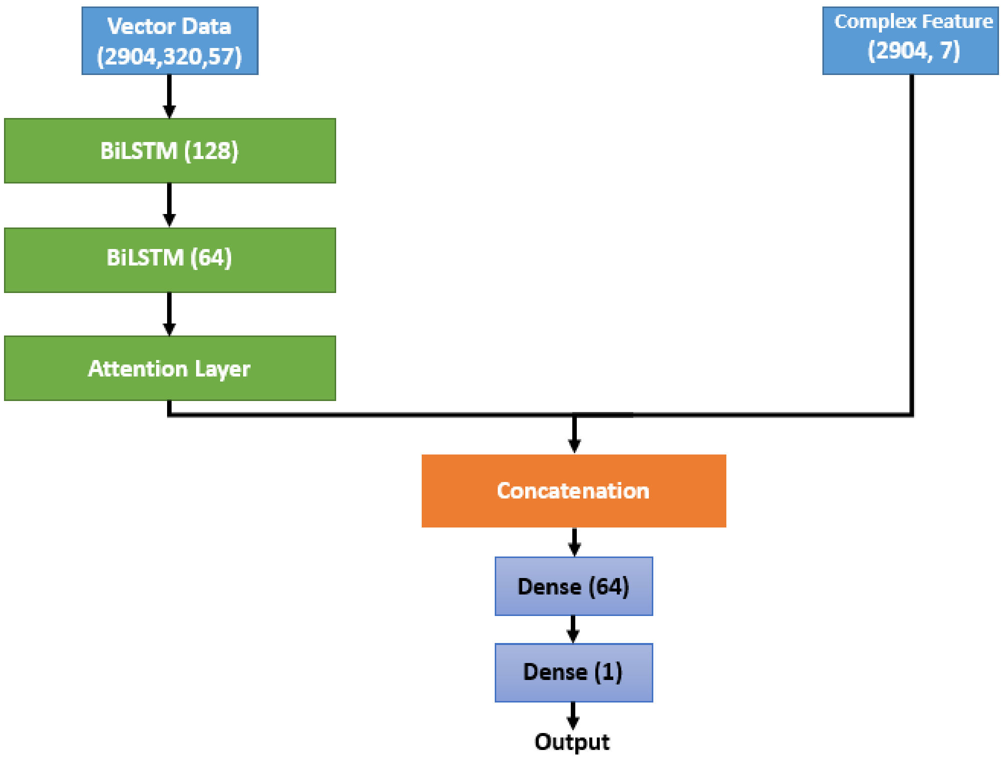



Figure 4.25 : Model_320.

As shown in figure 4.25, the model_320 which deals with data in shape(2904,320,57) and (2904,7) consistes from many layers as follow:

BiLSTM layer with 128 cell which receive input data vector (2904,128,57), then

BiLSTM layer with 64 cell which receive the ouput of previous BiLSTM(128) layer, then

Attention layer which receive the ouput of previous BiLSTM(64) layer,after that

Concatenation layer which connect the output of attention layer with the complex feature data in shape(2904,7), then

The model includes a dense layer with 64 cells that uses ReLu as an activation-function. This layer is used because the LSTM model also inputs a sequence of vectors, and the dense layer can convert this input into a single vector output. Additionally, there is a dense layer with one cell that uses sigmoid as an activation function. This layer serves as the output layer and predicts whether the input belongs to class 0 or 1. The model summary is shown in Figure 4.26, and it reveals that the total no. of trainable parameters is 364033.

```
 Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)            [(None, 320, 57)]    0           []

 bidirectional (Bidirectional)   (None, 320, 256)     190464      ['input_1[0][0]']

 bidirectional_1 (Bidirectional   (None, 320, 128)    164352      ['bidirectional[0][0]']
 )

 attention (Attention)           (None, 128)          448         ['bidirectional_1[0][0]']

 input_2 (InputLayer)            [(None, 7)]          0           []

 concatenate (Concatenate)       (None, 135)          0           ['attention[0][0]',
                                                                   'input_2[0][0]']

 dense (Dense)                   (None, 64)           8704        ['concatenate[0][0]']

 dense_1 (Dense)                 (None, 1)            65          ['dense[0][0]']

==================================================================================================
Total params: 364,033
Trainable params: 364,033
Non-trainable params: 0
```

Figure 4.26 : Model_320 summary.

#### 4.4.2.1. Training and validation model_320

The model_320 has been trained in each fold 30 times and during the training the MCC was continually being checked and at best MCC the model weights was saved in .h file.

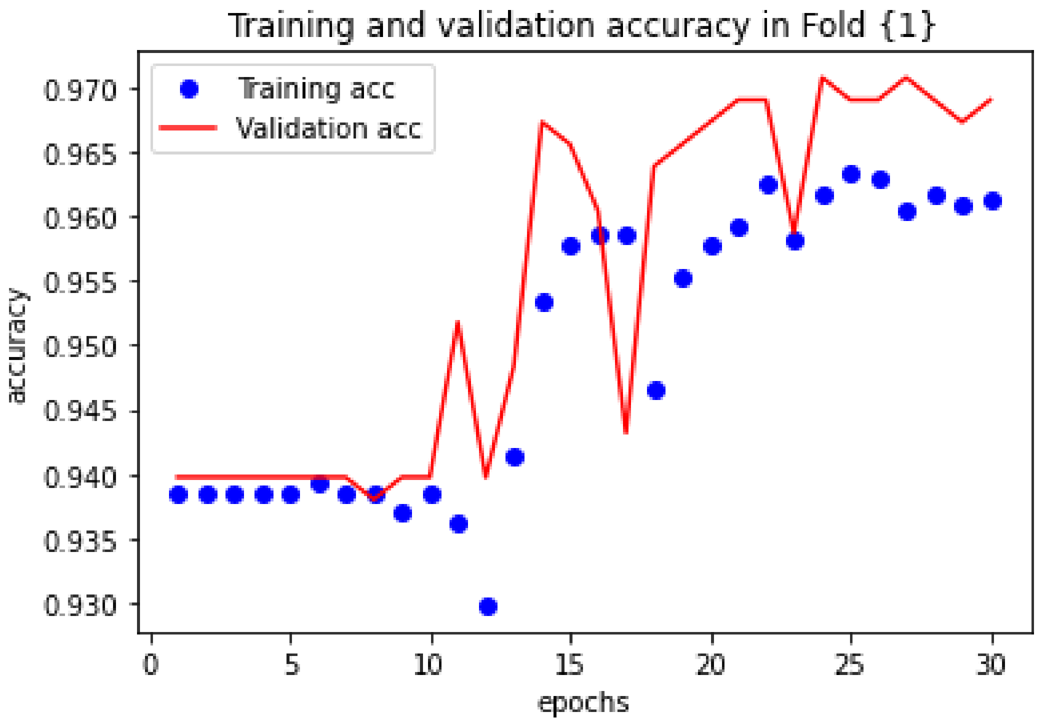



Figure 4.27 : The model_320 training and validation accuracy in fold 1.

The chart in figure 4.27 represents the ACC of the validation and training throughout the training in fold 1. The max training ACC achieved by the model is 0.96 at the 30th epoch, while the maximum validation accuracy is 0.97 at the 26th epoch. Additionally, the MCC for this fold is 0.56. The training and validation ACC are not so close but proximity points out that the system is well-fitted.

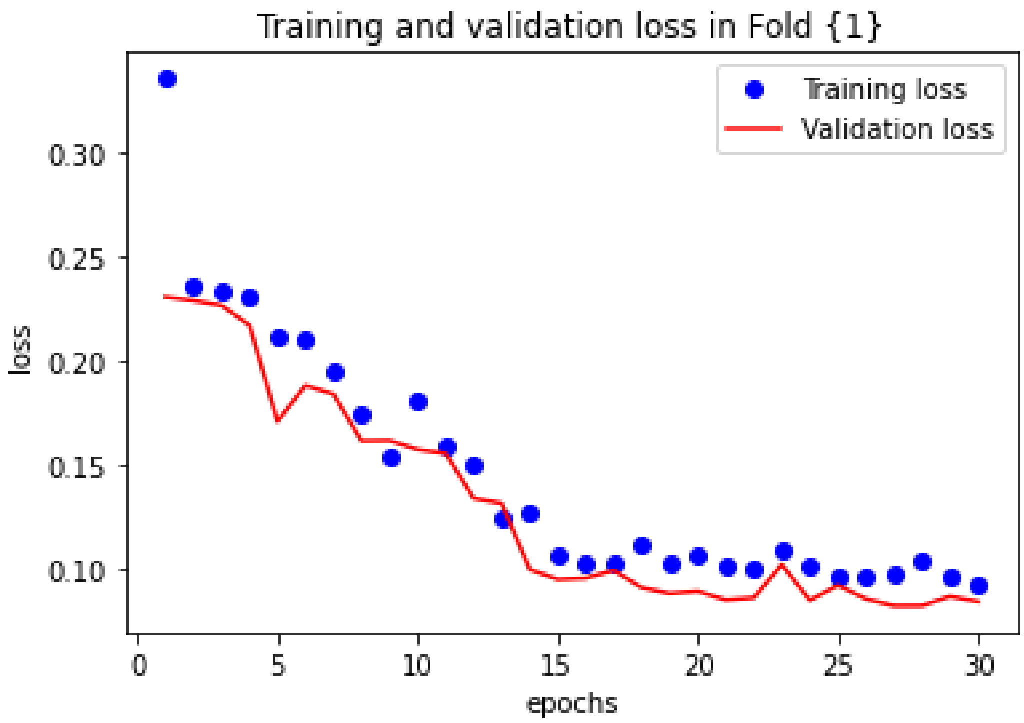



Figure 4.28 : The model_320 training and validation loss in fold 1.

Figure 4.28 displays the chart of the validation and training loss foduring the training of model in fold 1. It can be observed that the Max loss _training of the AI model was 0.11 at the 30th epoch, while the maximum validation accuracy was 0.09 at the 30th epoch. Additionally, both losss are close, pointing out that the system is best fitting.

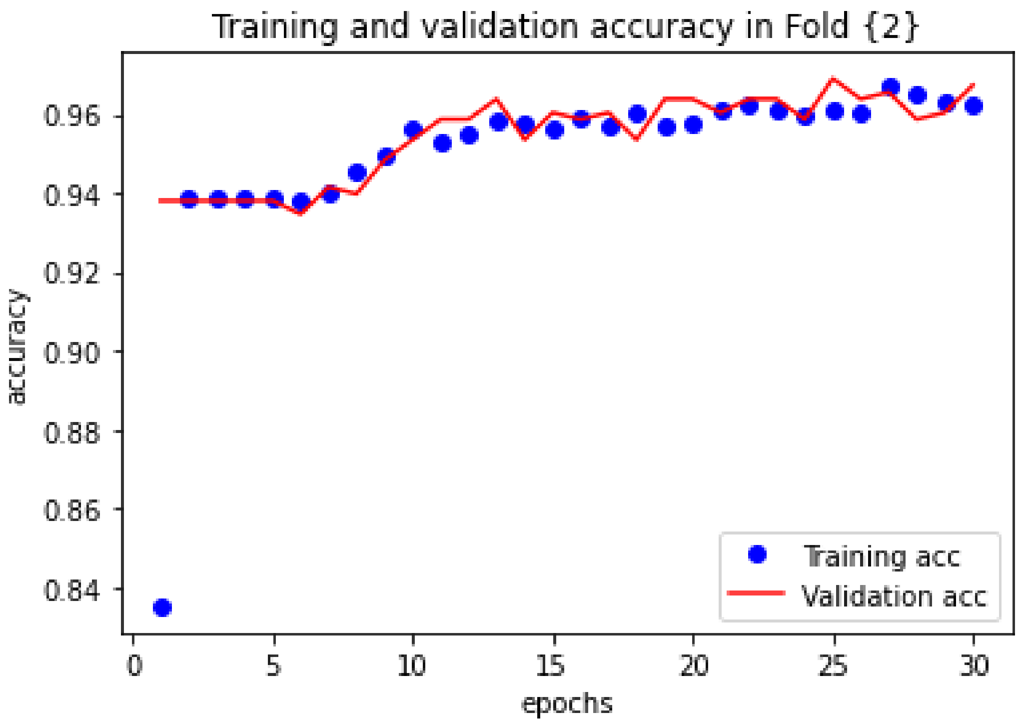



Figure 4.29 : The model_320 training and validation accuracy in fold 2.

The chart in figure 4.29 illustrates the ACC_val and ACC_train during fold 2. At the 30th epoch, the max training accuracy reached 0.96, whereas the max validation accuracy was 0.97 at the 26th epoch. Additionally, this fold has a MCC of 0.56. In general, the close both accuracy point out that the AI model is best fitting better than in fold 1.

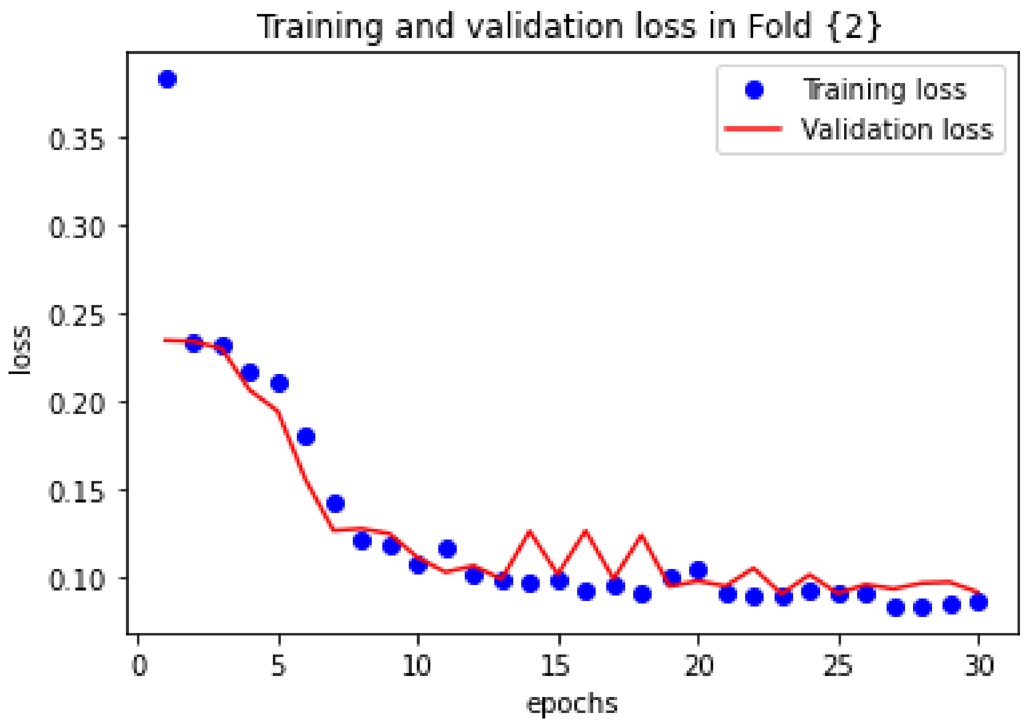



Figure 4.30 : The model_320 training and validation loss in fold 2.

Figure 4.30 depicts the chart of the loss_training and loss_validation throughout the training of the AI system in fold 2. The model's maximum training loss has been reached at the 30th epoch, reaching 0.11. However, the Max validation ACC is only 0.09 at the same epoch. Additionally, the loss_training and loss_validation are in close proximity.

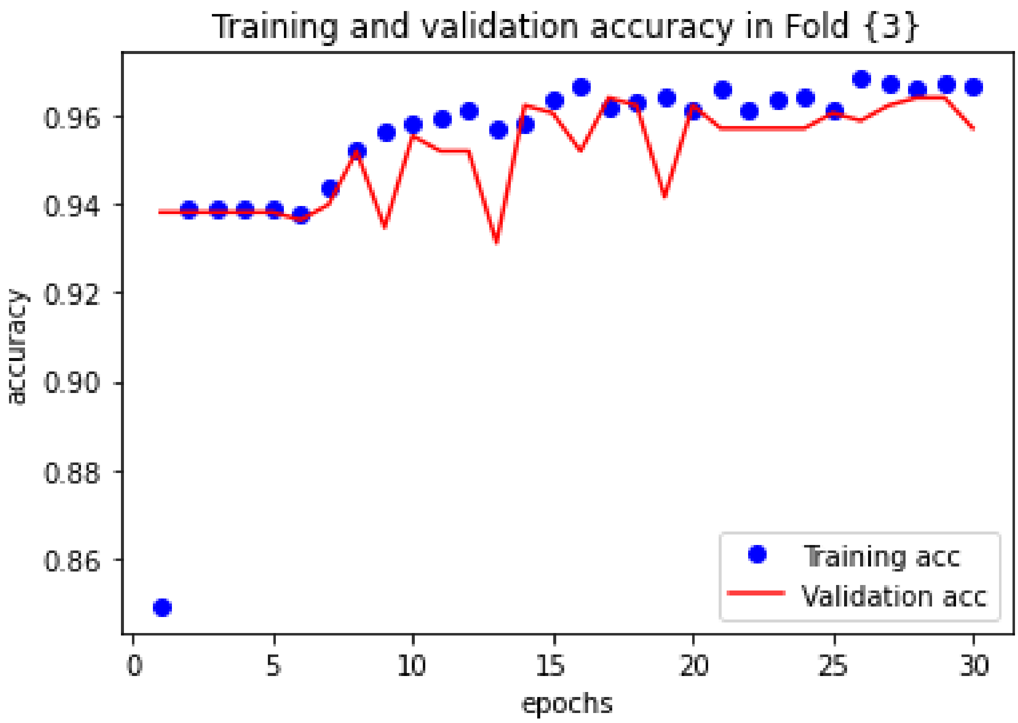



Figure 4.31 : The model_320 training and validation accuracy in fold 3.

Figure 4.31 shows the training and validation ACC chart in fold 3. The highest training accuracy achieved by the model is 0.96 at the 30th epoch, but the highest validation accuracy is 0.97 at the 26th epoch. Furthermore, the MCC for this fold is 0.56. The training and validation ACC are similar.

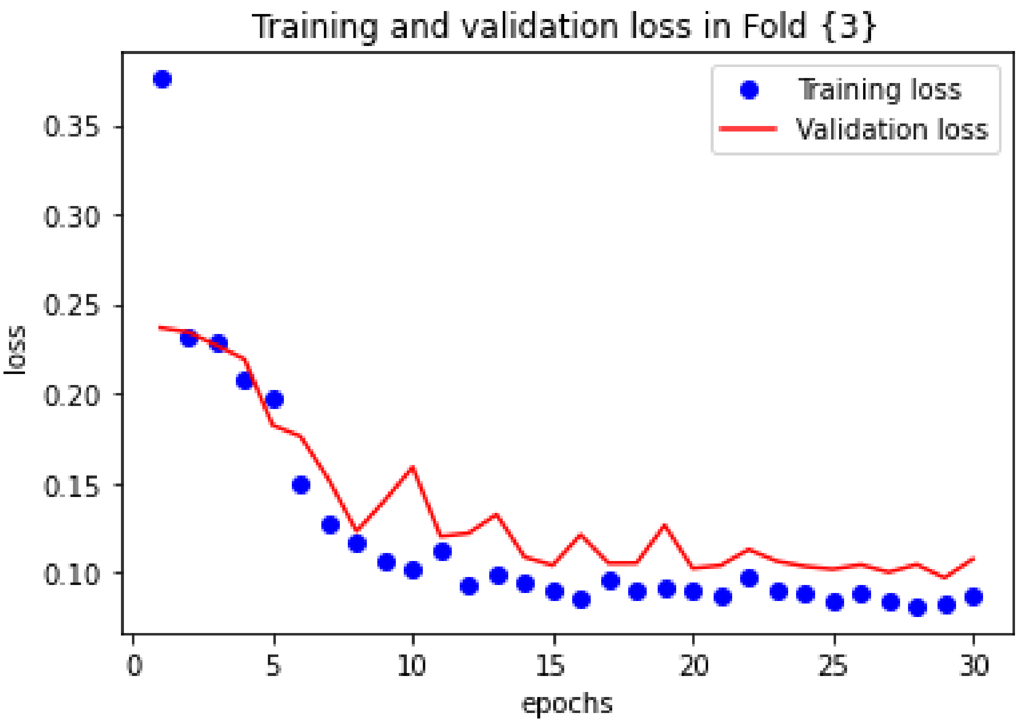



Figure 4.32 : The model_320 training and validation loss in fold 3.

Figure 4.32 depicts the loss_training and loss_validation chart in fold 3. At the 30th epoch, the model's maximum training loss has reached 0.11, whereas the maximum validation accuracy is 0.09 at the same epoch. Additionally, both loss is close like previous.

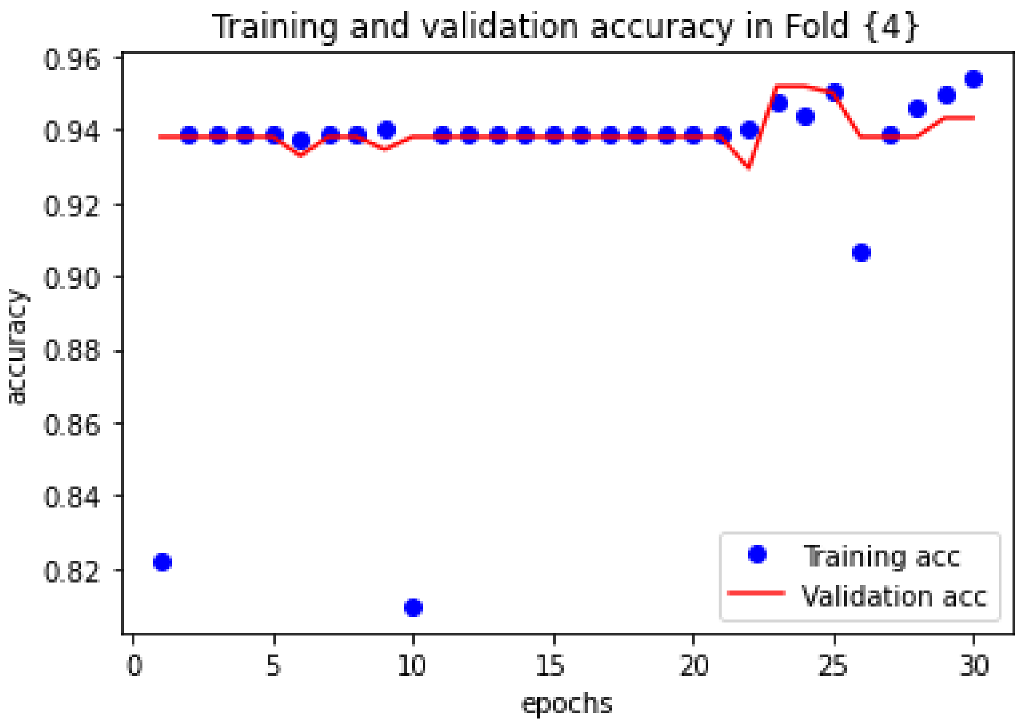



Figure 4.33 : The model_320 training and validation accuracy in fold 4.

This figure, depicted in 4.33, shows validation and training accuracy chart of the model in fold 4. The model's maximum training accuracy is 0.96 at the 30th epoch, but the Max validation ACC is 0.97 at the 26th epoch. Additionally, the MCC in this fold is 0.56. In general, both accuracy is similar.

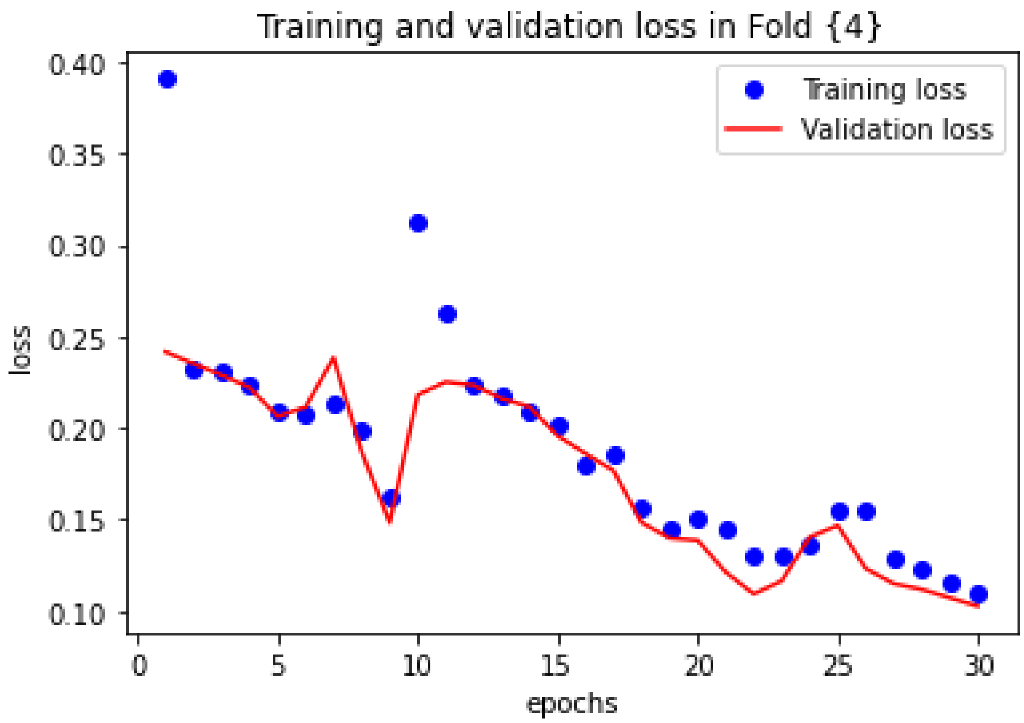



Figure 4.34 : The model_320 training and validation loss in fold 4.

The chart shown in figure 4.34 presents the loss values for both through the training of the system in fold 4. At the 30th epoch, the maximum training loss was 0.11, while the maximum validation accuracy was 0.09. Furthermore, both loss were similar, indicating that the AI system is well-fitted.

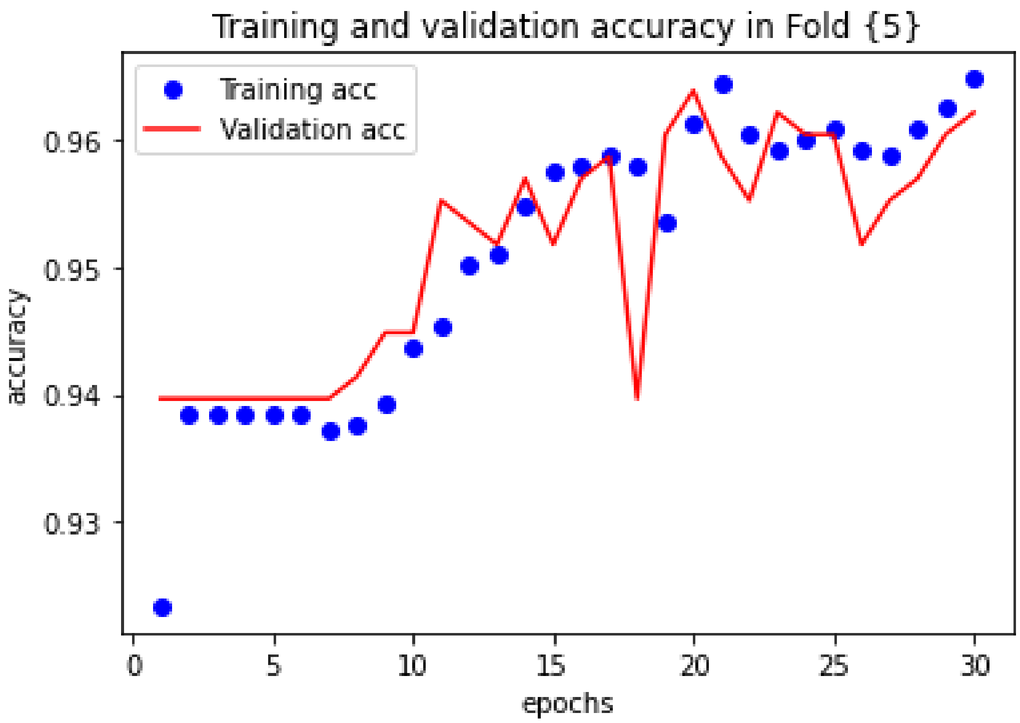



Figure 4.35 : The model_320 training and validation accuracy in fold 5.

Figure 4.35 depicts the accuracy chart during the model training in fold 5. At the 30th epoch, the Max-training accuracy of the system was 0.96, whereas the maximum validation accuracy was 0.97 at the 26th epoch. In addition, MCC in this fold was 0.56. The model's training and validation ACC are close, poiting out that the AI system is the best fitting like in fold 4.

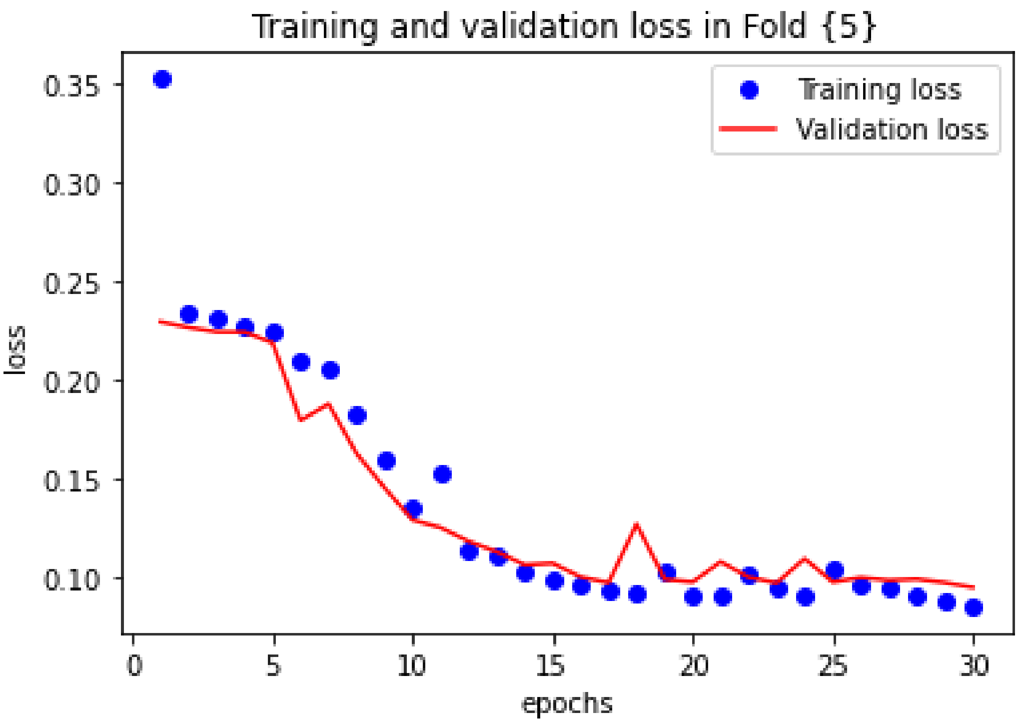



Figure 4.36 : The model_320 training and validation loss in fold 5.

The loss chart of the model in fold 5 during training is shown in figure 4.3, depicting the training and validation losses. At the 30th epoch, the model's maximum training loss reached 0.11, whereas the maximum validation accuracy was 0.09. Additionally, since the both loss are close, it points out that the model fits well.

#### 4.4.2.2. Testing model_320

Once the training of model_320 was completed and weights were saved at each fold, five .h files were generated to be used for testing the AI system. Upon testing, the classifier identified 1047 faulty lines. Figure 4.37 displays a bar chart illustrating the distribution of the classifier's output.

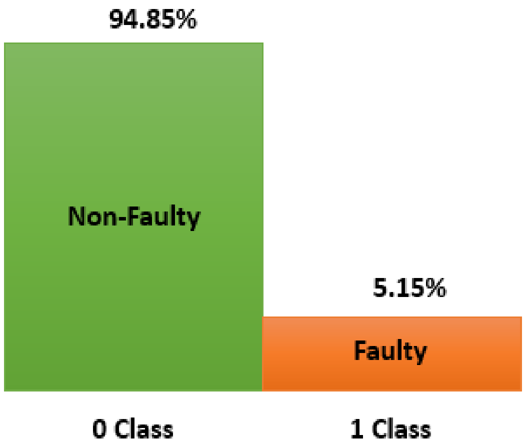



Figure 4.37 : The bar chart of the output distribution of the model_320.

### 4.4.3. Model_hybird

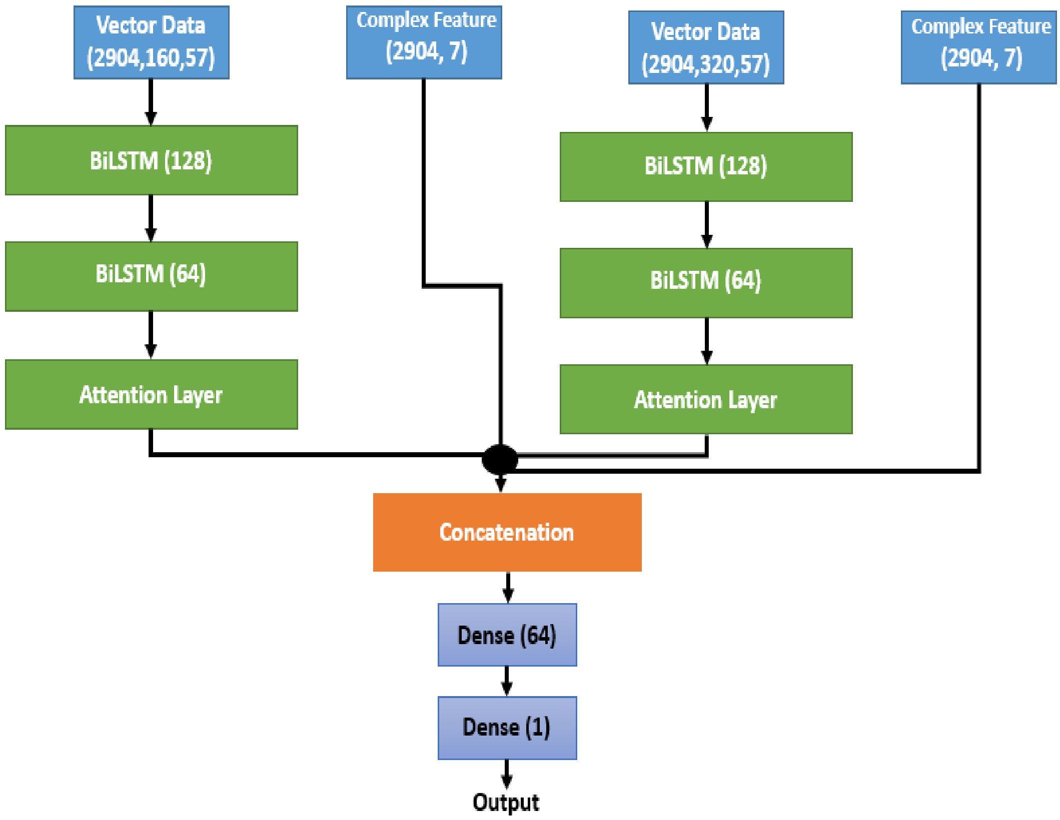



Figure 4.38 : Model_hybrid.

As shown in figure 4.38, the model_hybrid combines between model_160 and model_320 to compromise between the properties of both models. Hence the model_hybrid deals with data in shape(2904,160,57) with extra complex feature data (2904,7)and shape(2904,320,57)with extra complex feature data (2904,7).The model_hybrid consistes from many layers as follow:

BiLSTM layer accepts the input data in shape(2904,160,57 ) with 128 cell which receive input data vector (2904,160,57), then

BiLSTM layer with 64 cell which receive the ouput of previous BiLSTM(128) layer, then

Attention layer which receive the ouput of previous BiLSTM(64) layer.

The same thingaccepts the input data in shape(2904,320,57 ) with 128 cell which receive input data vector (2904,128,57), then

BiLSTM layer with 64 cell which receive the ouput of previous BiLSTM(128) layer, then

Attention layer which receive the ouput of previous BiLSTM(64) layer,after that

Concatenation layer which connect the output of two attention layers with the complex feature data in shape(2904,7), then

Dense layer with 64 cell using ReLu as an activation function, Since LSTM also inputs Sequence of Vectors as shown in the figure 4.38,so dense layer is used because it accepts input as Sequence of Vectors and convert it into 1-vector outputs.

Dense layer with 1 cell using sigmoid as an activation function and this layer is an output layer which predicts whether it is class 0 or 1.And the sammary of model is illistrated in figure 4.26 . Moreover, the count of the parameters that are trainable is 727777.

```
 Layer (type)                Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)        [(None, 160, 57)]    0           []

 input_4 (InputLayer)        [(None, 320, 57)]    0           []

 bidirectional (Bidirectional)  (None, 160, 256)  190464      ['input_1[0][0]']

 bidirectional_2 (Bidirectional  (None, 320, 256)  190464     ['input_4[0][0]']
 )

 bidirectional_1 (Bidirectional  (None, 160, 128)  164352     ['bidirectional[0][0]']
 )

 bidirectional_3 (Bidirectional  (None, 320, 128)  164352     ['bidirectional_2[0][0]']
 )

 attention (Attention)       (None, 128)          288         ['bidirectional_1[0][0]']

 input_2 (InputLayer)        [(None, 7)]          0           []

 attention_1 (Attention)     (None, 128)          448         ['bidirectional_3[0][0]']

 input_3 (InputLayer)        [(None, 7)]          0           []

 concatenate (Concatenate)   (None, 270)          0           ['attention[0][0]',
                                                               'input_2[0][0]',
                                                               'attention_1[0][0]',
                                                               'input_3[0][0]']

 dense (Dense)               (None, 64)           17344       ['concatenate[0][0]']

 dense_1 (Dense)             (None, 1)            65          ['dense[0][0]']

==================================================================================================
Total params: 727,777
Trainable params: 727,777
Non-trainable params: 0
```

Figure 4.39 : Model_hybrid summary.

#### 4.4.3.1. Training and validation model hybrid

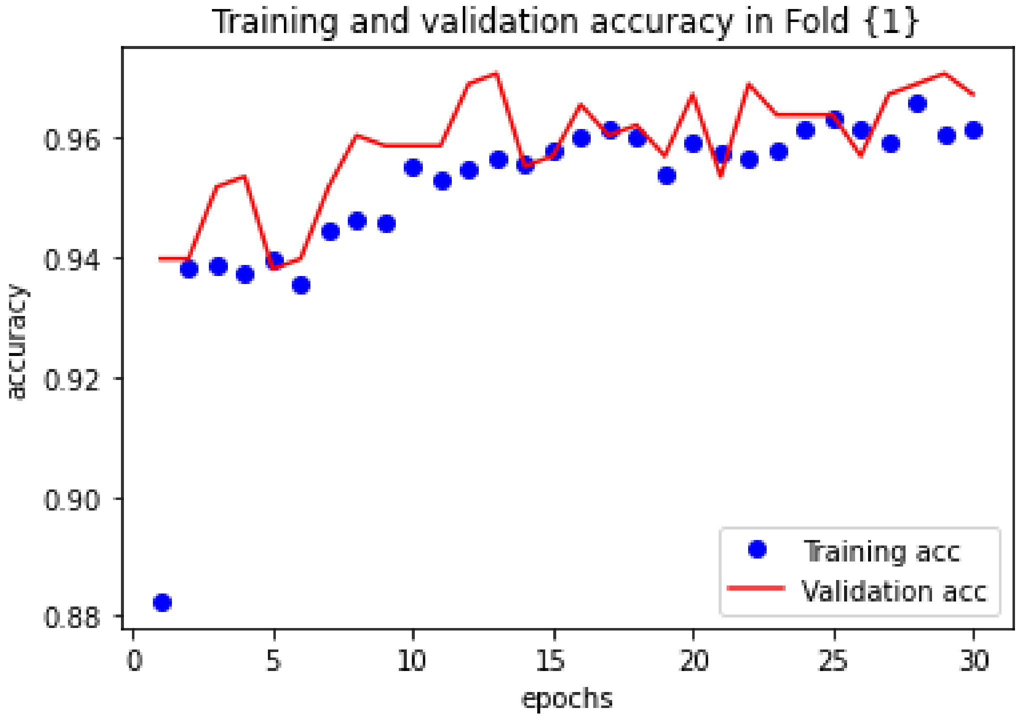



Figure 4.40 : The model_hybrid training and validation accuracy in fold 1.

The accuracy chart in fold 1 is displayed in Figure 4.40. The max ACC_training of the AI system was achieved at the 30th epoch, reaching 0.96, while the maximum validation accuracy of 0.97 was reached at the 26th epoch. Furthermore, the MCC for this fold is 0.56. Overall, the training and validation ACC are similar, pointing out that the model fits well.

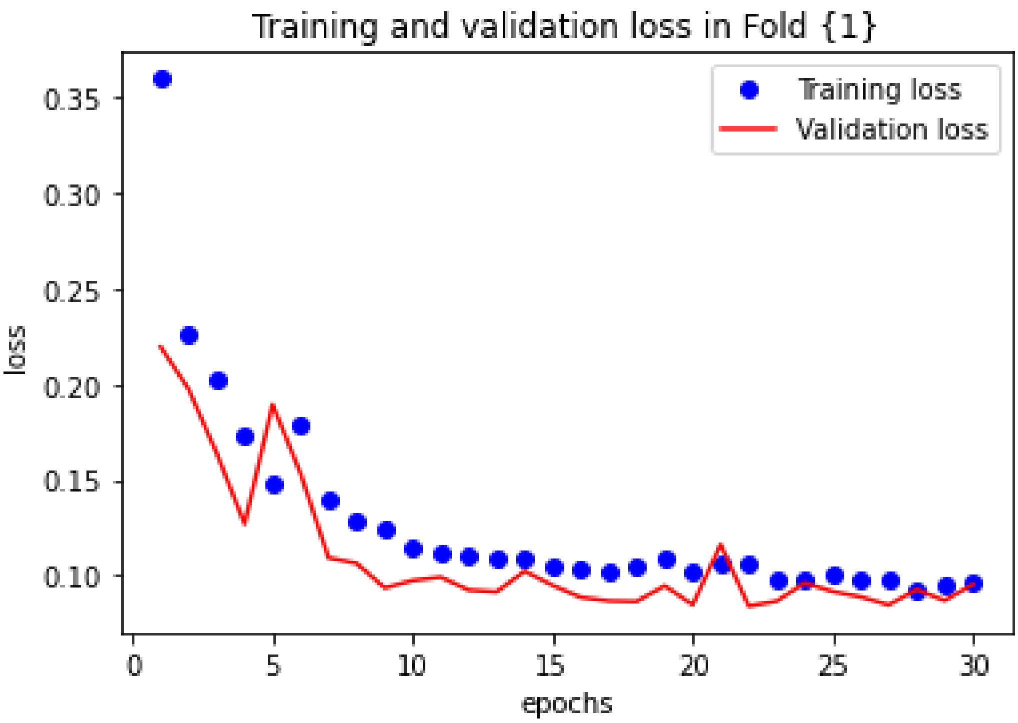



Figure 4.41 : The model_hybrid training and validation loss in fold 1.

Figure 4.41 illustrates a chart showing the loss in fold 1. The highest loss _training of the AI model was at 0.11 in the 30th epoch, while the maximum validation accuracy was at 0.09 in the same epoch. Additionally, the difference between the losses is small, pointing out that the AI system is well-fitted.

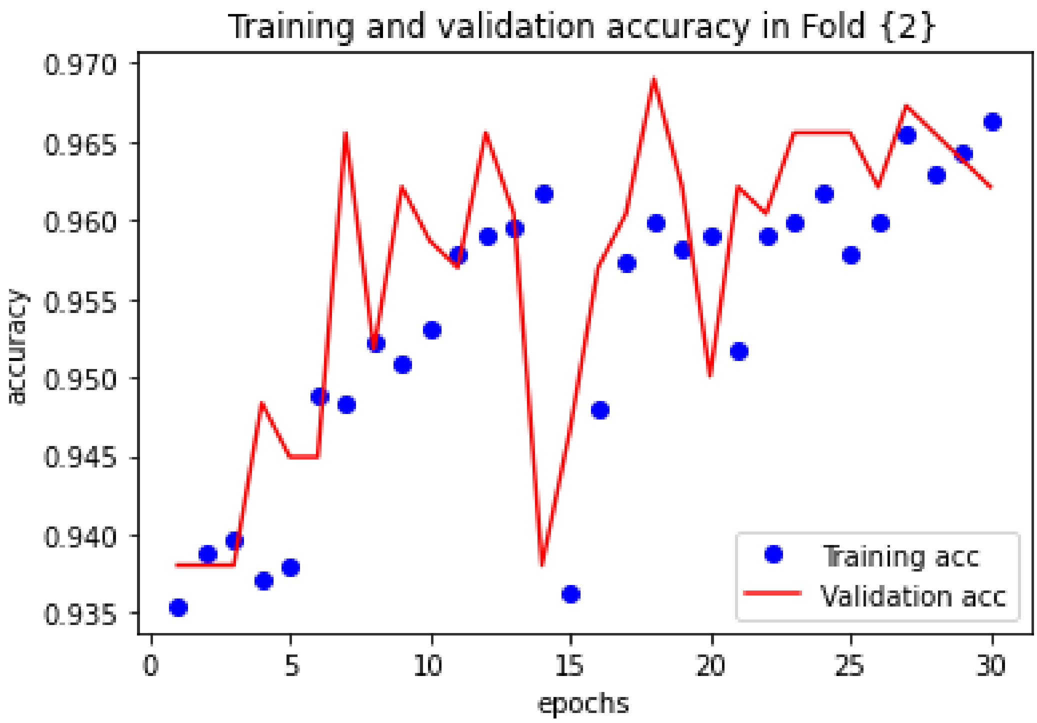



Figure 4.42 : The model_hybrid training and validation accuracy in fold 2.

Figure 4.42 shows the chart for both accuracy through the training of the AI system in fold 2. At the 30th epoch, the max training accuracy done by the model is 0.96, whereas the maximum validation accuracy is 0.97, which is achieved at the 26th epoch. Additionally, the MCC for this fold is 0.56. Furthermore, since the training and validation ACC are close, it points out that the system is a good fit as in fold 1.

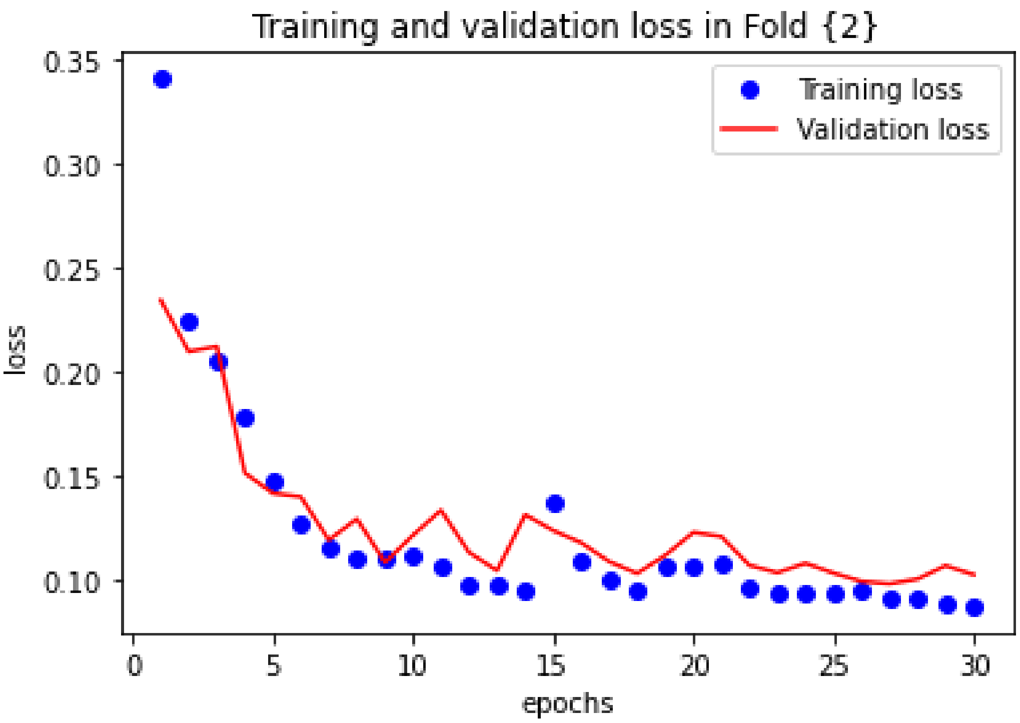



Figure 4.43 : The model_hybrid training and validation loss in fold 2.

Figure 4.43 shows a chart that displays the loss in fold 2. The maximum training loss that the model has achieved occurred at the 30th epoch and reached 0.11, while the maximum validation ACC was 0.09 at the same epoch. Additionally, the two losses are quite similar.

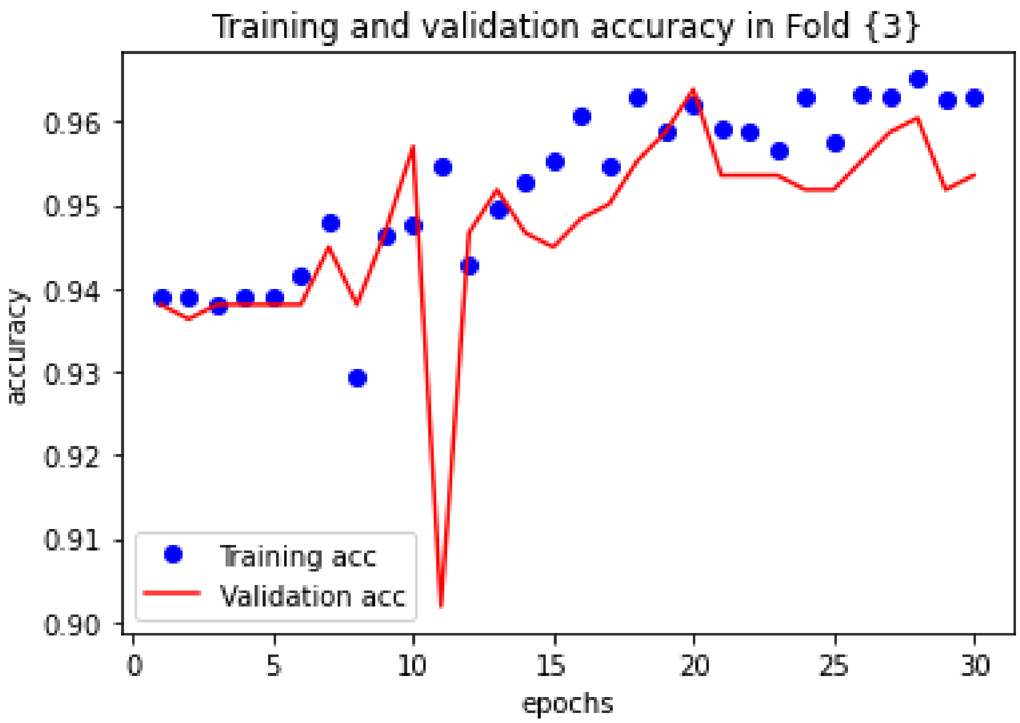



Figure 4.44 : The model_hybrid training and validation accuracy in fold 3.

Figure 4.44 depicts a chart that shows the accuracy in fold 3. The model's maximum training accuracy occurred at the 30th epoch and reached 0.96, whereas the maximum validation accuracy was 0.97 at the 26th epoch. Additionally, the MCC for this fold is 0.56. Overall, the training and validation ACC are quite similar, poiting out that the system is well-fitted.

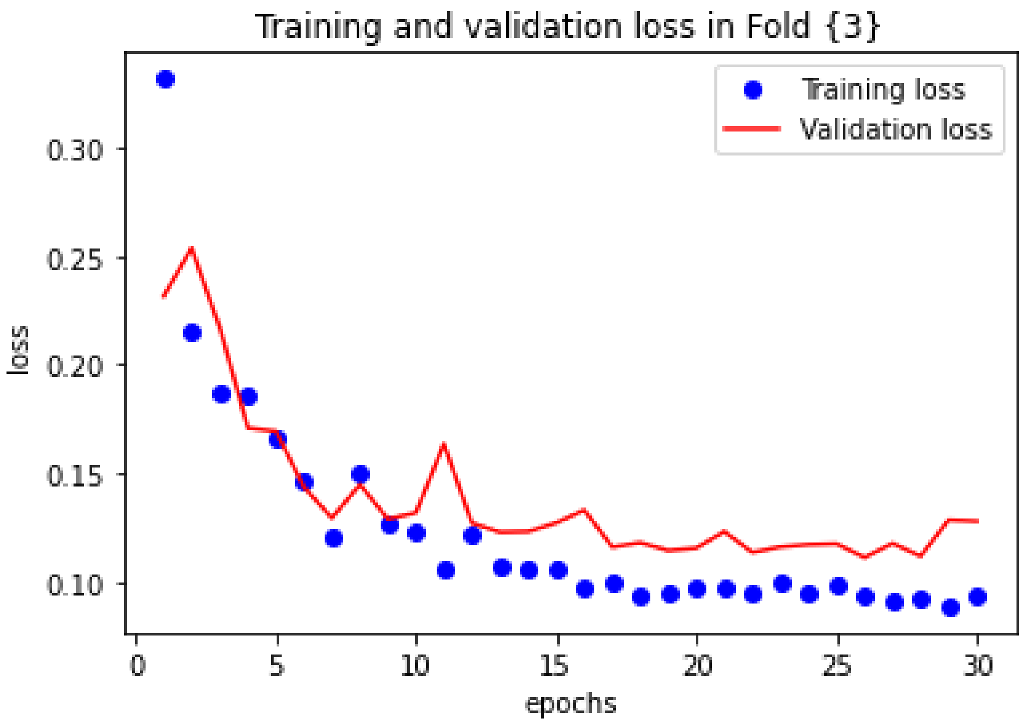



Figure 4.45 : The model_hybrid training and validation loss in fold 3.

Figure 4.45 presents a chart that shows the loss for both stages throughout the training of the system in fold 3. The model's maximum training loss occurred at the 30th epoch and reached 0.11, whereas the maximum validation accuracy was 0.09 at the same epoch. Additionally, the both loss are relatively similar.

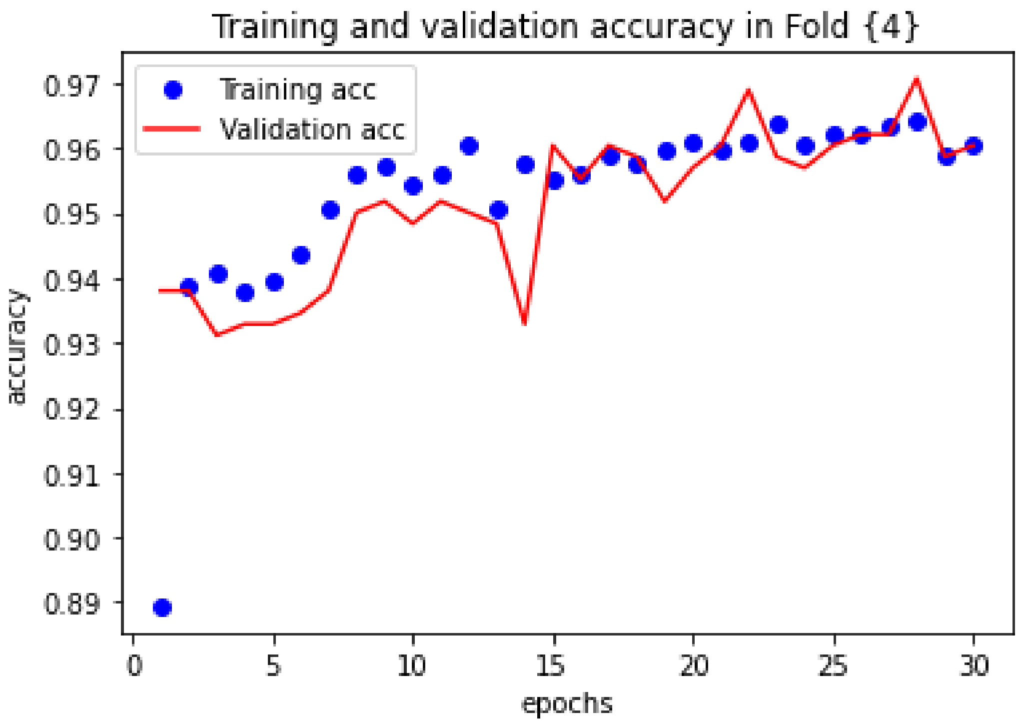



Figure 4.46 : The model_hybrid training and validation accuracy in fold 4.

Figure 4.46 depicts a chart that shows the ACC for both training and validation stages in fold 4. The model's maximum training accuracy occurred at the 30th epoch and reached 0.96, whereas the maximum validation accuracy was 0.97 at the 26th epoch. Additionally, the MCC for this fold is 0.56. Overall, the training and validation ACC are quite similar as previous.

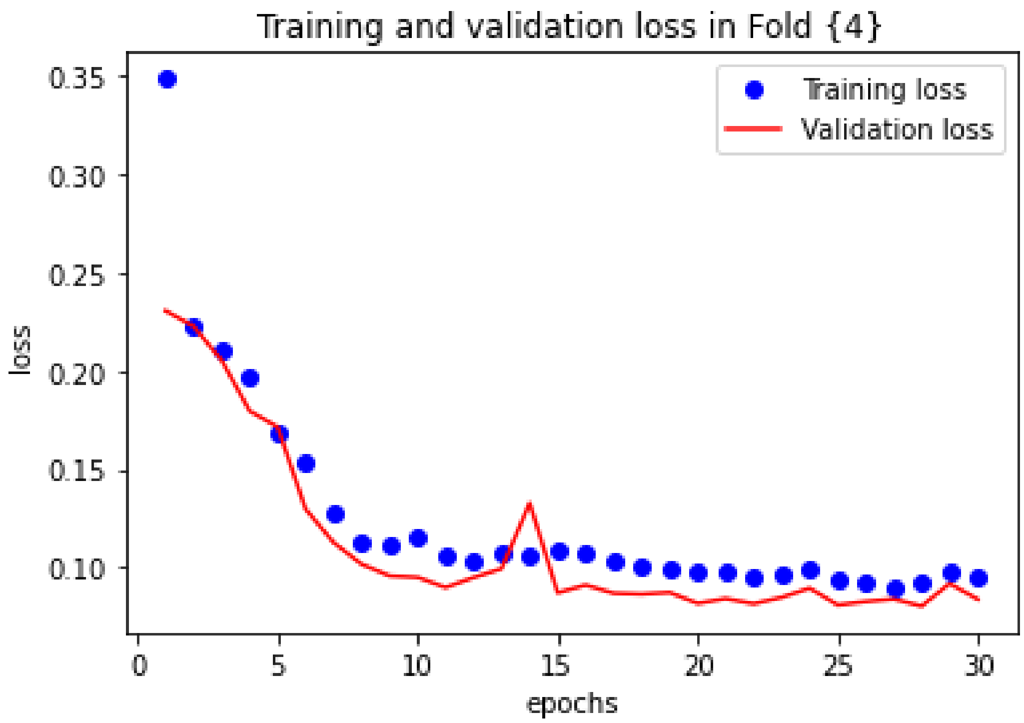



Figure 4.47 : The model_hybrid training and validation loss in fold 4.

Figure 4.47 presents a chart that shows the loss-training and loss-validation throughout the training of the system in fold 4. The model's maximum training loss occurred at the 30th epoch and reached 0.11, whereas the maximum validation accuracy was 0.09 at the same epoch. Additionally, the two loss are relatively close.

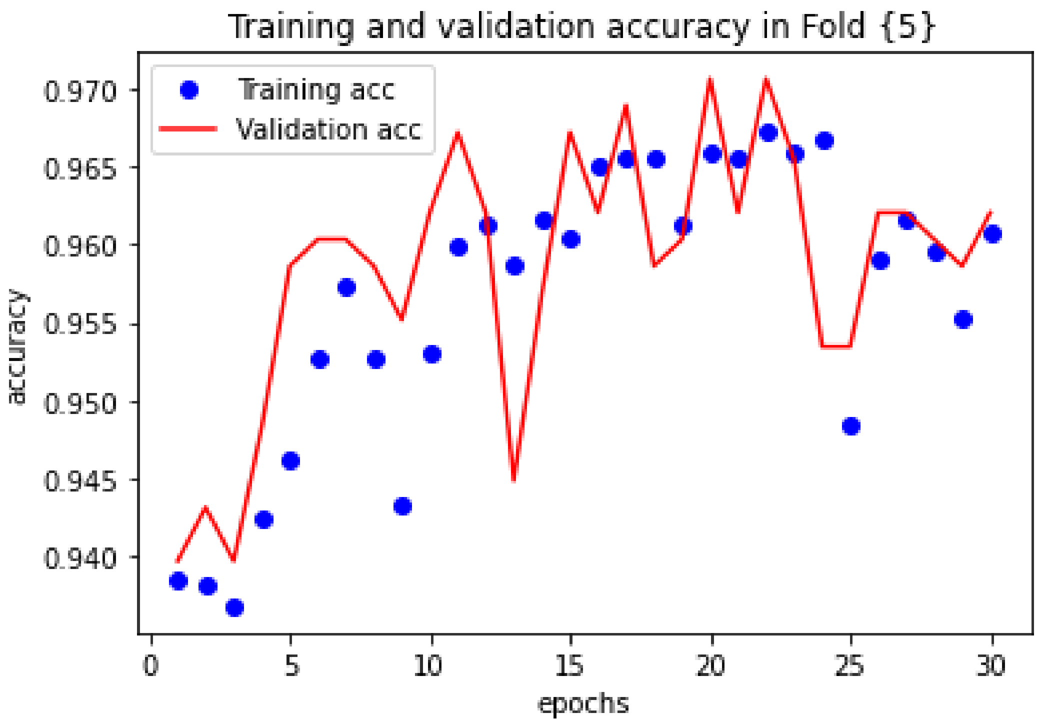



Figure 4.48 : The model_hybrid training and validation accuracy in fold 5.

Figure 4.48 depicts a chart that shows the accuracy in fold 5. The model's maximum training accuracy occurred at the 30th epoch and reached 0.96, whereas the maximum validation accuracy was 0.97 at the 26th epoch. Additionally, the MCC for this fold is 0.56. Overall, both accuracy are quite similar, indicating that the AI system is well-fitted.

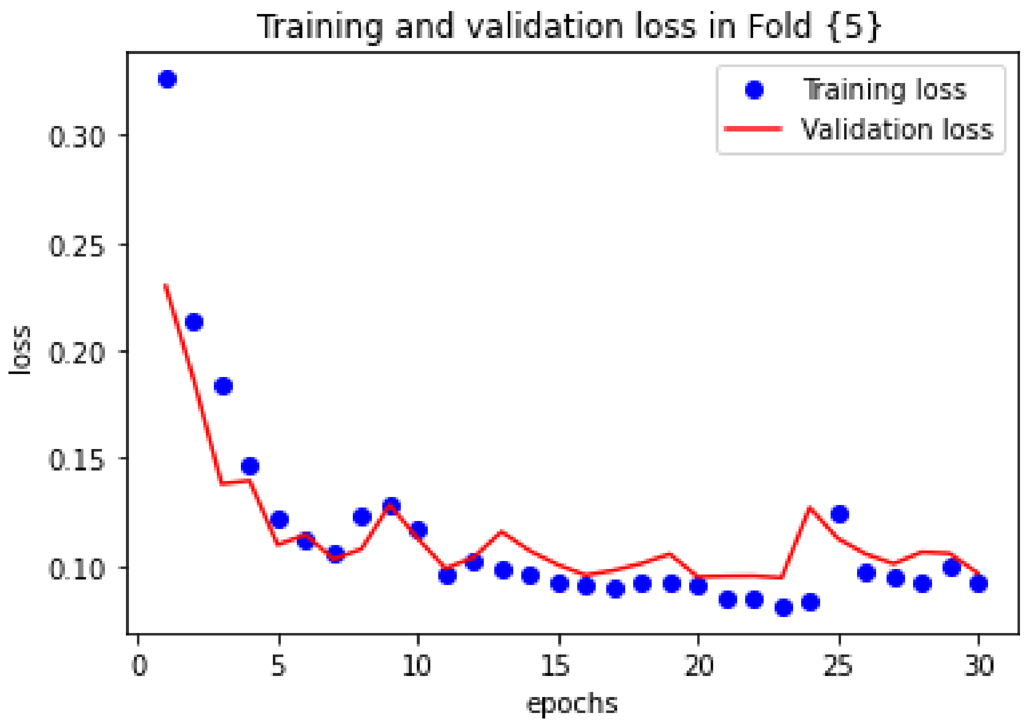



Figure 4.49 : The model_hybrid training and validation loss in fold 5.

Figure 4.49 presents a chart that shows the loss for both validation and training stages in fold 5. The model's maximum training loss occurred at the 30th epoch and reached 0.10, whereas the maximum validation accuracy was 0.09 at the same epoch. Additionally, the validation and training loss are relatively similar, indicating that the system is well-fitted.

#### 4.4.3.2. Testing model_hybrid

After training the model_hybrid and saving its weights for each fold, a set of five files with the .h extension is generated to facilitate the testing of the model. Upon conducting the testing process, the model successfully identifies 990 faulty lines. To visualize the distribution of the classifier output, a bar chart is utilized, which is depicted in Figure 4.50.

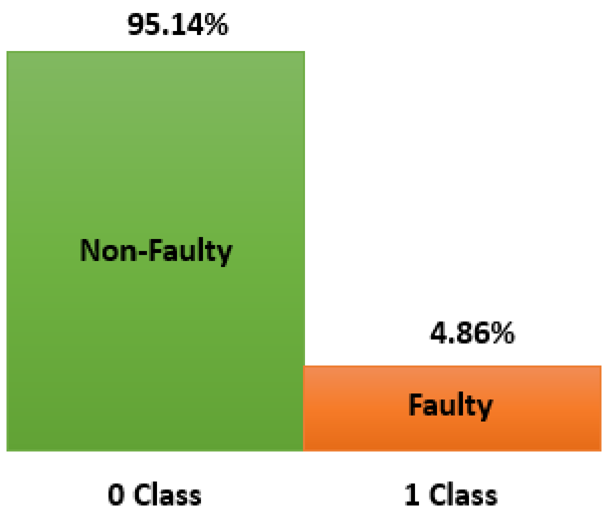



Figure 4.50 : The bar chart of the output distribution of the model_hybrid.

### 4.5. Models Comparison

During the research journey, many different artificial intelligence techniques were proposed and tested. One of the classifiers was designed based on the idea that the input data was as images of the voltage signal, and CNN was used for that. The use of this approach did not achieve satisfactory or even acceptable results because the nature of the data requires appropriate treatment for its nature, which is a study and measurement of its complexity and extraction of hidden information from it.

Another model was tried that depends on the input data being of two different types: data as images and as time series data, but it did not achieve good results. Finally, the thesis's proposed model was tested and achieved great results, and it was approved. Table 4.2 shows the comparison between several models that were tested during the research trip.

Once the deep learning models have been designed and trained using identical hyper-parameters such as no. of epochs and learning rate (LR), a comparison of the three AI models were presented in table 4.1. The results manifest that model_hybrid outperforms the other models in terms of MCC, as well as training and validation accuracy and loss. However, the other two models manifest advantages over model_hybrid in terms of shorter model training-time and fewer trainable parameters.

Table 4.1 : Model comparison.

| Comparison parameters | Model_160 | Model_320 | Model_hybird |
|---|---|---|---|
| **Max training accuracy** | 0.96 | 0.96 | **0.975** |
| **Max training Validation** | 0.96 | 0.97 | **0.975** |
| **Max training Loss** | 0.1 | 0.1 | 0.1 |
| **Max Validation Loss** | 0.1 | 0.1 | 0.1 |
| **MCC** | 0.75 | 0.71 | **0.79** |
| **No. of detected faulty Lines** | 1023 | 1047 | 990 |
| **Model training Time (Hour)** | 7 | 10 | 20 |
| **Best Threshold** | 0.26 | 0.38 | 0.35 |
| **No. of trainable parameters** | **363873** | **364033** | **727777** |

Table 4.2 : A comparison of many different AI approachs has been tested during the research journey.

| Tested techniques | MCC | Accuracy |
|---|---|---|
| **CNN** | 0.15 | 0.05 |
| **CNN+BILSTM** | 0.25 | 0.4 |
| **Thesis's propsed model** | **0.79** | **0.975** |

# CHAPTER 5. CONCLUSION AND RECOMMENDATIONS

## 5.1. Conclusion

In previous chapters, the problem of partial discharge was described and the importance of studying this phenomenon and the benefits of designing the AI model to detect this problem were discussed. Additionally, the provided training datasets were described and processed, and beyond then, they have been used for training the classifier model to distinguish which power line is faulty or not. Moreover, the structure of deep neural networks was discussed and three model models were built and trained at the same hyperparameters the Model_hybrid is used as the accountable model to detect the PD because has the best MCC value although it took much time in training. Hence, Relying on the transfer learning technique that was talked about in the second chapter, this trained model can be employed to address PD-related problem , even when the form and nature of the datasets are different, for instance.

## 5.2. Transfer Learning With Model_hybrid

Model Hybrid can be used with different input, output shapes, or even both but after necessary modification in the model is done to be suitable to the new purpose of the model. That technique is called transfer learning and using this technique in these situations can reduce the training time for a new model and can be useful in case, not enough training dataset is provided, for instance.

### 5.2.1. Different input model

The used input data is as voltage signal of the electric flows through each phase but as depicted in figure 5.1 and 5.2, the nature of the input data could be different and maybe can be as the temperature of the overhead power insulator, the sound which is generated as a result of PD or even the both. Consequently, the nature of the training dataset can be as sound and temperature signals and voltage signals can be besides them.

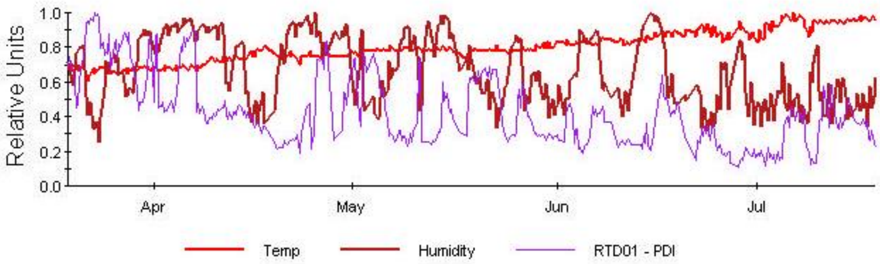



Figure 5.1 : PD affected by heat and humidity for sync motor [85].

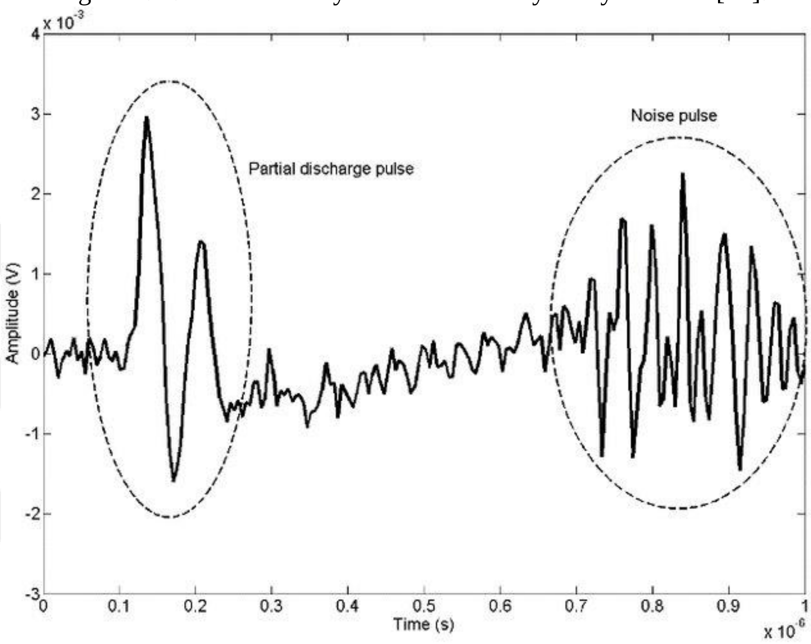



Figure 5.2 : Partial discharge with noise[86].

As shown in figure 5.3, BiLSTM model accepts both temperature and sound as I/O is stayed binary output.

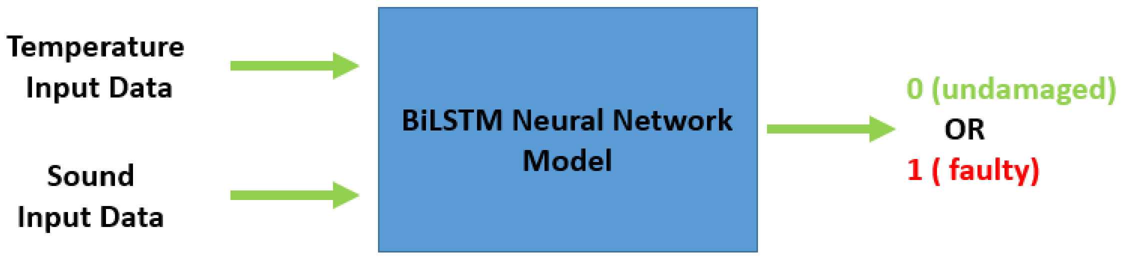



Figure 5.3 : Different input model.

### 5.2.2. Different output model

The output shape of the trained model is binary and it just classifies two states either faulty or not but the output shape can be categorical. That means the model can classify if the line is normal or faulty in levels one, two or so on as illustrated in figure 5.4. For example, faulty level one can mean that it is not serious partial discharge and the maintenance is not necessary right now maybe after 5 months, for instance.

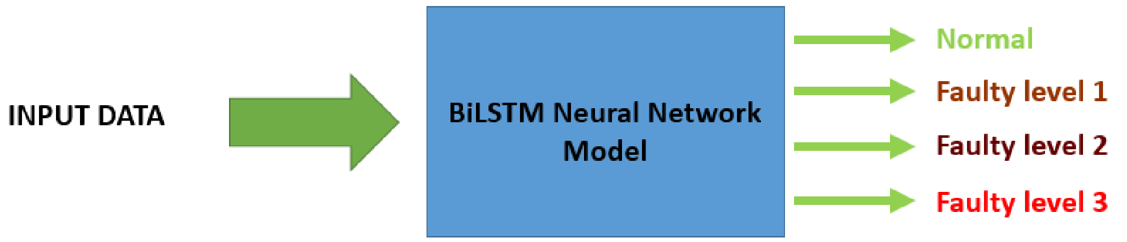



Figure 5.4 : Categorical output model.

### 5.2.3. Different input and output model

The new model can be with different input and output than model_hybrid as illustrated in figure 5.5.

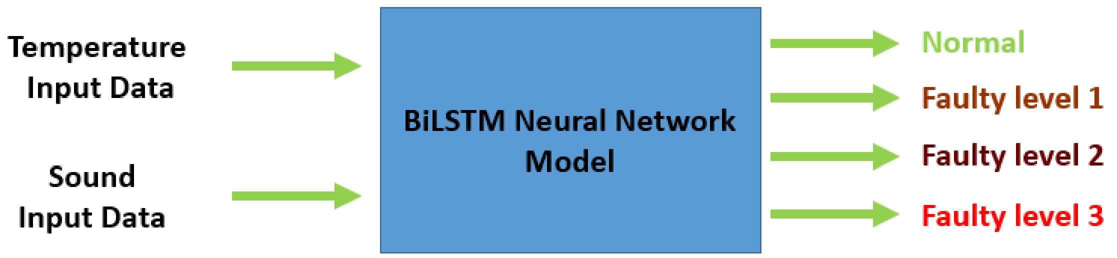



Figure 5.5 : Different input and output model.

Utilizing a pre-existing model offers advantages over creating a new model, as it enables the application of transfer learning methods, resulting in a notable decrease in the no. of parameters that require training. Consequently, this leads to a reduction in the training-time required.

## REFERENCES


[1] Kessler, O. (2020). The importance of partial discharge testing: PD testing has proven to be a very reliable method for detecting defects in the insulation system of electrical equipment and for assessing the risk of failure. *IEEE Power and Energy Magazine*, 18(2), 62-65.

[2] Thakor, K. and Shah, P. (2020), Partial Discharge Detection Techniques: A Review. *International Research Journal of Engineering and Technology (IRJET)*, 7, (10) PP 885 - 890.

[3] Densley, J. (1979), *Corona measurement and interpretation*, Chapter 11 in Engineering Dielectrics, Vol. 1, ASTM.

[4] Url-1 < *https://www.fluke.com/en-us/learn/blog/leak-detection/partial-discharge*>, date retrieved 29.11.2021.

[5] Beam, A. L., & Kohane, I. S. (2018). Big data and machine learning in health care. *Jama*, 319(13), 1317-1318.

[6] Friendly, A. I. (1974). Part of a series on. *artificial intelligence*, 10, 1.

[7] El Naqa, I., & Murphy, M. J. (2015). *What is machine learning?* (pp. 3-11). Springer International Publishing.

[8] Hu, J., Niu, H., Carrasco, J., Lennox, B., & Arvin, F. (2020). Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning. *IEEE Transactions on Vehicular Technology*, 69(12), 14413-14423.

[9] Bengio, Y., & LeCun, Y. (2007). Scaling learning algorithms towards AI. *Large-scale kernel machines*, 34(5), 1-41.

[10] Kulin, M., Kazaz, T., De Poorter, E., & Moerman, I. (2021). A survey on machine learning-based performance improvement of wireless networks: PHY, MAC and network layer. *Electronics*, 10(3), 318.

[11] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61, 85-117.

[12] Sarle, W. S. (1994). Neural networks and statistical models. In *Proceedings of the 19th annual SAS users group international conference* (Vol. 13).

[13] Url-2 < *https://viso.ai/deep-learning/deep-neural-network-three-popular-types/*>, date retrieved 29.11.2021.

[14] Gao, Z., Li, J., Guo, J., Chen, Y., Yi, Z., & Zhong, J. (2018). Diagnosis of diabetic retinopathy using deep neural networks. *IEEE Access*, 7, 3360-3370.

[15] Nix, R., & Zhang, J. (2017, May). Classification of android apps and malware using deep neural networks. In 2017 *International joint conference on neural networks (IJCNN)* (pp. 1871-1878). IEEE.

[16] Nattrass, D. A. (1993). Partial discharge. XVII. The early history of partial discharge research. *IEEE Electrical Insulation Magazine*, 9(4), 27-31.

[17] R. LaKoste,(1963), Review, Progress, and Trends in Partial Discharges in France, *ERA*, IB/2187,1963.

[18] Austen, A. E. W., & Hackett, W. (1944). Internal discharges in dielectrics: their observation and analysis. *Journal of the Institution of Electrical Engineers-Part I: General*, 91(44), 298-312.

[19] Ahmed, N. H., & Srinivas, N. N. (1998). On-line partial discharge detection in cables. *IEEE Transactions on Dielectrics and Electrical Insulation*, 5(2), 181-188.

[20] Fuchs, S. P. (1993). *Partial discharge and corona theory and measurement*, master thesis, University of Dayton,USA

[21] Wenrong, S., Junhao, L., Peng, Y., & Yanming, L. (2008). Digital detection, grouping and classification of partial discharge signals at DC voltage. *IEEE Transactions on Dielectrics and Electrical Insulation*, 15(6), 1663-1674.

[22] Howells, E. (1985). *Acoustic emission detection of partial discharges in power transformers*.

[23] Eleftherion, P. M. (1995). Partial discharge. XXI. Acoustic emission based PD source location in transformers. *IEEE Electrical Insulation Magazine*, 11(6), 22-26.

[24] Morshuis, P. H. (2005). Degradation of solid dielectrics due to internal partial discharge: some thoughts on progress made and where to go now. *IEEE Transactions on Dielectrics and Electrical Insulation*, 12(5), 905-913.

[25] Akbal, B. (2018). Applications of artificial intelligence and hybrid neural network methods with new bonding method to prevent electroshock risk and insulation faults in high-voltage underground cable lines. *Neural Computing and Applications*, 29, 97-105.

[26] Nyanteh, Y., Edrington, C., Srivastava, S., & Cartes, D. (2013). Application of artificial intelligence to real-time fault detection in permanent-magnet synchronous machines. *IEEE transactions on industry applications*, 49(3), 1205-1214.

[27] Wang, M., Tong, W., & Liu, S. (2017, June). Fault detection for power line based on convolution neural network. *In Proceedings of the 2017 International Conference on Deep Learning Technologies,China*, (pp. 95-101).

[28] Ghaderi, A., Ginn III, H. L., & Mohammadpour, H. A. (2017). High impedance fault detection: A review. *Electric power systems research*, 143, 376-388.

[29] Wang, S., & Dehghanian, P. (2020). On the use of artificial intelligence for high impedance fault detection and electrical safety. *IEEE Transactions on Industry Applications*, 56(6), 7208-7216.

[30] Url-3 < *https://eepower.com/technical-articles/what-is-partial-discharge/#*>, date retrieved 29.11.2021.

[31] Tehrani, P., & Levorato, M. (2020, November). Frequency-based multi task learning with attention mechanism for fault detection in power systems. *In 2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm*) , Tempe, AZ, USA ,(pp. 1-6).

[32] Mitchinson, P. M., Lewin, P. L., Strawbridge, B. D., & Jarman, P. J. I. E. I. M. (2010). Tracking and surface discharge at the oil/pressboard interface. *IEEE Electrical Insulation Magazine*, 26(2), 35-41.

[33] Jakarta State University, Engineer at PT. Tiara Vibrasindo Pratama, Electrical Enginerrring department. (2015). *Partial discharge at switchgear* [PowerPoint slides]. Retrieved from https://pt.slideshare.net/mauritzronigabemanur/partial-discharge-switchgear-ea-tech/4

[34] Url-4 < *https://www.elprocus.com/types-of-insulators-and-their-applications*/>, date retrieved 29.11.2021.

[35] Url-5 < *https://www.electricaleasy.com/2016/10/insulators-used-in-overhead-power-lines.html* >, date retrieved 29.11.2021.

[36] Url-6 < *https://www.electrical4u.com/types-of-electrical-insulator-overhead-insulator/*>, date retrieved 29.11.2021.

[37] Url-7 < *https://www.elprocus.com/types-of-insulators-and-their-applications/*>, date retrieved 29.11.2021.

[38] Jain, A., & Bhullar, S. (2018). Micro-phasor measurement units (μPMUs) and its applications in smart distribution systems. In ISGW 2017: Compendium of Technical Papers: *3rd International Conference and Exhibition on Smart Grids and Smart Cities* (pp. 81-92). Springer Singapore.

[39] Das, H. P., & Pradhan, A. K. (2016, December). Development of a micro-phasor measurement unit for distribution system applications. *In 2016 National Power Systems Conference (NPSC)* (pp. 1-5). IEEE.

[40] Jain, A., & Bhullar, S. (2018). Micro-phasor measurement units (μPMUs) and its applications in smart distribution systems. *In ISGW 2017: Compendium of Technical Papers: 3rd International Conference and Exhibition on Smart Grids and Smart Cities* (pp. 81-92). Springer Singapore.

[41] Poole, D. L., & Mackworth, A. K. (2010). *Artificial Intelligence: foundations of computational agents*. Cambridge University Press.

[42] Uncertain reasoning: Russell & Norvig (2003, pp. 452–644),Poole, Mackworth & Goebel (1998, pp. 345–395),Luger & Stubblefield (2004, pp. 333–381),Nilsson (1998, chpt. 19).

[43] Url-8 < *https://www.britannica.com/technology/artificial-intelligence*>, date retrieved 04.12.2021.

[44] Purkait, N. (2019). Hands-On Neural Networks with Keras: *Design and create neural networks using deep learning and artificial intelligence principles*. Packt Publishing Ltd.

[45] McCarthy, J. (2007). *What is artificial intelligence*.

[46] Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 255-260.

[47] Url-9 < *https://www.ibm.com/cloud/learn/machine-learning* >, date retrieved 29.11.2021.

[48] Wang, H., Lei, Z., Zhang, X., Zhou, B., & Peng, J. (2016). Machine learning basics. *Deep learning*, 98-164.

[49] Fukushima, K. (1980). A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol, Cybern*, 36, 193-202.

[50] Zhang, Y. D., & Sangaiah, A. K. (Eds.). (2021). *Cognitive Systems and Signal Processing in Image Processing*. Academic Press.

[51] Url-10 < *http://ozhangulal.com/hayat/yapay-zeka-derin-ogrenme-ve-makine-ogrenimi-arasindaki-farkliliklar*/>, date retrieved 29.11.2021.

[52] Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., & Pietikäinen, M. (2020). Deep learning for generic object detection: A survey. *International journal of computer vision*, 128, 261-318.

[53] Url-11 < *https://www.mathworks.com/discovery/deep-learning.html* >, date retrieved 29.11.2021.

[54] Url-12 < *http://www.global-engage.com/life-science/deep-learning-in-digital-pathology*/>, date retrieved 29.11.2021.

[55] Dongare, A. D., Kharde, R. R., & Kachare, A. D. (2012). *Introduction to artificial neural network. International Journal of Engineering and Innovative Technology (IJEIT)*, 2(1), 189-194.

[56] Url-13 < *https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53*>, date retrieved 29.11.2021.

[57] Medsker, L., & Jain, L. C. (Eds.). (1999). *Recurrent neural networks: design and applications*. CRC press.

[58] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.

[59] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv*:1412.3555.

[60] Url-14 <*https://www.tutorialexample.com/understand-the-effect-of-lstm-input-gate-forget-gate-and-output-gate-lstm-network-tutorial/*>, date retrieved 29.11.2021.

[61] West, J., Ventura, D., & Warnick, S. (2007). Spring research presentation: A theoretical foundation for inductive transfer. Brigham Young University, *College of Physical and Mathematical Sciences*, 1(08).

[62] Van Rossum, G. (2007, June). Python Programming Language. *In USENIX annual technical conference* (Vol. 41, No. 1, pp. 1-36).

[63] Venners, B. (2003). The making of python. Artima. com.

[64] Url-15 < *https://www.w3schools.com/python/python_intro.asp* >, date retrieved 29.11.2021.

[65] Point, T. (2016). Tutorials Point: Internet of Things, India: Tutorials Point (I) Pvt.

[66] Metz, C. (2015). Google just open sourced tensorflow, its artificial intelligence engine. *Wired. Accessed*, 10.

[67] Syed, I. (2020). Overview on open source machine learning platforms-tensorflow. *Journal For Innovative Development in Pharmaceutical and Technical Science* (JIDPTS) Volume, 3, 11-14.

[68] Url-16 < *https://dzone.com/articles/what-is-tensorflow-and-what-is-new-in-it*>, date retrieved 10.12.2021.

[69] Lardinois, J. F., Lynley, M., & Mannes, J. (2017). Google is acquiring data science community Kaggle. *Techcrunch. Archived from the original on March, 9*.

[70] Url-17 < *https://www.kaggle.com/competitions*>, date retrieved 10.12.2021.

[71] Roy, M., Kumar, V. R., Kulkarni, B. D., Sanderson, J., Rhodes, M., & Stappen, M. V. (2000). Simple denoising algorithm using wavelet transform. *arXiv preprint nlin*/0002028.

[72] Durand, S., & Froment, J. (2001, May). Artifact free signal denoising with wavelets. *In 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings* (Cat. No. 01CH37221) (Vol. 6, pp. 3685-3688). IEEE.

[73] Url-18 < https://en.wikipedia.org/wiki/Time_series>, date retrieved 10.12.2021.

[74]Url-19< *https://www.sciencedirect.com/science/article/abs/pii/S0960077915002817*

>, date retrieved 10.12.2021.

[75] Url-20 <*https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7516527/*>, date retrieved 10.12.2021.

[76] Albers, D., & Alexanderson, G. L. (2008). *Mathematical people: Profiles and interviews*. CRC Press.

[77]Url-21<*https://www.sciencedirect.com/science/article/pii/ S1110016819300845#b005* >, date retrieved 10.12.2021.

[78] Url-22 <*https://www.javatpoint.com/entropy-in-machine-learning*>, date retrieved 10.12.2021.

[79] Goh, C., Hamadicharef, B., Henderson, G. T., & Ifeachor, E. C. (2005, June). Comparison of fractal dimension algorithms for the computation of EEG biomarkers for dementia. *In 2nd international conference on computational intelligence in medicine and healthcare (CIMED2005),* Portugal.

[80] Mohan, A. T., & Gaitonde, D. V. (2018). A deep learning based approach to reduced order modeling for turbulent flow control using LSTM neural networks. *arXiv preprint arXiv:1804.09269.*

[81] Url-23 <*https://www.analyticsvidhya.com/blog/2019/11/comprehensive-guide-attention-mechanism-deep-learning/*>, date retrieved 10.12.2021.

[82] Url-24 <*https://towardsdatascience.com/the-basic-building-block-of-neural-networks-a9b2e8f5c056*>, date retrieved 10.12.2021.

[83] Cramer, H. (1946). *Mathematical methods of statistics, Princeton, 1946. Math Rev (Math-SciNet) MR16588 Zentralblatt MATH*, 63, 300.

[84] Kane, C., & Golubev, A. (2003, June). Advantages of continuous monitoring of partial discharges in rotating equipment and switchgear. In *Conference Record of the 2003 Annual Pulp and Paper Industry Technical Conference*, 2003. (pp. 117-122). IEEE.

[85] Parrado-Hernández, E., Robles, G., Ardila-Rey, J. A., & Martínez-Tarifa, J. M. (2018). Robust condition assessment of electrical equipment with one class support vector machines based on the measurement of partial discharges. *Energies*, 11(3), 486.