ABSTRACT

ARTIFICIAL NEURAL NETWORKS AND THEIR APPLICATIONS TO INTELLIGENT FAULT DIAGNOSIS OF POWER TRANSMISSION LINES

by
Fatemeh Mohammadi Shakiba

Over the past thirty years, the idea of computing based on models inspired by human brains and biological neural networks emerged. Artificial neural networks play an important role in the field of machine learning and hold the key to the success of performing many intelligent tasks by machines. They are used in various applications such as pattern recognition, data classification, stock market prediction, aerospace, weather forecasting, control systems, intelligent automation, robotics, and healthcare. Their architectures generally consist of an input layer, multiple hidden layers, and one output layer. They can be implemented on software or hardware. Nowadays, various structures with various names exist for artificial neural networks, each of which has its own particular applications. Those used types in this study include feedforward neural networks, convolutional neural networks, and general regression neural networks. Increasing the number of layers in artificial neural networks as needed for large datasets, implies increased computational expenses. Therefore, besides these basic structures in deep learning, some advanced techniques are proposed to overcome the drawbacks of original structures in deep learning such as transfer learning, federated learning, and reinforcement learning. Furthermore, implementing artificial neural networks in hardware gives scientists and engineers the chance to perform high-dimensional and big data-related tasks because it removes the constraints of memory access time defined as the von Neuman bottleneck. Accordingly, analog and digital circuits are used for artificial neural network implementations without using general-purpose CPUs. In this study, the problem of fault detection, identification, and location estimation of transmission lines is studied and various deep learning approaches are implemented and designed as solutions.

This research work focuses on the transmission lines' datasets, their faults, and the importance of identification, detection, and location estimation of them. It also includes a comprehensive review of the previous studies to perform these three tasks. The application of various artificial neural networks such as feedforward neural networks, convolutional neural networks, and general regression neural networks for identification, detection, and location estimation of transmission line datasets are also discussed in this study. Some advanced methods based on artificial neural networks are taken into account in this thesis such as the transfer learning technique. These methodologies are designed and applied on transmission line datasets to enable the scientist and engineers with using fewer data points for the training purpose and wasting less time on the training step. This work also proposes a transfer learning-based technique for distinguishing faulty and non-faulty insulators in transmission line images. Besides, an effective design for an activation function of the artificial neural networks is proposed in this thesis. Using hyperbolic tangent as an activation function in artificial neural networks has several benefits including inclusiveness and high accuracy.

ARTIFICIAL NEURAL NETWORKS AND THEIR APPLICATIONS TO INTELLIGENT FAULT DIAGNOSIS OF POWER TRANSMISSION LINES

by
Fatemeh Mohammadi Shakiba

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology – Newark
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Computer Engineering

Helen and John C. Hartmann Department of Electrical and Computer Engineering

August 2022

APPROVAL PAGE

ARTIFICIAL NEURAL NETWORKS AND THEIR APPLICATIONS TO INTELLIGENT FAULT DIAGNOSIS OF POWER TRANSMISSION LINES

Fatemeh Mohammadi Shakiba

Prof. MengChu Zhou, Dissertation Co-advisor Date
Distinguished Professor of Electrical and Computer Engineering, NJIT

Dr. Seyyedmohsen Azizi, Dissertation Co-advisor Date
Assistant Professor of Electrical and Computer Engineering, NJIT and School of Applied Engineering and Technology

---

Prof. Nirwan Ansari, Committee Member Date
Distinguished Professor of Electrical and Computer Engineering, NJIT

Dr. Hieu Nguyen, Committee Member Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Qing Liu, Committee Member Date
Assistant Professor of Electrical and Computer Engineering, NJIT

Prof. Zhi Wei, Committee Member Date
Professor of Computer Science Department, NJIT

BIOGRAPHICAL SKETCH

Author: Fatemeh Mohammadi Shakiba

Degree: Doctor of Philosophy

Date: August 2022

Undergraduate and Graduate Education:

- Doctor of Philosophy in Computer Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2022
- Master of Science in Computer Engineering,
  Southern Illinois University of Carbondale, Carbondale, IL, 2018
- Bachelor of Science in Hardware Computer Engineering,
  Iran University of Science and Technology, Tehran, Iran, 2014

Major: Computer Engineering

Presentations and Publications:

Fatemeh Mohammadi Shakiba and MengChu Zhou, "Novel analog implementation of a hyperbolic tangent neuron in artificial neural networks", *IEEE Transactions on Industrial Electronics, 68, 11, 10856-10867, 2020*

Fatemeh Mohammadi Shakiba, Milad Shojaee, S Mohsen Azizi, and MengChu Zhou, "Generalized fault diagnosis method of transmission lines using transfer learning technique", *Neurocomputing, Volume 500, 2022, Pages 556-566, ISSN 0925-2312*

Fatemeh Mohammadi Shakiba, S Mohsen Azizi, and MengChu Zhou, "Transfer Learning-based Method to Detect Insulator Faults of High-Voltage Transmission Lines via Aerial Images", *SMC-Mag-2022-05-0021, conditionally accepted*

Fatemeh Mohammadi Shakiba, S Mohsen Azizi, and MengChu Zhou, "Application of Machine Learning Methods in Fault Detection and Classification of Power Transmission Lines: A Survey", *Artificial Intelligence Review, Springer, Conditionally accepted, 2022*

Fatemeh Mohammadi Shakiba, Milad Shojaee, S Mohsen Azizi, and MengChu Zhou, "Robustness Analysis of Generalized Regression Neural Network-based Fault Diagnosis for Transmission Lines", *IEEE SMC 2022 conference, Accepted, Prague, Czech 2022*

Fatemeh Mohammadi Shakiba, Milad Shojaee, S Mohsen Azizi, and MengChu Zhou, "Real-time Sensing and Fault Diagnosis for Transmission Lines", *IEEE Sensors Journal, Under review, 2022*

This dissertation is dedicated to my dear father, who has been nicely my supporter while he was alive, and my beloved mother who always encourages me attentively with her fullest and truest attention to accomplish my work with truthful self-confidence. I also dedicate this thesis to the kindest sister in the world who was always a big supporter of me during my life.

ACKNOWLEDGMENT

This dissertation research is done as a part of my coursework. Throughout the writing of this dissertation, I have received a great deal of support and assistance. First and foremost, I am thankful to my advisor Dr. MengChu Zhou, Distinguished Professor, Department of Electronics and Communication Engineering, New Jersey Institute of Technology, Newark, for his valuable guidance and assistance, without which the accomplishment of this research work would have never been possible. Dr. Zhou has directed me with great enthusiasm and interest and has allowed me to have the freedom to exercise a thoughtful and scientific approach towards the problem.

I would like to thank Prof. Seyyedmohsen Azizi for guiding and supporting me over the last two years of my Ph.D.. He has set an example of excellence as a researcher, mentor, instructor, and role model.

I would like to thank Prof. Durgamadhab Misra for his kind support from the initial day of my program until the end of it. He will stay in my mind as the symbol of great management and mentoring forever.

I would also like to thank Prof. Nirwan Ansari, Prof. Hieu Nguyen, Prof. Qing Liu, and Prof. Zhi Wei for agreeing to be a part of the committee for my dissertation defense and helping me to improve my dissertation.

I have been given a unique opportunity as Teaching Assistant by the Department of Electrical and Computer Engineering. By this funding and through this position, I was able to touch the lives of so many students and motivate and guide them in both professional and personal ways. I would also like to thank the professors who have been my guides for the past years, without whom I would not have been able to share the knowledge in this dissertation.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

LIST OF TABLES
(Continued)

LIST OF TABLES (Continued)

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Problem Statement and Motivation

Nowadays, traveling wave (TW)-based approaches are used in industry to address the problem of detection, classification, and location estimation of the faults [3, 4, 5, 6, 7, 8, 9]. Although such techniques have advantages such as their independence from the network configuration, remaining unaffected by load variations, high grounding resistance, and series capacitor, they suffer from being expensive, requiring high sampling frequency for capturing the high frequency fault transients, the limited capability to distinguish between waves reflected from a fault and the remote end of a TL, and the fundamental limitation of not detecting the faults at zero crossing of the voltage waveform [10, 11, 12]. Moreover, the performance of TW-based methods depends on high impedance faults and source inductance variations. Such variations affect the shape of transient waves such that TW-based methods struggle in capturing the arrival time of the transient waves. For instance, when a double line-to-ground fault occurs at 75 (km), by changing the source inductance from 16.58 (mH) to 50 (mH), the performance of the TW-based method deteriorates dramatically, as shown in Figure 1.1.

The location of faults is calculated based on the TW frequency obtained from the frequency spectrum of a transient voltage signal as [13]:

$$x = \frac{v}{2f_1} \tag{1.1}$$

where v is the propagation speed of TWs and f_1 is the first transient frequency in the frequency spectrum (fast Fourier transform (FFT)) of the voltage signal.

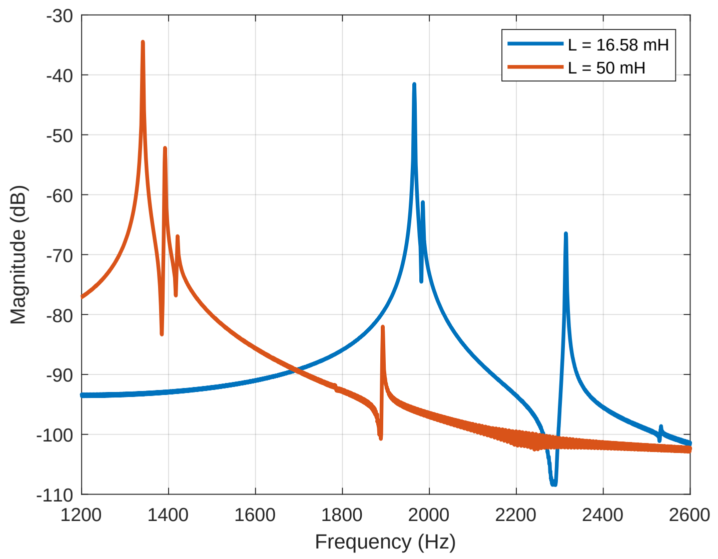


Figure 1.1: Frequency spectrum (FFT) of transient voltage signals for source inductances of 16.58 (mH) and 50 (mH) when a double line-to-ground fault occurs at 75 (km).

As indicated in Figure 1.1, when the source inductance value is 16.58 (mH) and 50 (mH), the first transient frequencies are equal to 1965.4 (Hz) and 1340.8 (Hz), respectively. These two frequencies are equivalent to the estimated distances of 74.64 (km) and 109.42 (km), respectively. Therefore, with the increase of the source inductance, the performance of the fault location estimation deteriorates drastically.

Another deficiency of TW-based techniques in the literature is that they have not reported the accuracy of their fault diagnosis system or any statistical results to show the performance of their approach in terms of mistakenly detected or undetected faults. There are many disturbances, uncertainties, and noise in the TLs that could make transient waves similar to the faulty waves and hence, cause false alarms and power outages. According to the U.S. Department of Energy, it is estimated that power outages cost American businesses around *$150 Billion* annually [14]. Therefore, it is required to have a high level of accuracy in fault detection in order to reduce the rate of false alarm power outages. The machine learning (ML) techniques discussed in this study make use of the amplitudes of the fundamental frequency component of the voltage and current signals to avoid dealing with the transient waves generated from faults. Thus, the performance of such methods is not adversely affected by the changes in current and voltage transients and this feature makes them more effective and economical.

Due to the above-mentioned drawbacks of TW-based approaches, several studies are conducted on fault detection, identification, and location estimation using ML methods such as NNs, SVMs, and Neuro-fuzzy networks., combining with signal processing schemes such as S-transform (ST), FFT, DWT, etc.

In this study, we try to take advantage of the newly emerged deep learning techniques to overcome the drawbacks of the existing old solutions for transmission line problems. We cover identification, detection, and location estimation of overhead transmission line faults. We also consider the problem of broken insulator detection using the aerial images insulators.

A significant amount of work has been done in developing simulation environments for ANNs on sequential machines. However, even the fastest sequential processor cannot provide real-time response and learning for networks with large number of neurons and synapses. Parallel processing with multiple simple processing elements (PEs), on the other hand, can provide tremendous speedups. When implemented in hardware, neural networks can take full advantage of their inherent parallelism and run orders of magnitude faster than software simulations. The accuracy of ANNs is also impacted by the approach used to convolve weights. Conventional ANN architectures rely on GPUs to speed up computationally expensive operations. However, such implementations suffer from imprecise calculations, because each analog input is approximated by a digital value and complex activation functions like Sigmoid and *Tanh* are approximated using look up tables which is inefficient in power dissipation and silicon area. Therefore, recent architectures are shifting to analog domain where the circuit that computes the convolution and the circuit that implements the activation function is analog in nature.

ANNs are usually processed on multicore processors such as GPUs. In GPU architectures, a larger portion of energy is consumed by floating point units (FPUs) in streaming cores. In order to reduce the energy consumption of the FPUs, content addressable memory blocks are used beside each FPU. Once a FPU operation is issued with input operands, the proposed architecture approximately checks whether similar input operands are stored in the associative memory block, while the pipeline stages of the FPU are processed in parallel.

Recently, enormous datasets have made power dissipation and area usage lie at the heart of designs for Artificial Neural Networks (ANNs). Considering the significant role of activation functions in the neurons and the growth of hardware-based neural networks like memristive neural networks, this work proposes a novel design for a hyperbolic tangent activation function (*Tanh*) to be used in memristive-based neuromorphic architectures. The purpose of implementing a CMOS-based design for *Tanh* is to decrease power dissipation and area usage. This design also increases the overall speed of computation in ANNs, while keeping the accuracy in an acceptable range. The proposed design is one of the first analog designs for the hyperbolic tangent and its performance is analyzed by using two well-known datasets including the Modified National Institute of Standards and Technology (MNIST) and Fashion-MNIST. The direct implementation of the proposed design for *Tanh* is proposed and investigated via software and hardware modeling.

1.2 Dissertation Goals and Objectives

Nowadays, more complex settings are required to protect classic transmission lines with high degree of sophistication. The goal of this study is to protect transmission lines in real-time and proposing novel ideas to make this protection as robust and reliable as possible. In this dissertation, we tried to show the great promise of artificial intelligence methods in detection, identification, and location estimation of transmission lines. Different procedures are used to cover and improve all aspects of power delivery network in power network.

Some of the intelligent fault diagnosis approaches that we took advantage of them for overhead transmission lines are deep learning methods including Convolutional Neural Network (CNN) and Generalized Regression Neural Network (GRNN), computer vision techniques such as image augmentation for insulator images, and transfer learning technique for generalizing the proposed solution and making them applicable to larger groups of transmission line. Besides, a novel architecture for hyperbolic tangent activation function is proposed to improve the normal speed and power efficiency of neural networks.

The existing methodologies in literature review, which are in chapter two, are divided into three main categories called generic ML, ANNs, and hybrid methods. The basic idea, fundamental equations, and relevant publications since 2015 are included and summarized for each method. The significance of this survey is highlighted as compared to the existing review studies in this study. Moreover, the advantages and disadvantages of the ML approaches are discussed in details and summarized.

The main goal of chapter three is to analyze the robustness of the proposed detection, identification, and location estimation techniques against the parameter changes in a transmission line, namely fault resistance, fault inception angle, source inductance, phase difference between the two buses, bus voltage amplitude variation, and measurement noise. In addition, a time delay analysis is performed to guarantee that these modules can successfully complete their tasks within the desired time window based on the IEEE standard before the tripping relays disconnect the transmission line.

Chapter four aims to show that a technique based of transfer learning idea can generalize the detection, identification, and location estimation of transmission line method. In this chapter, a specific length is taken into account and the faults of other variants of transmission line lengths are predicted based on the original length. One goal in chapter five is to expand the original CPLID dataset which has only 248 broken insulator images among 3,808 images to generate reliable accuracy, precision, recall and F1 Score. Using the data augmentation approach with different portions, a tremendous balanced dataset with 16,720 images is produced which is a more reliable dataset comparing to the dataset with only 3,808 images. Afterwards, a proposed transfer learning methodology based on VGG-19 CNN is implemented as the base model for transfer learning which is trained using the ImageNet dataset. In the second step, the weights of VGG-19 layers, except the two fully connected final layers, are kept frozen to perform the feature extraction task.

In chapter six, we focus on implementing ANNs in a hardware and a new generation MNN which provides us with high parallelism to run a large application faster than software methods that face von Neuman bottleneck limits. A novel design for *Tanh* activation function is proposed in this study to help the neuromorphic architectures overcome the heavily demanded computations through the layers in the training step. These computations make them the most time-consuming section among all.

Although we tried our best to cover a good depth of the existing research results in this dissertation, there are always some limitations and further studies which can be done in every branch of science. In Chapter seven, the purpose is to clarify some of these ideas and concepts and conclude the works in this dissertation.

1.3 Background

1.3.1 Artificial Neural Network

Computers are impeccable at finding solutions for algorithms and mathematical problems, but world's functions cannot always be defined in mathematical form. Image recognition and language processing are a couple of illustrations for problems that cannot effortlessly be quantified into mathematical concepts. However, recently these tasks are become so important to engineers [15].

In 1982, interest in Artificial Neural Networks (ANNs) was renewed. Their design enables us to analyze data in a comparative way as to our own biological brains, by drawing inspiration from how human nervous system functions. This makes them valuable apparatuses for solving such issues as image or voice recognition, and big data management, which our biological brain can perform efficiently [15, 16, 17].

ANNs consist of an input layer, multiple hidden layers, and one output layer [18]. Each layer has several neurons which have multiple inputs and one output, typically in real range. Each input signal is multiplied by a predetermined weight value known as the synaptic weight. The neuron sums up the convolved signals and maps them to output using an activation function. [19, 20]

Overtime, different algorithms in software and hardware domains are generated for ANNs. Each time, neuroscientists try to make their designs more efficient, especially according to heavy computations in a neuromorphic concept, they are faced with problems like large area of designs, lack of accuracy, and high power consumption. Despite the fact that these designs are costly, they may not be able to provide users with acceptable results.

Therefore, one objective of this dissertation work is to design a power efficient activation function which is compatible with new methods like memristors or crossbar array memories [21, 22, 23].

1.3.2 Biological Neural Network

“Neurons” or nerve cells are the basic functional units of human nervous system, which are located in the brain and other parts of human bodies. The term “Neural” derived from this basic unit, and the human brain can be described as a biological neural network, which is an interconnected web of neurons transmitting complex patterns of electrical signals [24, 25, 26].

Neural network, generally, is an extremely interconnected network of billions of neurons with their interconnections.

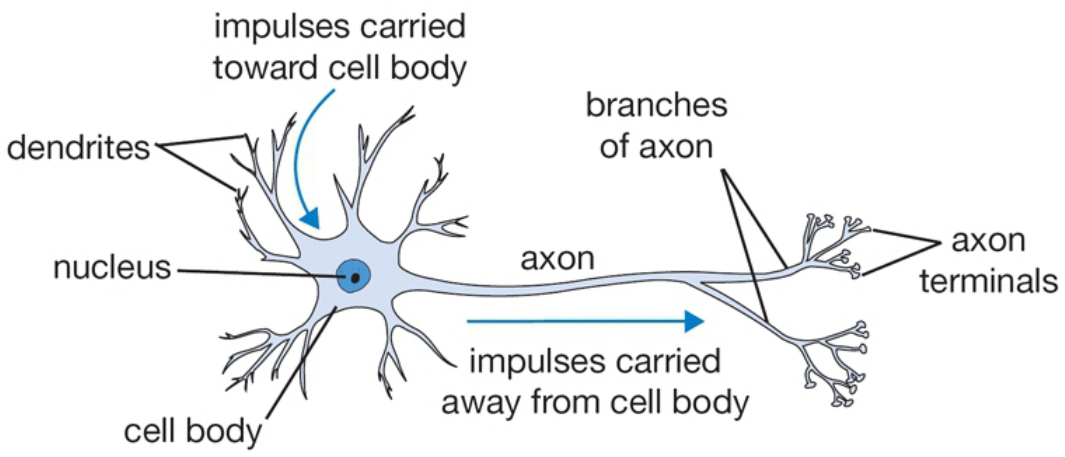

impulses carried toward cell body
dendrites
branches of axon
nucleus
axon
axon terminals
impulses carried away from cell body
cell body

Figure 1.2: A sample drawing of biological neural network [1].

The natural nerve cell of human brain, which is shown in Figure 6.3, consists of four main parts [25].

- Dendrite: This part is like the input port that receives signals from other neurons in the network.
- Soma or cell body: It sums all the incoming signals to generate input for the next neuron.
- Axon: Each neuron has a threshold value to compare it with the input signal summation. If the summation value reaches this threshold, the neuron is fired and the signal passes through the axon to deliver it to the other neurons.
- Synapses: These are like the output port of a neuron. In other words, synapses are the interconnection points among two neurons. This connections can have different levels of strength that determine the amount of signal transmission. Also, they can be increasing or decreasing in nature [27, 26].

Human brain structure has attracted computer scientist since a long time ago. In 1943, a neuroscientist (Warren S. Mcculloch) and a logician (Walter Pitts), formed the first model of an artificial neural network. But, there are always some sort of problems that are incredibly "easy for a human, difficult for a machine" to solve like pattern recognition. In order to use computers for solving this kind of problems, Artificial Neural Networks are invented, which are inspired by biological brain, to make computers capable of performing specific tasks like pattern recognition, data classification, clustering, and etc [25]. Figure 1.3 shows the structure of artificial neuron.

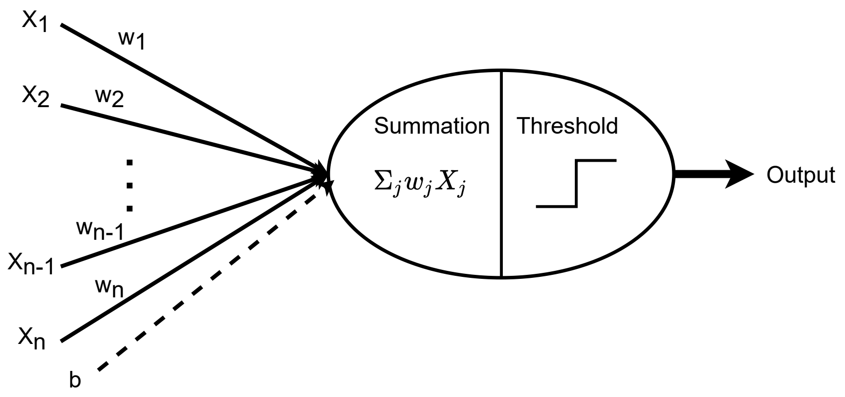


Figure 1.3: A schematic of an artificial neuron (perceptron).

Each artificial neural network receives inputs from the outside world in the form of pattern and image in vector (matrix) form. These inputs are shown as $[X_1, X_2, ..., X_n]$ for n number of inputs. Each of them is multiplied by a synaptic weight. These weights are the information produced by the neural network through learning step, to solve the problem. Literally, these weights present the strength of interconnections between neurons connected from two consecutive layers. These weighted inputs are all summed up in artificial neurons. If the result of this summation becomes zero, a bias will be added to change the output to a value which is not zero, or increase the scale of the system response. The weight and input of the bias is always equal to one. Although the value of summation in a neuron can be any real value, we have limits for the inputs of next layer. Especially, if we are not able to provide each layer with acceptable range of inputs, the network cannot work properly. To solve this problem, scientists set up a threshold value and introduced different activation functions for different kinds of neural networks [1].

1.3.3 Activation Functions in ANNs

Activation functions are used between layers to moderate the output values of previous layer to the acceptable range of inputs for next layer. In order to achieve this goal, people have used various functions.

Every activation function takes a single number and performs a certain fixed mathematical operation on it [1]. There are some useful characteristics in activation functions that should be considered while we want to choose the appropriate one for our model. We mentioned these features below [28]:

1. Non-Linearity

   The activation function goal is to provide the network with non-linearity that allows us to model a response variable such as target variable, class label, and etc. using the non-linearly with its explanatory variables. In general, non-linear means that one output cannot be reproduced from a linear combination of the inputs. Another way to express non-linearity is to say that without a non-linear activation function in a neural network, despite having multiple layers, it would behave like a single-layer perceptron, because adding these layers would result in another linear function.

2. Continuously differentiable

   For enabling gradient-based optimization methods, we need to choose the activation functions that have this property. For example, the binary step activation function that is not differentiable at 0, and its differentiation is 0 for the other values, should not be opted for gradient-based methods.

3. Range
   Gradient-based training methods are more stable when there is a finite range for the used activation function. The reason is that pattern presentations significantly affect only limited weights.

4. Monotonic
   In error calculation, when the used activation function is monotonic, there would be a guarantee that error surface associated with a single-layer model will finally converge.

5. Approximates identity near the origin
   For activation functions with this property, the neural network will be capable of learning efficiently when its weights are initialized randomly. But, when the activation function does not approximate identity near the origin, we cannot initialize weights randomly and we need special attention while initializing the weights.

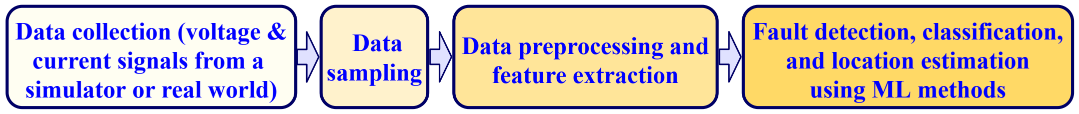
Artificial neural networks (ANNs) are utilized in a wide scope of applications. These architectures consist of an input layer, multiple hidden layers, and one output layer. ANNs can be implemented in a software or hardware domain [29, 30, 31, 2]. In ANNs, layers include several neurons, and each neuron can have multiple inputs and one output, which are real numbers most of the time. The output of each neuron is connected to neurons of the next layer through links called the synapses. Each synapse has a weight value to represent the strength of the connection between two neurons. Neurons compute the sum of all weighted inputs and transfer it into an output signal by an activation function. Based on recent studies, scientists are moving toward hardware replacements for ANN to reduce the existing calculation complexity of software implementations [2, 32, 33].

Implementing ANNs in hardware provides us with a boost for large networks because it removes the constraints of memory access time defined as von Neuman bottleneck. Accordingly, analog and digital circuits are used for ANN implementations without using general-purpose CPUs [34, 35, 36, 37, 38, 39]. Emerging ANN paradigms use memristors to implement a synapse and they benefit the crossbar array structures of memristors to perform analog multiplication in parallel by exploiting the Ohm's law [2, 40, 41, 42, 43, 44, 45, 46, 47]. These architectures use reconfigurable Memristive Crossbar Array (MCA) to perform power-efficient and high-performance multiplication and addition operations [47, 48, 49]. Memristive Neural Networks (MNNs) include a pair of memristors to store synaptic weights (positive or negative). There are different types of MNN architectures like Inverter-based MNN [50], MCA-based MNN [41], fully MNN [51], etc. This study considers a multilayer feedforward MNN [41] which is taking advantage of a novel analog design for hyperbolic tangent as an activation function of the layers. Feedforward MNN in [43] includes a dual column structure in which two adjacent memristors are placed in a row to store synaptic weights.

The mentioned approaches store a weight value in one of the two adjacent memristors while the other one is in a very high resistive state so no current can pass through it. The sign of the weight value determines which memristor is active and stores the weight value. Although several non-ideal characteristics of memristor synaptic devices are detected up to now [52, 53], MNN is one of the most efficient architectures in terms of speed and power consumption that can defeat its digital counterparts [2, 39, 54, 55].

1.3.4 Transmission Lines

Due to the accelerated expansion of power systems and microgrids in recent years, power system operators (PSOs) face various challenges to provide end-users with uninterrupted electrical power. Failures and interruptions are inevitable in all systems, and hence engineers try to protect power systems by providing accurate fault detection and identification modules. Four main steps should be performed for fault detection, classification, and location estimation of transmission line (TL) faults as indicated in Figure 1.4.


Data collection (voltage & current signals from a simulator or real world)
Data sampling
Data preprocessing and feature extraction
Fault detection, classification, and location estimation using ML methods
Figure 1.4: The overview of identification and localization of TL faults.

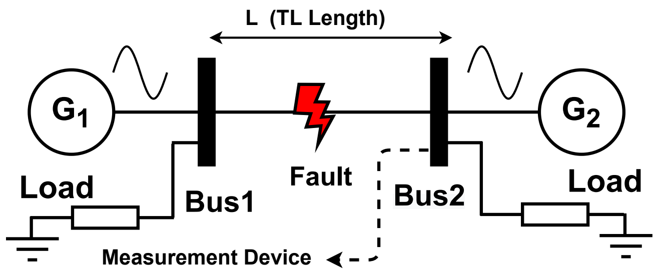


Figure 1.5: Schematic of a TL system.

The initial step is data collection from current and voltage signals of TLs generated by the real power networks or power system simulators such as Matlab/Simulink. Next, data sampling approaches are conducted. The third step is data preprocessing and feature extraction methods such as Discrete Fourier Transform (DFT) and Discrete Wavelet Transform (DWT) which are performed to extract the significant features. Last, an approach is chosen for fault identification (detection and classification of TL faults) and/or fault location estimation. Figure 1.4 shows these steps in sequence and Figure 1.5 shows a TL system including two generators connected through a three-phase TL with the length of L.

A TL fault diagnosis problem consists of three sub-problems, i.e., detection, classification, and location estimation of faults. Fault detection methods take 2-10 (mS) while classification methods normally take 30 (mS). The maximum time limit for these three steps of fault diagnosis should be less than 100 (mS) in total [56].

TLs are vulnerable to circumstances such as overloading, breakage, short circuits, icing, lightning strikes, tree interference, bird nesting, aging, human activity, hurricanes, and in general lack of protection and conservation [57, 58, 59]. TL faults caused by these phenomenons can be categorized into series (open conductor) and short-circuit (shunt) faults.

1.4 Dissertation Outline

The first chapter of this dissertation discusses the preliminaries which are utilized in the following chapters. Also, a transmission line as a system with imbalanced datasets is well discussed. In the second chapter, a comprehensive review of ML methods for TL fault detection, classification, and location estimation is presented. In the third chapter, the problem of fault detection, identification, and location estimation of transmission lines using two NNs, namely, GRNN and CNN is studied. In the fourth chapter, a deep learning methodology is proposed based on the transfer learning technique to improve the response time of the insulator image classification problem. The fifth chapter proposes a novel architecture for hyperbolic tangent activation function in neural networks. In chapter six, the contributions of this dissertation are summarized and the limitation of every work is discussed. Also, the potential extension for each chapter is presented to provide a clear pathway to future research directions.

CHAPTER 2

LITERATURE REVIEW

2.1 Classification

The expansion of power systems and smart grids calls for advanced fault diagnosis techniques to prevent undesired interruptions and expenses. One of the most important part of such systems is transmission lines. This study presents a survey on recent machine learning-based techniques for fault detection, classification, and location estimation in transmission lines. In order to provide reliable and resilient electrical power energy, faster and more accurate fault identification tools are required. Costly consequences of probable faults motivate the need for immediate actions to detect them using intelligent methods, especially emerging machine learning approaches that are powerful in solving diagnosis problems.

This chapter presents a comprehensive review of various machine learning methodologies including naive Bayesian classifier, decision tree, random forest, k-nearest neighbor, and support vector machine as well as artificial neural networks such as feedforward neural network, convolutional neural network, and adaptive neuro-fuzzy inference system that are used to detect, classify, and locate faults in transmission lines.

Classification is a procedure of categorizing a data set into 2 or more classes. This process can be performed on structured or unstructured data. It initiates with considering a labeled data set to train the system. After training level execution, the system will be able to predict the class of given unlabeled data points [60]. Categorization, effortless and instantaneous for people, stays a primary challenge for artificial intelligence [61]. It goes without saying that even a four year old baby can distinguish between a dog and a cat after teaching him/her once or twice, but interpreting such a task for machines requires a lot of work and knowledge. Consequently, although classification seems an easy process at the first glance, explaining a scene from image data, detecting and describing objects and their relationship demands much effort and research. Classification problems might be so simple and binary with two classes or multi-classes. There are some classification problem examples like: speech recognition, handwritten recognition, image classification, illness diagnosis and etc. [60]. For the time being, there are countless types of data sets that can be generated regarding any little thing around us. The ones that we deal with under classification subject can be classified for purposes like efficiency in time or power consumption, extracting useful knowledge from a bunch of useless data, categorizing based on different necessitates, and more instances like this.

Nowadays, imbalanced data sets attract data scientists' attention because of the major differences that exist between these data sets and the normal (balanced) ones. One noticeable point about different existing classes in every data set is their proportion, or the ratio of each class in the whole data set. Sometimes, we have normal cases such as one tenth to one, but in some rare cases (which can include important data sets) classes are represented absolutely unevenly which are called imbalanced data sets.

In fact, imbalanced data classification is a name that alludes to problems with categorization where the number of instances in different categories have a huge proportional difference, such as 1000 to 1. In such a case, if we assume all given data points are from the major class, the accuracy value will be 99.9% which is excellent in the first sight, but such a model cannot detect the faulty data points at all. In these cases, we have to consider other parameters such as recall, ROC, F1 score, and etc. This problem is predominant in scenarios that faulty point detection is vital such as electricity pilferage, fraudulent transactions in banks, identification of rare diseases, etc. In this obstacles, the predictive classification model using conventional machine learning procedures or neural networks could be problematic and inaccurate [61]. Furthermore, skewed distribution of imbalanced data set causes conventional machine learning and neural networks algorithm facing a lot of challenges such as less efficiency, inaccuracy in results and misrepresentation of common ML, or data classification of existing functions or methods. Therefore, classification and regression algorithms are not able to predict minor classes precisely. These problems become more noticeable when we consider multi-class classification problems [62].

Classification is a procedure of categorizing a data set into 2 or more classes. This process can be performed on structured or unstructured data. It initiates with considering a labeled data set to train the system. After the training level execution, the system will be able to predict the class of given unlabeled data points. The classification predictive modeling is a concept to express approximation of the mapping function from input data to clustered output data points. These models aim to figure out which new given data belongs to which category or class. For instance, a received email needs to go to the inbox or the spam folder (Binary classification) [61].

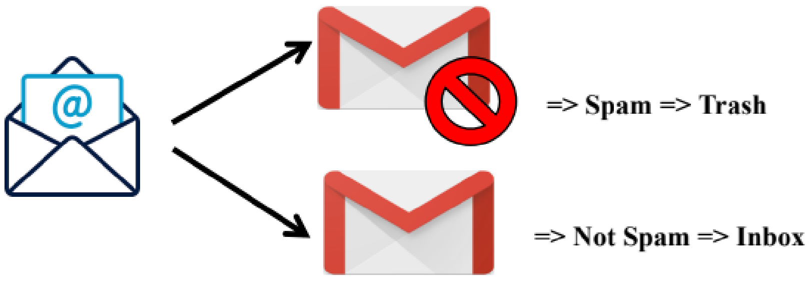


Figure 2.1: Binary classification of emails.

Below we name few types of classification algorithms in machine learning [62]:

- Linear Classifiers: Logistic Regression, Naive Bayes Classifier
- Nearest Neighbor
- Support Vector Machines

- Decision Trees
- Boosted Trees
- Random Forest
- Neural Networks

Naive Bayes Classifier (Generative Learning Model) This is a classification algorithm based on Bayes' Theorem with hypothesis of independency among predictors. A Naive Bayes classifier considers no relation among data set feature. This technique is easy to build and works good enough with big data because of its scalability [62].

2.1.1 Nearest Neighbor

K-nearest-neighbors algorithm is a supervised classification method which uses nearness as a parameter of similarity. This algorithm takes some labeled data points and uses them to label the new given points based on their closeness to existing points in each class. After checking with the "K" nearest neighbors, the new data point will be categorized with the label which most of the neighbors have. For this algorithm, choosing a distance metric is the critical point based on the application, because in some cases such as text-based data sets geometric distance is not applicable [62].

2.1.2 Logistic Regression (Predictive Learning Model)

This method is a statistical model to analyze data sets with one or more independent features that result in the outcome. Results are measured by dichotomous variables (each includes only two possible outcomes). Logistic regression aims to find the best fitting model to explain the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor) variables. This algorithm works better than some others such as k-nearest-neighbors for binary classification because of considering factors that lead to proper quantitative classification [62].

2.1.3 Decision Trees

This method works based on a tree structure. In this algorithm, data sets are broken down into smaller and smaller subsets while the tree is developed and its depth increases. Therefore, final outcome is a tree with decision and leaf nodes. Each leaf node stands for a classification or decision and each decision node has at least two branches to decide. The root node (topmost decision node) is the best predictor. Decision trees are able to handle categorical and numerical data [62].

2.1.4 Random Forest

Random forests or random decision forests are a learning algorithm for classification, regression, and similar applications which work by building up a multitude of decision trees at training step and resulting the class which is the status of classes (classification) or mean prediction of each tree. The advantage of random decision forests over decision trees is decreasing the chance of over fitting [62].

2.1.5 Neural Networks

A neural network is built with multiple units called neuron which are arranged in layers. This structure transform the input data (vector) into output data points. Each neuron in each layer is supposed to take an input, apply a function (activation function) to it and then pass the output to the next layer. Usually, neural networks in this concept are feed-forwarded and so there is no feedback to prior layers. Weights, which are tuned in training step base on the application usage, apply to passing signals forward in these neural networks [62].

2.2 Imbalanced Data Set

Every person that has done some studies in machine learning and classification concepts, definitely has come across imbalance data distribution. In such data sets, number of data points in one or more classes are significantly less than the instances in the other classes. Generally, if the events existing in a minority class are less than 20% of the total data set, that data set is identified as an imbalanced data set. For calculating the exact threshold line between balanced and imbalanced data sets, we can use the Shannon entropy as a measure of balance. On a data set of n instances, if you have k classes of size c_i, one can compute entropy as follows:

$$H = -\sum_{i=1}^{k} \frac{c_i}{n} \log \frac{c_i}{n} \tag{2.1}$$

This is equal to:

- 0 when there is one single class. In other words, it tends to 0 when your data set is very unbalanced
- $log(k)$ when all your classes are balanced of the same size n/k

Therefore, one could use the following measure of Balance for a data set:

$$Balance = \frac{H}{\log k} = \frac{-\sum_{i=1}^{k} \frac{c_i}{n} \log \frac{c_i}{n}}{\log k} \tag{2.2}$$

Which is equal to:

- 0 for a unbalanced data set
- 1 for a balanced data set

Considering the fact that machine learning algorithms aim to decrease the errors and develop better accuracy, they cannot distinguish this faulty points as a class and work based on its prediction. In other words, they are prone to wipe them out. What a mess? Instead of detecting the faulty points in a system data set, or sick people among all tested patients, we just ignore them and this is of course going to be a tragedy. So what is the solution then? Seeing regular classification algorithms cannot perform accurately when faced with imbalanced data sets, some regularization, modification and adjustment are useful to solve these problems. Foe instance, Decision Tree or Logistic Regression tend to predict majority class data and treat the below 5% minority classes as noises. Classification methods' efficiency is measured by a Confusion Matrix which consists of information regarding actual and predicted data points. Table 2.1 show a confusion matrix and the driven metrics from it come after as Equations 2.3 - 2.6 [63]:

Table 2.1: The Confusion Matrix

Actual Data	Prediction Data	
True Label	Class 1	Class 2
Class 1	TP	FN
Class 2	FP	TN

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.3}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.4}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.5}$$

$$F1\ Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{2.6}$$

To deal with imbalanced data sets, some strategies such as data preprocessing or classification algorithm modification emerged. Various methods are proposed to solve the imbalanced data classification problem [64, 65], and they can be categorized into three kinds [60]: (i) Sampling based methods: They use under sampling or oversampling techniques to transform the class-imbalanced data set into a balanced one [66]. In summary, the main solution to handle these data sets is to increase the density of minority classes, or on the other hand, decreasing the number of data points in the major classes in order to achieve almost same proportion of data in all classes; (ii) Cost-sensitive learning-based algorithms [67, 62]: This kind of methods work with the costs associated with miss-classifying data points; And (iii)Ensemble learning based approaches [68, 61]: This type of methods aim to increase the efficiency of each classifier by taking advantage of multiple classifiers and combining them to create a new and more efficient classifier [64].

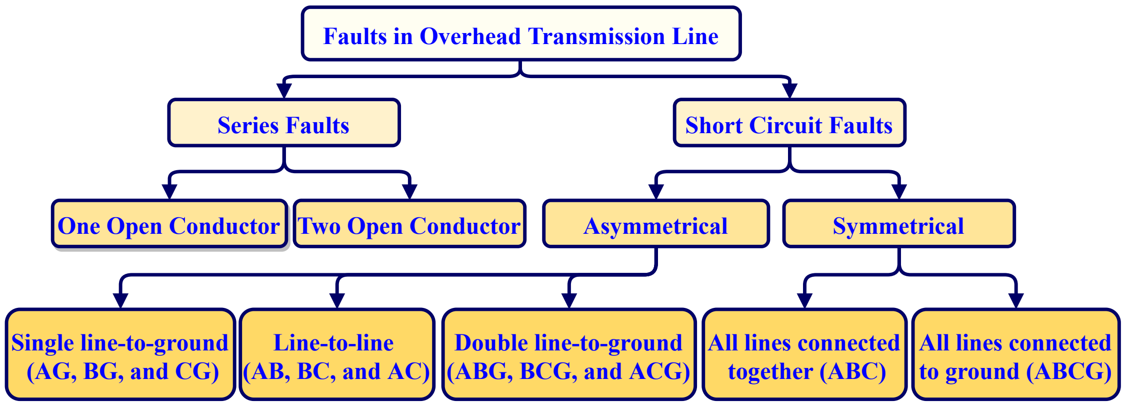
TLs are vulnerable to circumstances such as overloading, breakage, short circuits, icing, lightning strikes, tree interference, bird nesting, aging, human activity, hurricanes, and in general lack of protection and conservation [57, 58, 59]. TL faults caused by these phenomenons can be categorized into series (open conductor) and short-circuit (shunt) faults.

Series faults can be recognized easily by observing every phase voltage. If an open conductor fault happens, the voltage values would change, which is a sign of a series fault. These faults are classified into two types including one open conductor, and two open conductor faults. These faults can rarely occur in TLs and therefore usually are not considered in the studies [75].

In this study, short-circuit faults, as more common faults than open-circuit ones in TLs, are discussed. These faults can be classified into four types: single line-to-ground, double line-to-ground, line-to-line, and all-lines-connected (-to-ground), as shown in Figure 2.2.

Table 2.2: Comparison Between Existing Survey Papers and This work

Ref.	Focusing Aspects	Before Year
Haque et al. [69]	An overview on neural network (NN) techniques used in power systems	2007
Hare et al. [57]	A survey on various failure modes in microgrid components consisting both clean and conventional energy generation systems, and on the existing fault diagnosis approaches	2016
Prasad et al. [70]	A review on the proposed technologiesfor fault classification in power TLs	2016
Prasad et al. [71]	A review on the fault classification methods of TLs and a brief introduction to TL faults	2017
Mishra et al. [72]	A review on all methods including signal processing techniques, impedance-based measurement and traveling wave phenomenon-based methods, artificial intelligence-based method and some special technique for the detection, classification, and location estimation of faults in TLs	2017
Prasad et al. [73]	A survey on the location estimation of faults in overhead TLs and different fault location estimation techniques based on ANN.	2017
Parihar et al. [74]	An overview of cable fault detection methods in 3 main categories namely, Impedance base, traveling waves, and magnetic field	2018
Raza et al. [56]	A review on fault diagnosis methods in power TLs including fault detection, classification, and location estimation, and a comparison between different artificial intelligence and ML algorithms.	2020
This survey	A survey on ML methods and their usage and implementations in TLs fault detection, classification, and location estimation.	2022


Faults in Overhead Transmission Line
Series Faults
Short Circuit Faults
One Open Conductor
Two Open Conductor
Asymmetrical
Symmetrical
Single line-to-ground (AG, BG, and CG)
Line-to-line (AB, BC, and AC)
Double line-to-ground (ABG, BCG, and ACG)
All lines connected together (ABC)
All lines connected to ground (ABCG)
Figure 2.2: Types of faults in a three-phase transmission line (A, B, and C represent the three phases, and G stands for ground connection).

This survey covers the majority of Artificial Intelligence (AI) approaches used in the studies published since 2015 and discusses their advantages and disadvantages in TLs fault diagnosis problems. We divided this study into two main sections based on Figure 2.3 due to numerous ML methods used in this area. Based on Figure 2.3, we defined the ML methods, which are not known as deep learning (artificial neural networks (ANNs)), as generic machine learning methods, and deep learning techniques are presented as ANNs.

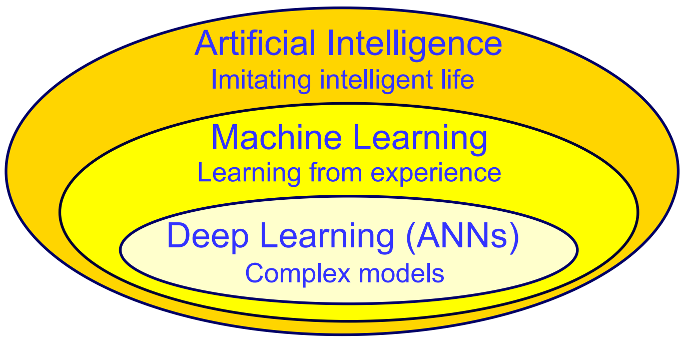


Figure 2.3: Deep learning is a subcategory of machine learning, which is a subcategory of artificial intelligence.

Table 2.2 reviews some of the previous surveys and compares them with this survey study.

This chapter is organized as follows. Section 2.3 explains different generic ML methods. Section 2.5 describes various ANN techniques used for the TL protection. Section 2.6, reviews a few studies using hybrid approaches. Section 2.7 compares all methods and discusses their characteristics. Section 2.8 summarizes the over all ideas in this chapter.

2.3 Generic Machine Learning Methods

2.3.1 Naive Bayesian Classifier (NBC)

NBCs and Bayesian Classifiers (BCs) are well-known for their power in describing the extracted features of signals and generating solid results about these features in uncertain situations. They are directed acyclic graphs that assume a conditional probabilistic distribution for each node. Each node represents one variable's domain and can connect with the others by using its probabilistic dependency [76, 77]. To solve a classification problem using a Bayesian network, nodes are used as the fundamental features. BCs optimize the correlation between core features and other casual dependencies among non-fundamental ones. Consequently, this algorithm is capable of detecting the most probable output generated by a specific input.

The advantages of BCs in comparison with binary classifiers such as decision trees, neural networks, etc. are their compatibility with Bayesian network systems for data modeling, capturing the informal dependencies in the learning process, readaptation based on modifications of a variable, combining the existing information and newly added data, and taking advantage of statistical models besides Bayesian networks to overcome over-fitting issues. In BCs, the probability shows the level of correctness, and NBC operates based on the class conditional independence assumption that is shown as [77, 78, 79, 80] (2.7) below.

$$P(y \mid x) = \frac{P(x \mid y)P(y)}{P(x)} \tag{2.7}$$

In equation (2.7), $P(x)$ and $P(y)$ stand for the independent probabilities (predictor prior probability) of x and y, respectively. $P(x \mid y)$ shows the conditional probability (likelihood) of x when y is considered as a true value, and vice versa for $P(y \mid x)$.

Swetapadma et al. [81] use an NBC to protect parallel TLs from inter-circuit faults by detecting and classifying the faults. The fundamental components of given three-phase currents from two circuits are obtained by discrete Fourier transform (DFT) and fed into NBC for fault detection. After the fault detection step, the classification module becomes active. Zero sequences are added to the previous data, and then for each phase a separate classifier is applied. Finally, based on each classifier output, the type of fault is determined.

Adly et al. [82] use a similar fault detection and classification method. This study presents sensitive and automated fault identification scheme to solve the existing difficulties such as high-impedance faults (HIFs), non-linear modelling of arcing, etc. An Adaptive Wavelet Algorithm [83, 84, 85] is used for classifying transitory events caused by TL faults based on combining DWT and Bayesian Linear Discrimination (BLD) analysis. This algorithm uses DWT to decompose signals into coefficients with the sampling frequency of $500(kHz)$ and then uses BLD analysis for classification. Aker et al. in [86] decompose three-phase fault current waveforms into several levels by using Daubechies mother wavelet of Wavelet Daubechies (db4) to extract the features, such as standard deviation and energy values. Then, the extracted features are used to train classifiers such as Multillayer Perceptron (MLP) Neural Network, BC, and NBC to classify the types of faults.

In [87], voltage and current signals are given to a DWT for feature extraction, and NBC is used to classify TL faults. Another study [77] presents a technique for TL multiples faults diagnosis using both of the experimental and calculated leakage current signals' harmonics as residuals. This approach integrates the qualitative trend analysis (QTA) for the features extraction with an NBC for the fault diagnosis. In this procedure an NBC is designed to identify the most critical fault in a multiple faults scenario using the leakage current data.

Although all of the NBC-based studies have achieved 95-100% accuracy in TL fault classification, none of them predicts the location of faults. DFT and DWT are two essential signal processing approaches to complete NBC-based classification, and the sampling frequencies for the above-mentioned studies vary from $1(kHz)$ to $20(kHz)$.

2.3.2 Decision Tree Classifier (DTC)

DTC provides classification and regression models by imitating a tree and generating branches equal to the number of classes. The algorithm starts from the root of a tree, and in each step moving downward, it splits the given dataset into smaller subsets based on the feature. Therefore, a decision tree is developed gradually with decisions and leaf nodes [88, 89, 90].

In general, decision trees are divided into two categories, namely, classification trees with 'fit' or 'unfit' outcomes, and regression trees when the target value is continuous. Their core algorithm is ID3 [88], which is a recursive algorithm that selects the best attribute to split the data and develops the leaf nodes of the tree until the classes become complete. ID3 uses entropy and information gain for decision tree construction. Entropy is a methodology to calculate the homogeneity of data points. It gets any real value between zero and one for the maximum and minimum homogeneity level. Entropy (E) is represented as (2.8) below [91, 92].

$$E(S) = -\sum_{x \in X} P(x) log_2 P(x) \tag{2.8}$$

In (2.8), $Entropy(S)$ is the entropy of dataset S, X shows the set of all existing classes in S, and $p(x)$ is the probability of existing elements in class x. The definition of information gain (I_G) is shown in (2.9) [91, 92].

$$I_G(A,S) = E(S) - \sum_{t \in T} P(t)E(t) \tag{2.9}$$

where T stands for the created subsets from dividing set S based on attribute A such that $S = \bigcup_{t\in T} t$ and $P(t)$ is the probability of existing elements in class t belonging to dataset S. Higher information gain in branches of DT means a better model and therefore a more accurate classification is achieved.

Many studies take advantage of DTC to classify the existing faults. A recent one [93] considers current signals and extracts DC offset parameters and fundamental components to feed them to a bagging DTC. This process considers one decision tree for each training subset of the dataset. Chaitanya et al. [94] propose a decision tree aided traveling wave-based methodology for multi-terminal TLs. They consider each fault at its inception time that is the origin of traveling waves to propagate along the line in both directions getting away from the fault point.

The work [95] applies a singular value decomposition principle besides the Fast Discrete Orthonormal S-transform (FDOST) and bagged DTC to detect and classify TL faults.

Another approach is presented in [96] where magnitude of differential power (MODP) is used. Differential power is defined as the multiplication of the difference between the magnitude of the real and estimated voltage and current phasors. In this study, MODP is given to DTC as input data.

By using DTCs based on the above studies, one could classify the TL faults with 99.29-100% accuracy possessing sampling frequencies in the range of $200(Hz)$-$200(kHz)$. Various signal processing methods are used in DTCs including Stockwell transform, Adaline algorithm, FDOST, and DWT.

2.3.3 Random Forest Classifier (RFC)

RFC is a supervised learning algorithm that consists of multiple trees as classifiers. In other words, random forest (RF) merges decision trees to obtain more accurate and reliable predictions. In RFCs, each classifier uses a random vector sampled independently from the input vector, and then each tree generates a unit vote for the associated class to classify an input vector. This classifier is used for classification problems considering randomly selected attributes or a combination of attributes at each node to grow a tree based on that origin. RFC takes advantage of the Gini Index as an attribute selection measure that quantifies the amount of impurity of an attribute corresponding to a class. Gini Index is calculated based on the subtraction of the sum of squared probabilities of classes from one, which is expressed as follows [97]:

$$Gini\ Index = 1 - \sum_{x \in X} P(x)^2 \tag{2.10}$$

In (2.10), $P(x)$ is the probability of class x belonging to a set of classes X in dataset S.

In the process of building an RFC, every time a tree with maximum depth is grown based on a combination of attributes. These trees are not pruned. Because the generalization error converges even without pruning the trees, over-fitting is not an issue according to the strong law of large numbers [98, 99, 100] in this classifier. In RFCs, the number of attributes to be used at every node and the number of trees to be grown are two parameters that users determine based on the applications [101].

Yin et al. [102] presents a methodology to solve the problem of uncertainty and diversity of faults, and to enhance the fault detection accuracy. This detection method is designed based on integrated RFC, improved multi-scale permutation entropy

(IMPE), and wavelet packet transform (WPT). In the first step, a singular value decomposition is applied to filter the current signal, and then the high-dimensional fault features are constructed using the wavelet packet energy and the wavelet packet energy-entropy. Finally, the high-dimensional fault features are used to train the RFC to detect the type of faults.

Wu et al. [103] propose an intelligent fault detection and classification technique for high voltage DC TLs. They use RFC to categorize the faulty current signals. Koochi et al. [104] present an approach to predict the types of faults in real-time. They use pre-disturbance and post-disturbance current and voltage phasors as measured by Phasor Measurement Units (PMUs) to generate the useful attributes and feed them to DTC to determine the coherent groups of generators precisely. In this study, mathematical morphology (MM) is applied to voltage signals in the presence of disturbance to extract important features.

Fonesca et al. [105] show the use of RF method with a simple preprocessing step using the notch filter to distinguish the type of faults in TLs. The performance of this scheme is compared with a NN to show its efficiency. Using k-fold cross-validation to train, test, and compare the models, this study achieved the mean accuracy of 89.59% for the NN and 91.96% for the RF for testing data.

The accuracy range obtained by the above-mentioned studies is 93.54-100% and the range of sampling frequencies is $200(Hz)$-$200(kHz)$ in RFC-based approaches. Hilbert transform, DWT, and S transform are the signal processing methods used for TL fault identification and location estimation alongside RFCs.

2.3.4 Support Vector Machine (SVM) Classifier

One of the commonly-used classification algorithms is SVM that is a supervised ML method and is able to solve regression and classification problems [106]. Its advantages over other ML algorithms include being fast and simple to be implemented, having a low computation load, and producing high accuracy level even with a limited amount of data. SVM finds a hyperplane in an N-dimensional space (N is the number of attributes) to classify the data.

Finding the maximum margin for all classes is the optimization objective in SVM. Maximum distances generate higher classification accuracies for future data points. In this algorithm, the closer data points to a specific hyperplane will have more impact on the hyperplane position and the margins of the classifier. These crucial data points are called support vectors. Although SVMs are initially introduced for binary classification, nowadays they can solve multi-class classification problems such as TL fault classification problem. This is a one-versus-all problem where each fault is compared to the rest of fault types every time. SVMs are able to classify the faults in TLs and take advantage of an alternative loss function to estimate the fault locations. SVM uses linear or non-linear kernels such as polynomial, Gaussian, and radial basis function (RBF) for various kinds of problems depending on the dataset. It aims to minimize the following cost function (2.11) [103, 107, 108, 106].

$$[\frac{1}{n}\sum_{i=1}^{n} max(0, 1 - y_i(w^T x_i - b))] + \lambda \mid\mid w \mid\mid^2 \tag{2.11}$$

In (2.11), y_i is the i^{th} target, w is the weights matrix, and $w^T x_i - b$ is the i^{th} output. λ stands for the trade-off between enhancing the margin size and keeping the data point x_i on the correct class. There are numerous studies that use SVM to perform location estimation and classification of TL faults.

Figure 2.4 shows a naive idea of SVM-based approaches. In this figure, the black dotted lines are the hyperplanes that separate fault types that are shown as different shapes such as triangle, circle, square, and etc. The thick gray lines show the margins between classes and the solid black lines stand for the maximum margin decision hyperplane.

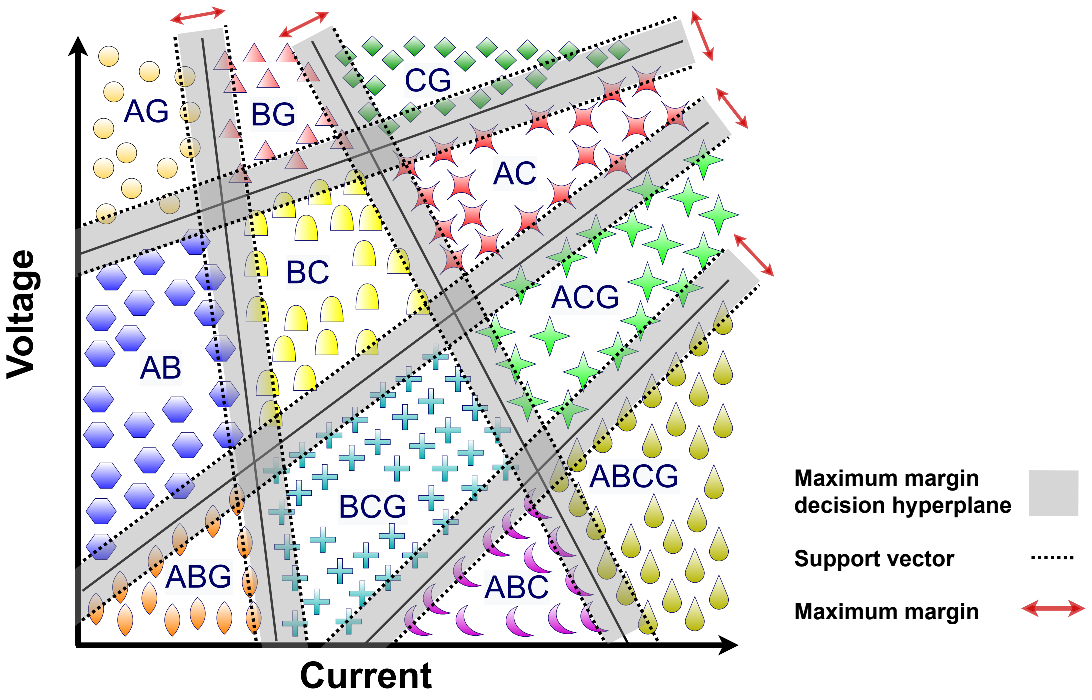


Figure 2.4: Schematic of 11 transmission line faults (AG, BG, CG, AB, BC, AC, ABG, BCG, ACG, ABC, and ABCG) classification using SVM.

Patel [109] proposes a methodology for fault detection, classification, and location estimation of overhead and underground cables in TL networks. The entropy principle and fast discrete orthogonal S-transform (FDOST) are used for feature extraction, and SVM with a regression model is applied for pattern generation to determine the fault types and locations. Reddy et al. [110] present an approach that extracts the energy coefficients from the transient current signals when different types of faults occur. Then, discrete orthogonal stockwell transform (DOST) is applied for real-time fault detection and classification. Finally, SVM determines the location of faults based on extracted coefficients. In another study [111], current signals from TLs are preprocessed by using wavelet packet transform (WPT) and the selected attributes are classified by using SVM. A pattern recognition method is presented by Moravej et al. in [112] based on extracted features from one cycle of post fault data by

FDOST. Then the most important features are fed to 11 SVMs for the classification of faults. In [113], different fault impedance values and fault types are considered in training and testing data. The Gaussian RBF kernel is used to train SVMs for precise classification of faults precisely. Singh et al. [114] propose a method that uses SVMs for the detection and classification of a single line-to-ground fault. They also utilize SVM for the estimation of fault locations in TLs.

The study [115] focuses on an SVM-based fault detection and localization to improve the efficiency of power grids using phasor measurement units (PMUs) that are generated only from a single generator bus. Huang et al. [116] classify ten types of TL faults by using an empirical wavelet transform (EWT) and local energy feature vectors and taking advantage of SVMs. The work [117] combines method of DWT and SVM for fault classification of inter-circuit, cross-country, and high resistance faults. Johnson et al. [118] monitor the rectifier-side AC RMS voltage, the DC voltage, and the current of TLs continuously, and feed them to SVM as inputs for fault detection. Their SVM multi-class classification module distinguishes the types of faults, and SVM regression algorithm detects fault locations.

The accuracy range for SVM-based studies varies from 70% to 100% because of different training approaches and various signal processing methods, and their sampling frequency range is 100 (Hz) to 200 (kHz). SVM is a user-friendly method and can be implemented easily. Therefore, almost one-tenth of all considered papers in this survey belong to this category.

2.3.5 k-Nearest Neighbor (k-NN)

k-NN algorithm was first introduced in the early 1950s. Since then, it is widely used in the field of pattern recognition and data classification. This algorithm works based on the similarity of a given test sample with training samples [119, 120]. There are two important parameters in this algorithm: distance matrix and the number of nearest neighbors which is shown with k. In k-NN algorithm, a suitable nearest neighbor is chosen based on the calculated distance using several matrices such as Euclidean, Cityblock, Minkowski, Chebychev, and Manhattan distance. Having two data matrices with mn size, one with row vectors $x_1, x_2, ..., x_m$ and another with row vectors $y_1, y_2, ...y_m$, different distance matrices between the vectors x_i and y_j can be described as below. For example, Minkowski distance, $D_{Minkowski}$ can be defined as Eq. (2.12).

$$D_{Minkowski} = \sqrt[p]{\sum_{j=1}^{n} \mid x_{sj} - y_{tj} \mid^p} \tag{2.12}$$

In the above equation, $D_{Minkowski}$ is the distance between vectors s and t, x and y are twopints, p is a positive integer describing the exponent of Minkowski distance. Euclidean, Cityblock, Chebychev, Manhattan distances are defined in Equations 2.13, 2.14, 2.15, and 2.16, respectively.

$$D_{Euclidean} = \sqrt{(x_s - y_t)(x_s - y_t)^T} \tag{2.13}$$

$$D_{Cityblock} = \sum_{j=1}^{n} \mid x_{sj} - y_{tj} \mid^p \tag{2.14}$$

$$D_{Chebychev} = max_j \mid \{x_{sj} - y_{tj} \mid\} \tag{2.15}$$

$$D_{Manhattan} = \sum_{j=1}^{n} \mid x_{sj} - y_{tj} \mid \tag{2.16}$$

The process of training is based on different distance matrices in order to find an appropriate one. Next, k should be chosen to train the network. In a regression process, k-NN calculates the average value of its outputs. It should be noted that its performance depends on the resubstitution and the errors of test samples, for which mean square error is calculated [121, 122].

k-NN is a classifier that usually performs detection and classification of faults in TLs, and only a few papers have presented location estimation of faults [121, 123]. In Figure 2.5, the general idea of TL fault classification by using k-NN is shown. In this figure, there are 4 classes for labeled faults in training step (line-to-ground, line-to-line, double line-to-ground, and all-lines-connected (-to-ground)). To detect the class of a new fault, we measure its distances from these classes. d_i stands for the distance and k is the number of considered neighbors to classify the new fault based on their average (5 or 21).

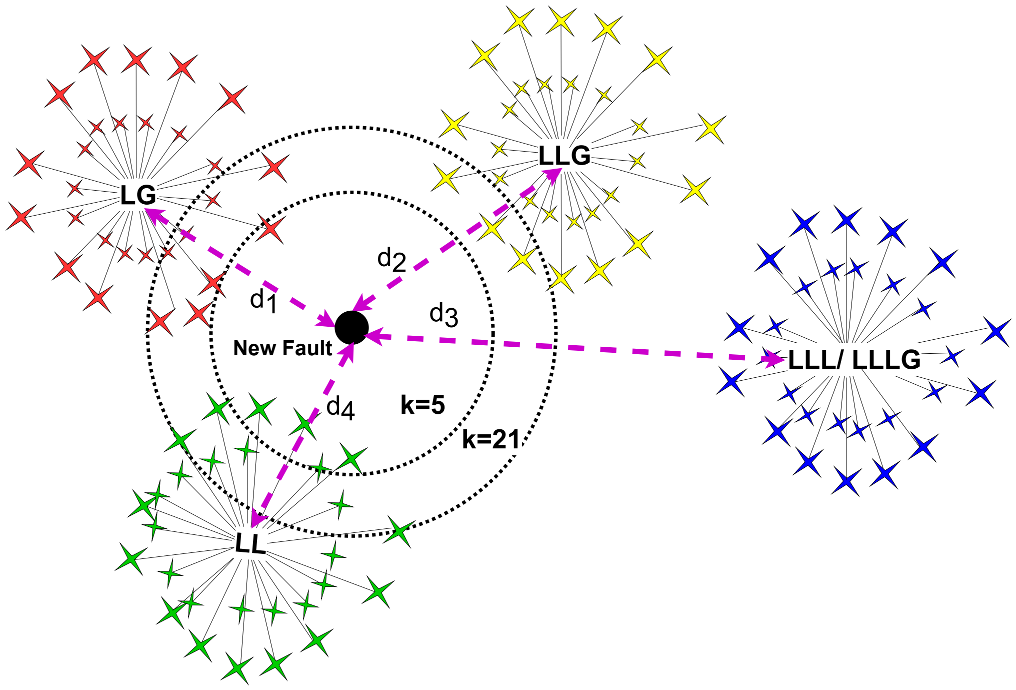


Figure 2.5: Schematic of transmission line fault classification using k-NN (d_i is the distance of the new fault from existing classes and k is the number of considered samples for classification).

Swetapadma et al. [121] take advantage of DFT to preprocess voltage and current signals to feed them as input data to k-NN with $k = 2$. Majd et al. [124] present an approach based on measuring the distance between each data point and its fifth nearest neighbor in a predefined window. These windows help them to detect the occurrence time of faults and the faulty phases. The maximum value of distances is compared with predefined threshold values to detect and classify faults. The study [125] proposes a scheme for fault classification based on fuzzy k-NN and continuous wavelet features. Two wavelet features are considered as the fundamental attributes: Wavelet Mean (μ) and Wavelet Standard Deviation (σ).

An approach for TL fault detection and classification is presented by Dasgupta et al. [126] using cross-correlation and k-NN. The proposed method takes advantage of synthetic fault data within half cycle of pre-fault and half cycle of post-fault to diagnose the types of faults in TLs in various situations and considering different parameter variations.

Singh et al. [127] propose an ML-based intelligent approach for classification of faults in a series compensated power network based on k-NN with various k values including 1, 3, and 5. DWT is applied to three-phase post-fault current signals for extracting the crucial features of the shunt fault. Moreover, k-NN is used for fault classification in a TL network. A semi-supervised ML methodology is proposed in [128] that co-trains DTC and k-NN to generate the classification model for labeled and unlabeled data to automate the fault classification process.

The study [129] presents a robust protection scheme for series capacitor compensated TLs using DWT and k-NN algorithm. All the protective relaying functions such as detection, classification, phase identification, and location estimation of faults are considered in this study. The signal processing and feature extraction are done using DWT which is able to distinguish between high and low frequency transient components. For fault detection and classification, only the approximation of wavelet coefficient for the current signals up to level 1 is used; while for k-NN location estimation, voltage and current signals of the circuits are decomposed up to level 3. Finally, the standard deviation of one cycle pre-fault and one cycle post-fault samples of the wavelet coefficient approximations are computed to form the feature vector for the k-NN-based algorithm.

Gashghaei et al. [130] present an integrated ML-based system architecture for the fault diagnosis of TLs in which various ML models of SVM and K-NN are used for fault. The K-NN fault type classifier is modeled as a dual-purpose module, which detects the fault type and acts as a redundant module for uncertainties from the startup unit. Gradients and standard deviations of DC current, voltage, harmonic current, and a correlation coefficient among the aerial and zero modes of DC current are the extracted features from single-end signal measurement.

Although k-NN is a common approach to solve TL fault identification and location estimation problems, it is mostly used in locating the faults in power systems. The mentioned studies based on this algorithm have an accuracy range of 97.61% to 100% for detecting TL fault types while most of them can achieve 100% accuracy. Signal processing methods used in the above-mentioned studies are DFT, DWT, and continuous wavelet transform (CWT). The sampling frequencies for k-NN-based methods are in the range of 1.2 (kHz) to 50 (kHz) which is smaller than those of DTC, RFC, and SVM. The main advantage of k-NN is that a large number of training samples do not affect its performance, unlike most ML methods.

2.3.6 ML Methods Comparisons and Discussions

This section summarizes the differences between the above-mentioned ML methodologies. Table 2.3 shows each study with its used techniques, input signals, immunity to noise, robustness analysis, the performed tasks with their average accuracy for classification task, and average relative error for their location estimation task.

Table 2.3: Comparison of Different Machine Learning Methods **(C)** Classification, **(L)** Location, **(I)** Current, **(V)** Voltage

ML Method	**Ref.**	**Used Techniques**	**Input Signal**	**Performance with Noise**	**Robustness Analysis for Generators**	**Task (C&L)**	**Avg. ACC**	**Avg. Relative Error**
NBC	[81]	DFT + NBC	I	Not mentioned	Not Mentioned	C	100%	NA
	[82]	DWT + NBC	I	Immune	Not Mentioned	C	100%	NA
	[86]		I					
	[87]		I&V					
	[77]	QTA + NBC	I&V	Not mentioned	Not mentioned	C	95%	NA
DTC	[93]	Least Square +Adaline Alg. +DTC	I	Immune	Not mentioned	C & L	99.29%	±5%
	[94]	ST + TW + DTC	V	Not mentioned	Not mentioned	L	NA	0.31%
	[95]	FDOST + DTC	I	Immune	Not mentioned	C	100%	NA
	[96]	MODP + DTC	I&V	Not mentioned	Not mentioned	C	99.5%	NA
RFC	[103]	RFC	I	Immune	Not mentioned	C	100%	NA
	[104]	MM+RFC	I&V	Not mentioned	Not mentioned	C	93.54%	NA
	[102]	IMPE+WPT +RFC	I&V	Not mentioned	Not mentioned	C	96.42%	NA
	[105]	MLP+RFC	I&V	Immune	Not mentioned	C	96.49%	NA
SVM	[109]	FDOST+SVM	I	Immune	Not mentioned	C & L	99.53%	0.47%
	[110]		I	Not mentioned	Not mentioned		100%	0.81%
	[112]		I&V	Immune	Provided		100%	0%
	[111]	WPT+SVM	I	Not mentioned	Not mentioned	C & L	99.21%	0.21%
	[113]	SVM	I&V	Not mentioned	Not mentioned	C	93.25%	NA
	[114]		V			C	70%	NA
	[118]		I&V			C & L	100%	0.03%
	[115]	FFT+SVM	I&V	Not mentioned	Not mentioned	L	NA	0%
	[116]	EWT+SVM	V	Immune	Not mentioned	C	100%	NA
	[117]	DWT+SVM	I	Not mentioned	Provided	C	99%	NA
k-NN	[121]	DFT+k-NN	I&V	Not mentioned	Provided	L	NA	1%
	[124]	k-NN	I	Not mentioned	Not mentioned	C	98%	NA
	[125]		I&V	Immune			100%	
	[127]	DWT+k-NN	I	Not mentioned	Not mentioned	C	99.4%	NA
	[129]		I&V			L	NA	1%
	[128]	CWT+k-NN	I	Immune	Not mentioned	C	97.61%	NA

The first method is NBC that has a simple structure to be implemented and to solve TL fault classification problems, where a signal processing method such as DFT, DWT, QTA is needed to do data preprocessing and feature extraction task. Although NBC is a good approach for classification of the types of faults, none of the considered studies have done location estimation for TL faults which questions NBC ability to solve this problem.

The second approach in Table 2.3 is DT algorithm which can be combined with other approaches in fault diagnosis of TLs such as TW. The main drawback of DT-based methodologies is over-fitting, while DTCs are able to do both classification and location estimation tasks for faults. None of the studies using DTC have done

Table 2.4: Parameters of Under Study Transmission Lines

Parameter	**Description**
d ($\boldsymbol{km}$)	Fault Distance
ϕ ($^\circ$)	Fault Inception angle
R_f (Ω)	Fault Resistance
δ ($^\circ$)	Phase Difference
L_s (mH)	Source Inductance
ΔV_i (kV), $i = 1, 2$	Voltage Fluctuations

robustness analysis with respect to the parameters mentioned in Table 2.4 because DTC's ability to handle datasets with high number of features is limited.

The third approach in Table 2.3 is RFC. The studies using this scheme only solved the classification problem of TLs, and the robustness analysis with respect to the parameters mentioned in Table 2.4 are not considered in them. These studies used MM, IMPE, WPT, and MLP to do the feature extraction function before RFCs are applied.

SVM-based studies come after RFC-based approaches in Table 2.3. Various signal processing approaches can be applied besides SVM to do the feature extraction task such as FDOST, WPT, FFT, DWT, and EWT. The results show that SVM-based approaches have satisfactory results and are compatible with any type of input signals such as voltage, current, and both voltage and current signals. Some of the studies in this category show their reliability by providing robustness analysis or proven immunity to noise.

The last technique in Table 2.3 is k-NN which is used mostly for classification problems. Even the studies such as [129] and [121] used segmentation scheme to solve the location estimation problem of TLs. DFT, DWT, and CWT are used with k-NN to extract the significant features of input signals.

2.4 Robustness

A system is said to be robust if it has an acceptable performance under different parameter variations. In other words, not only should the system work under a specific set of parameters, but it also should work under a different set of parameters.

The larger the range of the parameter variations, the more robust performance the system will have. Robustness is a crucial property of any system because the parameters of the system do not always have their nominal values. For example, voltage in a transmission line can have $\pm 10\%$ variations and it can negatively affect the performance of the identification and estimation procedure.

2.5 Artificial Neural Networks (ANNs)

ANN is an Artificial Intelligence (AI) concept that emulates the structure of the human brain to analyze data and perform pattern recognition to make decisions and classifications. ANNs with more layers are categorized as deep learning methods. Deep learning is a branch of ML and is capable of solving unsupervised and supervised learning problems.

ANNs work based on parallel computing phenomenon and are powerful systems consisting of a large number of processors with numerous interconnections [131, 132]. Deep learning takes advantage of a hierarchical level of ANNs to accomplish the ML methodologies. ANNs have three primary applications in the real world: classification of data such as image matching, feature extraction, and pattern recognition; noise reduction such as recognizing the discordant patterns in data and generating noise-free outputs; and extrapolation which means predictions based on previous data history [23].

In this study, the main focus is the indigenous capability of ANNs for the multi-label classification of TL faults. There are various types of ANNs that follow common fundamentals such as including 3 or more layers with neurons and activation functions for rectifying the values. Beside their similar structures, every type of ANN has its own characteristics such as computational methods for updating the value of weights, the connection of neurons among layers, storing significant values, removing the unwanted data, etc. Among various types of ANNs, some of them are used in

TL fault detection, classification and localization such as feedforward neural network (FNN), convolutional neural network (CNN), adaptive neuro-fuzzy inference system (ANFIS), extreme learning machine (ELM), and probabilistic neural network (PNN).

2.5.1 Feedforward Neural Network (FNN)

FNN is one of the initial structures presented for ANNs. FNNs are multilayer and fully connected ANNs that consist of one input layer, one or more hidden layers, and one output layer. Figure 3.3 shows a 4-layer FNN with two hidden layers.

From the second layer of an FNN, each node takes the weighted sum of the neurons from the previous layer and feeds it to a nonlinear activation function in order to generate the input of the next layer. The equation for each node of the hidden or output layer is given as follows [133, 134]:

$$\begin{gathered} z = f(b + x.w) = f(b + \sum_{i=1}^{N}(x_i w_i)) \\ x \in R_{1\times n}, w \in R_{n\times 1}, z \in R_{1\times 1} \end{gathered} \tag{2.17}$$

In (3.1), z is the output of current node, f represents the activation function, x_i stands for the input from the previous layer node i with the weight w_i, and b is the bias to correct the range of sum value. FNN is a model to mostly solve supervised

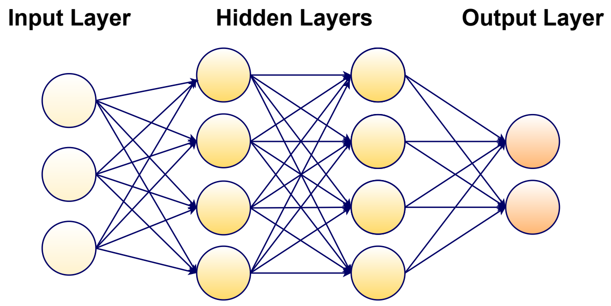


Figure 2.6: Structure of an example feedforward neural network with one input layer, two hidden layers, and one output layer.

learning problems. Having a portion of the dataset as labeled data points for the training step helps the system to learn how to classify the rest of the unlabeled data points. There are studies for TL fault detection, classification, and localization that take advantage of FNNs to achieve higher accuracy and performance.

Koley et al. [12] present a methodology using voltage and current waveforms of TLs that are fed to DFT and DWT for feature extraction. In this study, the fault detection and classification phase is done with a microcontroller classifier, and for the fault distance estimation a 4-layer FNN is implemented. A combined DWT and FNN approach [135] is presented for double circuit TLs based on the single-end data for classification and location estimation of TL faults. Zin et al. in [136] propose a scheme to detect and classify TL faults using discrete wavelet transform and an FNN based on Clarke's transformation. In this study, Clarke's transformation generates alpha and beta (mode) currents for conversion of the DWT signal to calculate the Wavelet Transform Coefficients and the Wavelet Energy Coefficient. Thwe et al. [137] use FNNs to do fault detection and classification for high voltage TLs to provide high-speed protection for digital power systems. Gowrishankar et al. [138] propose a technique utilizing DWT for decomposition of TL fault transients, and for applying an FNN to detect and classify them.

Yu et al. [139] propose an intelligent fault detection methodology for microgrids using wavelet transform and a specific FNN that works based on the gated recurrent unit. In this methodology, fast fault type recognition, phase detection, and location estimation for microgrids protection and service recovery are presented.

Koley et al. [140] present an algorithm including two steps. In the first step, an FNN-based algorithm is used to detect and classify possible types of shunt faults within one cycle, and in the second step, the locations of shunt faults are predicted. In this algorithm, they consider the voltage and current signals filtered by a low-pass filter to extract the fundamental components from them using DFT and feed them into an FNN. The work [141] presents a scheme based on an FNN for protecting TLs by detection and classification of one-conductor-open faults in parallel TLs.

Koley et al. [142] present a methodology using a hybrid wavelet transform and FNN model to detect, classify, and locate six-phase TL faults using single-end data. In this scheme, they gather the standard deviation of the approximate coefficients of voltage and current signals using DWT, and feed them as inputs to the modular FNN for fault identification and location estimation. Another study [143] presents a robust fault classification method by utilizing Wavelet Transform (WT) for feature extraction, and a 2-Tier (MLP) network. Jamil et al. [144] perform a study to detect and classify the TL faults using FNN. For this purpose, three-phase current and voltage signals of one end of a TL are taken as the inputs to FNN with a number of hidden layers for efficiency enhancement.

According to the aforementioned studies, FNN is the most common solution for TL faults classification and location estimation in recent years. More than one-fifth of all of the reviewed papers belong to this category, and some approaches are focusing on location estimation only. The accuracy range for these studies on TL fault classification is 90.60-100% while most of them have reached 100% with sampling frequencies in the range of 1 (kHz) to 1 (MHz). In some cases, signal processing methods such as DFT, DWT, continuous wavelet transform (CWT), and DOST are used with FNNs.

2.5.2 Convolutional Neural Network (CNN)

CNN is a subset of ANNs and is mainly used for analyzing imagery datasets and image classification problems. CNNs can handle high dimensional datasets faster, and with more details and minimum need for data preprocessing [145].

A CNN architecture consists of different layers including input layer to obtain data from the datasets; convolutional layers to create a feature map for feature class probability prediction (this step is done by applying a filter that slides over the whole data block); pooling layers for downsampling the data; fully connected layers to flatten the outputs from prior layers to generate a single vector (fully connected layers involves weights, biases, and neurons and by using feature analysis perform label predictions precisely); Softmax/Logistic layer which resides at the end of fully connected layers (Logistic is used for binary classification and Softmax is for multi-classification); and the connected output layer to produce the final probabilities for class determination. The architecture of a CNN is a vital factor to determine its performance and efficiency. The way that the layers are organized, the number of layers, the utilized elements in every layer, and their design affect the speed and accuracy of CNNs. Some of the well-known CNN architectures are LeNet-5 [146], AlexNet (2012) [147, 148], GoogleNet (2014) [149], and VGGNet (2014) [150]. Figure 3.5 shows the typical architecture of a CNN.

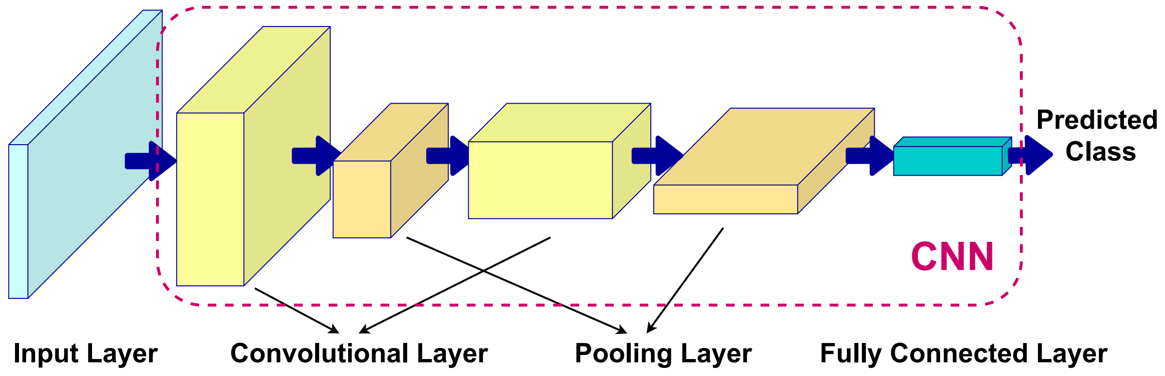


Figure 2.7: Typical architecture of a CNN.

There are numerous studies in TL fault detection, classification, and localization problems that use CNNs to achieve higher accuracies. These studies can be divided into two main categories: The first category includes those with focus on the image-based datasets taken from outdoor TLs [151, 152, 153], and the second category includes those that consider the generated time-series voltage and current signal waveforms from generators to be fed to the CNNs. In this survey, we focus on the second category.

In one study [154], a self attention CNN framework and a time series image-based feature extraction model is presented for fault detection and classification of TLs using a DWT for denoising the faulty voltage and current signals. Rai et al. [155] propose a customized CNN for fault detection and classification of TLs integrated with distributed generators. The work [156] presents an ML-based CNN for TL fault detection and classification that takes advantage of DWT for feature extraction.

Shukla et al. [157] present a CNN-based methodology which is able to distinguish between power swing (both symmetrical and asymmetrical) and faults besides their detection, classification, and location estimation. This ability removes the possibility of maloperation during the non-faulty stressed conditions to overcome the limitation of the protection model. Another study [158] proposes a scheme to detect and categorize the faults in power TLs using convolutional sparse auto-encoders. This approach has the ability to learn features extracted from the dataset of voltage and current signals, automatically, for fault detection and classification. To generate feature vectors, convolutional feature mapping and mean pooling methods are applied on multi-channel signal segments.

CNN-based approaches are usually better in the classification of TL faults, and they generate 99.58-100% accurate results within the range of $3.84(kHz)$ to $60(kHz)$ sampling frequencies. One of the advantages of CNN-based schemes is that they either need a simple and fast data preprocessing or do not need it at all, which increases their speed and performance. In CNN-based approaches, all studies focus on TL fault identification and classification problems and none of them considered the location estimation issue.

2.5.3 Adaptive Neuro-Fuzzy Inference System (ANFIS)

ANFIS is a type of ANN that utilizes a fuzzy inference system model to generate output from the input dataset. In this transformation of input to output data, membership functions, fuzzy logic operators, and $if - then$ rules are included [159, 160].

ANFIS is based on a special Sugeno model developed from the Fuzzy Inference System. There are five principal processing steps in ANFIS namely input fuzzification, fuzzy operators, application method, output aggregation, and defuzzification. ANFIS is a combination of both NN and fuzzy systems and so it benefits from their advantages such as simple implementation and high learning capability. The architecture of ANFIS includes five layers: fuzzy layer that consists of adaptive nodes with membership functions; product layer including node outputs to show the firing strength of a rule; a normalized layer consisting of nodes that represent the normalized firing strength of each rule; de-fuzzy layer that consists of adaptive nodes with node functions determining the contribution of the rules toward the general output; and the output layer that is one node to compute the sum of all rules [161, 162, 163]. Figure 2.8 shows a basic structure of ANFIS.

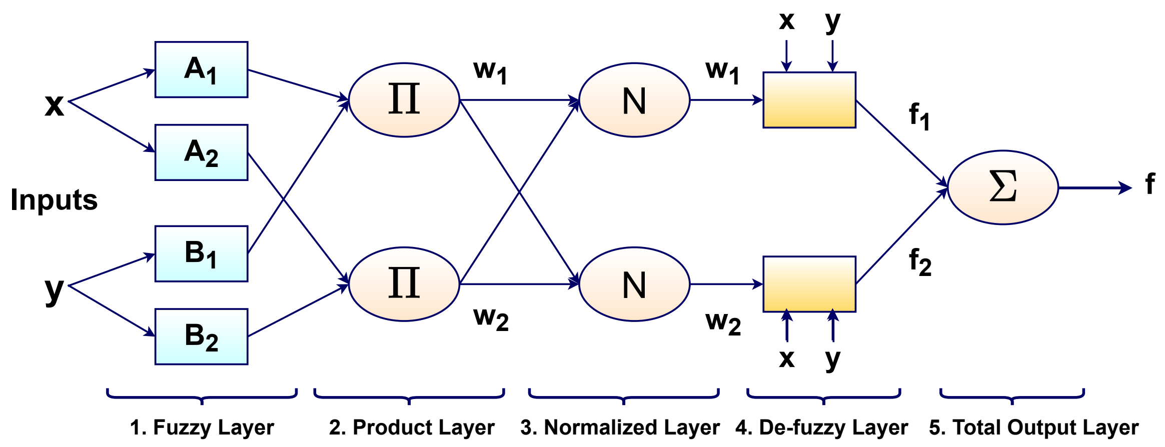


Figure 2.8: Typical ANFIS structure diagram.

Figure 2.8 represents the ANFIS architecture for two inputs x and y, and a single output f in a first order Sugeno fuzzy model. The common rule set with two $if - then$ fuzzy rules is explained below [164, 161].

Rule I: **if** x is A_1 and y is B_1, **then**:

$$f_1 = p_1 x + q_1 y + r_1 \tag{2.18}$$

Rule II: **if** x is A_2 and y is B_2, **then**:

$$f_2 = p_2 x + q_2 y + r_2 \tag{2.19}$$

In (2.18), x and y are the inputs, A_i and B_i are the fuzzy sets, f_is are the outputs of the fuzzy region determined by the fuzzy rule, and p_i, q_i, and r_i are the design parameters that are calculated in the training step [165].

To analyze and improve the performance of the built model, different values for significant models' parameters usually are tested experimentally. For every model, the best-result output with the minimum estimation error is detected based on the coefficient of determination (R2), Root Mean Square Error (RMSE), and Mean Bias Error (MBE) as follows [166]:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{\sum_{i=1}^{n}(y_i - \bar{y}_i)}(0 \leq \mathrm{R}^2 \leq 1) \tag{2.20}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i}{n}}(0 \leq \mathrm{RMSE} \leq \infty) \tag{2.21}$$

$$MBE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)(0 \leq \mathrm{MBE} \leq \infty) \tag{2.22}$$

where n is the number of data points, y_i is the predicted value, and $\hat{y}_i$ is the average value of observed data points. The best fit can be chosen in the case that $RMSE$ values are closer to 0 and R^2 values are closer to 1. The MBE value shows the negative and positive calculation error, determining the similarity of the predicted values with actual observed values [166].

Some studies are performed in recent years on the TL fault detection, classification, and location estimation based on ANFIS. The most interesting ideas are discussed in this subsection briefly.

Jarrahi et al. [162] propose another methodology to classify the faults generated by a double-circuit TL using ANFIS considering three-phase current samples of only one terminal. This method does not have any dependency on the impacts of fault inception and load angle, location, and resistance.

The study [167] proposes an application of ANFIS for fault classification and location estimation of TLs based on measured data from one terminal. The fundamental values of the voltage and current measurements are given to DFT to produce the dataset.

Another study [168] presents a robust fault detection, classification, and location estimation approach for 132 (kV), 100 (km) TLs. This approach integrates the feature extraction capability of the DWT with the intelligent classification capability of ANFIS. The RMS values generated by five detail levels of the DWT decomposition for the current signals are utilized to train the designed ANFIS models.

Khaleghi et al. [169] propose an approach to estimate the location of a single line-to-ground fault in four-circuit TLs. They use wavelet analysis for feature extraction and record the dynamic attributes of a fault signal by taking sampled current data from one side of the line. Also, ANFIS is used to find the relationship between the collected attributes from wavelet signal analysis and the variation of fault conditions, without knowing the type of faults.

The study [170] presents a three-phase distance relay model based on ANFIS to protect TLs, where they are exposed to the faults caused by the destruction of electrical equipment due to high amplitude electrical currents. In this study, ANFIS is designed in two parts. The first part is to detect the faults in a TL by measuring the voltage and current signals for every phase and compute the value of line impedance for fault detection and location estimation. The second part is to determine fault type and location by measuring the peak value of currents.

Veerasamy et al. [171] propose a scheme for detection and identification of high impedance fault (HIF) in medium-voltage 13.8 (kV) distribution networks by taking advantage of DWT and intelligence classifiers such as ANFIS and SVM. In this study, the obtained fault current signals for various types of three-phase faults such as line-to-line, line-to-ground, double line-to-ground, and HIF are sampled in 1st, 2nd, 3rd, 4th and 5th levels of coefficients. Then, they are approximated by DWT analysis for feature extraction which is called Standard Deviation (SD) values. These SD metrics that are generated by the DWT technique are used to train different ML classifiers such as Bayes, FNN, ANFIS, and SVM.

Although the ANFIS-based methods for solving TL fault classification, and location estimation problems have very high accuracies in the range of 99.4-100%, their implementations are complex and costly. Their sampling frequency range is 1 (kHz)-20 (kHz), and various signal processing approaches such as FFT, DFT, DOST, WT, and DWT are used besides ANFIS models. These methodologies can solve both identification and location estimation problems for TL faults.

2.5.4 Other Types of ANNs

There are two other types of ANNs that are used for TL fault detection, classification and localization. These two types are described below.

The first type is Extreme Learning Machine (ELM) which is FNN including one or more layers of hidden nodes whose parameters do not need to be tuned. These hidden nodes are assigned randomly and never get updated. They also can be inherited from their ancestors without adjustment. In ELM, the output weights of hidden nodes are updated in a single step to train a linear model [172]. Akmaz et al. [173] present a scheme to estimate the location of faults in TLs. In this study, they use traveling wave frequencies and an ELM to predict fault locations. Also an FFT is applied to transient signals in the time domain to transform them to the frequency domain and detect the traveling wave frequencies from their transient frequency spectrum. The work [174] proposes an integrated framework that combines fault classification and location estimation of TLs. In this study, the authors apply a summation-wavelet extreme (SWE) learning machine to add feature extraction to the learning process, and propose a Summation-Gaussian Extreme Learning Machine for the problem of TL fault diagnosis [174].

The second type is Probabilistic Neural Network (PNN) [175] which is a feed-forward NN. This category of ANNs is mostly known for its application in classification and pattern recognition problems. PNN is based on the parent probability distribution function of every class that is approximated by a Parzen window and a non-parametric function. By taking advantage of the mentioned function for each class, the class probability of a new input data is predicted and the Bayes' rule will be applied to assign the most probable class to the new input data. This architecture minimizes the probability of misclassification. The idea of PNN was initially acquired from the Bayesian network and a statistical algorithm (Kernel Fisher discriminant analysis) [176]. The structure of PNN includes four layers, namely, input layer, pattern layer, summation layer, and output layer [177, 178, 179, 180].

The study in [181] proposes the application of autonomous NNs for mapping the correlation between electrical signals at one terminal and TL fault information.

The presented approach provides NN models that can perform fault detection, classification, and location estimation with the ability to adapt automatically by taking advantage of probabilistic NNs. Roy et al. [182] present a scheme that utilizes PNN to diagnose the fault types and faulty phase of overhead TLs. Also, an approach for the computation of the fault locations is represented in this study and the authors use multi-resolution S-transform to generate complex S-matrices of the current signals measured at both ends of a TL. Mukherjee et al. [183] demonstrate an approach to detect and classify the type of a fault in three-phase overhead single-end-fed TLs. In this study, multivariate statistical methods are applied, such as Principal Component Analysis alone, and as an integrated method with PNN, to classify faults.

2.5.5 Comparisons and Discussions of ANN Methods

In this section, the differences between the above-mentioned ANNs are presented. Table 2.5 presents the details of all studies classified based on their types of NNs.

The first presented type of NN in Table 2.5 is FNN which is the most common type of NNs and is able to solve all sorts of TL-related problems. The signal processing schemes used besides FNN include WT, DWT, DFT, and CWT. The results of considered studies show that FNNs can perform very well in both classification and location estimation problems while the input signals can be only current signals or current and voltage signals together.

The second approach in Table 2.5 is CNN. The input signals in all studies based on CNN are both voltage and current. DWT can be used as a feature extraction technique before feeding the data to CNN. The accuracy for fault classification in the considered studies is almost 100% which shows the power of CNN in solving TL faults classification problem.

The third technique is ANFIS which is an efficient trade-offs between NNs and fuzzy systems working based on hybrid learning rules. Among the studies belonging to this category, none of them considered the effect of noise or robustness analysis of generators, while they show good performances in average relative error of fault location estimation or classification accuracy.

The fourth NN in Table 2.5 is ELM which uses current signals for classification or location estimation of TL faults. ELM-based studies have not considered the effect of noise or robustness analysis of TLs with respect to the parameters mentioned in Table 2.4.

The fifth method in Table 2.5 represents PNNs which is able to solve both classification and location estimation of TL faults. Most of the presented studies based on the PNN approach mentioned their robustness to noise and resulted in almost 100% average accuracy for classification of TL faults.

Table 2.5: Comparison of ANN Methodologies.**(C)** Classification, **(L)** Location, **(I)** Current, **(V)** Voltage

ANN Method	**Ref.**	**Used Techniques**	**Input Signal**	**Performance with Noise**	**Robustness Analysis for Generators**	**Task (C&L)**	**Avg. ACC**	**Avg. Relative Error**
FNN	[12]	WT+FNN	I&V	Not mentioned	Provided	C &L	100%	0.07%
	[135]		I&V		Not mentioned	C& L	99%	0.08%
	[137]		I		Not mentioned	C	100%	NA
	[136]	DWT+FNN	I&V	Not mentioned	Not mentioned	C	99%	NA
	[138]		I	Not mentioned	Not mentioned	C	90.60%	NA
	[139]		I	Immune	Not mentioned	C & L	99.31%	5.9%
	[142]		I&V	Immune	Provided	C & L	99.9%	0.68%
	[140]	DFT+FNN	I&V	Not mentioned	Provided	C & L	100%	0.73%
	[141]		I	Not mentioned	Not mentioned	C	99.8%	NA
	[143]	CWT+FNN	I&V	Immune	Not mentioned	C	99.52%	NA
	[144]	FNN	I&V,	Not mentioned	Not mentioned	C	100%	NA
CNN	[156]	DWT+CNN	I&V	Not mentioned	Not mentioned	C	100%	NA
	[158]			Immune			100%	
	[154]			Immune			99.58%	
	[155],	CNN	I&V	Not mentioned	Not mentioned	C	99.97%	NA
	[157]						99.72	
ANFIS	[162]	FFT+ANFIS	I&V	Not mentioned	Not mentioned	C	99.4%	NA
	[167]	DFT+ANFIS	I&V	Not mentioned	Not mentioned	C & L	100%	2.16%
	[168]	DWT+ANFIS	I&V	Not mentioned	Not mentioned	C & L	99.78%	0.22%
	[169]	WT+ANFIS	I	Not mentioned	Not mentioned	L	NA	0.61%
	[170]	ANFIS	I&V	Not mentioned	Not mentioned	C & L	100%	2.06%
	[171]	DWT+ANFIS	I	Not mentioned	Not mentioned	C	100%	NA
ELM	[173]	FFT+ELM	I	Not mentioned	Not mentioned	L	NA	5.31%
	[174]	SW+ELM	I	Not mentioned	Not mentioned	C & L	98 %	3.2%
PNN	[181],[183]	PNN	I&V	Not mentioned	Not mentioned	C & L	100%	6.5%
			I	Immune		C	100%	NA
	[182]	ST+PNN	I	Immune	Not mentioned	C & L	99.6%	4.35%

2.6 Hybrid Methods

This section addresses the hybrid schemes that take advantage of more than one ANNs or a combination of one of them with at least one generic ML method to achieve efficient and precise fault diagnosis results. The methodologies in this category usually have high performance because they take advantage of multiple approaches to overcome the weaknesses of the individual solutions. Table 2.6 presents these approaches.

The study [184] proposes a hybrid methodology for detection, classification and location estimation of short-circuit faults in power TLs. In this study, the authors take advantage of a hybrid framework including a two-stage finite impulse response filter, four SVMs, and eleven support vector regressions (SVRs) that are implemented in Proteus 6 and Matlab environments. In this approach, a two-stage finite impulse response filter integrated with SVMs is used for detection and classification of the faults, and the SVRs are implemented to estimate the location of them. Another study [185] presents an identification and categorization methodology for TL fault types and location estimation of the faults. In this study, to increase the sensitivity to faults, the frequency response curves related to the implemented faults in the TLs are calculated and used. Then, a deep learning technique is developed to design a hybrid model of the CNN and LSTM called C-LSTM for interpretation of the frequency response curves related to faults and determine the type of faults, and estimate their locations.

Table 2.6: Comparison of Hybrid Methods**(C)** Classification, **(L)** Location

Ref.	**Used Techniques**	**Input Signal**	**Performance with Noise**	**Robustness Analysis for Generators**	**Task (C&L)**	**Avg. ACC**	**Avg. Relative Error**
[184]	FIR + SVM + SVR	I&V	Not mentioned	Not mentioned	C& L	100%	4%
[185]	CNN + LSTM	I&V	Not mentioned	Not mentioned	C & L	98%	0.27%
[186]	LDA + RFC +FNN	I&V	Not mentioned	Not mentioned	C	98%	NA
[187]	DWT + ANFIS + FNN	I&V	Not mentioned	Not mentioned	C & L	100%	0.11%
[188]	SVM + GNN	I&V	Not mentioned	Not mentioned	L	NA	0.17%

Balakrishnan et al. [186] present a hybrid methodology for detection and classification of TL faults based on RF algorithm. They use linear discriminant analysis to extract the data from voltage and current signals and detect the fault types using an RF algorithm that is based on the cuttlefish optimizer. Gayathri and Kumarappan [188] present a hybrid approach for double circuit TLs subject to obstacles such as the mutual coupling between the two circuits under fault conditions. The study [187] proposes a numerical relaying method that is based on DWT, ANN, and ANFIS. The authors design a NN for fault detection and classification that is trained by samples of voltage and current signals generated by a simulated test system. They take advantage of ANFIS trained by the extracted summation values of discrete wavelet coefficients of a three-phases TL to estimate the location of TL faults.

In this methodology, the fault location of extra high voltage TLs is estimated by using a radial basis function-based SVM applied to the reconstructed input data points, and a gradient based NN (GNN) using scaled conjugate gradient.

Table 2.7: Comparison of Different ML Methods **(C)** Number of Papers Considering Identification of Faults Including Detection and Classification, **(L)**: Number of Papers Solving Location Estimation Problem, and **(C&L)** Number of Papers Considering Both Problems

Method	**Task**	**Complexity**	**Advantages**	**Disadvantages**
NBC	5 (C)	Simple$O(np)$	• Easy implementation • High dimensional datasets support • Accurate results for independent features	• No robustness to noise • Low performance for datasets with dependent attributes • Unsupportive for unobserved datapoints in training step • High computational complexity
DT	2 (C) 1 (L) 1 (C&L)	Simple$O(n^2p)$	• Easy rule setup • Automatic attribute selection • Simple implementation and understanding • Compatibility with other decision methods • Fast preprocessing procedure and no need for normalization	• Risk of over-fitting • Computationally expensive • Unreliable results for imbalanced datasets • Complex calculation in case of high uncertainty • Generating biased results for repetitive features
RF	4 (C)	High$O(n^2pn_t)$	• Capable of handling imbalanced datasets • Handling high dimensional and large datasets • Supportive for unobserved datapoints in training step	• Overfitting possibility • Computationally expensive • Time consuming training step • Cannot be effective for real-time problems • Users have little control on the model performance
SVM	4 (L) 1 (L) 5 (C&L)	Complex$O(n^2p + n^3)$	• Low probability of misclassification • Working well in case of clear margins • Accurate classification and regression results • Efficient for datasets with more dimensions than samples	• Unable to do estimation tasks directly • Weak performance for large or noisy dataset • Computationally expensive for classification problems
k-NN	3 (C) 2 (L)	Simple$O(np)$	• Intuitive and simple • No training phase • Easy implementation • Considering various distance criteria for selection	• Time consuming • Sensitive to outliers • Requiring homogeneous features • No robustness to noise or imbalanced classes • Unable to handle datasets with missing cases • Inaccurate prediction for high dimensional datasets
FNN	9 (C) 1 (L) 6 (C&L)	Complex	• Easy implementation • Robust to noise • Easy usage by adjustment of a few parameters • Having various applications in real-life problems • Well performance for data classification problems • Ability to learn by itself and no need for reprogramming	• Convergence is dependent on initial values • Not following the parameter sharing concept • Possibility of getting trapped in local optima • Slow convergence in the backpropagation algorithm • Complex training process for high dimensional datasets

Table 2.8: Comparison of Different ML Methods **(C)** Number of Papers Considering Identification of Faults Including Detection and Classification, **(L)**: Number of Papers Solving Location Estimation Problem, and **(C&L)** Number of Papers Considering Both Problems

Method	**Task**	**Complexity**	**Advantages**	**Disadvantages**
CNN	6 (C) 1 (L)	Complex	• Robust to noise • Data preprocessing is not mandatory • Following the parameter sharing concept • Ability to learn convolutional filters automatically	• High computational cost • Parameter tuning is essential • Slow training for complex datasets • Requiring high amount of training data
ANFIS	4 (C) 4 (C&L)	Medium	• Reduced search space dimension • Tuning parameters by the hybrid learning rules • Good trade-off between neural and fuzzy systems • Smooth and adaptable to a wide range of applications	• No robustness to noise • Sensitive to outliers • Computationally expensive • Needing experts for membership determination function and fuzzy rules, for large amount of training data
ELM	1 (L) 1 (C&L)	Complex	• Support noisy datasets • Easy implementation • Very short training time • High prediction accuracy • Higher performance speed comparing to FNN	• Limited applications • Overfitting or underfitting possibility
PNN	1 (C) 2 (C&L)	Complex	• Fast learning time • Robust to noise • No need for learning process • Insensitive to outliers • Automatic weights initialization • Faster and more accurate than MLP networks • Guaranteed convergence with Bayesian classifier	• Time consuming for large networks • Slower than MLP in case of complex datasets • More memory space usage for storing the model • Hard to find an optimum architecture configuration
Hybrid	2 (C) 2 (L) 2 (C&L)	Complex	• Ability to solve any kind of problem • Flexibility for high dimensional datasets • High performance for classification problems • Performing feature extraction via unsupervised procedures	• No robustness to noise • High computational load • Inefficient for noisy datasets • Slow training and testing processes • Sensitive to feature selection methods

2.7 Comparisons and Discussions

In this section, the differences between the above-mentioned methodologies are presented. is a key concept in ML and deep learning approaches, and is divided into two steps: feature extraction and feature selection. In feature extraction, all the features of the dataset understudy will be extracted, and in feature selection, the significant ones will be selected. Although deep learning methods perform these two steps as a combined step, ML approaches have an order for them, first feature extraction, and then classification.

The detailed description of each methodology is already explained in the previous sections, and Table 2.7 and Table 2.8 compare the advantages and disadvantages of them. The second column of these Tables show the number of papers that are reviewed for each method in this study. “C” stands for the number of papers that considered the identification of faults including detection and classification, ”L” stands for the papers solving the location estimation problem, and “C&L” shows the number of papers that considered both problems. The third column shows the complexity of the training step for each method based on its requirements such as the number of training samples, number of features, number of trees (for RF), and number of layers for deep learning approaches. Usually, the testing step complexity is considerably lower than the training step, and so the reported complexity belongs to the training step only [56].

The first method is NBC that has a simple structure to be implemented and is able to classify independent high dimensional data accurately. To solve TL fault classification and location estimation problems, DWT is needed to do data preprocessing for the Bayesian classifiers. The second line of this table belongs to the DT algorithm with its easy interpretation and understandable structure while not being in need of data normalization. The main drawback of DT-based methodologies is over-fitting. Besides, if the number of faults in TLs is too small and the dataset is considered imbalanced, DT cannot be an effective classifier.

By using RF, scientists have tried to compensate the drawbacks of DT. The RF-based studies in this review provide several advantages such as handling imbalanced datasets and supporting datasets with missing cases. Although RF is proposed to overcome the over-fitting problem of DT by using extra computational power and resources, it fails to guarantee a solution to those problems.

k-NN does not need any training step and is mostly used for fault type classification and range-based estimation of fault location. This algorithm is slow, unable to deal with missing data, and very sensitive to outliers.

SVMs are complex, accurate, and widely used algorithms for TL fault detection, classification, and location estimation. Due to SVM's high computational load, a large memory is required that makes this methodology costly. Also, this approach cannot work effectively with noisy datasets, which undermines the reported accuracy for TL fault classification using SVM.

FNN is the most common type of NNs that is able to solve all sorts of TL-related problems. This method is easy to implement and can precisely detect the exact type of faults even in the presence of noise. The only concern about FNNs in the TL fault identification process is its time-consuming training step, and the need for data preprocessing methods such as DFT and DWT that makes the whole process complex.

The outstanding feature of CNN is that it does not need any data preprocessing approach, and so raw data can be given to the system directly. CNN is a powerful technique that supports noisy data and learns the functionality of filters automatically without mentioning explicitly. Depending on the number of layers, this approach can be computationally expensive and costly.

ANFIS is one of the most efficient trade-offs between NNs and fuzzy systems that works based on hybrid learning rules and provides a solution with smoothness and adaptability. On the other hand, ANFIS is not a robust method and is unreliable for noisy datasets. It is also computationally complex, expensive and very sensitive to outliers.

ELMs are used in some recent studies because of their fast training, high accuracy against noisy datasets, and easy implementation. However, ELMs suffer from the over/under-fitting problems due to their structure and do not apply to all sorts of applications.

Another common type of NNs is PNN which does not need the learning process and is much faster and more accurate than MLP networks. Although the PNN structure is insensitive to outliers, it is really hard to be implemented and needs a lot of memory space. This structure cannot classify new cases as fast as MLP networks.

The last line of Table 2.8 belongs to some hybrid ML methods that perform feature extraction via an unsupervised procedure. These methods have better performances and are immune to noisy datasets. They usually have complicated architectures that are not efficient in cost, memory, and time.

Considering the advantages and disadvantages of all the above-mentioned classifiers, it can be concluded that the deep learning-based methods are superior solutions to TL fault detection, classification, and location estimation problems. Moreover, the efficiency of the proposed techniques in terms of cost and memory requirement has not yet been studied [117].

In this review chapter, more than 150 papers are included among which only a few studied the robustness of their proposed approaches. There is only one paper [112] from 2015 that proposes an SVM-based method for detection, classification, and location estimation of TL faults considering the robustness analysis of TLs with respect to the parameters mentioned in Table 2.4. This technique suffers from sensitivity to noise due to the nature of SVM.

2.8 Summary

In this chapter, a comprehensive review of ML methods for TL fault detection, classification, and location estimation is discussed. The used methodologies are divided into three main categories, namely, generic ML, ANNs, and hybrid methods. Generic ML approaches are classified into five subsections, namely, Naive Bayes Classifier, Decision Tree, Random Forest, Support Vector Machine, and k-Nearest Neighbour. Artificial Neural Networks are divided into four main categories based on their various structures, namely, Feed forward Neural Network, Convolutional Neural Network, Adaptive Neuro Fuzzy Inference System, and other small groups such as Extreme Learning Machine and Probabilistic Neural Network, and the third category is assigned to hybrid methods. The original idea, fundamental equations, and relevant publications since 2015 are included and summarized for each method. Last but not least, the advantages and disadvantages of the ML approaches is presented in details and summarized.

CHAPTER 3

ROBUSTNESS ANALYSIS OF NEURAL NETWORK-BASED FAULT DIAGNOSIS TECHNIQUES FOR TRANSMISSION LINES

3.1 Introduction

Protection of high voltage transmission lines is one of the most crucial problems in the power system industry. Accurate and timely detection, identification, and location estimation of line-to-ground, line-to-line, line-to-line-to-ground, and line-to-line-to-line faults can considerably improve and simplify the recovery process of transmission lines and hence save the costs associated with the downtime of a power system. Therefore, it is essential that a robust, affordable, and accurate fault diagnosis system completes its operation within an acceptable time window after a fault occurs in the presence of system uncertainties and disturbances.

The significant cost of mistakenly detected or undetected faults in the conventional techniques motivated us to present a robust detection, identification, and location estimation system by using two different neural networks, namely generalized regression and convolutional neural networks. The robustness of these techniques are analyzed with respect to the variations of fault resistance, phase difference between three connected buses, fault inception angle, local bus voltage fluctuations, source inductance fluctuations, and measurement noise. The time delay analysis is also conducted to show that the proposed techniques can detect, identify, and estimate the location of faults before tripping relays and circuit breakers disconnect a faulty region.

Due to the drawbacks of TW techniques, several studies are conducted on fault detection, identification, and location estimation using artificial intelligence methods such as NNs, SVMs, Neuro-fuzzy networks, etc., and signal processing methods such as FFT, discrete wavelet transform (DWT), etc. In the study [189], authors use DWT for extracting the features and SVM for identifying faulty sections which is a computationally expensive approach. In [190], authors employ a decision tree regression-based method together with FFT and DWT for fault distance estimation which is a noise sensitive methodology.

There are several crucial situations that may cause a protection system to have a poor performance, namely, high resistance faults, power fluctuations, heavy load switching, measurement noises, fault inception time, fault location, phase difference between sources, etc. They may affect the voltage and current signals after the occurrence of faults, leading to malfunctioning of the fault detection, identification, and location estimation system. Therefore, a reliable, accurate, and fast fault detection and location estimation technique should be given a high precedence [191]. Although the reviewed techniques from literature have acceptable performances in fault identification and distance estimation under different conditions such as fault inception time, fault resistance, load variations, etc., only a few of them has investigated the robustness of the detection and location estimation performance to the variations of amplitude voltages of the generators and the phase difference between neighboring generators [12, 121, 112, 192, 140, 142, 117].

In this study, we focus on this robustness importance and the effects of each above-mentioned parameter on the final accuracy of detection and localization method to generate realistic results and a comprehensive examination on TL faults.

To the best of our knowledge, the GRNN method has not been applied to the fault detection of TLs. The main goal of this study is to design a fault detection, identification, and location estimation system based on GRNN, and assess its performance in terms of fault identification accuracy and location estimation error in high voltage TLs. This system utilizes the voltage and current waveforms recorded from one end of a two-bus TL. Moreover, FFT is applied to both current and voltage waveforms to extract the amplitudes of their fundamental frequency components and then feed them to the fault identification and location estimation systems. The changes to the operating points of a TL can cause considerable effects on the accuracy of the fault detection and location estimation GRNNs. As a result, the proposed technique is assessed in terms of robustness to changes of operating points that include voltage amplitude and phase difference fluctuations in generators. To this end, not only the network is trained for different types of faults, fault locations, and fault inception times, but also it is trained for different operating points to maintain an acceptable robust performance in terms of identification accuracy and distance estimation error. Moreover, time delay analysis is carried out in this study to show that the cumulative detection, identification, and location estimation times are within the timing standard of the IEEE report [193].

There are several crucial situations that may cause a protection system to have a poor performance, namely, high resistance faults, power fluctuations, measurement noises, fault inception time, fault location, phase difference between sources, etc. They may affect the voltage and current signals after the occurrence of faults, leading to dysfunction of the detection and location estimation system. Therefore, a reliable, accurate, and fast fault detection and location estimation technique should be given a high precedence [191].

The main goal of this work is to present a fault detection, identification, and location estimation procedure based on the two NNs, namely GRNN and CNN, and assess their performances in terms of fault identification accuracy and distance estimation error in high voltage transmission lines. The fault diagnosis system utilizes the voltage and current waveforms recorded from one end of a two-bus transmission line. Moreover, FFT is applied to both current and voltage waveforms to extract the amplitudes of their fundamental frequency components and then feed them to the fault identification and location estimation system. FFT is chosen in this study because it does not need high sampling frequency which is an advantage as compared to the methods using other TW techniques such as DWT [173].

The proposed GRNN and CNN techniques are assessed in terms of robustness against the possible variations of different factors such as fault types, fault distances, fault resistances, fault inception angles, source inductance, as well as system operating points including voltage amplitude and phase difference between generators. Moreover, time delay analysis is carried out in this study to show that the detection, identification, and location estimation times are within the IEEE timing standard[193]. It should be noted that the effect of noise on fault identification and location estimation is also investigated. The performances of the two NNs are compared.

The contribution of this work are as follows:

1. Designing GRNN and CNN for the fault detection, identification, and location estimation problems in a transmission line based on the time series signals after applying FFT which results in a highly robust performance;
2. Conducting robustness analysis against variations of fault resistance, fault inception angles, source inductance, phase difference between two connected buses, bus voltage fluctuations, and measurement noise for the two NNs;
3. Performing time delay analysis (cumulative detection, identification, and location estimation delays) based on the IEEE timing standard[193] for the two NNs.

The chapter is organized as follows. Section 3.2 presents a TL model and its waveform measurements. Section 3.3 presents the two NNs used in this study, generation of features, and fault detection/identification and location estimation procedures. In Section 3.4, the simulation results are presented, and the performances of both NNs are assessed in terms of cumulative detection/identification and location estimation time delays, fault identification accuracy, fault location estimation error, and robustness to variations of the system parameters and measurement noise. Finally, the conclusion of our work is made in Section 3.5.

3.2 Transmission Line Model

The system used for this study is modelled as two three-phase generators connected by a 120 (km) TL with a voltage rating of 240 (kV) and frequency of 60 (Hz) as depicted in Figure 3.1. The model is simulated in MATLAB Simulink's Simscape Power System, and all the eleven fault scenarios are considered including no-fault, Line to Ground (LG), Line to Line (LL), Line to Line to Ground (LLG), and Line to Line to Line (LLL). The TL model parameters are shown in Table 3.1, and the values of the two generators and loads are given in Table 3.2. It should be noted that the transmission line model considered in this study is based on the *IEEE 39*-Bus system which has 10 generators and 46 lines. In this work, only two of the generators with a three phase transmission line in between are considered.

Table 3.1: TL Nominal Parameters

Parameter	Zero Sequence	Positive Sequence
R (Ω/km)	0.3864	0.01273
L (mH/km)	4.1264	0.9337
C (μF$/km$)	7.751×10^{-3}	12.74×10^{-3}

Figs. 3.2 indicates faulty voltage and current waveforms of one-end of the TL for an LL (phase A to phase B) fault. As observed, with the occurrence of the fault, the voltage amplitude of phases A and B decrease while the voltage amplitude of phase C remains unchanged. The current waveforms behave differently. Based on Figure 3.2(b), with the occurrence of the fault, the current amplitudes of phases A and B increase significantly, while it remains unchanged for phase C.

Table 3.2: Source and Load Nominal Parameters

Nominal Parameter	Source 1	Source 2	Load
Phase to Phase Voltage (kV)	240	240	240
Frequency (Hz)	60	60	60
Resistance (Ω)	0.08929	0.08929	—
Inductance (mH)	16.58	16.58	—
Active Power (kW)	—	—	100
Inductive Reactive Power ($kVAR$)	—	—	< 100
Capacitive Reactive Power ($kVAR$)	—	—	< 100

3.3 Proposed NN Techniques and Comparison

In this section, a brief introduction to the two NNs used in this study is given, and their main characteristics are compared with each other.

3.3.1 Feedforward Neural Network (FNN)

FNNs are multilayer and fully connected NNs that consist of several layers, i.e., input layer, one or more hidden layers, and output layer as shown in Figure 3.3. Each node in the second layer of FNN receives the weighted sum of the nodes from the previous layer which are then fed into a nonlinear activation function. Each node in the hidden or output layer follows [133, 134]:

$$
\begin{gathered}
z = f(b + x.w) = f(b + \sum_{i=1}^{N}(x_i w_i)) \\
x \in R_{1\times n}, w \in R_{n\times 1}, z \in R_{1\times 1}
\end{gathered}
\tag{3.1}
$$

where z is the current node output, f is the activation function (which is the Sigmoid function in this study), x_i is the input i from the previous layer which has the weight w_i, and b is the bias used to correct the range of the sum of the values.

Several studies in TL fault detection, identification, and localization make use of FNNs to achieve better performance in terms of identification accuracy and location estimation error. In this study, two types of structures are considered in designing the FNNs for identification and location estimation. These structures are obtained based on several examinations in train and test errors. A five-layer 7-30-20-20-4 configuration is used for the identification stage in which the numbers represent the number of nodes in input layer, three hidden layers, and output layer, respectively. The FNN configuration for the location estimation stage is 7-10-10-10-1. It should be noted that the optimization method to update the weights in both structures is considered to be Bayesian regularization backpropagation. This function updates the weight and bias values using Levenberg-Marquardt optimization. The learning rate for both FNNs is 0.6. The architectures for the identification and location estimation phases give the Mean Squared Error (MSE) of 0.0012 and 0.086, respectively, for the test data.

3.3.2 Generalized Regression Neural Network (GRNN)

The GRNN belongs to the family of radial basis function (RBF) NNs. It works on the basis of sampled data, makes use of the Parzen non-parametric estimation, and gives output based on the maximum probability principle. GRNN training process is more convenient as compared to the RBF NNs and has more advantages in the approximation ability and learning speed. This network converges to the optimal regression surface where most of the samples are accumulated. The regression nature of GRNN is used to eliminate the adverse effect of the source inductance value because it provides faster and better performance than a one hidden-layer feed-forward neural networks. Therefore, it can solve the problems in many different areas such as TL fault diagnosis [194].

A GRNN consists of four layers which are the input layer, the pattern layer, the summation layer and the output layer as shown in Figure 3.4. Contrary to the backpropagation based NN, the GRNN does not need an iterative learning process. The main goal of the learning process in GRNN is to find the best smoothing parameter which is also called the spread of the exponential functions used in the pattern layer.

A brief explanation on this NN is given as follows. Let $f(X, Y)$ be the joint density of the random variables X and Y. Given the observed values x of a vector random variable X and the regression of a scalar random variable Y, the condition mean of Y on a given x is calculated by

$$\hat{Y}(x) = E[Y|x] = \frac{\int_{-\infty}^{\infty} y f(x,y) dy}{\int_{-\infty}^{\infty} f(x,y) dy} \tag{3.2}$$

where $\hat{Y}(x)$ is the predictive output of Y. The function $f(x, y)$ is unknown in general and should be estimated based on a set of observations of X and Y using the non-parametric consistent estimator suggested by Parzen [195] and is calculated as follows:

$$\hat{f}(X,Y) = \frac{1}{n(2\pi)^{\frac{(p+1)}{2}} \sigma^{(p+1)}} \sum_{i=1}^{n} e^{-\frac{(X-X_i)^T (X-X_i)}{2\sigma^2}} e^{-\frac{(Y-Y_i)^2}{2\sigma^2}} \tag{3.3}$$

where n is the number of sample observations, p is the dimension of the vector variable X, and σ is the smoothing parameter, which is actually the standard deviation of the Gaussian function. The PDF estimate $\hat{f}(X,Y)$ in (3.3) is substituted into the conditional mean in (3.2), which results in the desired conditional mean of Y given x. Interchanging the order of integration and summation yields the desired conditional mean, $\hat{Y}(x)$, as follows:

$$\hat{Y}(x) = \frac{\sum_{i=1}^{n} e^{-\frac{(x-x_i)^T (x-x_i)}{2\sigma^2}} \int_{-\infty}^{\infty} y e^{-\frac{(y-y_i)^2}{2\sigma^2}} dy}{\sum_{i=1}^{n} e^{-\frac{(x-x_i)^T (x-x_i)}{2\sigma^2}} \int_{-\infty}^{\infty} e^{-\frac{(y-y_i)^2}{2\sigma^2}} dy} \tag{3.4}$$

The two integrals in the numerator and denominator of the above equation are calculated analytically as $\sqrt{2\pi} y_i \sigma$ and $\sqrt{2\pi}\sigma$, respectively.

For simplicity, the scalar function D_i^2 is defined as follows:

$$D_i^2 = (x - x_i)^T (x - x_i) \tag{3.5}$$

Based on the integral values calculated above, and the definition of D_i^2 given in (3.5), Equation (3.4) becomes:

$$\hat{Y}(x) = \frac{\sum_{i=1}^{n} y_i e^{-\frac{D_i^2}{2\sigma^2}}}{\sum_{i=1}^{n} e^{-\frac{D_i^2}{2\sigma^2}}} \tag{3.6}$$

The estimate $\hat{Y}(x)$ in (3.6) can be interpreted as a weighted average of all of the observations y_i $(i = 1, ..., n)$, where each observed value y_i is weighted exponentially based on the Euclidean distance from x_i to x. The smoothing parameter σ plays an important role in the estimation process so that when it is made larger, the estimated PDF $\hat{f}(X, Y)$ becomes smoother. In the limit, $\hat{f}(X, Y)$ becomes a multivariate Gaussian with covariance σ^2 I, where I is an identity matrix, so that $\hat{Y}(x)$ is estimated to be the sample mean of the observed values $y_i(i = 1, ..., n)$. On the other hand, when σ takes smaller values, $\hat{f}(X, Y)$ becomes closer to a non-Gaussian PDF, so that $\hat{Y}(x)$ is estimated to be the value of y_i corresponding to the observation x_i closest to x. When σ takes an intermediate value (between the two extremes), $\hat{Y}(x)$ is estimated to be a weighted average of all values of y_i, while heavier weights are given to those values of y_i with x_i closer to x. It should be noted that after examining several values of σ based on the MSE measure, the optimal values for detection and location estimation were chosen to be 0.2 and 0.8, respectively. It should also be noted that the MSEs for the identification and location estimation are 0.00038 and 0.019, respectively.

3.3.3 Convolutional Neural Network (CNN)

CNN as a subset of deep learning methods is mainly used for analyzing imagery datasets and image classification problems. The advantage of CNNs is that they can handle high dimensional datasets with higher speed, and are more efficient with minimum requirement for data preprocessing [145].

A CNN architecture consists of different layers including input layer to obtain data from the datasets; convolutional layers to create a feature map for feature class probability prediction (this step is done by applying a filter that slides over the whole data block); pooling layers for downsampling the data; fully connected layers to flatten the outputs from prior layers to generate a single vector; fully connected layers which involve weights, biases, and neurons to perform label predictions precisely by using feature analysis; Softmax/Logistic layer which resides at the end of fully connected layers (Logistic is used for binary classification and Softmax is for multi-classification); and the connected output layer to produce the final probabilities for class determination. The architecture of a CNN is a vital factor to determine its performance and efficiency. The way that the layers are organized, the number of layers, the utilized elements in every layer, and their design affect the speed and accuracy of CNNs.

Studies on TL fault detection, identification, and location estimation problems that use CNNs are classified into two main categories. The first category includes those with a focus on the image-based datasets taken from TLs above the ground [151, 152, 153]. The second category includes those that consider the generated time-series voltage and current signal waveforms as blocks from generators to be fed to the CNN.

In this study, a CNN architecture is designed based on LeNet5 [196] and the existing time series dataset is considered as images generated from numerous blocks. This is done because of the important features of TL datasets and the ability of generating precise outcomes. The utilized CNN consists of two convolution layers including 64 and 128 neurons and kernel sizes of 3×7 and 5×7, respectively, and uses "relu" activation function followed by a max pooling layer. Then a dropout layer with 0.25 value is added and its output is flattened by a flat layer. Moving towards the output layer, two dense layers with 512 and 256 neurons, respectively, and another dropout with the value of 0.1 exist. The output layer is a dense layer with 16 neurons. This architecture generates 100% accuracy for TL fault identification and the average of MSE for all types of faults is 0.006. Figure 3.5 shows the general architecture for CNNs.

In this study, two different NNs are implemented and their robust performance in terms of fault detection, identification, and location estimation are analyzed. The first architecture is GRNN which is based on non-parametric regression and single-pass learning. GRNNs can generate higher accuracy in location estimation and identification problems than a single-layer feedforward NN because of their Gaussian functions. The main drawbacks of GRNNs are being computationally expensive and having no optimal method to improve their efficiency. In this study, it is shown that CNN can outperform GRNN in both identification and location estimation problems because of its power in feature extraction and resilience to noise. Among these two NNs, CNN is the only one which is sensitive to spatial features which makes it the best solution to TL fault diagnosis problem in this study.

Table 3.3: Parameter Values for The Generation of Training Data set

Parameter	**FNN, GRNN, CNN**
Fault Distance ($\boldsymbol{km}$)	$0.01, 30, 60, 90, 110$
Fault Inception angle ($^\circ$)	$1, 20, 50, 100, 150$
Fault Resistance $R_f(\Omega)$	$0.1, 1, 5, 10, 20, 50, 100, 150, 200, 300, 500$
Source Inductance L_s (mH)	$5, 10, 15, 20, 25, 30, 35, 40, 45, 50$
Phase Difference $\Delta\phi$ ($^\circ$)	$-30, 0, 30$
Voltage Fluctuations ΔV_i (kV), $i = 1, 2$	$-40, 0, 40$

3.3.4 Generation of Features

The data used for this study include the amplitudes of the main harmonics of the voltage and current waveforms calculated using FFT, which can compute frequency components with reduced computational complexity as compared to traditional Fourier transform. To this end, 1.5 cycles of the post-fault voltage and current signals are selected, their main harmonics are calculated, and the normalized amplitudes of the main harmonics are fed to the detection/identification and location estimation modules. In order to generate training data, several variations in the fault model were considered that included fault type, location, inception time, and resistance. In addition, different values for voltage amplitudes of the two generators as well as their phase difference were used to generate the training data. Table 3.3 shows the parameters that were changed for the generation of training dataset. In this table V_i $(i = 1, 2)$ represents the voltage amplitude of generator $\#i$ and $\Delta\phi$ is the phase difference between the two generators.

Distinguishing between LL and LLG faults cannot be accurately done by a fault diagnosis system only based on phase voltage and current signal measurements. Therefore, the zero-sequence current is calculated to provide a better indication of a ground fault since there is a considerable amount of zero-sequence current for an LLG fault. Such current is calculated from the mean of the phase currents.

Figs. 3.6 and 3.7 show the zero-sequence current without and with ground fault happening at 0.5 (s), respectively. As observed in Figure 3.6, when the ground is not involved in a fault, the zero-sequence current is almost zero. As shown in Figure 3.7, when the ground is involved, the oscillation amplitude increases significantly. Consequently, this significant difference is used in the NN training process for detection of ground involvement in faults.

3.3.5 Fault Detection, Identification and Location Estimation

The first step in fault detection/identification and location estimation is to acquire data from one end of the TL. As seen in Figure 3.1, the measurement device is placed at bus #2. The current and voltage signals of the three-phase TL are recorded via a 30-sample time window with the sampling frequency of 1.2 (kHz). For each window at every step, the FFT is applied to all three phases to calculate the amplitudes of the fundamental frequency components of current and voltage signals. In addition, the zero-sequence component is calculated for ground fault detection. The normalized extracted features are fed to the fault detection/identification and location estimation systems which are based on GRNN and CNN. The flowchart of the detection and location estimation procedure is shown in Figure 3.8. The solid lines represent the causal relationship among the modules which means that a module starts running after another module finishes. The dash lines indicate the data dependency which means that the extracted features are continuously and concurrently fed to the detection, identification and location estimation modules. However, the output of the location estimation module is ignored and masked before the faults are identified.

The fault detection/identification module generates four outputs. The first three outputs correspond to A, B, and C phases. The last output is associated with ground fault. In a non-faulty scenario, the values of all these outputs are zero. However, in a faulty situation, their values change. The time when at least one of these outputs (flags) switches to 1 is called "detection time". At this point, the type of a fault has not been determined yet. In other words, other outputs may switch to 1 after waiting for more samples. Based on extensive number of simulations for different fault scenarios for the GRNN (as an example), it is found out that the longest identification is achieved within 30 samples or 0.025 (s) from the fault occurrence time. Thus, we let the system wait for 30 samples so that the type of a fault can be identified accurately. This 0.025 (s) (30 samples) wait time is called "identification delay".

At the same time, all of the NNs in the fault location estimation module try to estimate the fault location, but the results are ignored and masked before the identification of a fault as demonstrated in Figure 3.9, because it is not clear yet as which location estimation NN to switch to and use the results from. When the type of a fault is identified, the corresponding location estimation NN is selected to estimate the fault location. As indicated in Figure 3.9, there is one location estimation NN specifically designed for each of the 4 category of fault cases (LG, LL, LLG, and LLL). It should be mentioned that each location estimation NN waits for 90 samples or 0.075 (s) after the fault occurrence time to reach a steady state value. This 0.075 (s) (90 samples) wait time is called "location estimation delay". After a fault is identified, the associated location estimation NN is selected, for which the average value of the most recent 30 samples $D_i(i = 1, 2, ..., 30)$ is calculated for the estimate of the fault location.

3.4 Simulation Results

In this section, the simulation results for the performance analysis of fault detection, identification, and location estimation system in terms of time delays and robustness are presented. The following results are based on the model demonstrated in Figure 3.1 with the parameters specified in Tables 3.1 and 3.2. It should be noted that all the simulations were run on Matlab/Simulink using the real-time simulator OPAL-RT (OP5700). OP5700 contains a powerful computer which has a linux-based real-time operating system and its CPU specifications are Intel Xeon E5, 8 Cores, 3.2 (GHz), and 20 (MB) Cache. The TL model shown in Figure 3.1 is simulated in the Simulink environment which is connected to RT-Lab software in OP5700. Then, the generated real-time faulty data are fed to the NN which is simulated in Matlab environment connected to RT-Lab. The fault detection, identification, and estimation results are transmitted to a regular PC using a LAN cable and shown in Matlab environment in Windows operating system as demonstrated in Figure 3.10.

3.4.1 Time Delay Analysis

The faults should be detected and estimated before the tripping relays and circuit breakers start to disconnect a faulty section of the TL. According to [193], this time interval is between 0.3 to 0.5 (s) from occurrence of a fault. Therefore, the location estimation step which is the last stage of the fault diagnosis process should be completed within this time interval. As explained in the previous section, the longest identification and location estimation delays are calculated based on Monte Carlo simulations for different sets of fault parameters.

Figure 3.11 demonstrates the times/delays of detection, identification and location estimation stages for *CNN*, and for the BCG fault. It also indicates the time evolution of the detection/identification and location estimation processes after the occurrence of a fault. The corresponding delays for the CNN are 0.0008 (s), 0.020 (s), and 0.068 (s), and for the GRNN are 0.0017 (s), 0.025 (s), and 0.075 (s), respectively. These results confirm that both of the NN-based fault diagnosis systems studied in this work are able to estimate the location of faults well before the circuit breakers disconnect the faulty region. Moreover, it is observed that the CNN-based fault diagnosis system can detect, identify, and localize the faults faster than GRNN.

3.4.2 Robustness Analysis

The main parameters (uncertainties) that may change and hence impact the performance of a TL fault detection, identification, and location estimation are fault resistance (R_f), phase difference between two sources ($\Delta\phi$), voltage fluctuations of the two sources (ΔV_i), source inductance (L_s), and fault inception angle. The fault identifier and location estimator NNs are trained to be robust against these parameter variations. For analyzing the detection/identification performance, the "accuracy" criterion is defined as the total number of correctly identified faulty or non-faulty scenarios divided by the total number of simulation experiments.

The fault identification and location estimation schemes should perform accurately for different fault resistances. Table 3.4 indicates the average accuracy of fault identification stage with respect to the variations of the fault resistance and for different fault types. The variations are considered to range from 0.5 (Ω) to 470 (Ω) [112]. It can be seen that CNN with average accuracy of 99.78% outperforms GRNN (average accuracy of 99.24%) in terms of the identification accuracy with various fault resistance values.

Table 3.4: Average Accuracy of Fault Identification With Respect to Fault Resistance (R_f) Variations for Two NNs

Fault	R_f ((Ω))	GRNN %	CNN %
LL		99.12	99.65
LG	0.5, 22, 43, 95, 120, 470	99.25	99.66
LLG		99.11	99.84
LLL		99.51	100

Table 3.5: Average Accuracy of Fault Identification With Respect to Phase Difference ($\Delta\phi$) Between Two Buses For Two NNs

Fault	$\Delta\phi$ (%)	GRNN (%)	CNN (%)
LL		99.87	99.96
LG	-50 to +50	99.76	99.79
LLG		99.96	99.81
LLL		99.94	99.82

Phase difference between the two generators varies from time to time due to the various operating conditions. In Table 3.5, the average accuracies of the three techniques are compared with respect to the variations of phase difference ($\Delta\phi$) between two generators from -50% to $+50\%$. It is observed that with the increase of the phase difference between the two generators in both directions, the performance of the three networks deteriorates. However, the CNN (average accuracy = 99.84%) shows to have better accuracy as compared to GRNN (average accuracy = 99.88%).

The fault diagnosis scheme should identify and localize the fault occurring at any fault inception time or angles. This parameter is varied from 27° to 180° [121]. Table 3.6 demonstrates the average accuracies of the three NNs with respect to the variations of the fault inception angle. The average accuracies for CNN and GRNN are 99.98% and 99.85%, respectively.

The voltages of generators can change due to the different operating conditions of the generators. Based on the IEEE standard 1250 [197], the voltage fluctuations of a bus cannot exceed 10% of its nominal voltage level. In this study, the voltage changes

Table 3.6: Average Accuracy of Fault Identification With Respect to the Variations of Fault Inception Angle

Fault	Inception angle (°)	GRNN (%)	CNN (%)
LL		99.95	100
LG	27 to 180	99.85	99.97
LLG		99.83	99.99
LLL		99.80	99.96

Table 3.7: Average Accuracy of Fault Identification With Respect to Bus Voltage Variations ($\Delta V_i, i = 1, 2$) For Two NNs

Fault	$[\Delta V_1, \Delta V_2]$ $\boldsymbol{(kV)}$	**GRNN(%)**	**CNN(%)**
LL		99.51	99.91
LG	-55 to 55	99.69	100
LLG		99.52	99.93
LLL		99.63	100

increase beyond this limit in order to show the robustness of the fault diagnosis system with respect to this variation as shown in Table 3.7. The average accuracies for GRNN and CNN are 99.58% and 99.96%, respectively.

Table 3.8: Average Accuracy of Fault Identification With Respect to Source Inductance (L_s) for Two NNs

Fault	$\boldsymbol{L_s(mH)}$	**GRNN**	**CNN**
LG		99.63	99.97
LL	7, 17, 27, 37, 47	99.79	99.99
LLG		98.86	100
LLL		99.52	99.94

As mentioned in the introduction, the source inductance plays an important role in the shape of faulty waveforms which negatively affects the estimated location of faults in TW approaches. The fault diagnosis system should be able to detect and identify the faults with the variations of source inductances. Table 3.8 shows the average accuracies of the two NN techniques for source inductances from 7 (mH) to 47 (mH). The average accuracy for GRNN and CNN techniques is 99.70% and 99.95%, respectively.

Similar to the accuracy analysis, the location estimation error analysis is performed with respect to the parameter variations of the transmission line. For this purpose, the absolute "Relative Error" is defined as follows:

$$R_E = \frac{|\text{Fault Distance} - \text{Estimated Fault Distance}|}{\text{Total TL Length}} \times 100 \qquad (3.7)$$

Table 3.9 shows the average relative error of fault localization with respect to the variations in fault resistance for fault different types and locations. The average of relative errors for GRNN and CNN is 0.25% and 0.052%, respectively, which shows the superior performance of CNN over GRNN.

The average relative error analysis with respect to the changes in the phase difference between the two generators is shown in Table 3.10 for different types of faults and for different fault locations. The average of relative error for GRNN and CNN are 0.27% and 0.017%, respectively. Their performance in location estimation with respect to the inception angle variations is shown in Table 3.11. Their average of relative errors are 0.07% and 0.02%, respectively. Their average relative errors with respect to bus voltage variations are demonstrated in Table 3.12. Their average of relative errors is 0.26% and 0.03%, respectively.

The performance of both NN techniques with respect to source inductance variations is shown in Table 3.13. It can be seen that the average of relative errors for GRNN and CNN techniques is 0.15% and 0.02%, respectively.

3.4.3 Effect of Noise

In real scenarios, the measurement noise in power systems may affect the accuracy of the fault diagnosis system. In order to assess the robustness of the fault detection/identification and location estimation system against the noise, a Gaussian noise is added to the measurement data (voltages and currents) so that the signal to noise ratio is 15 (dB). The results of the identification accuracy and location estimation errors for the two NNs subject to the above noise are shown in Table 3.14. Based on Table 3.14, it is clearly seen that adding the Gaussian noise weakens the performance of all the three NNs. However, the CNN still has a better performance than GRNN.

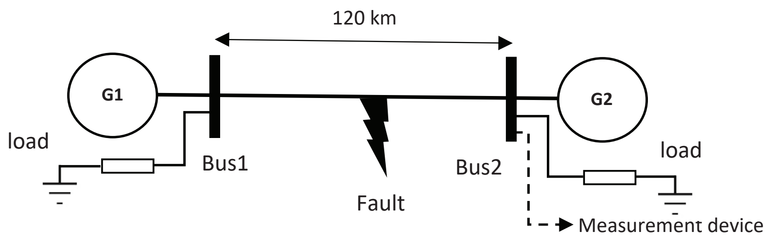


Figure 3.1: A two-bus power system.

Table 3.9: Average Relative Error of Fault Location Estimation with Respect to Fault Resistance (R_f) Variations for Two NNs

Fault	$R_f(\Omega)$	GRNN (%)	CNN (%)
LG		0.23	0.06
LL	0.5, 22, 43, 95, 120, 470	0.17	0.05
LLG		0.36	0.06
LLL		0.25	0.04

Table 3.10: Average Relative Error of Fault Location with Respect to Phase Difference ($\Delta\phi$) Between Two Buses for Two NNs

Fault	$\Delta\phi$ (%)	GRNN	CNN
LG		0.22	0.02
LL	-25 to +25	0.17	0.01
LLG		0.30	0.02
LLL		0.42	0.02

Table 3.11: Average Relative Error of Fault Location Estimation with Respect to the Fault Inception Angle for Two NNs

Fault	Inception angle (°)	GRNN	CNN
LL		0.08	0.01
LG	27 to 180	0.05	0.01
LLG		0.09	0.03
LLL		0.06	0.03

Table 3.12: Average Relative Error of Fault Location Estimation with Respect to Bus Voltage Variations (ΔV_i) for Two NNs

Fault	$V_1 - V_2(kV)$	GRNN	CNN
LG		0.41	0.04
LL	-55 to 55	0.08	0.04
LLG		0.36	0.03
LLL		0.19	0.04

Table 3.13: Average Relative Error of Fault Location Estimation with Respect to Source Inductance (L_s) for Two NNs.

Fault	$L_s(mH)$	GRNN	CNN
LG		0.07	0.03
LL	7, 17, 27, 37, 47	0.18	0.02
LLG		0.11	0.01
LLL		0.25	0.02

Table 3.14: Performance of Two NNs Subject to the Gaussian Noise.

NN	Identification accuracy(%)		Location estimation relative error(%)	
	Without noise	With Noise	Without noise	With Noise
GRNN	99.69	99.08	0.50	0.23
CNN	99.87	99.26	0.04	0.06

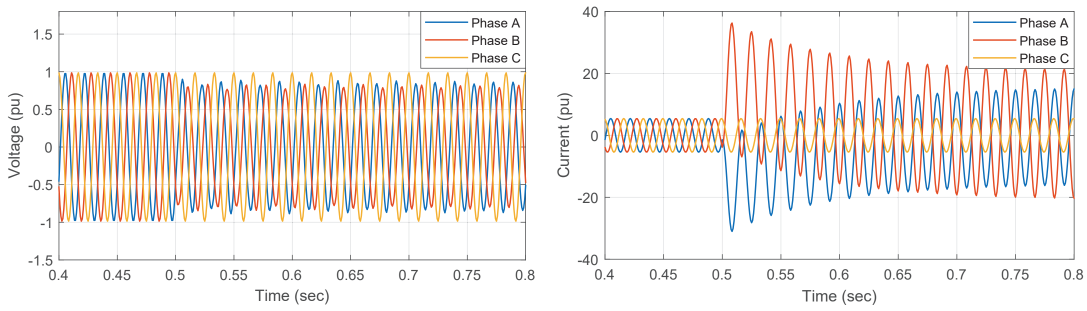


Figure 3.2: The measured (a) voltages (b) currents of one end of the TL for an LL (phase A to phase B) fault at t = 0.5 (sec).

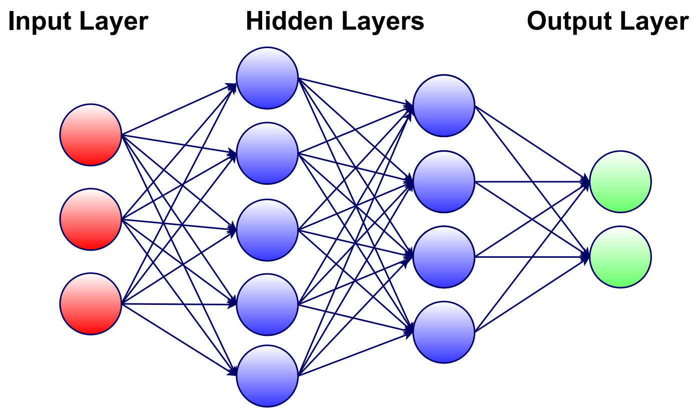


Figure 3.3: Structure of an FNN.

Table 3.15: Comparison Between the Proposed Classifiers in this study and the Other Schemes in the Literature (I Current and V Voltage)

Reference	**Used Techniques**	**Input Signal**	**Performance With Noise**	**Robustness Analysis**	**Average Accuracy**	**Average Relative Error**
[111]	WPT+SVM	I	Not Mentioned	Not Mentioned	99.2%	0.21%
[198]	WT+ELM	I	Not Mentioned	Not Mentioned	96.5%	0.5%
[182]	ST+PNN	V	Immune	Not Mentioned	99.6%	4.46%
[110]	RTU+ANFIS	V+I	Not Mentioned	Not Mentioned	100%	0.81%
[154]	WT+SAT-CNN	V+I	Immune	Not Mentioned	99.5%	Not Mentioned
[12]	WT+ANN	V+I	Not Mentioned	Incomplete	100%	0.27%
[142]	WT+ANN	V+I	Immune	Incomplete	100%	0.68%
Proposed Methods I	GRNN	V+I	Immune	Complete	**99.5%**	**0.20%**
Proposed Methods II	CNN	V+I	Immune	Complete	**99.9%**	**0.03%**

Table 3.15 shows the comparison results among the existing methods and the proposed techniques. Among all the mentioned approaches only [142] presents immunity to noise and robustness analysis with respect to the parameter changes in Table 3.1. However, it does not consider the voltage fluctuations. Therefore, both of our proposed methods, considering all parameters mentioned in Table 3.1, generate the best results for the classification and location estimation of the TL faults.

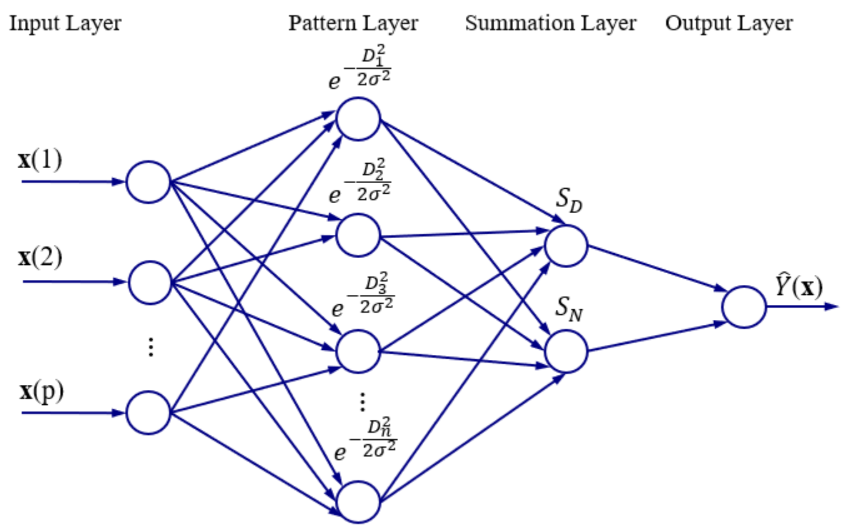


Figure 3.4: The structure of a GRNN.

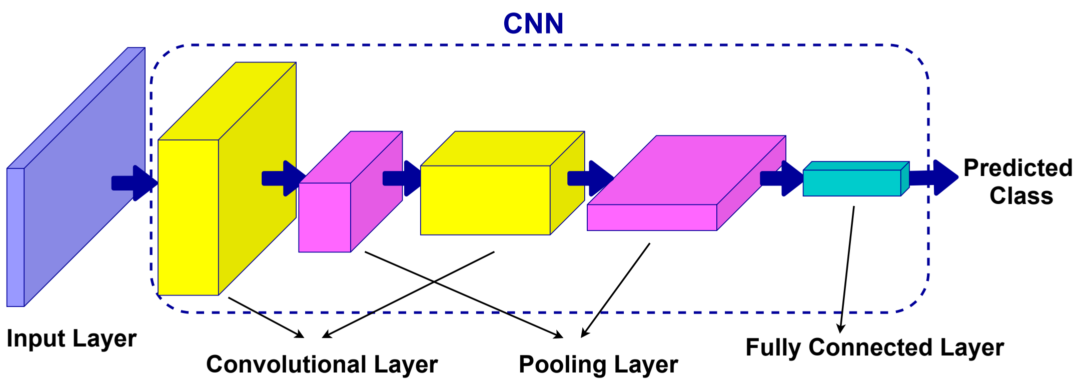


Figure 3.5: General architecture of a CNN.

3.5 Summary

In this chapter, the problem of fault detection, identification, and location estimation of transmission lines using two NNs, namely, GRNN and CNN is presented. The data used in this study are the phase current and voltage measured from one end of the transmission line. An FFT is used to extract the amplitude of the fundamental frequency components from the current and voltage waveforms, and feed them to both of the NNs for the detection, identification, and location estimation of faults. The main focus of this chapter is that the robustness of the proposed detection, identification, and location estimation techniques against the parameter changes in a transmission line, namely fault resistance, fault inception angle, source inductance, phase difference between the two buses, bus voltage amplitude variation,

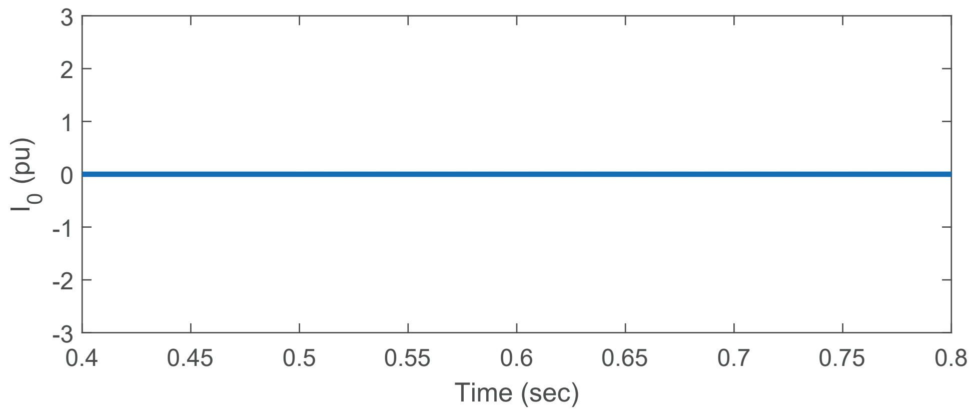


Figure 3.6: I_0 waveform when ground is not involved in an LL fault.

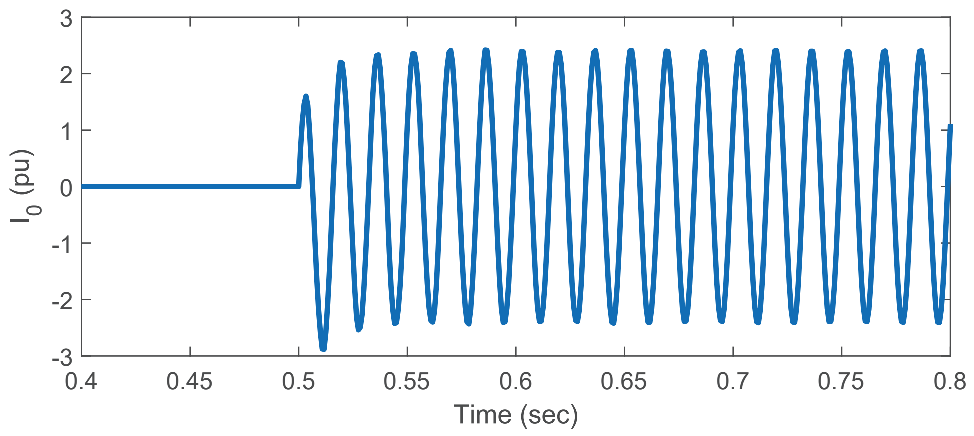


Figure 3.7: I_0 waveform when ground is involved in an LL fault.

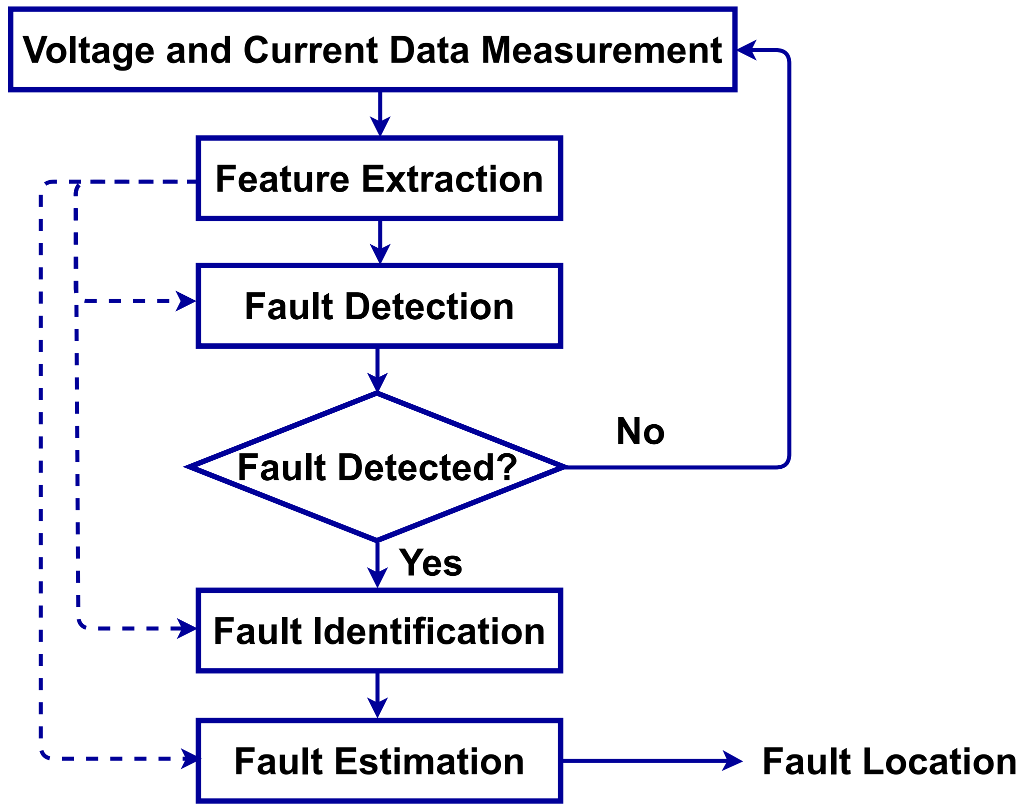


Figure 3.8: Flowchart of the detection, identification, and location estimation procedure.

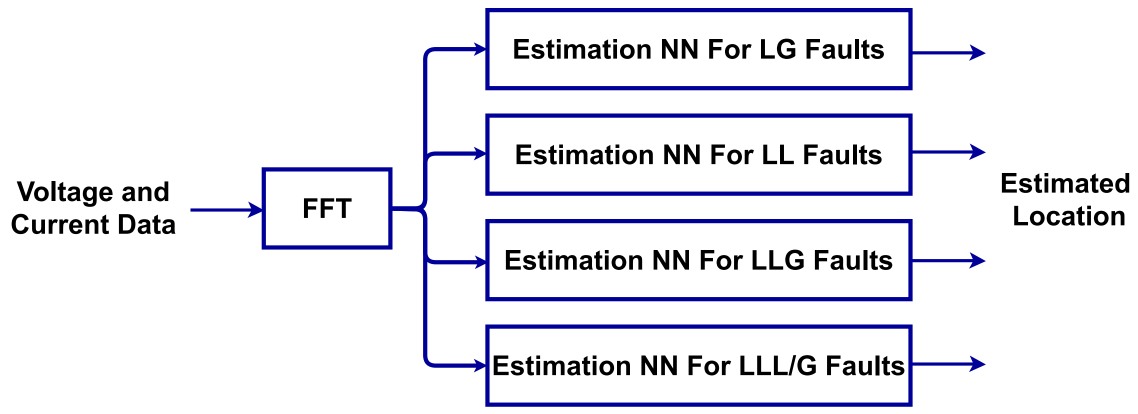

Voltage and Current Data
FFT
Estimation NN For LG Faults
Estimation NN For LL Faults
Estimation NN For LLG Faults
Estimation NN For LLL/G Faults
Estimated Location

Figure 3.9: The fault location estimation module including multiple location estimation NNs.

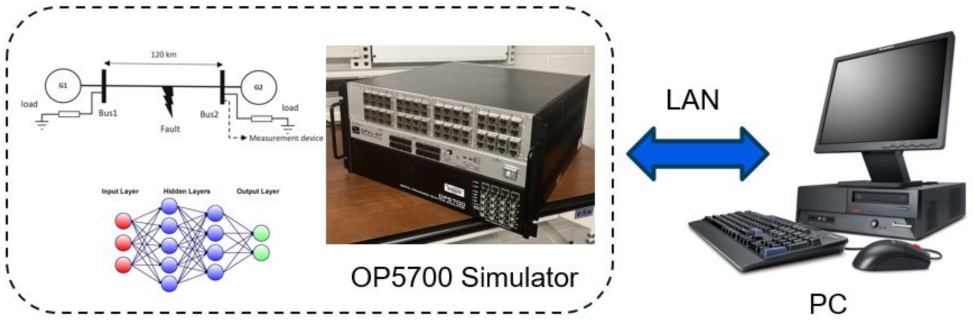


Figure 3.10: Real-time simulation using OP5700.

and measurement noise are analyzed. Besides, a time delay analysis is performed to guarantee that these modules can successfully complete their tasks within the desired time window based on the IEEE standard before the tripping relays disconnect the transmission line. In overall, CNN has better performance in both identification and localization of the faults in comparison with GRNN. It should also be noted that both GRNN and CNN have better robust performance compared to other techniques in the literature.

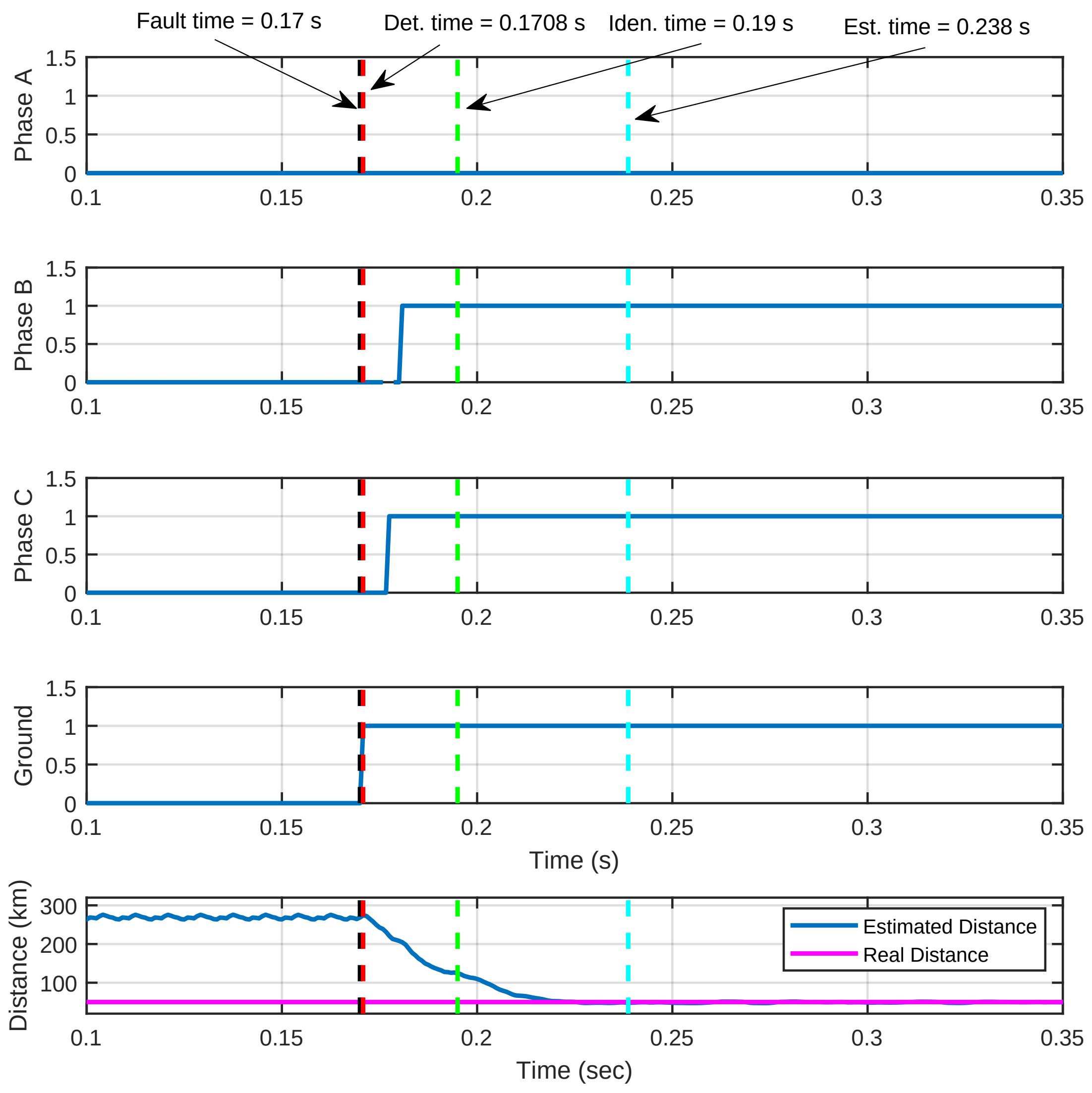


Figure 3.11: CNN location estimation, identification, and detection outputs with the corresponding time. "Est.", "Iden.", and "Det." represent these three tasks, respectively.

CHAPTER 4

GENERALIZED FAULT DIAGNOSIS METHOD OF TRANSMISSION LINES USING TRANSFER LEARNING TECHNIQUE

4.1 Introduction

Recent artificial intelligence-based methods have shown great promise in the use of neural networks for real-time detection of transmission line faults and estimation of their locations. The expansion of power systems including transmission lines with various lengths have made the fault detection, classification, and location estimation process more challenging. Transmission line datasets are stream data which are continuously collected by various sensors and hence, require generalized and fast fault diagnosis approaches. Newly collected datasets including voltages and currents for faulty and non-faulty situations might not have adequate and accurate labels that are useful to train neural networks.

In this chapter, a novel transfer learning framework based on a pre-trained LeNet-5 convolutional neural network is proposed. This method is able to diagnose faults for different transmission line lengths and impedances by transferring the knowledge from a source convolutional neural network to predict a dissimilar target dataset. By transferring this knowledge, faults from various transmission lines, even without sufficient data samples with labels, can be diagnosed faster and more efficiently than the existing methods. To prove the feasibility and effectiveness of this methodology, seven different datasets that include various lengths of transmission lines are used. The robustness of the proposed methodology against the generator voltage fluctuations, variations in fault locations, fault inception angle, fault resistance, and phase difference between the two generators are well studied to prove the reliability of this technique for fault diagnosis of transmission lines..

A powerful technique to handle the predictive modeling for different but somehow related problems is transfer learning in which partial or complete knowledge of one model, e.g., a convolutional neural network (CNN) [199], is transferred and reused to increase the speed of the training process and improve the performance of another model (e.g., CNN) [200]. The transfer learning technique is generally used to save resources such as time, data, and computing power which are used to train multiple machine learning models from scratch to complete new tasks related to the existing ones. This technique also resolves a lack of labelled training data by using pre-trained models. In this study, a transfer learning-based CNN is used for the first time to detect, identify, and locate the faults for various lengths of TLs. It is demonstrated in this work that our methodology is reusable, fast, and accurate.

There are several studies in the literature that use CNNs for fault diagnosis of TLs. In [154], a self-attention CNN framework and a time series image-based feature extraction model are presented for fault detection and classification of TLs with length of 100 (km) using a discrete wavelet transform (DWT) for denoising the faulty voltage and current signals. In the study [155], authors present a customized CNN for fault detection and classification of 50 (km) TLs integrated with distributed generators. The work done in [156] proposes a machine learning-based CNN for TLs with length of 280 (km) to perform fault detection and classification using DWT for feature extraction.

Shiddieqy et al. [201] present another methodology that considers all features of the TL faults to generate various models for robust fault detection. They take advantage of various AI methods including CNN to achieve a 100% detection accuracy. The length of the TLs in this study is 300 (km). The study in [158] presents a scheme to detect and categorize faults in power TLs with length of 200 (km) using convolutional sparse auto-encoders. This approach has the capability to learn extracted features from the dataset of voltage and current signals, automatically, for fault detection and classification. To generate feature vectors, convolutional feature mapping and mean pooling methods are applied to multi-channel signal segments.

There are also several studies that apply transfer learning methods to time series datasets in different applications [202, 203, 204, 205, 206, 207]. Fawaz et al. [208] show how to transfer deep CNN knowledge for time series dataset classification. In [209], an intelligent method is proposed as a deep convolutional transfer learning network which diagnoses the dynamic system faults using an unlabeled dataset. Shao et al. [210] propose an intelligent fault diagnosis method for a rotor-bearing system which is based on a modified CNN with transfer learning. Li et al. [211] also presented a deep adversarial transfer learning network to diagnose new and unlabeled emerging faults in rotary machines.

In this chapter, we take advantage of transfer learning method to propose a generalized solution for TL fault diagnosis. The contributions of this work are as follows:

1. Proposing a generalized approach to detect, identify, and locate faults in TLs with various lengths using transfer learning technique for the first time.
2. Comparing the proposed methodology with three benchmarks and showing its effectiveness in prediction performance.
3. Performing robustness analysis against variations of fault resistance, fault inception angle, source inductance, phase difference between two connected buses, bus voltage fluctuations, and measurement noise during the process of generalization.

The rest of the study is organized as follows. Section 5.2 presents preliminary concepts used in this study including time series classification, CNNs, and transfer learning technique. In Section 4.3, the TL model used in this study as well as the feature generation approach are discussed. Section 4.4 describes the dataset generation procedure and the proposed transfer learning-based method. Section 5.4 discusses the results and compares our proposed method with three benchmarks including K-means clustering, CNN without transfer learning, CNN with transfer learning technique but without fine tuning process. Finally, conclusions are made in Section 5.5.

4.2 Preliminaries

In this section, an overview of time series data classification, CNNs, and transfer learning procedure is provided.

4.2.1 Time Series Classification

In general, time series data can be defined in two different ways as described below [208].

- Definition 1. A time series data is an ordered (time dependant) set of real values such as $X = [x_1, x_2, ..., x_n]$ where n is the number of real values and the length of X [208].

- Definition 2. A time series dataset D is defined as $D = \{(x_1, y_1), ..., (x_n, y_n)\}$ with length of n which consists of a collection of pairs (x_i, y_i) where x_i is a time series data point with its corresponding label (class) as y_i [208].

Time series classification is defined as classifying the dataset D by considering every input x_i to train a classifier which maps the given inputs to the given labels based on every class variable y_i [208]. It is clear that sometimes D consists of a set of pairs in which the inputs and labels are vectors. In this study, the system utilizes voltage and current waveforms recorded from one end of a two-bus TL as inputs, and considers 10 types of faults as well as the non-faulty case as the labels (classes). FFT is applied to both current and voltage waveforms to generate the amplitude of the main frequency component of the signals. These amplitude data are used as the inputs to the CNN. Therefore, the dataset used to train the CNN includes seven different features from which the first three features are associated with voltages, the second three features are associated with currents, and the last one is related to the zero sequence signal which is the average of the phase currents. The reason for having the last feature will be discussed later in Section 4.3.

4.2.2 Convolutional Neural Network

Having the capability of learning hierarchical features independently from inputs, CNNs are widely used for image datasets. CNNs have a minimum need for pre-processed data and can handle high dimensional datasets faster and with more details in comparison with most of artificial CNNs [212, 145]. In general, CNNs include three types of layer: convolution layer, pooling layer, and fully connected layer. Convolutional and pooling layers incorporate convolution blocks which are stacked for feature extraction purpose. Fully connected layers are used as classifiers and the output layer, which is a fully connected one, performs the classification or regression task.

The main advantages of CNN architecture are local receptive fields, shared weights, and the pooling operation. CNNs take advantage of the concept of a local receptive field which means that only a small focused area of the input data is connected to each node in a convolution layer. Because of this trait, the number of parameters is reduced considerably in CNN which in turn decreases the training computational expenses of the NN [213].

Kernel convolution is used in many Computer Vision algorithms. In this process we take a small matrix of numbers (called kernel or filter), we pass it over our image and transform it based on the values from filters. The feature map values are computed based on Equation 4.1, where the input image is denoted by f and our kernel by h. The indexes of rows and columns of the result matrix are marked with m and n respectively.

$$G[m,n] = (f \times h)[m,n] = \sum_{j} \sum_{k} h[j,k] f[m-j, n=k] \tag{4.1}$$

Numerous studies are performed in TL fault detection, classification, and localization problems using CNNs to achieve higher accuracies. These studies are divided into two different categories: The methodologies with a focus on image-based datasets recorded from outdoor TLs [151, 152, 153], and the ones that consider time-series voltage and current waveforms recorded from generators and are fed to CNNs as blocks of data points. The methodology proposed in this study belongs to the second category.

4.2.3 Transfer Learning

Transfer learning is proposed to solve the learning problems between two or multiple domains. The combination of deep learning and transfer learning shows notable improvements in the accuracy and time of fault diagnosis approaches. Transfer learning consists of two steps: First, the process of training a source neural network on a source dataset and task, and second, transferring the learned features and knowledge to a new network to help the training process of the new related (target) dataset. Two main concepts are used in transfer learning, namely, domain and task which are defined below.

- Definition 3 (Domain [214, 215]). A domain $\mathcal{D} = \{\mathcal{X}, \mathcal{P}(X)\}$ consists of two elements, namely, a feature space $\mathcal{X}$ and a marginal probability distribution $\mathcal{P}(X)$ where $X = \{x_i\}_{i=1}^{n} \in \mathcal{X}$ is a dataset in which every $x_i \in \mathbb{R}^D$ is sampled from this domain.

- Definition 4 (Task [214, 215]). A task $\mathcal{T} = \{\mathcal{Y}, \mathcal{P}(Y|X)\}$ consists of two elements given a domain $\mathcal{D} = \{\mathcal{X}, \mathcal{P}(X)\}$, where $\mathcal{Y}$ stands for the label space and $\mathcal{P}(Y|X)$ is the conditional probability distribution in which $Y = \{y_i\}_{i=1}^{n}$ shows the label vector of X with $y_i \in \mathcal{Y}$ as the label of x_i.

There are two domains in transfer learning, namely, a source domain $\mathcal{D}_s = \{\mathcal{X}_s, \mathcal{P}_s(X_s)\}$ and a target domain $\mathcal{D}_t = \{\mathcal{X}_t, \mathcal{P}_t(X_t)\}$ where $\mathcal{X}_s$ and $\mathcal{X}_t$ show the feature spaces of the source and target domains, respectively, and $\mathcal{P}_s(X_s)$ and $\mathcal{P}_t(X_t)$

stand for the marginal probability distribution of them. Based on definitions 3 and 4, the definition of transfer learning is presented below.

- Definition 5 (Transfer Learning [215]). Considering the source domain $\mathcal{D}_s$, learning task $\mathcal{T}_s$, target domain $\mathcal{T}_s$, and learning task $\mathcal{T}_t$, the goal of transfer learning is to promote the performance of target predictive function $f_t(.)$ in $\mathcal{D}_t$ by inducing the knowledge to $\mathcal{D}_s$ and $\mathcal{T}_s$ while $\mathcal{D}_s \neq \mathcal{D}_t$ or $\mathcal{T}_s \neq \mathcal{T}_t$.

In the case of $\mathcal{T}_s = \mathcal{T}_t$, a common subproblem of transfer learning takes place which is called domain adaptation. In this study, the relationship between the source and target data is the domain adaptation because the seven aforementioned features (currents, voltages, and zero sequence current) repeat in both the source and target TLs which only differ in length [216].

Transfer learning provides the ability to distribute learned features across different learning applications. The focus of transfer learning is on the deep learning training step to enhance its capability in common feature extraction and adoption among multiple datasets of a similar problem. Transfer learning is based on using the pretrained layers on a source task to solve a target task. For this purpose, a pretrained model in which its fully connected layers are cut off is used and its convolutional and pooling layers become frozen (their weights are not updated) to perform the role of feature extractors. To adjust the given pretrained network with the target dataset and efficient target classification, the fully connected layers (classifier section) need to update their weights. There are two different approaches in transfer learning including feature extraction and fine tuning. In the first approach, the fully connected layers are completely removed and based on the target dataset, the new fully connected layers are integrated with the frozen pretrained layers. In the fine tuning process, the structure of fully connected layers from the pretrained model is saved and only their weights are updated. Figure 5.2 shows the basic concept of transfer learning representing two different datasets which have similarities. These datasets are fed

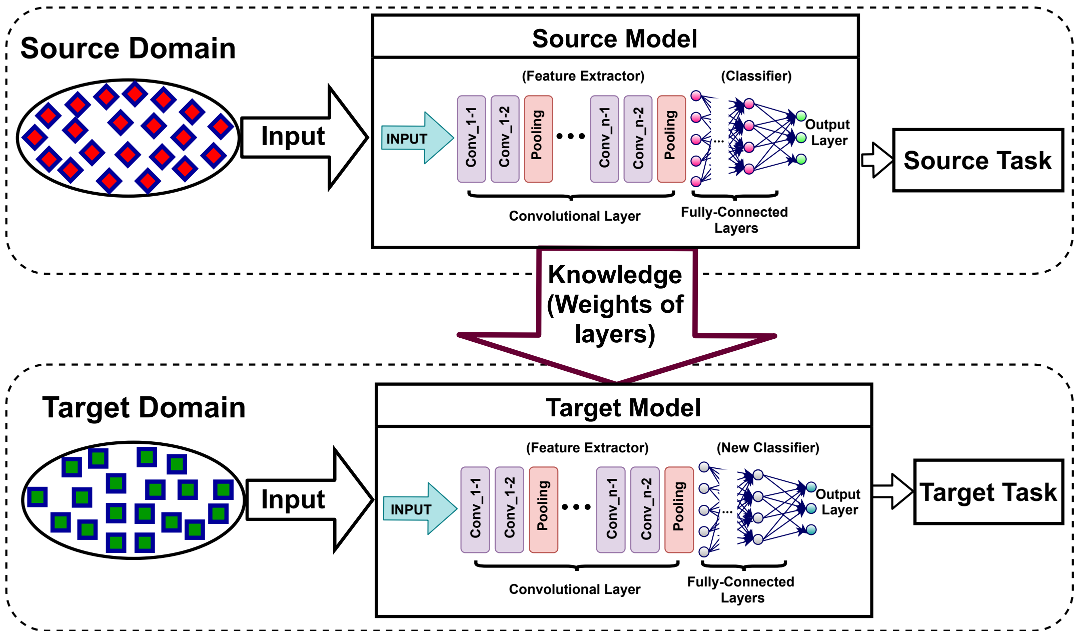


Figure 4.1: A typical model of transfer learning.

into source and target models. The knowledge is transferred to the target model to perform the training of target task faster. In this study, the fine tuning approach is used to achieve satisfactory results. Transfer learning method is used in this study for TL fault diagnosis problems to detect the rare occurrence of failures which are difficult or impossible to be labeled. It should be noted that a transfer learning-based CNN has not been used before for TL fault diagnosis problems.

4.3 Transmission Line Model and Feature Generation

In this study, a power system with two generators which are connected through a 100 (km) three-phase TL is used. Based on some recent studies [168, 141, 143, 154], a common length is chosen which lies in the medium range of transmission line lengths and its multiplications can reside in short or long TLs. The voltage of both generators is 240 (kV) and their frequency is 60 (Hz) as shown in Figure 4.2. This model is simulated in MATLAB Simulink's Simscape Power System, and all the ten short-circuit faults i.e., single Line-to-Ground (LG), Line-to-Line (LL), double

Line-to-Ground (LLG), and all-Lines-connected (-to-Ground) (LLL/LLLG) as well as no-fault state are taken into account.

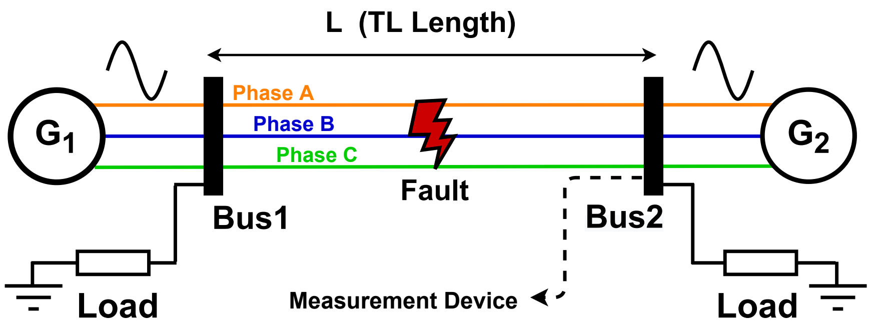


Figure 4.2: A three-phase and two-generator power system.

The parameters of the TL model and the features of generators are shown in Table 4.1 and Table 4.2, respectively. The model studied in this work is based on *IEEE 39*-Bus System which includes 10 generators and 46 lines.

Table 4.1: TL Nominal Parameters

Parameter	Zero Sequence	Positive Sequence
R $(\Omega/(km))$	0.3864	0.01273
L $(mH/(km))$	4.1264	0.9337
C $(\mu\text{F}/(km))$	7.751×10^{-3}	12.74×10^{-3}

Table 4.2: Source and Load Nominal Parameters

Nominal Parameter	Source 1 & 2	Load
Phase to Phase Voltage (kV)	240	240
Frequency (Hz)	60	60
Resistance (Ω)	0.08929	—
Inductance (mH)	16.58	—
Active Power (kW)	—	100
Inductive Reactive Power $(kVAR)$	—	< 100
Capacitive Reactive Power $(kVAR)$	—	< 100

The general datasets used in this study consist of the amplitudes of the fundamental frequency component of the voltage and current waveforms calculated by FFT which computes the frequency components faster and more efficiently than the conventional Fourier transform. For this purpose, 1.5 cycles of the most recent time-series data of voltage and current waveforms are selected. The 1.5 cycle window is shown to be effective since the overall time delay for the fault detection, identification, and location estimation procedures is within 0.3 to 0.5 (sec) as required by the IEEE standard [193]. Then, their fundamental frequency components are computed, normalized and fed into the fault diagnosis module.

To assess the robustness of the proposed methodology, some variations in the TL model are considered such as fault type, location, inception angle, resistance, voltage amplitudes of the generators, source inductance, and the phase difference between them. Such variations help the system to generate datasets that are large enough to generate reliable results. In general, the dataset used in this work includes 7 features that are 3 phase voltages, 3 phase currents, and a zero-sequence current which is a detector for ground faults. In other words, a fault diagnosis model cannot distinguish between LL and LLG faults only by considering phase voltage and current signal values. In such cases, a zero-sequence current, which is the average value of the phase currents, is considered to detect ground faults.

Table 4.3: Parameter Values for the Generation of Training Dataset

Parameter	**Variations**
Fault Distance (km)	1.2, 10, 24, 40, 60, 95
Fault Inception angle ($^\circ$)	1, 20, 50, 100, 150
Fault Resistance R_f (Ω)	0.1, 1, 10, 20, 30, 40, 50, 60
Phase Difference$\Delta\phi$ ($^\circ$)	-30, 0, 30
Voltage Fluctuations Δ V_i $(kV) = V_1 - V_2$	-40, 0, 40

Table 4.3 shows the parameters with their variations to generate robust results. It should be noted that because the length of a TL is the critical parameter in this study, the variation of fault location is defined as a dependent parameter to the TL length ($L = 100km$), and for different lengths, the initial fault distances for the reference dataset would be different, as well.

Table 4.4: The Average Accuracy Results of K-means Clustering for 11 Types of Faults Based on Statistical Testing

Length (km)	**12.5**	**25**	**50**	**100**
Accuracy (%)	82.42 ± 0.10	82.47 ± 0.08	83.47 ± 0.13	85.09 ± 0.15
Length (km)	**200**	**400**	**800**	**Average**
Accuracy (%)	87.12 ± 0.12	87.67 ± 0.04	83.87 ± 0.17	84.58 ± 0.12

4.4 Proposed Fault Diagnosis Approach

In this section, the proposed method based on transferred CNN is described. Acquiring data from one end of a TL is the initial step in fault detection/ classification and location estimation. As shown in Figure 4.2, the sensing/measurement device is placed at the second bus to record the current and voltage of the three-phase TLs. This process is performed via a 30-sample time window with the sampling frequency of 1.2 (kHz), and FFT is applied to each window at every step to extract the features.

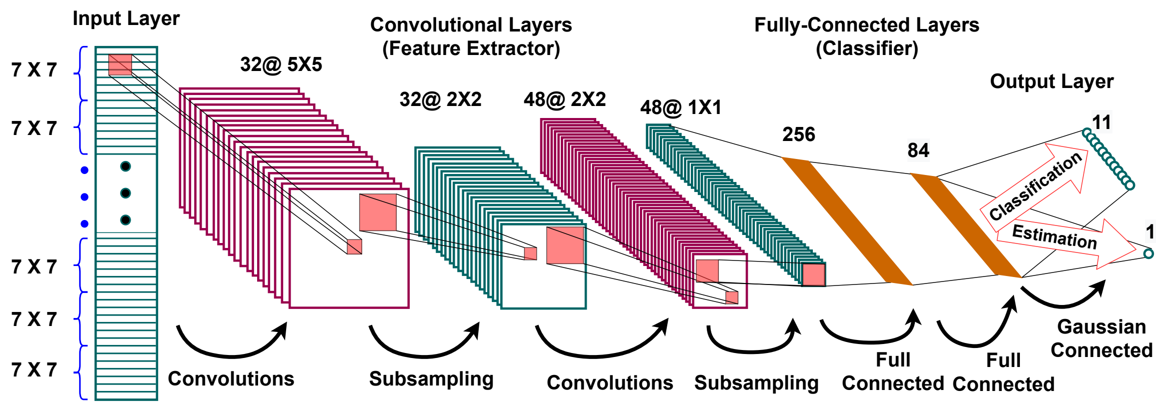


Figure 4.3: The structure of LeNet-5.

After normalizing the extracted features, the data points are fed to the fault diagnosis and location estimation networks. In the fault diagnosis process, 4 outputs are generated for each data point which are associated with phases A, B, C, and the ground G. When there is no fault, the value of all these 4 outputs are zero and by switching each output to one, the faulty state of that phase is detected. If not all outputs are zero, then at least two of them are one that shows the connection of those two outputs together (or to the ground). Therefore, in the output layer of the CNN, eleven states can occur including 0000 (or 1111), 0011, 0110, 1100, 1001, 0101, 1010, 0111, 1011, 1101, 1110. A parallel process using another CNN is performed for the location estimation of the faults. In this system, the input dataset is the same as that of the classification procedure; however, the outputs are the locations of the faults which are real numbers, not binary.

The detection, classification, and location estimation of TL faults are done by considering the 7-dimension input dataset divided into 7×7 small blocks. The features are extracted by striding over these blocks and performing the convolution computation for each window. For this purpose, one of the earliest pretrained CNNs called LeNet-5 [217] is designed by using Keras library [218, 219] in Python. LeNet-5 is chosen for this study because of its simple and straightforward architecture including 2 sets of convolutional and average pooling layers followed by a flatten layer, 2 fully connected layers, and ultimately a Softmax classifier. Other advanced CNNs such as ResNet [200] are not chosen in this study, because they have higher computational expenses than LeNet-5, and need more time and memory to make only slight changes in the accuracy level which are not worth it. In LeNet-5 structure, there is only one channel, the kernel size is 3×3, and the filter sizes for the first and second convolutional layers are 32 and 48, respectively. These parameters are determined experimentally with the purpose of achieving highest possible accuracy. To obtain high accuracy, ReLu activation function is used for all the layers except the output which takes advantage of Softmax function for classification and the linear regression function for location estimation. This architecture is depicted in Figure 4.3. As shown in this figure, the input data points are given by 7×7 matrices to the network in order to emulate the image behaviors, and the output layer consists of 11 neurons (equal to the number of classes) for classification or one neuron for the location estimation tasks using CNN.

To apply the transfer learning method to the TL fault diagnosis problem, the LeNet-5 network is trained with the dataset of a TL with length L which is initially equal to 100 (km). Then, the weights of convolutional and pooling layers of this network are saved as ".npy" files and a new LeNet-5 with the same architecture uses these files to apply the generated weights to the corresponding layers. Therefore, the weights of convolutional and pooling layers in the new LeNet-5 are frozen and equal to the first four trained layers of the initial LeNet-5. These 4 layers perform the feature extraction, and the rest of the layers, which are free to be updated based on the new datasets (for TLs with different length of $\frac{L}{8} = 12.5$, $\frac{L}{4} = 25$, $\frac{L}{2} = 50$, $2 \times L = 200$, $4 \times L = 400$, and $8 \times L = 800$), perform the classification and adaptation tasks. As it is shown in the next section, this process reduces the training time considerably as compared to the case that a new CNN is trained individually for each specific TL length.

4.5 Results and Discussions

This section presents the results of the proposed transfer learning-based CNN method. Simulations are run on a PC with Microsoft Windows 10. This PC uses an Intel Corei7-4710 MQ @ 2.50 (GHz) processor with 8 (GB) of RAM. Keras 2.3 library [218] with TensorFlow 2.0 backend [220] is used to design LeNet-5, and Scikit-learn library [221] is used for classification modules. An LeNet-5 is used with the architecture depicted in Figure 4.3 to classify the faults and estimate their locations.

To achieve reliable results, a statistical testing method is implemented and each experiment is performed 30 times. The values in the tables and figures in this section are all based on 30 iterations of running each analysis [222]. The accuracy fluctuations of these analyses are shown in the corresponding tables. This section is divided into two main subsections including classification and location estimation of TL faults.

4.5.1 Fault Classification

In this subsection, four different simulations are performed to prove the reliability and accuracy of the proposed approach. Different classification and clustering approaches are performed including (a) K-means algorithm for clustering without knowing the labels, (b) a dedicated CNN (a NN that is independently and specifically trained for one TL) for each specific TL length, (c) transfer learning method without fine tuning, and (d) transfer learning method with fine tuning which is the main focus in this study.

First, a K-means clustering algorithm is implemented to categorize the faults without knowing their true labels. Then, the results are compared with the true labels to calculate the accuracy which is reported in Table 4.4. The number of iterations needed for the convergence of TL length variants are between 15 to 23 in this algorithm [223].

In the second step, a dedicated CNN method is used for each TL length variant. The chosen CNN is LeNet-5 which produces acceptable results and is used for the comparison with the proposed transfer learning-based method. For this purpose, 70% and 30% of the data are used for training and testing, respectively.

Table 4.5: Classification Results of Various Lengths for TLs Using a Dedicated LeNet-5 NN.

Length (km)	**Accuracy** (%)	**Precision** (%)	**Recall** (%)	**F1 Score** (%)	**Training Time** (sec)
12.5	99.7	97.61	97.54	97.57	3169.12
25	99.69	97.57	97.45	97.50	3493.55
50	99.46	95.84	95.58	95.70	3441.43
100	99.42	95.5	95.23	95.36	3167.27
200	99.15	93.38	93.08	93.22	2783.07
400	99.1	93.08	92.56	92.81	3217.381
800	98.62	89.85	87.83	88.82	3168.63

Table 4.5 demonstrates the accuracy, precision, recall, F1 Score, and training time for seven different lengths of TLs without using the transfer learning technique. The parameters of performance evaluations are defined as Equations (4.2)-(4.5).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.2}$$

$$Precision = \frac{TP}{TP + FP} \tag{4.3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.4}$$

$$F1 = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{4.5}$$

The provided results are obtained after 64 epochs and with 3×10^{-4} learning rate for the "adam" optimizer.

In the third step, a transfer learning-based method without fine tuning process with 64 epochs is used. In other words, all layers of LeNet-5 hold on to the weights of pretrained layers and become frozen. The purpose of this step is to show the effect of fine tuning process on the result of TL fault classification. Table 4.6 shows the results of this step.

Table 4.6: Classification Results for Various Lengths of TLs Using LeNet-5-based Transfer Learning Method <u>without</u> Fine Tuning Process

Length (*km*)	**Accuracy** (%)	**Precision** (%)	**Recall** (%)	**F1 Score** (%)	**Training Time** (*sec*)
12.5	90.80	74.06	50.59	60.11	1101.04
25	90.96	75.3	52.62	61.94	1111.66
50	90.89	72.49	55.37	62.78	1103.21
200	91.19	74.25	59.7	66.18	1123.83
400	90.90	71.99	56.18	63.10	1195.01
800	90.11	65.07	42.35	51.30	1193.73

In the fourth step, the proposed method is implemented using transfer learning method and LeNet-5 with fine tuning approach. Table 4.7 demonstrates the fault classification results of the transfer learning method for various lengths of TLs. In order to perform a fair comparison among the transfer learning-based and non-transfer learning-based methods, the number of epochs (64) and learning rate (3×10^{-4}) for "adam" optimizer) remain the same in all of them.

According to the results given in Tables 4.5 and 4.7, the training time of the transfer learning-based method is almost <u>half</u> of the training time of the dedicated CNN methodology (which is a LeNet-5 NN without transfer learning and fine tuning), while the accuracy values are almost similar. This result is achieved with negligible loss in accuracy level. As it is shown in Figure 4.4, the difference between the proposed transfer learning-based method accuracy and the dedicated CNN method is less than 0.5 % for all various lengths of TLs. Figure 4.5 shows the comparison of the training time between the transfer learning-based and non-transfer learning-based approaches.

Table 4.7: Classification Results for Various Lengths of TLs Using LeNet-5-based Transfer Learning Method with Fine Tuning Process

Length (km)	**Accuracy** (%)	**Precision** (%)	**Recall** (%)	**F1 Score** (%)	**Training Time**(sec)
12.5	99.48±0.04	95.9±0.3	96.0±0.3	96.0±0.3	1307±46
25	99.36±0.04	95.5±0.3	95.1±0.3	95.3±0.3	1414±46
50	99.26±0.03	94.3±0.2	93.8±0.3	94.0±0.2	1328±49
200	99.01±0.03	92.2±0.2	91.6±0.2	91.9±0.2	1406±43
400	98.82±0.04	91.2±0.2	89.8±0.3	90.5±0.2	1371±39
800	98.29±0.05	87.4±0.3	80.9±0.4	84.1±0.3	1483±47

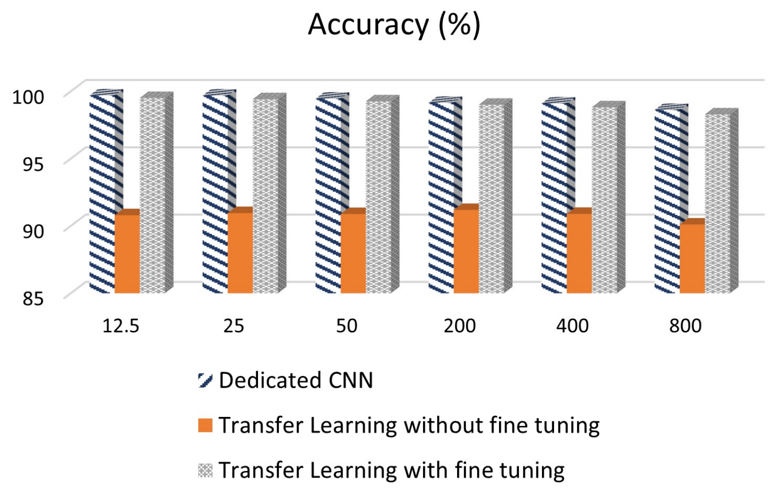


Figure 4.4: Accuracy comparison between three methods including using the transfer learning method with and without fine tuning process, and dedicated CNN for classification of TL faults.

As it is clear, transfer learning decreases the training time of the classification to less than half of the training time of the dedicated CNN approach. The k-means clustering technique generates much lower accuracy results (in average 85%) which is another proof for the significance and reliability of using transfer learning method when labels are missing or inadequate.

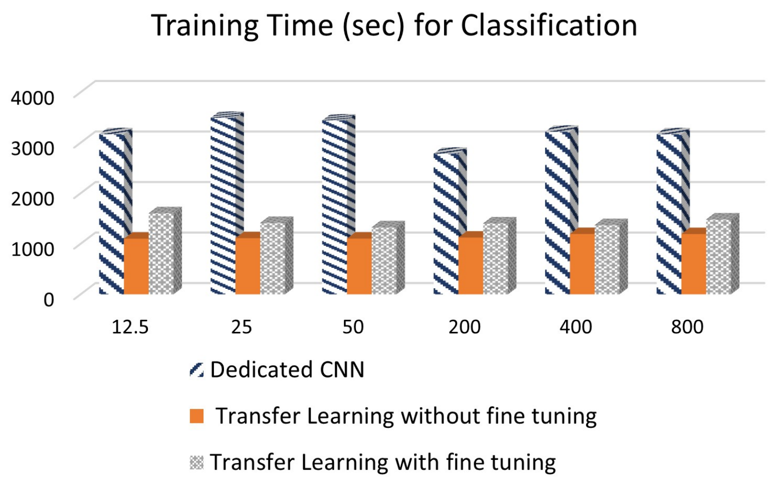


Figure 4.5: Training time comparison between three methods of using the transfer learning with and without fine tuning process, and the dedicated CNN for classification of TL faults.

4.5.2 Fault Location Estimation

To evaluate the accuracy of location estimation methodologies, the mean square error (MSE) which is defined as (4.6), is calculated for each length of TL individually in the three considered approaches: A dedicated CNN for each length variant of TLs, and with and without the fine tuning process in transfer learning-based technique. The number of epochs is 32 for all of these approaches.

$$MSE = \sum_{i=1}^{n} (x_i - y_i)^2 \tag{4.6}$$

Tables 4.8 - 4.10 indicate the MSE values and the training times of these three approaches in locating the TL faults. Based on these tables, it can be concluded that the proposed transfer learning method decreases the training time of faults location estimation to one-fourth of the training time of the dedicated CNN approach (without transfer learning).

Table 4.8: Location Estimation Results for TL Length Variants Using Dedicated LeNet-5 NN

Length (km)	**# Epochs**	**MSE** (m^2)	**Training Time** (sec)
12.5		$1.55\times10^{-}1$	2687.36
25		$1.03\times10^{-}1$	2645.50
50		$1.13\times10^{-}1$	2948.40
100	32	$8.73\ \times10^{-}2$	2842.66
200		$8.03\times10^{-}2$	2835.40
400		$5.15\times10^{-}2$	2668.42
800		$5.52\times10^{-}2$	2558.84

Table 4.9: Location Estimation Results for TL Length Variants Using LeNet-5-based Transfer Learning <u>without</u> Fine Tuning

Length (km)	**# Epochs**	**MSE** (m^2)	**Training Time** (sec)
12.5		$3.21\times10^{-}1$	550.86
25		$3.01\times10^{-}1$	562.62
50	32	$2.98\times10^{-}1$	561.61
200		$2.48\times10^{-}1$	555.35
400		$2.59\times10^{-}1$	550.65
800		$2.80\times10^{-}1$	537.17

Furthermore, although the fine tuning process takes negligible amount of time, it reduces the overall error considerably. Based on Figure 4.6, the MSE differences between these two methodologies (dedicated CNN and transfer learning technique with fine tuning) for each length variant is less than 0.1 (m^2) which proves the reliability of the proposed method. One general conclusion that can be made from Figures 4.4 and 4.6 is that as the difference between the TL lengths of the source and target datasets increases, the transfer learning-based accuracy decreases and the validation loss for location estimation increases which is expected considering the feature differences. Figure 4.7 shows that transfer learning with fine tuning method decreases the training time of the TL fault location estimation to less than <u>one-fourth</u> of the state where a dedicated CNN is applied.

Considering the test time of detection, classification, and location estimation of TL faults, it is concluded that the cumulative test time does not exceed 0.3 (sec) which satisfies the IEEE standard specifications [56, 193].

Figs. 4.8 shows the variations of accuracy for 6 different TLs using the transfer learning methodology, and compares the training and validation accuracy values for the detection and classification steps. In Figure 4.9, the loss variations of TLs for

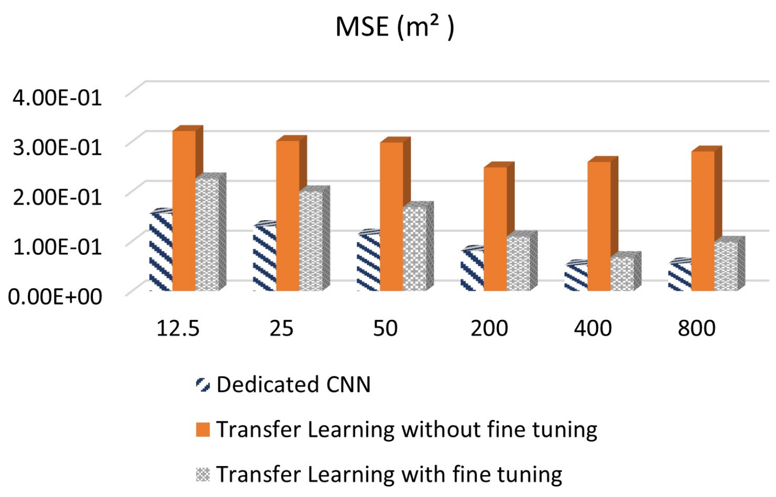


Figure 4.6: MSE comparison between two states of using the transfer learning method and without its usage.

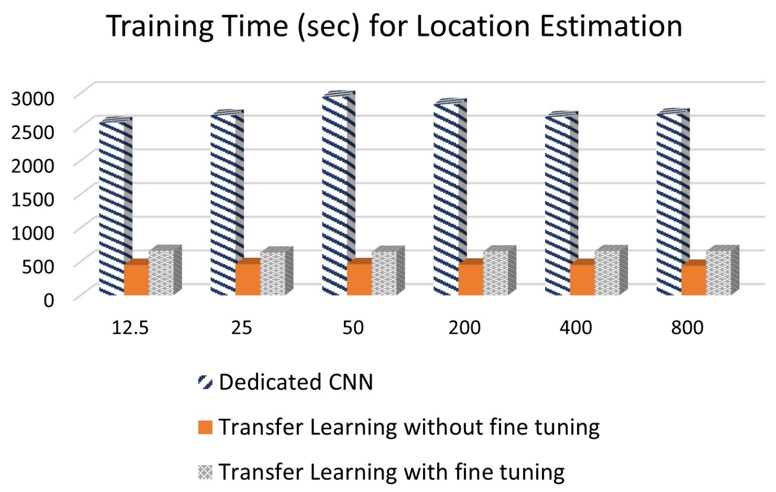


Figure 4.7: Training time comparison between two states of using the transfer learning method and without its usage for location estimation of TL faults.

Table 4.10: Location Estimation Results for TL Length Variants Using LeNet-5-based Transfer Learning Method with Fine Tuning

Length (km)	**# Epochs**	**MSE** (m^2)	**Training Time** (sec)
12.5	32	$2.26\times10^{-}1$	654.99
25		$2.00\times10^{-}1$	654.45
50		$1.69\times10^{-}1$	643.90
200		$1.10\times10^{-}1$	649.84
400		$6.79\times10^{-}2$	632.82
800		$9.88\times10^{-}2$	658.35

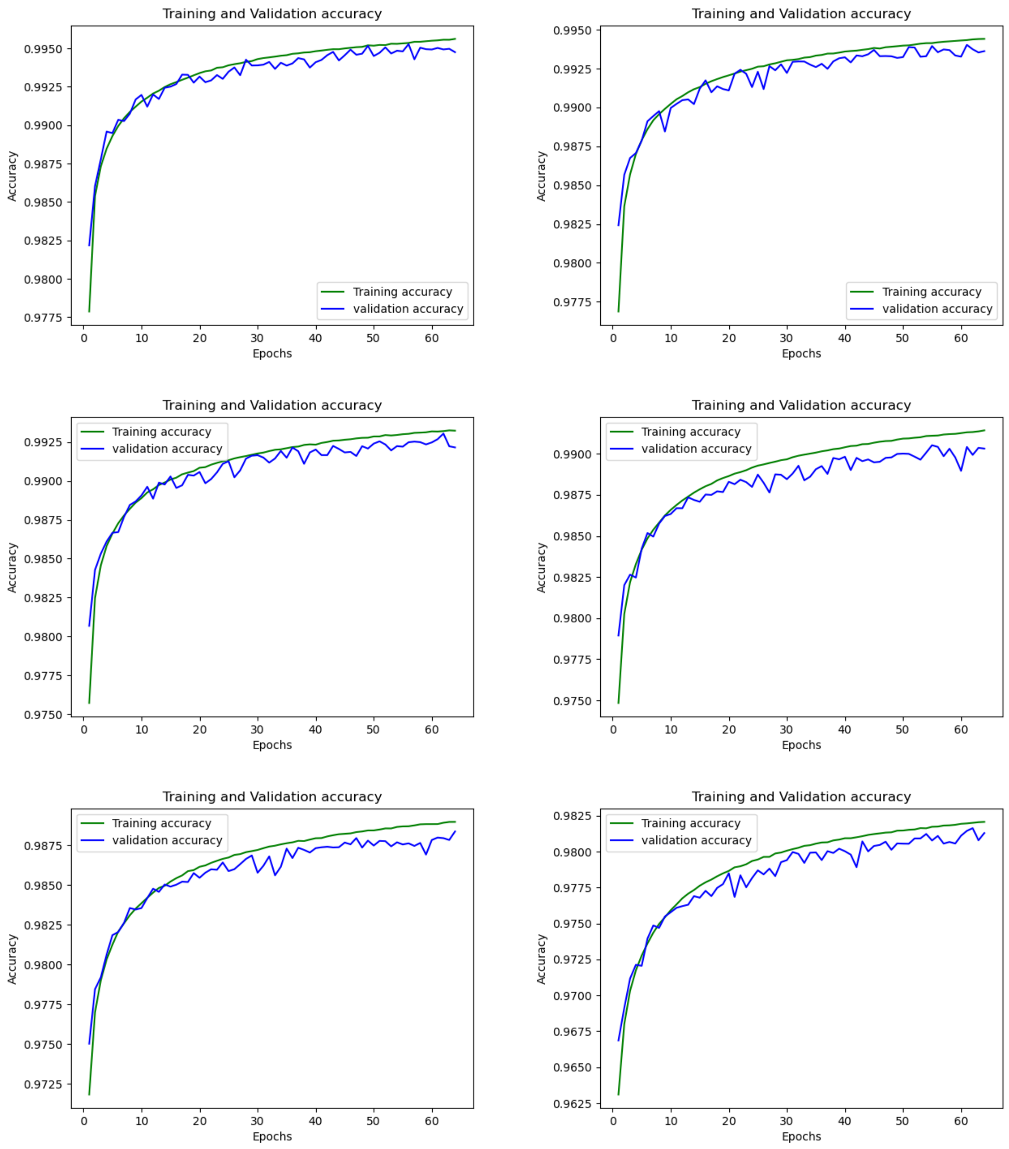


Figure 4.8: Training and validation accuracy of TLs with the lengths of a. L/8=12.5 (km) b. L/4=25 (km) c. L/2=50 (km) d. 2L=200 (km) e. 4L=400 (km) f. 8L=800 (km) based on LeNet-5 using transfer learning method with fine tuning for classification of TL faults.

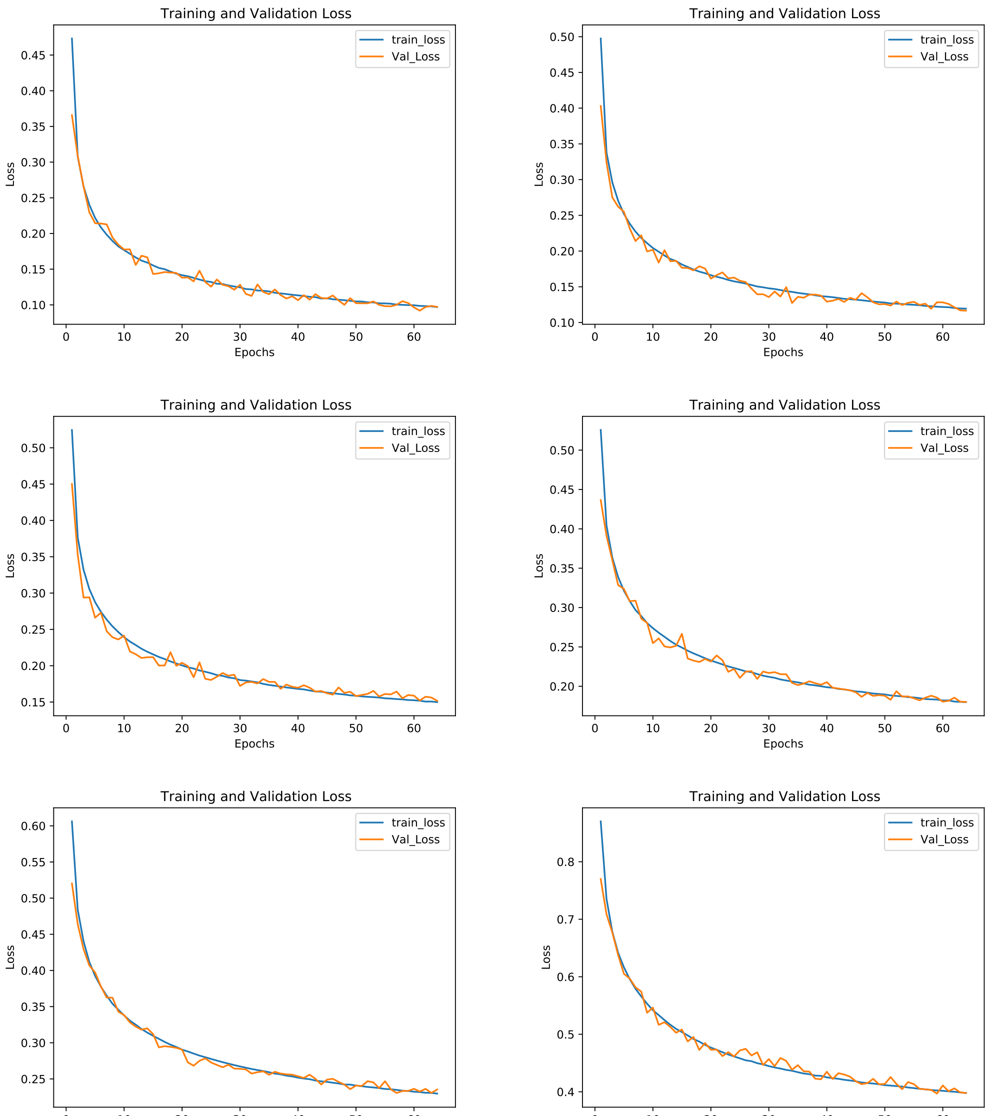


Figure 4.9: Training and Validation loss of TLs with the lengths of (a.) L/8=12.5 (km) (b.) L/4=25 (km) (c.) L/2=50 (km) (d.) 2L=200 (km) (e.) 4L=400 (km) (f.) 8L=800 (km) based on LeNet-5 using transfer learning method with fine tuning for location estimation of TL faults.

Table 4.11: Noise Tolerance (Validation Accuracy) of the Proposed Method

Noise	**Noise Free**	**White**	**Salt&Pepper (20%)**	**Salt&Pepper (50%)**
Accuracy (%)	98.2	98.9	97.9	94.9
Training Time (sec)	1384	1371	1368	1389

training and validation steps are shown for all 6 variants of the TLs. These figures prove the reliability and efficiency of the proposed approach in location estimation of TL faults.

To show the robustness of the presented approach, its accuracy is tested by adding white (Gaussian) noise with 0.05 standard deviation, and Salt & pepper noise with two different percentage to the TL datasets. Table 4.11 shows the accuracy results after 10 epochs with the batch size of 128.

4.6 Summary

The need for a general solution that is low cost, fast, reliable, and compatible with more than one type of TLs has become viral more than before. In this chapter, a generalized, high-speed, and accurate approach based on the transfer learning methodology is presented for the first time to diagnose TL faults and estimate their locations. This approach makes use of a pre-trained CNN called LeNet-5 which is trained based on the dataset of a TL with the length of $L = 100(km)$, and then the weights of feature extractor layers are frozen and transferred to a new similar CNN. In the second CNN, the fully connected layers update their weights based on the new datasets specific to each length variant of TLs to produce more accurate results. The simulation results show that the training time of the transfer learning-based approach is <u>half</u> of the required training time of the dedicated CNN (without using transfer learning) approach for the classification of TL faults, and for the location estimation of TL faults the training time of the proposed method is <u>one-fourth</u> of the required training time of the dedicated CNN (without using transfer learning). Such remarkable reduction in training time proves the reduction in computational load and therefore, memory usage, which are considered to be significant issues for the associated online datasets. Moreover, the accuracy of the fault classification and the MSE of the fault location estimation are almost similar to those given by a specifically trained (dedicated) CNN for each length of TLs. These results prove the generality and reliability of the proposed transfer learning methodology for the TL fault diagnosis problems.

CHAPTER 5

INSULATOR FAULTS DETECTION IN AERIAL IMAGES FROM HIGH-VOLTAGE TRANSMISSION LINES USING TRANSFER LEARNING TECHNIQUE AND IMAGE AUGMENTATION

5.1 Introduction

Deep learning methods have shown great promises in the intelligent inspection of high-voltage transmission lines. The expansion of power systems including transmission lines has brought the problem of insulator fault detection into account more than before. In this chapter, a novel transfer learning framework based on a pre-trained VGG-19 deep convolutional neural network (CNN) is proposed to detect broken insulators in aerial images. In this procedure, first a well-known large imagery dataset called ImageNet is used to train VGG-19 and then the knowledge of this deep CNN is transferred, and by using a few layers for fine tuning purpose, the newly built deep CNN is capable of distinguishing the corrupted and intact insulators. This method is able to diagnose these faults using the aerial images taken from transmission lines in different environments. The original dataset used in this study is the Chinese power line insulator dataset which is an imbalanced dataset and includes only 3,808 insulator images. Therefore, a random image augmentation procedure is applied to generate a more reliable dataset with 16,720 images. This new dataset not only performs better than the original dataset in providing the accuracy results, but also is a balanced dataset which is more reliable than imbalanced ones. Training the deep CNN using the new generated large dataset gives more power to the system for detecting the corrupted insulators in different situations such as rotated, dark, and blurry images with complex background. The results of this study are compared with

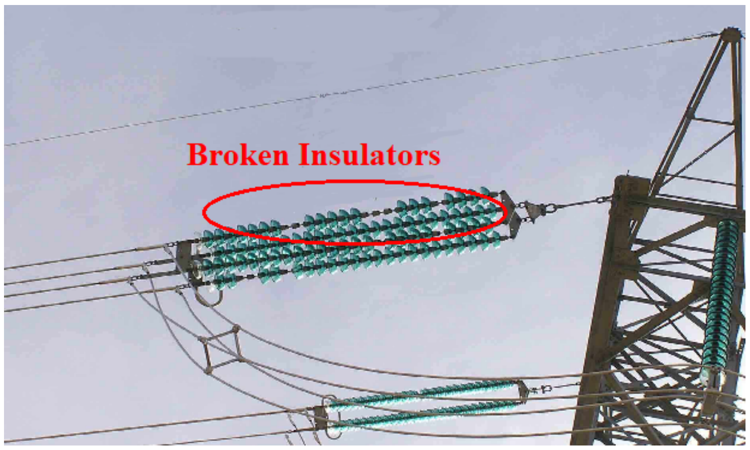


Figure 5.1: Missing faults in glass insulators of transmission lines.

various existing benchmarks in the literature and some other implemented ones to prove the reliability and efficiency of the proposed method.

Increasing demand for electrical energy results in expansion of power systems including transmission lines (TL). Considering the vital role of insulators in TLs for mechanical support and electrical insulation during the power grid operations, and their proneness to faults and breakage, intelligent methodologies are proposed to enhance their safety and reliability [224, 225, 226]. Due to insulators' exposure to outdoor environments for long period of time, they face missing faults as shown in Figure 5.1.

Insulator faults are variant and random and their occurrences interrupt the safety and stability of the entire TL operations, and therefore impose tremendous economic losses. To guarantee the safety and stability of TLs and intact operation of power grids, insulator fault detection and intelligent inspection are considered as salient tasks [227, 224].

Maintaining insulator faults using traditional manual patrol is inefficient and time consuming, and waste a lot of human resources. Hence, it is gradually replaced by unmanned aerial vehicle (UAV) patrol which is discussed in [228, 229]. In this new process, the workers do not require to investigate TL insulator faults by using telescopes. However, the complexity and variability of UAV application scenarios cause several challenges in the autonomous detection of insulator faults [224]. Therefore, artificial intelligence-based methodologies became viral for such problems.

There are many studies on insulator fault identification using aerial images and traditional or new image processing methodologies. In the first step, traditional image processing algorithms classify insulator images into classes with specific features such as texture, color, shape, etc. Then they take advantage of matching algorithms to implement fault detection procedures. The drawback of these methodologies is the fact that the features are designed manually, and therefore, they are inefficient in complicated power grids with a variety of image features. Besides, correlation features of insulators in aerial images are not clear and the accuracy of traditional image processing algorithms is highly dependent on these feature [230, 231, 232, 233].

Recently, the tremendous progress in artificial intelligence theory has led to a noticeable progress in image processing methods which are based on deep neural networks. CNNs are the most common deep neural networks which perform pattern recognition and object detection tasks, and are able to extract image features automatically and learn under various environmental conditions. These CNN-based techniques overcome the limitations of the traditional image processing approaches and conclude higher performance and better accuracy in object detection tasks [234, 151, 152]. Although CNNs have completely dominated the image processing area in recent years, and their results are a lot better than traditional methodologies, they still have some drawbacks such as requiring big datasets for training, being time consuming, having high computational load, and being computationally expensive in general. Therefore, some advanced technique such as transfer learning approaches are proposed, which make the deep CNNs capable of generating results faster and transferring the knowledge from an already trained CNN to a new CNN which has common features with the base dataset for the purpose of saving time and resources.

Transfer Learning is a machine learning approach whereby a model gets trained and developed for one dataset and is then re-used for a second related one. It describes the situation whereby a learnt matter in one setting is exploited to improve optimisation in another one. Transfer learning is generally used when a new dataset is smaller than the primary dataset, which is used to train the base model.

There are several studies in the literature that use CNNs for fault diagnosis of TLs. The study [154] proposes a self-attention CNN framework and a time series image-based feature extraction model for fault identification and classification of TLs with length of 100 (km) using a discrete wavelet transform (DWT) for denoising the faulty voltage and current signals. The work [155] proposes a customized CNN for fault detection and classification of 50 (km) TLs integrated with distributed generators. In [156], a machine learning-based CNN for TLs with length of 280 (km) is proposed to perform fault detection and classification using DWT for feature extraction.

Shiddieqy et al. [201] propose another methodology that investigates all features of the TL faults to generate various models for robust fault detection. They use various AI methods including CNN to obtain a 100% detection accuracy. In [158], a scheme to detect and categorize faults in power TLs with length of 200 (km) using convolutional sparse auto-encoders is proposed. This approach can learn the extracted features from the dataset of voltage and current signals, automatically, for fault detection and classification.

There are multiple studies that apply transfer learning methods to various datasets and applications [202, 203, 204, 205, 206, 207]. In [208], the authors show how to transfer deep CNN knowledge for real-time dataset classification. The study in [209] proposes an intelligent method based on a deep convolutional transfer learning network which detects the dynamic system faults using an unlabeled dataset. Shao et al. [210] present an intelligent fault diagnosis approach for a rotor-bearing system that works based on a modified CNN with transfer learning. Li et al. [211] also propose a deep adversarial transfer learning network to investigate new and unlabeled emerging faults in rotary machines.

In this study, the model VGGNet is used as a basic model to get trained on a base dataset (ImageNet), and then re-used to learn/transfer features to be trained on an insulator imagery dataset. Taking advantage of the initial training, transfer learning allows us to start with the learnt features on the ImageNet dataset, and then tune the weights and possibly the structure of the base model to match the new dataset/task instead of starting the learning process on the new data from scratch using random weights initialization [235]. There are also similar studies which perform transfer learning on various datasets using VGGNet CNNs. Huang et al. [236] used ImageNet to pre-train VGG19 network for DenseBox initialization, which is defined as a unified end-to-end fully CNN framework for object detection. Li et al. [237] adjust the VGG-19 pre-trained on ImageNet for hierarchical convolutional feature extraction of visual object tracking images. Gatys et al. [238] take advantage of VGG-19 network for object recognition of texture synthesis.

In this work, we utilize transfer learning method to develop a fault detection system for distinguishing intact and broken insulator images. The contributions of this work are as follows:

1. Implementing an image augmentation procedure to generate a large labeled dataset for insulator images classification based on the Chinese Power Line Insulator Dataset (CPLID).
2. Balancing the CPLID to generate reliable results by performing image augmentation with different portions on broken and intact classes of insulator images.
3. Proposing a transfer learning technique for the generated reliable dataset using VGG-19 CNN and ImageNet dataset whose performance outperforms than the existing benchmarks.

The rest of this chapter is organized as follows. Section 5.2 reports preliminary concepts used in this study including CNNs, transfer learning technique, ImageNet dataset, and image augmentation procedure for the CPLID. Section 5.3 describes the proposed methodology and describes the architecture of the used CNNs. Section 5.4 presents the simulation results and analysis. Finally, Section 5.5 presents the conclusion of this chapter.

5.2 Preliminaries

5.2.1 Convolutional Neural Network (CNN)

CNNs are capable of learning hierarchical features independently from inputs, and therefore, they are widely used for imagery dataset problems. The structure of CNNs empowers them to have the minimum need for data pre-processing, and handle high dimensional datasets faster with more details in comparison to most of the artificial neural networks [212, 145]. In general, CNNs consist of three types of layers: convolution layer, pooling layer, and fully connected layer. Convolutional and pooling layers incorporate convolution blocks which are stacked for the purpose of feature extraction. Fully connected layers are built as classifiers and the role of the output layer (a fully connected one) is to perform the classification or regression tasks.

The CNN architectures are made of local receptive fields, shared weights, and the pooling operation. Taking advantage of local receptive fields, meaning that only a small focused area of the input data is connected to each node in a convolutional layer, is one of the main advantages of CNNs. Because of this feature, the number of parameters is reduced considerably in CNN which in turn reduces its training computational load [213].

Numerous studies are performed in image classification problems using CNNs to achieve better performances. CNNs are also used for protecting the TLs in various aspects such as their fault identification and distinguishing the intact waveforms from corrupted ones in waveform images. These studies are divided into two different categories: The methodologies with a focus on image-based datasets recorded from outdoor TLs [151, 152, 153, 224], and those that consider time-series voltage and current waveforms recorded from generators and fed to CNNs as blocks of data points. The methodology proposed in this study belongs to the first category.

In this work, a deep CNN introduced by Visual Geometry Group (VGG) is used which is called VGG-19. Two successful architechture of VGG are VGG-16 and VGG-19 which perform well on ImageNet dataset. VGGNets are an improved version of AlexNet [239], which takes advantage of large kernel-sized filters besides various small kernel-sized filters, and results in 13 and 16 convolutional layers for VGG-16 and VGG-19, respectively [150, 240].

5.2.2 Transfer Learning

Transfer learning is a methodology which provides the capability of transferring the learnt knowledge from one trained neural network to another one. The emerging concept of transfer learning besides deep learning structures shows remarkable improvements in the performance of image classification and pattern recognition. Transfer learning consists of two general steps: First, the process of training a base neural network on a source dataset, and second, transferring the learnt features and knowledge to a new neural network to help the training process of the new related (target) dataset. The second step also can be achieved by only adding some extra layers for the purpose of adjustment and tuning to the base neural network. Two main definitions are used in transfer learning, namely, domain and task which are described below.

- Definition 3 (Domain [214, 215]). A domain $\mathcal{D} = \{\mathcal{X}, \mathcal{P}(X)\}$ includes two elements, namely, a feature space $\mathcal{X}$ and a marginal probability distribution $\mathcal{P}(X)$ where $X = \{x_i\}_{i=1}^{n} \in \mathcal{X}$ is a dataset in which every $x_i \in \mathbb{R}^D$ comes from this domain.

- Definition 4 (Task [214, 215]). A task $\mathcal{T} = \{\mathcal{Y}, \mathcal{P}(Y|X)\}$ includes two elements given a domain $\mathcal{D} = \{\mathcal{X}, \mathcal{P}(X)\}$, where $\mathcal{Y}$ stands for the label space and $\mathcal{P}(Y|X)$ is the conditional probability distribution in which $Y = \{y_i\}_{i=1}^{n}$ stands for the label vector of X with $y_i \in \mathcal{Y}$ as the label of x_i.

Transfer learning includes two domains, namely, a source domain $\mathcal{D}_s = \{\mathcal{X}_s, \mathcal{P}_s(X_s)\}$ and a target domain $\mathcal{D}_t = \{\mathcal{X}_t, \mathcal{P}_t(X_t)\}$ where $\mathcal{X}_s$ and $\mathcal{X}_t$ present the feature spaces of the source and target domains, respectively, and $\mathcal{P}_s(X_s)$ and $\mathcal{P}_t(X_t)$ show the marginal probability distribution of them. Based on definitions 3 and 4, the concept of transfer learning is presented below.

- Definition 5 (Transfer Learning [215]). Considering the source domain $\mathcal{D}_s$, learning task $\mathcal{T}_s$, target domain $\mathcal{T}_s$, and learning task $\mathcal{T}_t$, the purpose of transfer learning is to boost the performance of target predictive function $f_t(.)$ in $\mathcal{D}_t$ by inducing the knowledge from $\mathcal{D}_s$ and $\mathcal{T}_s$ while $\mathcal{D}_s \neq \mathcal{D}_t$ or $\mathcal{T}_s \neq \mathcal{T}_t$.

In the case of $\mathcal{T}_s = \mathcal{T}_t$, a general sub-problem of transfer learning emerges, which is called domain adaptation. In this study, the relationship between the source and target data is not the domain adaptation because the understudy datasets are both images but different ones [216].

The transfer learning technique helps deep learning methods by distributing the learned features across different learning applications. Transfer learning focuses on the deep learning training step to promote its ability in common feature extraction and adaptation among multiple datasets of a comparable problem. Transfer learning uses the pre-trained layers on a source task to perform a target task. For this purpose, a pre-trained neural network, whose fully connected layers are cut off, is used and its convolutional and pooling layers become frozen (their weights are not updated) to perform the feature extraction. To adapt the given pre-trained neural network to the target dataset and efficient target classification, the fully connected layers (classifier section) require to update their weights and build the desired model.

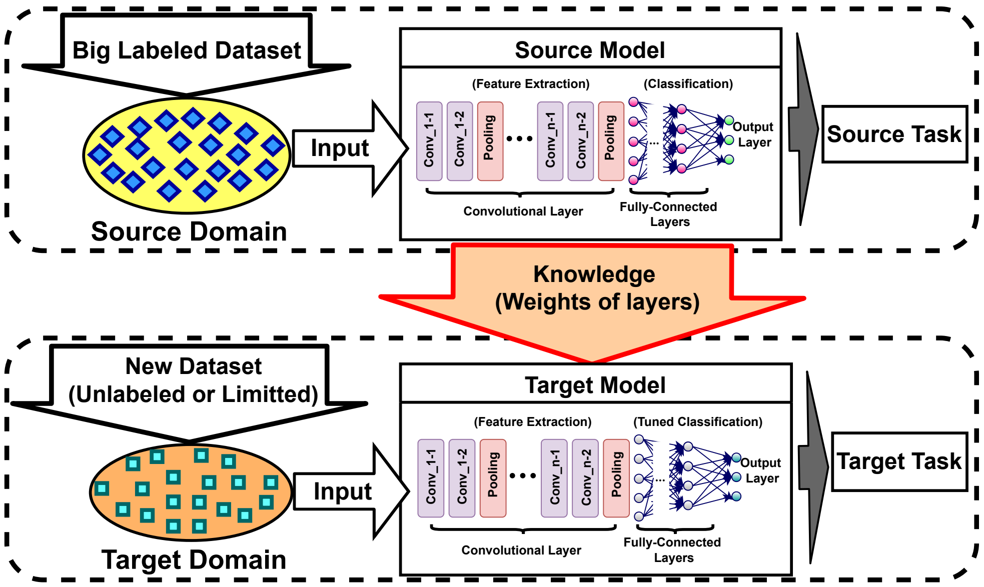


Figure 5.2: A schematic of typical transfer learning model.

There are two distinct concepts in transfer learning including feature extraction and fine tuning. In the first one, the fully connected layers are eliminated and based on the target dataset, the adjusted fully connected layers are integrated with the frozen pre-trained layers. In the fine tuning process, the structure of the fully connected layers from the pre-trained model is stored and only their weights are updated. Fine tuning is the process that is used in this study for the purpose of adapting the ImageNet extracted features to the under study insulator dataset. Figure 5.2 depicts the basic idea of transfer learning representing two different datasets which have resemblance. These datasets are fed into source and target deep neural networks. The knowledge is transferred to the target model to perform the training of target task at a higher pace.

The transfer learning method is used in this study for TL insulator fault diagnosis problems to detect the rare occurrence of failures which are difficult or impossible to be labeled. It should be noted that a transfer learning-based CNN has not been used before for TL insulator fault detection, and is proposed and studied in this study for the first time.

5.2.3 ImageNet

ImageNet is a large-scale ontology of images built upon the hierarchical structure provided by WordNet [241]. ImageNet aimed to collect the majority of the 80,000 synsets of WordNet with an average of 500-1000 clear and high resolution images, and this resulted in tens of millions of annotated images categorized by the semantic hierarchy of WordNet [242, 241]. ImageNet provides the most comprehensive and diverse coverage of the image world. The current 12 sub trees consist of a total of 3.2 million cleanly annotated images spread over 5247 classes. Over 600 images are collected for each synset of ImageNet. Nowadays, ImageNet consists of over 15 million annotated images. Some CNNs have shown great promises in classifying images in the ImageNet dataset into its corresponding categories, namely, AlexNet, VGGNet (used in this study), CaffeNet, ResNet, and etc. [243]. Based on a review reported by Cheplygina et al. [244], ImageNet is the most commonly used dataset for transfer learning based image analysis and classification methods [244, 242, 240]. In this study, ImageNet is used as a source dataset for the proposed transfer learning technique to perform the feature extraction for insulator images fault classification.

5.2.4 Insulator Dataset Generation Using Data Augmentation

In classical discriminative examples such as the study of broken insulators vs intact ones in this work, the image recognition software has to overcome the issues of viewpoint, lighting, occlusion, background, scale, and more. The task of data augmentation is to prepare these translational invariances for consideration of the dataset such that the resulting models can perform better despite the existing challenges. The concept of having a larger dataset results in better deep learning models and performances is a generally accepted nation [245, 246]. However, collecting large datasets can be a complicated task because of the need for manual efforts to collect and label data especially in the field of image processing. The existing images from TL insulators are not an exception to this fact, and by searching in google images it can be observed that there are not enough broken or defective insulator aerial images [245].

To the best of our knowledge, there is no standard dataset for TL insulator faults that consists of broken and intact insulators. Therefore, because the aerial images of insulator faults are rare and almost impossible to collect, to obtain adequate insulator faults images, the simulated insulator faults samples are created on the basis of the Chinese Power Line Insulator Dataset ('CPLID') [247]. In these images which are created by using the software Photoshop in [224], the normal insulator strings are erased and replaced by their nearby pixels. The process of using Photoshop to generate the desired dataset is so time consuming and requires a lot of efforts. Therefor, only 248 faulty images are produced with this procedure while the number of images including intact insulators are 3,560 [224].

Having this dataset from [224], image augmentation is the approach which helps to generate a large sufficient dataset with $248 \times 9 + 248 = 2,480$ faulty insulator images and $3,560 \times 3 + 3,560 = 14,240$ intact insulator images in variant backgrounds. The image augmentation process is done randomly such that from each broken insulator image, nine augmented images, and from each intact insulator image, three augmented ones are generated with arbitrarily various features including brightness, angles, zooming level, quality, and etc. Therefore, this augmented insulator image dataset includes 16,720 images. Figures 5.3 and 5.4 show some examples of these images.

5.3 Proposed Method

This study takes advantage of transfer learning technique to improve the fault classification results for TL insulators dataset. To perform this approach, a common deep CNN called VGG-19 is used for the purpose of feature extraction. This CNN is consisted of 19 layers out of which 13 layers are convolutional.

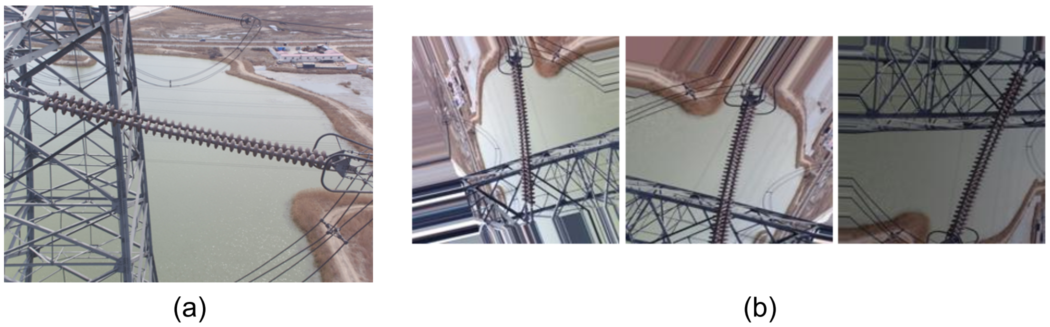
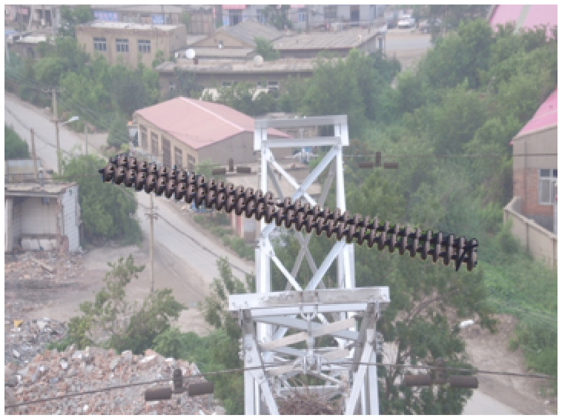


Figure 5.3: Images of an intact insulator. (a) The original image from CPLID, and (b) three augmented images generated from the original image.

Although the benchmark CNNs for ImageNet dataset include numerous layers, CNN models used in fault diagnosis methods are relatively shallow. The depth of a CNN model for fault diagnosis is almost up to 5 hidden layers only because of the simple structure of the existing faults. This can underestimate the effectiveness of CNN models in fault diagnosis. Regardless of this difference among the required CNN models' architectures, the volume of labeled samples is always limited in fault diagnosis applications, and it is complicated to train a deep CNN model without having a tremendous amount of well-organized datasets such as ImageNet. Feature transferring by taking advantage of trained deep CNN models as a feature extractor has attracted a lot of attention these days and has become an alternative solution to these sort of limitations. By reusing the knowledge (finalized weights) of pre-trained networks as the feature extractor, the deep CNNs can perform well on the small datasets in other domains [248]. VGG-19 is one of the most common architectures which is used in transfer learning-based methodologies for fault diagnosis in the literature [236, 237, 238].

In this study, the transfer learning technique is proposed for TL insulator fault diagnosis by reusing the pre-trained VGG-19 on ImageNet dataset as the feature extractor. The structure of VGG-19 is shown in Figure 5.5. The notion of "Conv3-64" implies that the filter (kernel) of the layer is 3×3, and its depth is 64. The pooling


(a)

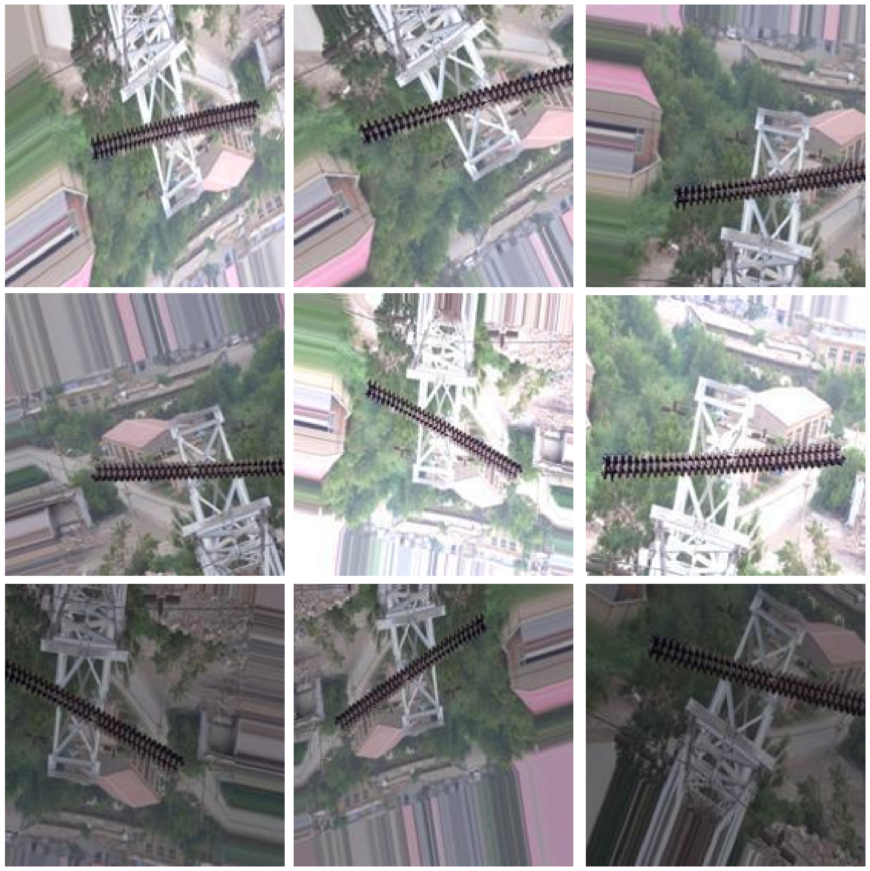

(b)

Figure 5.4: Images of a faulty (broken) insulator. (a) The original image from CPLID, and (b) nine augmented images generated from the original image.

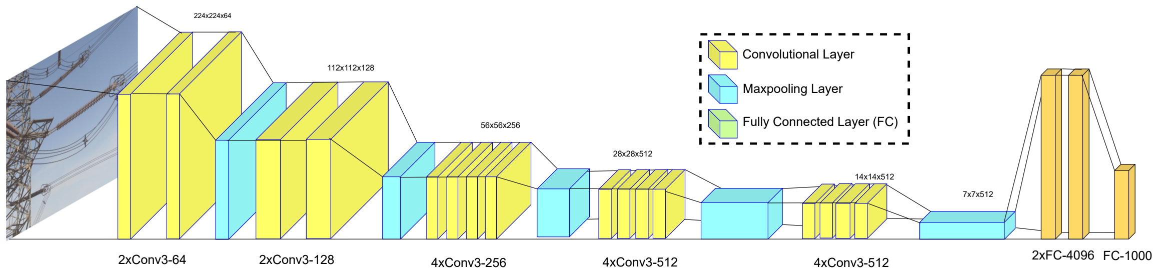


Figure 5.5: The architecture of VGG-19 CNN.

layers in this architecture are maxpooling which store the maximum values to be fed into sebsequent layers.

To take advantage of transfer learning technique, the VGG-19 is implemented and transferred, and its layers remain frozen during the training process of insulator fault diagnosis system. All the convolutional and maxpooling layers and the first fully connected layer perform as feature extraction layers. Then, the weights of the two fully connected layers, which are located at the end of the above-mentioned feature extraction layers, are free to become updated for the purpose of fine tuning and adaptation to the new dataset. The last fully connected layer is a Softmax classifier to adjust the pre-trained VGG-19 to the insulator dataset for fault identification. Two datasets are used in this experiment including ImageNet for the purpose of VGG-19 pre-training and feature extraction, and the augmented insulator dataset including 13,160 aerial images of insulators in two classes of defective (broken) and intact. Because the size of images in ImageNet dataset are 224×224, we resize all the images in the augmented insulator dataset to this size using Python Numpy and PIL libraries. Accordingly, we are able to feed the new images to the same pre-trained VGG-19.

The VGG-19 CNN is implemented using Keras, and the weights of convolutional, maxpooling, and first fully connected layers are restored. Then, the two newly added fully connected layers including the Softmax classifier are randomly initialized.

The weights of other layers of VGG-19 remain frozen and untrained during the new training process, which is called fine tuning.

Using the above mentioned datasets, we could distinguish aerial images of broken and intact insulators with the accuracy of 99.93% which is a reliable result. In the next section, we compare the proposed methodology with the existing benchmarks on these datasets. To have a descent comparison between the outcome of existing CPLID dataset and the generated large dataset, we trained the transferred CNN with these two datasets and test them with a common variant dataset. We also generated imbalanced large dataset to prove the effect of making this dataset balanced in the next section.

5.4 Simulation Results and Comparisons

This section describes the results of the proposed transfer learning-based method for classification of insulator images into two categories of broken and intact insulators. Simulations are run on a PC with Microsoft Windows 10. This PC uses an Intel Corei7-4710 MQ @ 2.50 GHz processor with 8 GB of RAM. The programming language used in this study is Python version 3.8. Keras 2.4 library [218] with TensorFlow 2.3 backend [220] is used to design VGG-19, and Scikit-learn library [221] is used for classification modules. In this study, we compared our proposed methodology using transfer learning and image augmentation techniques with five benchmarks from the literature and implemented fo this study which are proposed in [224].

In [224], a modified model based on You Only Look Once (YOLO) is presented for insulator fault detection in aerial images with compound backgrounds. Initially, aerial images with few faults are collected in various scenes, and then a new dataset is established using Photoshop. In order to enhance feature reusing and propagation in the low-resolution feature layers, a Cross Stage Partial Dense YOLO (CSPD-YOLO) model is presented based on YOLO-v3 and the Cross Stage Partial Network. Finally, the feature pyramid network and improved loss function are applied to the CSPD-YOLO model to improve the accuracy of insulator fault detection. The CSPD-YOLO model and compared models in [224] are trained and tested on the generated dataset for comparison. We used these three networks to perform the first step of comparison in this study. Table 5.1 shows these comparison results.

Table 5.1: Comparison Among the Existing CNNs Used for the CPLID and the Proposed Method Using the Original CPLID Data set

Neural Network	# of Images	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Time (s)
YOLO-v3 [249]	3,808	93.31	94.00	94.00	94.00	0.01
YOLO-v4 [249]	3,808	96.38	98.00	95.00	97.00	0.01
CSPD-YOLO [224]	3,808	98.18	99.00	98.00	99.00	0.011
Transfer Learning and VGG-19	3,808	99.28	99.62	99.00	99.30	0.006

As it is indicated in Table 5.1, not only the accuracy, precision, and F1 Score are improved using our proposed transfer learning method, but also this methodology reduced the fault detection time of insulator datasets. In this table, the running time shows the required time for one image classification between two classes of intact or broken insulator.

To be able to compare the proposed transfer learning based methodology with the existing CNN benchmarks, namely, YOLO-v3, YOLO-v4 [249], and CSPD-YOLO [224], the proposed method is examined on the original CPLID dataset as well.

In the next step, to prove the reliability of the proposed methodology for generating a larger dataset with image augmentation, we trained the proposed transferred CNN two times, once with the original CPLID dataset and another time with 80% of the large generated dataset. For testing, the remaining 20% of the generated dataset is given to the transferred CNN trained with the two different datasets separately. Table 5.2 shows the results for this step.

Table 5.3 shows that with the same testing datasets, the larger inclusive dataset generates better results as it was expected.

Table 5.2: Comparison Among the Original CPLID and Augmented Dataset Using Transferred CNN While Keeping the Test Data Same

Neural Network	Train Dataset	Test Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Time (s)
Transferred VGG-19	3,808 (Original)	3,344 (Augmented)	96.04	95.10	95.19	95.14	0.007
Transferred VGG-19	13,376 (Augmented)		99.93	99.20	99.41	99.20	0.007

In the third step, the effect of balancing the under study dataset is tested. For this purpose, we do the image augmentation with the equal portion for broken insulator images and the intact ones. Therefore, although the dataset becomes larger and more robust to the feature variation of the images, it would be still an imbalanced dataset. To implement this step, 4 augmented images are generated from each image in CPLID dataset. Hence, the overall number of images including the original images and the augmented ones become 17,800, which is close enough to the number of unequally augmented large dataset with 16,720 insulator images. Table 5.3 shows the results of this step. For both generated datasets (equally and unequally augmented), 80% of the images are used for training and the remaining 20% of the images are used for testing. As it is understandable from Table 5.3, the results for the balanced dataset are a lot better than the case that the dataset is imbalanced.

Table 5.3: Comparison Among the Balanced and Imbalanced Augmented Datasets Based on CPLID Using the Proposed Transferred CNN Method

Neural Network	# of Images	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Time (s)
Transferred VGG-19	17,800 (Imbalanced)	93.22	92.90	92.11	92.50	0.007
Transferred VGG-19	16,720 (Balanced)	99.93	99.20	99.41	99.20	0.007

The results of these three steps show that the proposed method can detect each insulator image category within almost half of the time consumed by the other methods in the literature because of the used transfer learning technique. The fact that the specification of the utilized computer in this study are lower than those used computers in the benchmarks, makes the result improvements in our proposed method more noticeable. The comparison parameters show that the generated dataset outperforms the other benchmarks and it is the most reliable approach up to now.

5.5 Summary

In this chapter, a deep learning methodology is proposed based on the transfer learning technique to improve the results of insulator image classification problem. One contribution of this chapter is that the original CPLID dataset has only 248 broken insulator images among 3,808 images, which puts this dataset in an imbalanced dataset category. Using the data augmentation approach with different portions, a tremendous balanced dataset with 16,720 images is produced, which is a more reliable dataset comparing to the dataset with only 3,808 images.

In this presented transfer learning methodology, first a VGG-19 CNN is implemented as the base model for transfer learning, which is trained using the ImageNet dataset. In the second step, the weights of VGG-19 layers, except the two fully connected final layers, are kept frozen to perform the feature extraction task. The weights of the fully connected final layers are updated using the insulator image dataset for the fine tuning. This transferred VGG-19 CNN generates better accuracy results as compared to benchmarks.

The results show that the proposed transfer learning technique is able to distinguish the intact and broken insulator images with more than 99.9% accuracy, and the required time for insulator image classification training in the proposed technique is about half of the reported time in existing studies.

CHAPTER 6

CMOS-BASED TANH, AN ACTIVATION FUNCTION FOR ANNS

6.1 Introduction

Activation functions in neurons play a significant role in ANN efficient realization, especially because of immense computation in modern applications such as pattern matching, image and speech recognition, natural language processing, and anomaly detection in data. There are various methods to improve ANN accuracy.

Depending on the applications and ANN architectures, both analog and digital activation functions can be used. These functions are used between layers to moderate the output values from the previous layer to the endurable range of input values for the next layer. To achieve this goal, scientists have used various activation functions. Every activation function takes a single value and performs a certain fixed mathematical operation on it [250, 251].

As a part of any hardware implementation of ANNs, non-linear activation functions are the most crucial, expensive, and hard to implement [252]. Many studies consider hardware implementations of activation functions. They focus on various aspects like accuracy, applied approximation methods, cost of implementation, and analog or digital bases [253]. Studies in [254] and [255] show that accurate nonlinear activation function implementations can improve the learning and generalization abilities of ANN. Architectures with higher accuracy cause more silicon area usage and computational speed reduction. Accordingly, having nonlinear activation function hardware designs with a small area, high speed and an acceptable range of accuracy has become a critical issue. To address these challenges, approximation methods are proposed. These methods are mainly categorized into piecewise linear approximation (PWL), piecewise nonlinear approximation, lookup tables (LUT), bit-level mapping, and hybrid techniques. PWL divides function into multiple segments and uses a linear approximation in each segment. Authors in [256, 257, 258] use this method for hyperbolic tangent and sigmoid function implementation. In [259] and [260], a different PWL is introduced. It exploits the centered recursive interpolation based on the lattice algebra-based algorithm instead of similar studies that use input domain segmentation methods. Piecewise nonlinear approximation method works similarly to the discussed PWL. The only difference is that a nonlinear approximation is used in each segment instead of a linear one to result in higher accuracy. This method is used in [261] and to approximate both Sigmoid and hyperbolic tangent functions in [258]. Lookup Table (LUT)-based methods divide the input range into equal sub-ranges and each sub-range is approximated and assigned by a value stored in a LUT [262]. In a bit-level mapping method, the output is approximated based on a direct bit-level mapping of input using purely combinational circuits [263]. A combination of the aforementioned methods is used in hybrid procedures. For example, the methods in [264, 265] have used a combination of PWL and LUT methods for *Tanh* implementation. In a nutshell, there are three important metrics in all hardware implementation of ANNs: accuracy, gate complexity, and processing speed. Each proposed design should be governed by balancing these criteria [256]. Finding a fast activation function that can provide engineers with acceptable accuracy is one of the main challenges in ANN architectures [37, 266, 267, 268].

Among all different types of activation functions, *Tanh* is chosen because of its ideal steep derivative which allows a wider range of values for fast learning and grading methods. This activation function has become famous because of its particular features as explained in [36]. First of all, it covers an entire negative to a positive range of inputs and generates the output values from -1 to 1. Another feature is its symmetric graph, which makes it easier to be implemented in hardware.

Mathematically, *Tanh* is defined as follows [35, 269]:

$$Tanh = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{6.1}$$

where x is an input variable and its graphical representation is given in Figure 1. Because CMOS aging affects the total synaptic current of neuromorphic architectures and this can reduce the activation function's efficiency in each neuron [270, 271, 272, 273, 274, 275], a novel design for hyperbolic tangent (6.1) with high speed and a small area is proposed in this study.

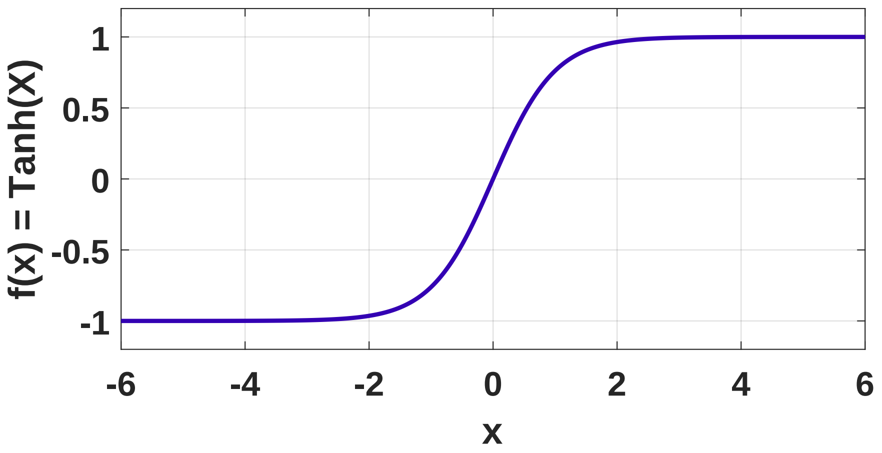


Figure 6.1: The hyperbolic tangent activation function.

In the next section, an analog CMOS-based activation function is proposed which is purely hardware-based, and instead of using any of the above-mentioned approximation techniques, it generates the output from the input data point directly.

6.1.1 Implementation of the Proposed *Tanh* in Two Sides

This chapter is organized as follows. Section 6.2 introduces the hardware design for (6.1) and specifies its elements. The design compatibility with a recently emerged MNN [2] is discussed in Section 6.3. This Section also explains the steps of reaching an optimal design concerning accuracy and power consumption. Section 6.4 describes the operation regions of this design. Section 6.5 shows simulation results, examinations, the accuracy analysis for two datasets based on the proposed design, power consumption behaviors, Monte Carlo, and all other hardware and software tests. Section 5.5 concludes this work.

The hyperbolic tangent is one of the well-established activation functions in ANNs. Therefore, this study presents a CMOS-based design in 45 (nm) technology using the Cadence virtuoso simulator. Applying this design to ANNs causes low power consumption, fast computation ability, and an acceptable range of accuracy in image classification. The proposed design is capable of producing a reliable approximation of the mathematical *Tanh* activation function (1). In this study, *Tanh* is divided into two sub-circuits including its positive half which is shown in Figure 6.2(a) and the negative half that is presented in Figure 6.2(b). These are two straightforward circuits including a few critical elements. The components of each are chosen as follows. The positive half of this activation function includes a P-type metal-oxide-semiconductor (PMOS) transistor with a width size of 435 (nm) in a 45 (nm) technology, a resistor with the resistivity value of 9.2 ($k\Omega$), and a direct current source that provides the circuit with various currents in the range of 0 (A) to 450 (μA). The purpose of this circuit is to resemble the positive half of *Tanh*. On the other side, the negative half of *Tanh* consists of an N-type metal-oxide-semiconductor (NMOS) transistor with the width size of 320 (nm) in the same technology and a resistor with the same resistivity of 9.2 ($k\Omega$), and a direct current source that provides the circuit with various currents in the range of -450 (μA) to 0 (A). This current is resembling the input current when the input to the circuit is in the opposite direction. Figure 6.2 shows the schematic of two sides of *Tanh*.

By taking advantage of the symmetry behavior of *Tanh*, a design with similar schematics for negative and positive sides is presented. Consequently, this simply-

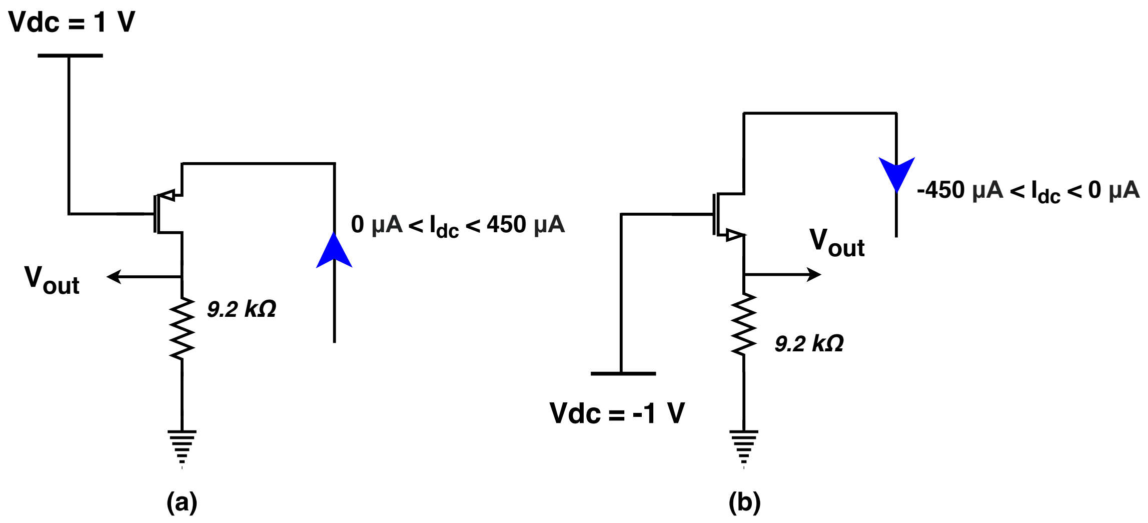


Figure 6.2: The design of (a) positive half and (b) negative half of *Tanh*.

designed *Tanh* outperforms the similar circuits with its low power dissipation and small area usage of its circuitry. Whenever an input current goes through the proposed *Tanh* circuit, based on the sign unit (to be explained in Section 6.3) only one side with a limited power consumption gets activated [276, 277, 278].

6.2 Proposed Architecture

As we mentioned above, in the proposed design for *Tanh*, we need to detect the direction of the current before the activation function operation with a sign unit and propagate the input current through its corresponding section of the neuron to generate the appropriate value for the next layer [29, 279].

Figure 6.3(a) shows the circuit diagram of the proposed neuron. It consists of an Interface Module (IM) and a complete hyperbolic tangent activation circuit. IM uses a domain wall (DW) spintronic device that takes I_j^+ and I_j^- to map their difference into a resistive state. IM is a three-terminal device that consists of a thin domain wall strip connecting to two anti-parallel fixed magnetic domains. The transition area between these two fixed domains (called DW) can be moved by injecting current along the nano-strip. Hence, the resistive state of the device can be changed. A fixed magnet and domain wall strip forms the Magnetic Tunnel Junction (MTJ) for reading the resistive state of the device [280].

The sign unit in IM (Figure 6.3(a)) generates a control signal based on the polarity of the total synaptic current $I_j^+ - I_j^-$. This control signal activates only one of the activation circuits in the hyperbolic tangent neuron, i.e., the positive or negative half of *Tanh* circuit [281]. The positive half is enabled when the control signal is on; otherwise, the negative half is active.

IM operation is described by using three clock cycles: $Clk_1 - Clk_3$. The duty cycles of each clock are different because the reset, write, and read times of a DW device are different. When Clk_1 is low, the sign unit generates the sign of total synaptic current. When $I_j^+ > I_j^-$ logic value of the sign is high, and it is low otherwise. When Clk_2 and Clk_1 are both low, the position of DW is reset so that the resistive state of the device becomes low when the sign is high; and high when the sign is low. When Clk_1 is high, I_j^+ and I_j^- flow into DW. Thus, the difference between the positive and negative convolutional currents programs the DW spintronic device and causes DM to move. Clk_2 and Clk_3 are both high during this period. When Clk_1 and Clk_3 are low, while Clk_2 being high, the resistive state of a spintronic device is read and the absolute difference between total positive and negative synaptic currents flows into a *Tanh*-based neuron. Figure 6.3(b) shows the timing diagram of the IM operation. On the right side of Figure 6.3(a), we have the proposed *Tanh* design divided into two-half circuitry. As we see in Figure 6.2, we have two elements for each half: one MOSFET and a 9.2 ($k\Omega$) resistor. This resistor is a shared component to lower the area usage and therefore we use only one resistor in our *Tanh* circuit to minimize the area.

6.2.1 Proposed Design in Memristive Neural Network (MNN)

In this section, we show the position of the proposed hyperbolic tangent design in MNN. Only two layers of a neural network are shown in Figure 6.4 as an example, which is connected to the rest of the MNN. In Figure 6.4, each $x_n^{[i]}$ stands for the input of neuron n belonging to layer $i+1$ and output of layer i. This architecture uses two MCAs for each layer of the ANN.

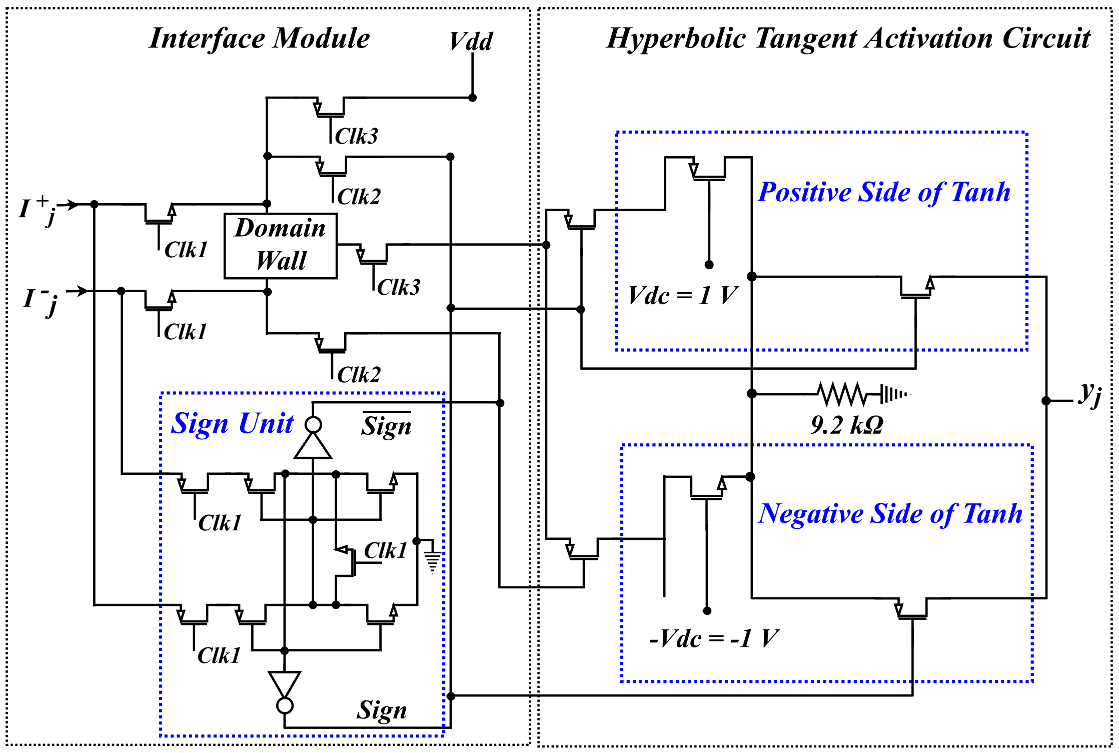
One of them is used to store the updated weights and the other one is used to store the output of gradient calculator circuitry (GCC). This circuit is described in detail in [2] and $f(x)$ in this architecture can be replaced by any activation function like the proposed *Tanh*. Gnawali et al. in [2] implement an on-chip design for a gradient descent backpropagation algorithm by taking advantage of MCAs.

6.3 Circuit Resolution Process

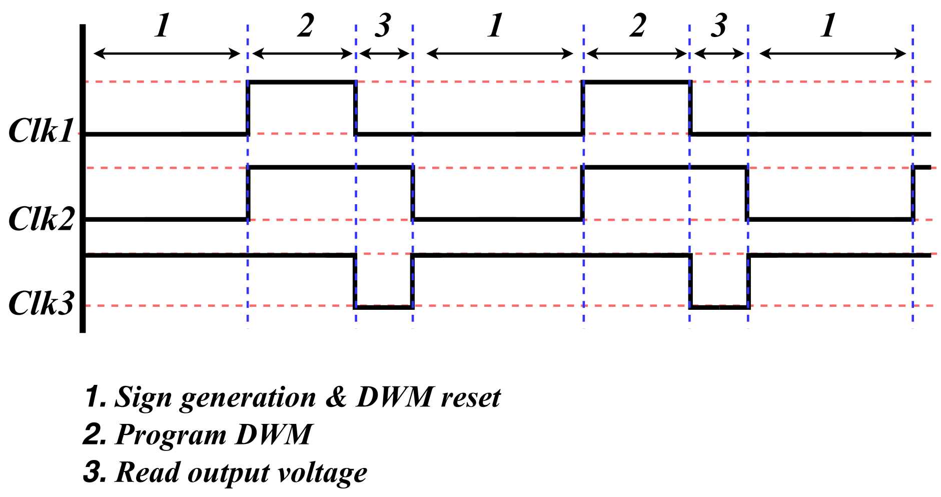
In this section, the circuit resolution of the proposed design for*Tanh* activation function is described. By increasing the resistivity value starting from 0 (Ω), and changing the width size of the MOSFETs, the behavior of the circuit is examined to reach the appropriate value for drain (output) voltage at the right time.

6.3.1 Positive Half of Tanh Design

In the first step, pure resistivity effects are analyzed by keeping the transistor width size fixed at its minimum value (120 (nm)). The output voltage reaches 1 (V) when the resistor value is around 28 ($k\Omega$). Table 6.1 shows the ranges of the generated voltage based on various resistor values. Not only 28 ($k\Omega$) resistivity is irrational and highly power consuming, but also observing the beginning step derivatives of the output graph shows that there is a significant difference between the obtained output voltage and the original *Tanh* function. Accordingly, the idea of changing the PMOS

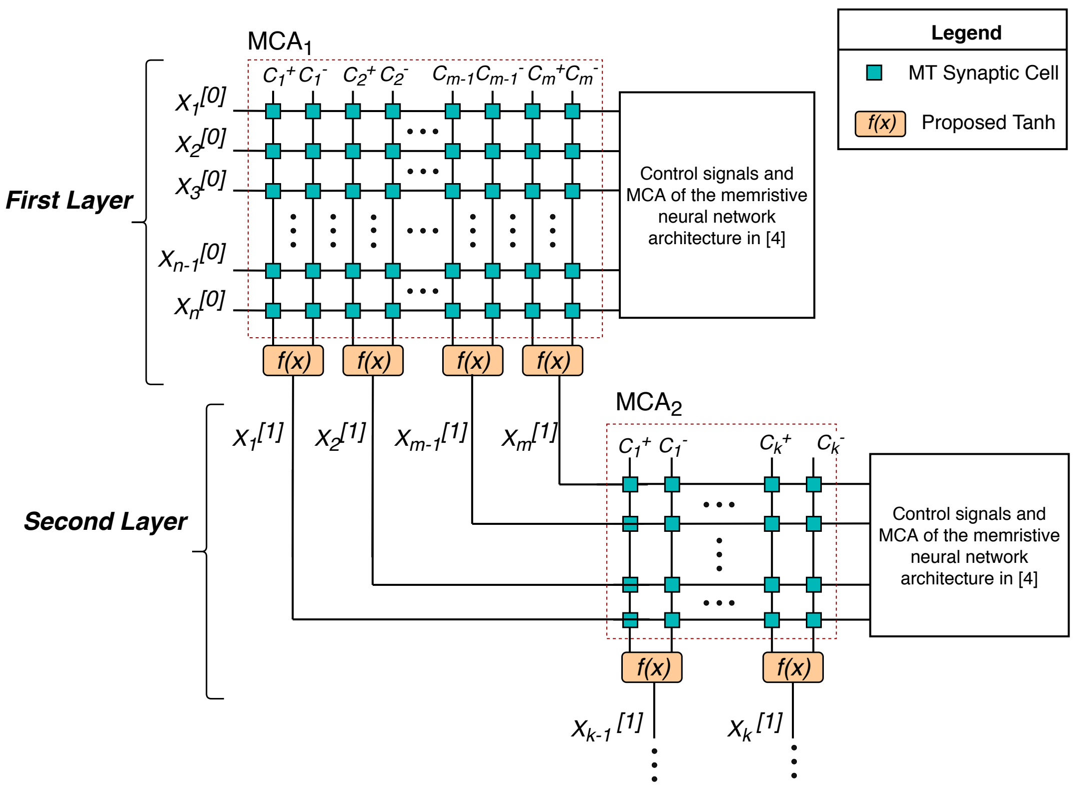


(a)



(b)

Figure 6.3: (a) The proposed *Tanh* neuron and (b) The timing diagram.



Legend
MT Synaptic Cell
f(x) Proposed Tanh
MCA1
First Layer
Second Layer
MCA2
Control signals and MCA of the memristive neural network architecture in [4]
f(x)

Figure 6.4: Position of the proposed design of *Tanh* in MNN [2].

transistor width and considering its fluctuation efficacy to achieve premium results is generated.

In the next step, the resistor value is set up to 10 ($k\Omega$) initially and the correlation of the transistor width size with output voltage is analyzed. By enlarging the width of this transistor, the length of the linear part for the output voltage increases, which means a higher output voltage is achieved with less input current value than the previous case that resistivity was the only parameter affecting the voltage. As mentioned before, the negative half of *Tanh* from its positive half is separated in this design. Therefore, the desired design for the positive half is a circuit with an increasing output value from 0 to 1 (V) and maximum similarity (minimum root mean square error (RMSE)) to *Tanh* function concerning the additive input current.

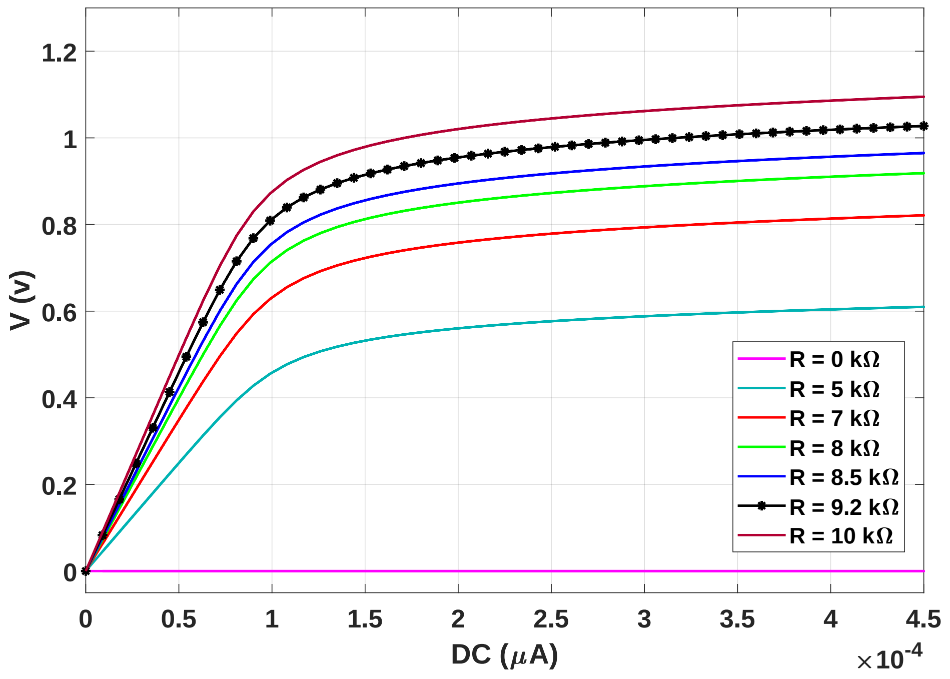


Figure 6.5: Voltage value according to the drain resistor (PMOS width = 435 (nm)).

Figure 6.5 represents how the output voltage varies according to a resistivity enhancement process while the width of PMOS remains constant at 435 (nm). The optimal value of the resistor for this circuit considering RMSE is 9.2 $k\Omega$ which is determined with a dotted black line in Figure 6.5.

Looking at the previous behavior of the circuit while modifying PMOS transistor width, and comparing it to the original hyperbolic tangent function determines that the simulation accuracy with previous values for PMOS transistor width is not enough. For example, in the original function of hyperbolic tangent, $Tanh(3) = 0.9950$; so, 1 (V) should be reached at around 350 (μA), or for input currents with less than 350 (μA) lower values of voltage are expected. Put differently, a line at the beginning is needed with a smaller slope. To achieve this goal, the process is started from the minimum width size in the 45 (nm) technology for the PMOS and increased until the lowest RMSE comparing with the mathematical function is achieved. Besides this, the resistivity value is changed between 9 to 10 ($k\Omega$).

Table 6.2 and Figure 6.6 show the results for the proposed circuit based on various width of the PMOS transistor with a rigid value for the resistor (9.2 ($k\Omega$)). In Figure 6.6 the optimal based on the minimum RMSE and power consumption for PMOS width is shown with a dotted black line. Real-hardware experiments and comparing the accuracy of this circuit with the ideal *Tanh* proves that the best result emerges when the circuit has a PMOS transistor with a width 435 (nm) and a resistor value of 9.2 ($k\Omega$).

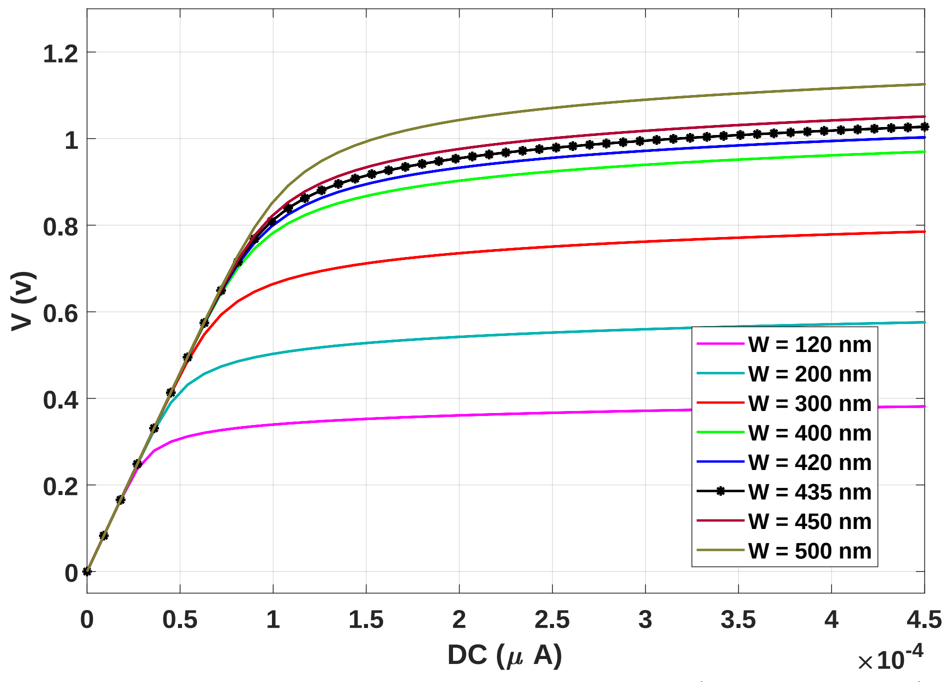


Figure 6.6: Voltage value according to PMOS width (Resistor = 9.2 ($k\Omega$)).

6.3.2 Negative Half of Tanh Design

Implementation of the negative side of the hyperbolic tangent activation function circuit is easier because of its symmetrical attitude. To do so with the least number of elements and area usage, this side is designed by using a 5 ($k\Omega$) resistor, an NMOS with 120 (nm) width, and bias gate voltage (V_{dc}) equal to -1 (V) initially. Then a similar procedure is performed to find the optimal values for the NMOS transistor width to result in the highest similarity to the mathematical *Tanh*, lowest power consumption and RMSE based on the input current range from – 450 (μA) to 0 (A).

Our experiments on this half of the design ended up with 320 (nm) value for NMOS width and 9.2 ($k\Omega$) for the resistivity. Tables 6.3 and 6.4 in addition to Figs. 6.7 and 6.8 show the process of finding the best values for the resistor and the width of the NMOS.

After achieving the optimal values based on RMSE, power consumption, and area usage for the elements for the proposed design, we go through the circuit operation details next.

6.4 Operating Regions

In this design, for the positive half of *Tanh*, a PMOS transistor with the width of 435 (nm) and the length of 45 (nm) is used, which results in threshold voltage at -472.256 mV.

For the negative side, an NMOS transistor with the width 320 (nm) and the same length as PMOS based on our 45 (nm) technology is considered, which leads to a threshold voltage at 721.895 mV. The operating region of a transistor depends on the difference between its gate and source voltages. As shown in Figure 6.2 (a), the PMOS transistor gate is connected to $V_{dc} = 1$ (V) and its source voltage increases with respect to the input current value enhancement. Hence, the output voltage which is the drain voltage of the PMOS transistor varies like a hyperbolic tangent function in its positive region. Figure 6.2 (b) shows that the gate voltage of NMOS (V_{dc}) is set to -1 (V) to make this circuit able to simulate *Tanh* activation function accurately for the negative inputs. The output voltage is the drain of the NMOS transistor with an increasing voltage value considering the sign (direction) of the input current. Next paragraphs explain how this neuron works and discuss the features that make it the desired design.

Table 6.1: Voltage Value According to the Drain Resistor for the Positive Side (PMOS Width = 435 (nm))

Resistor Value $(k\Omega)$	**The range of output voltage** (mV) **(For the input current 0** (A) **to 450** (μA)**)**
0	0 - 0
10	0 - 401.8
20	0 - 765.6
28	0 - 1002

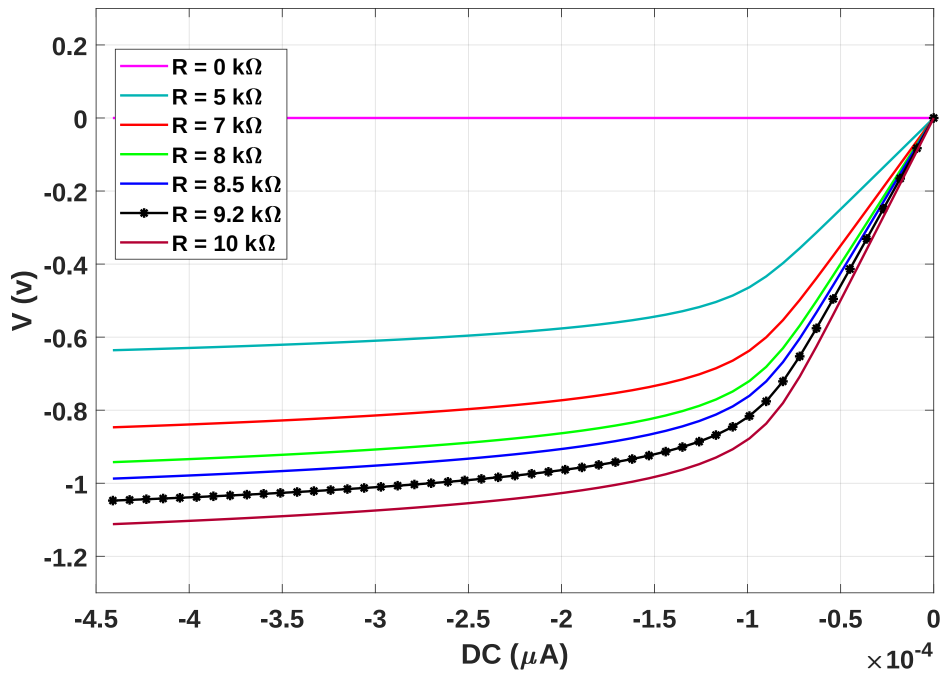


Figure 6.7: Voltage value according to the drain resistor (NMOS width = 320 (nm))

Table 6.2: Voltage Value According to PMOS Width for the Positive Side (Resistor =9.2 ($k\Omega$))

PMOS width ((nm))	**The range of output voltage (mV) (For the input current 0 (A) to 450 (μA))**
120	0 - 370
400	0 - 939
435	0 - 995
440	0 - 1002

Table 6.3: Voltage Value According to Drain Resistor for the Negative Side ((nm)OS Width = 320 (nm))

Resistor value ($k\Omega$)	**The range of output voltage (mV) (For the input current -450 (μA) to 0 (A))**
0	0 - 0
5	- 637 - 0
8	- 943.94 - 0
10	-1.11 - 0

As shown in Figure 6.9, gate voltages for PMOS and NMOS are fixed at 1 (V) and -1 (V) respectively. By increasing the input current, the source voltage goes up from 0.715 (V) to 1.86 (V) on the positive half, and from -2.09 (V) to -0.69 (V) on the negative half.

In our design, the gate voltage of the PMOS transistor is constant 1 (V) and surprisingly not 0 (V) as the expected value for the gate of a normal PMOS. The reason for this phenomenon is to boost up the maximum voltage to reach 1 (V) similar to the maximum value of mathematical *Tanh*.

If 0 (V) voltage value would be assigned to this PMOS, the final value of the output voltage cannot go higher than 650 (mV) and copy the hyperbolic tangent behavior. On the other hand, Figure 6.9(b) depicts the NMOS gate voltage is set to -1 (V) to bring the output voltage down to -1 (V) on the negative half when the input current amount is negative and high. Based on gate and source voltages, the drain voltage of the PMOS transistor (output voltage) varies from 0 (V) to 1.01 (V), and for the NMOS varies from -1.07 (V) to 0 (V). Therefore, we have the following ranges for each voltage as shown in Table 6.5.

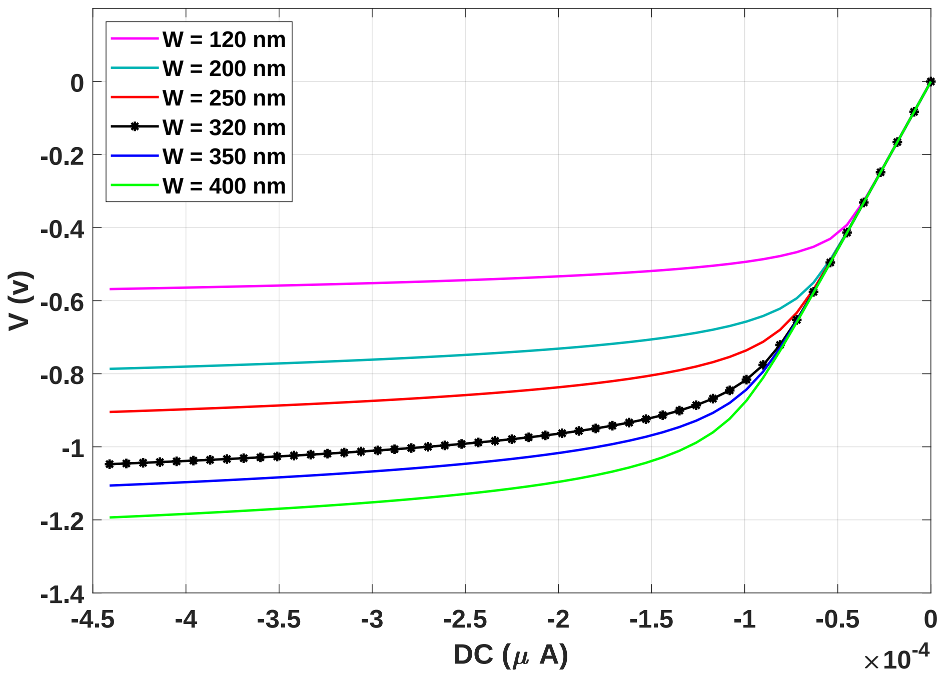


Figure 6.8: Voltage value according to NMOS width (Resistor =9.2 ($k\Omega$))

Table 6.6 details gain, drain, and source voltage values. At the initial level, the PMOS transistor is in its cutoff region. The gate voltage is 1 (V) and the source voltage, which is increasing with respect to the input current, has not reached $|V_g| + |V_{th}|$ to turn the PMOS on yet. Hence, it works in the cutoff region and there would be no current between the source and drain, and the drain voltage becomes 0 (V). Cutoff status is negligible throughout the whole process.

In the next stage, this transistor enters the saturation region. During this state, the circuit operates in two different modes. First, the drain voltage varies linearly based on the current transition, which is inspired by the original hyperbolic tangent that behaves almost linearly in the beginning. Second, the ramp of this voltage is decreasing more and more, until it becomes almost constant 0 at voltage 1.01 (V). Again, we have the same behavior in the hyperbolic tangent when it reaches the ultimate value of 1 (V). These two modes of characteristics are explained next.

For the negative half of *Tanh*, voltage ranges are shown in Table 6.7. Starting from input current at -450 (μA), NMOS works in its saturation region up to the point the input current becomes around -9 (μA). Then NMOS is in its cutoff region without any current flowing through it. Figure 6.9(b) shows the operation regions of NMOS.

6.4.1 Changing Linearly

There is a current flowing from the source to drain of PMOS and from the drain to source of NMOS, which causes the absolute value of drain voltage to increase.

$$I_D = -\frac{1}{2} K_p \frac{W}{L} (|V_{gs}| - |V_{th}|)^2 (1 + \lambda |V_{ds}|) \tag{6.2}$$

$$V_D = R I_D \tag{6.3}$$

Equations (6.2) and (6.3) show the current and voltage dependencies and parameters, where I_D is drain current, W and L are the width and length of the transistor respectively, K_p is the transconductance of MOSFET, V_{gs} is the gate-to-source voltage of MOSFET, V_{th} is the threshold voltage, and λ is the channel length modulation coefficient. According to (6.2) and (6.3), this change is linear because the absolute value of the input current that produces source voltage increases linearly. Therefore, the drain current value is calculated through (6.2). Note that we write the above equations based on the absolute value of the voltage to become compatible with both PMOS and NMOS.

6.4.2 Reaching desired value and getting saturated

We know that in MOSFET structures, there is a diode between the source and body of the transistor. For PMOS, until the source voltage reaches the value around 1.7 (V), the only existing current is from the source to the drain of this transistor which we call this linear behavior. The same phenomenon happens for NMOS in the negative half when the source voltage reaches -1.75 (V). But after that, despite that the absolute value of input current rises for both sides, the absolute value of the drain voltage increases at a slower rate, until it becomes almost fixed at 1.01 (V) in the positive half or -1.04 (V) in the negative half. These voltages are the ultimate values for the output of the circuit. The reason for this occurrence is that more than 1.7 (V) voltage at the source of PMOS transistor or less than -1.75 (V) at the source of the NMOS transistor turns on their inner diodes. Accordingly, we have currents in two directions. The first path is the source to the drain voltage current that we had before in the linear part, and the latter one is the source to the body current through the interior diode which does not let the drain-source voltage continue to increase linearly. Figure 6.9 demonstrates the characteristics of source voltage that varies like diode characteristics graph for PMOS after 1.7 (V) or for NMOS after -1.75 (V). An almost constant voltage at the source of each MOSFETS is observed which holds the output voltage (the drain) in a constant value equal to around 1 (V) for the positive half or -1 (V) for the negative half of the proposed *Tanh* design. Tables 6.6 and 6.7 summarize the above explanations for these two MOSFETs.

6.5 Experiments and Comparisons

In this Section, the proposed design is compared with its mathematical ideal version, to prove its efficiency, accuracy, and reliability. To implement the circuit and calculate the average power consumption, the Cadence virtuoso is used as a simulator.

To examine the accuracy of the proposed implementation for *Tanh*, we use two well-known datasets from the image recognition category for ANNs: MNIST and Fashion-MNIST. Each of these datasets has 70K images among which 60K of them are labeled and used for the training phase and the rest for the test. They both are categorized into 10 groups. MNIST images are handwritten digits from 0 to 9 and Fashion-MNIST consists of 10 fashion classes including T-shirt/Top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag and Ankle boot.

The above-mentioned datasets are used to prove that not only the proposed design works for an easy and common dataset like MNIST, but also results in very similar accuracy for Fashion-MNIST which is known as a complex one [281, 282, 283]. To evaluate the precision of the proposed design for the chosen datasets, we take advantage of the software methods explained below. The idea of using software methods is supported by activation function roles in ANNs. Activation functions determine the output behavior of each neuron to meet the constraints of input neurons in the next layer. Therefore, no matter if we work in a software or hardware environment, the process of assigning an $f(x)$ value to a given input x is the same.

The ANN architecture for this study is implemented in Python language, using Keras frameworks and then the existing activation function is replaced with our custom activation function which is built based on the proposed design for *Tanh* using a look-up table; meaning once this activation function is the mathematical equation of *Tanh* which is shown previously in Figure 6.1 and (6.1), and the other time the proposed design results are used from Cadence virtuoso simulator in the form of a lookup table which assigns a voltage value to each given current amount. The steps of producing these lookup tables are explained below. After implementing the circuit on the Cadence tool and plotting the output graphs, the ".csv" files are generated from plots. These files save the output of the circuit in the form of a two-dimensional array or a table with two columns and n rows. In this experiment, one column would be the input current steps, and the other one is their respective voltage values. Therefore, a lookup table is built to play the role of an activation function in the Python program.

Figure 6.12 shows the plot generated by Matlab. The red dotted line shows the graph of ideal hyperbolic tangent generated by its mathematical equation, and the thick black line represents the proposed activation function design. As is shown in Figure 6.1, the value of *Tanh* reach $f(x) = 1$ around $x = 3$ while this happens at the input current of almost 300 (μA) for the proposed design. To comparing the mathematical graph of the hyperbolic tangent with the proposed design, the values are scaled in Figure 6.12. Considering the fact that our input current varies between 0 (A) to 750 (μA) and for this range of value for x, the mathematical function value is almost zero, we have to scale our values in the applied lookup table and multiply them by 10^4 to show an appropriate range of this estimation. The calculated RMSE for this design is ***0.0294*** based on Figure 6.12.

6.5.1 A Fully-connected NN for MNIST & Fashion-MNIST

For the learning process, in the first phase, synaptic weight values are produced by training the network with 60,000 classified sample images in MNIST and Fashion-MNIST each. After the training (learning) process, we test the accuracy of this neural network with the other unclassified test images for these two datasets. Each of these images is a 28×28 matrix with the entries value from 0 (the brightest possible pixel, white) to 255 (the darkest possible pixel, black) [284]. To work in an analog domain, the value of each pixel is divided by 255 to generate value in the range from 0 to 1. In the output layer of ANNs, there are 10 neurons each of which is determined to be one digit from 0 to 9. Each neuron in this layer generates a voltage between 0 and 1 which is considered as the probability of being one specific digit in the MNIST dataset or belonging to that specific category in Fashion-MNIST.

For example, if the fifth neuron has the maximum value, it means the questioned digit is 4 in the MNIST or that picture belongs to the coat category in the Fashion-MNIST. To test the proposed design from multiple aspects, different ANNs with different characteristics are taken into account. By training parameters like epoch (the number of times that the model cycle through the data) and epsilon (the minimum error rate) [282, 283] modification, the maximal precision is achieved.

In this ANN, we have three fully connected layers including an input layer with 784 neurons, one hidden layer with 128 neurons, and an output layer with 10 neurons for providing us with ten categories. We train this ANN by using MNIST with batches of size 128 for 10 epochs and the Adam optimizer with the learning rate of e^{-3}.

Additionally, according to the existing studies [285, 286, 287, 288] on MNIST and Fashion-MNIST, we know that pure *Tanh* cannot give us more than ***89%*** accuracy for MNIST and ***77%*** for Fashion-MNIST when it is applied to all layers. To solve this issue, the Softmax activation function is applied to the output layer and the hyperbolic tangent is used for the other two layers. So, the ANN is upgraded to have such functionality, and consequently, the use of the proposed *Tanh* design has an accuracy of ***95.58%***, which is very close to ***95.51%*** achieved by using mathematical *Tanh*. Such an accuracy makes our design a competitive one [282].

For evaluating the accuracy using Fashion-MNIST dataset, considering the fact of having the same binary files and the same size of pictures, a similar procedure to MNIST is performed. The only difference is the higher complexity of images. Consequently, an output layer with 10 neurons for 10 fashion classes, as coded in Table 6.8 is built to generate a voltage between 0 and 1 for each input image, which means the probability of being classified in one specific class. The Softmax activation function is used for the output and *Tanh* for the other two layers. The use of the proposed *Tanh* design results in an accuracy of ***85.76%***, which exhibits a 0.07% lower value than ***85.83%*** achieved by using mathematical *Tanh* [283]. The reason is the complexity of this dataset that makes the proposed design a slightly better match to it. Table 6.8 and Figure 6.13 summarize the accuracy results for these datasets.

6.5.2 Adding Noise to ANN Inputs

To prove the robustness of the presented design, its accuracy is tested by adding white (Gaussian) noise and Salt & pepper noise to MNIST and Fashion-MNIST datasets. Table 6.9 shows the accuracy results after 10 epochs with the batch size of 128.

As we mentioned in Section 6.2, the proposed design is a part of recent MNN architectures and the mentioned validation methods [2, 41] are applicable for this activation function neuron design. For emerged MNNs, there are various methods to regulate the weight values to be stored in the memristors of synaptic cells (no matter which technology is applied). Optimization algorithms, such as stochastic gradient descent and batch gradient descent, calculate the synaptic weight value based on the cost function minimization. This cost function is usually defined by the existing error between the expected and generated values.

Recent conventional methods train the network using the Central Processing Unit (CPU) and Graphics Processing Unit (GPU) at the first step and then weights are stored into MCAs. These training strategies are called ex-situ training [2, 289, 290, 291]. One of these efficient methods is proposed in [2] recently. This method is called in-situ training or on-chip learning that improves an MNN based on the hardware basis to decrease the impact of Process Variation (PV) on architecture precision. This approach uses extra elements in the circuit to reserve the neuron outputs and compute the gradient of the cost function. Any flaw in the chip due to PVs may result in imprecise computation by neurons. Gnawali et al. in [2] consider the effect of Process Variation (PV) (intrinsic and extrinsic) at the output layer. In other words, the cost function carries the impact of PV toward output neurons, and the optimization algorithms in training level slander the effect of PV by adjusting the values of weights properly.

Base on the aforementioned techniques, by using the in-situ training method [2] any PV effect in the proposed design is being considered in the output layer of the ANNs which works based on the Softmax activation function. The accuracy analysis of the proposed design is done in a software program with usage of a lookup table based on the on-chip learning methodology [2]. The actual hardware architecture can be implemented as an MNN with the proposed hyperbolic tangent design that brings the imprecision of the whole layers to the output layer to be taken care of.

Figure 6.14 displays the noise analysis for both sides of the proposed design between 1 (Hz) to 1 (GHz). As expected, in low frequency, a higher noise rate is observed, and as frequency increases, noise values decrease. The input noise graphs are depicted based on the input current source that varies from -450 (μA) to 450 (μA) and the output noise plots show the negligible value for noises from drain voltage of the sub circuits.

6.5.3 Monte Carlo Analysis

To simulate the fabrication process variations besides the above-mentioned global on-chip learning method, the Monte Carlo simulation method is applied to vary the parameters of each element in the circuit for examination of transistor mismatches, integration issues, and process variations. Because in each case only one side of *Tanh* is activated, Monte Carlo is run only for one half of the design each time and the results are placed together as shown in Figure 6.15. The Cadence Virtuoso divides the variation into two different processes.

The first simulation process adds the same variation for the whole circuit at once and is called the process variation and the second simulation option is mismatch variation to vary every device individually. Both of these variations allow precise and realistic simulations for the proposed design of *Tanh* by defining the deviation of the parameters for each simulation process separately. In Figure 6.15 both mentioned procedures are considered by analyzing the value of the resistor (9.2 ($k\Omega$)), width and length of MOSFETs as the global variables besides the inner manufacturing parameters.

We run Monte Carlo with 1000 samples for each half of the design. The behavior of the circuits show the RMSE ranges for the positive and negative half are [0.017, 0.061] and [0.019, 0.058] which results in the average RMSE of 0.0309 and 0.0304 for them respectively [292].

6.5.4 Power Consumption of Proposed Design

To prove that this is a current-based low power design, the mean of power consumption for this design is calculated from the start point of -450 (μA) to the endpoint of 450 (μA) current by taking step size equal to 10 (μA) current. This power consumption calculation is based on the whole neuron circuitry in Figure 6.3. In this implementation, it is important to know the input current absolute value does not exceed 75 (μA) in most applications (more than 95%) [29, 278, 281] which is in the power-efficient range of current for the proposed design.

Finally, power consumption for all steps is summed up to result in total power. Experimental results show that the mean power consumption of the proposed design, in the complete range from -300 (μA) to 300 (μA) is 118.79 μW, while for the applicable range between -75 (μA) to 75 (μA) (based on experimental results in [29, 278]), it is 65.2 (μW). Hence the power consumption of the proposed design for *Tanh* is low and lets this design to be categorized as a low-power implementation for *Tanh*. Figure 6.16 shows the total ranges of the power consumption for this study with the focus on the applicable part which is determined by a red two-sided arrow.

6.5.5 Comparison With Others

Table 6.10 shows the comparison results among some existing hardware implementations of *Tanh* and our proposed design. Note that most of the existing designs are digital and only one analog implementation for *Tanh* is found beside the proposed design in this study. The two last rows of Table 6.10 are compatible with recent CMOS-based MNNs while the others are in the digital domain. Since the previous hardware implementations [262, 264, 257] were based on a CMOS 0.18 (μm) process, the area of the proposed *Tanh* design is scaled from 45 (nm) to 180 (nm). This process is performed by determining a factor using the geometric mean of three aspects (minimum feature size, metal I half pitch size, and 4 transistor logic size) to provide us with the ability to compare our novel design with existing ones [267].

In Table 6.10 the first line shows scheme-1 proposed by Lin and Wang who estimate the initial order derivative of *Tanh* with an isosceles triangular function [257]. The second row presents the approximation of a lookup table for *Tanh* activation function, and the third row shows the Range Addressable LUT (RALUT) estimation [262]. In the fourth row, the area and delay of a lookup table combined with a subtractor (LUT-Sub) [264] are shown. The fifth row shows the specification of CR Spline design in [293] that uses PWL method and LUTs. A 3-in-1 architecture for *Tanh*, ReLu, and Sigmoid is proposed by Chang et al. in [294] that although it takes a very small area for implementation, it has a very high delay for the critical path according to the used MUX and bit-level mapping process. The specification of this design is shown in the seventh row of Table 6.10. All the above-mentioned designs use the approximation methods that are explained in Section 6.1. In the eighth line of Table 6.10, an analog implementation of NN proposed by Yildiz et al. in [40] is proposed which is based on MCAs. In their *Tanh* design, the mentioned architecture includes 2 MOSFETs, 2 resistors, and an operational amplifier that needs a big area to be implemented. The final row shows the area and the delay estimation of the proposed design after scaling [296].

Minimizing circuit delay and area usage of any hardware implementation are two influential inseparable parameters and therefore, the fourth column is generated considering these two factors together as a combined parameter [264]. Based on Table 6.10, the area of the proposed design is not the minimum which is expected because of the high value of the used resistor; but the critical path delay of the proposed design is the lowest which makes the product of these two parameters the second minimum in the last column of Table 6.10. Also, only the two last designs of this table are compatible with recent MNNs and low power designs without taking advantage of any approximation method. N/P stands for Not Provided in Table 6.10 and is used for the studies that did not report the area or delay for their implementations.

6.6 Summary

Implementing ANNs in hardware and new generation MNN provides us with high parallelism to run a large application faster than software methods that face von Neuman bottleneck limits [34, 273, 281]. Memristive architectures as in [48, 275, 297, 40] are attracting a lot of attention because of the vital and time-consuming arithmetic operations like multiplications and additions that can be performed by simple components in hardware that by taking advantage of the recent designed resistive devices. In these methods, power dissipation and execution time are much lower in comparison with multiprocessor-based systems that utilize neuromorphic calculations [270, 298], or GPGPU-based architectures [299].

Considering the above-mentioned methods, a novel design for *Tanh* activation function is proposed to help the neuromorphic architectures overcome the heavily demanded computations through the layers in the training step. These computations make them the most time-consuming section among all.

The proposed architecture provides scientists with a pure hardware realization opportunity. It is a low-power analog design for the hyperbolic tangent activation function which has the simplest architecture up to now including only two MOSFETs and one resistor after a sign unit. Accuracy results on MNIST and Fashion-MNIST datasets, power consumption computation, reliable noise frequency analysis, and easy implementation of this design outperforms most of the similar architectures in literature. The other important advantage of the proposed design is its compatibility with CMOS-based architectures especially the recent MNNs [29, 300]. It is experimentally proved that the power dissipation of the proposed analog design for *Tanh*, and its small area usage is competitive against the existing digital designs for this important activation function while it does not affect accuracy.

Table 6.4: Voltage Value According to NMOS Width for the Negative Side (Resistor = 9.2 ($k\Omega$))

NMOS width ((nm))	**The range of output voltage (mV) (For the input current -450 (μA) to 0 (A))**
120	-566.96 - 0
200	-787.52 - 0
250	-906.14 - 0
350	-1.10 - 0

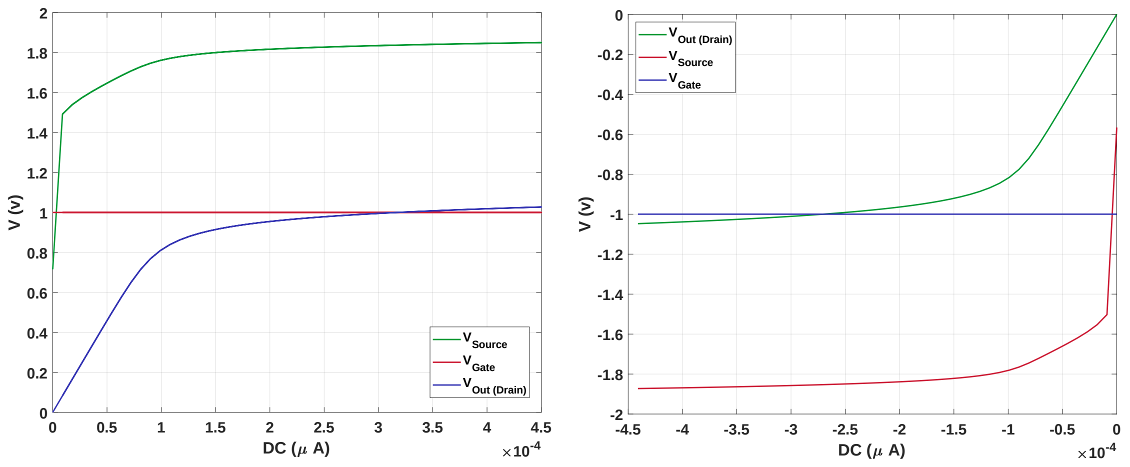


Figure 6.9: Source, gate and drain voltage variation of (a) the positive half and (b) the negative half of the proposed design.

Table 6.5: Voltage Values Range for Positive and Negative Sides of MOSFETs According to the Circuit Resolutions

Voltage (mV)	V_{th}	V_{gs}	V_{ds}	V_{gd}
Positive side (PMOS)	-471.896	[-761, 285]	[-1398.2, -715]	[-10, 1000]
Negative side (NMOS)	721.895	[430, 870]	[570, 1416]	[-1000, 49]

Table 6.6: Operation Regions of PMOS in Positive Side Tanh

Input Current (μA) / Voltage (mV)	**$0-8$ (Cutoff)**	**$8-100$ (Linear Section)**	**$100-450$ (Saturated V)**
V_g	1000	1000	1000
V_s	[715, 1471)	[1471, 1761)	[1761, 1845)
V_d	[0, 73)	[73.6, 813)	[813,1010)
V_{gs}	(-471, 285]	(-761, -472]	[-845, 761)
V_{gd}	(926, 1000]	(187, 926]	(-10, 187]
V_{ds}	(-1398, -715]	(-948, -1398]	[-948, -835)
Operation region	$V_{sg} <\|V_{th}\|$ $\Rightarrow$ Cutoff	$V_{sd} <V_{sg}-\|V_{th}\|$ $\Rightarrow$ Saturation	$V_{sd} <V_{sg}-\|V_{th}\|$ $\Rightarrow$ Saturation

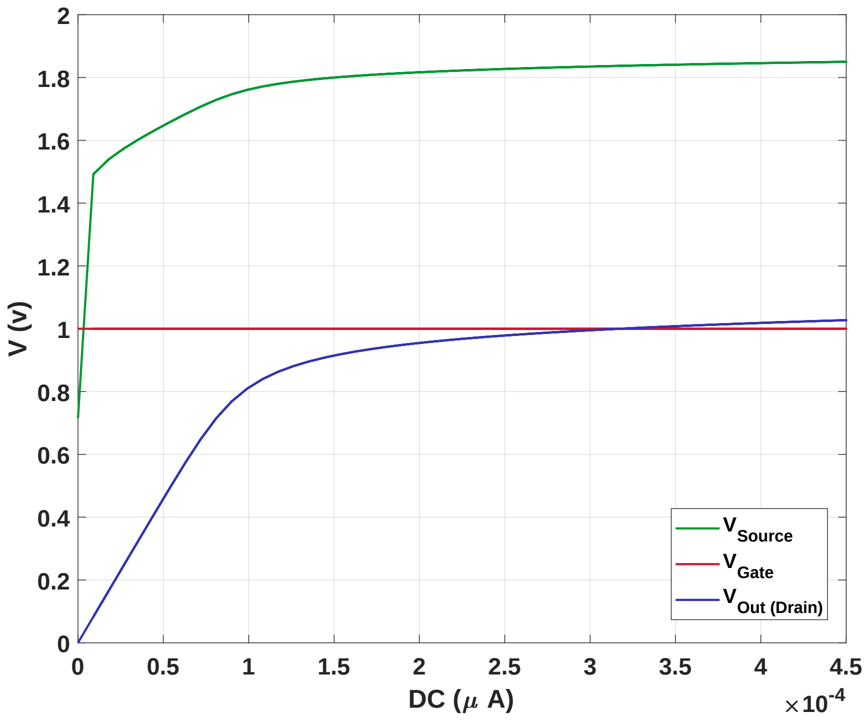


Figure 6.10: Source, gate and drain voltage variation of the positive half.

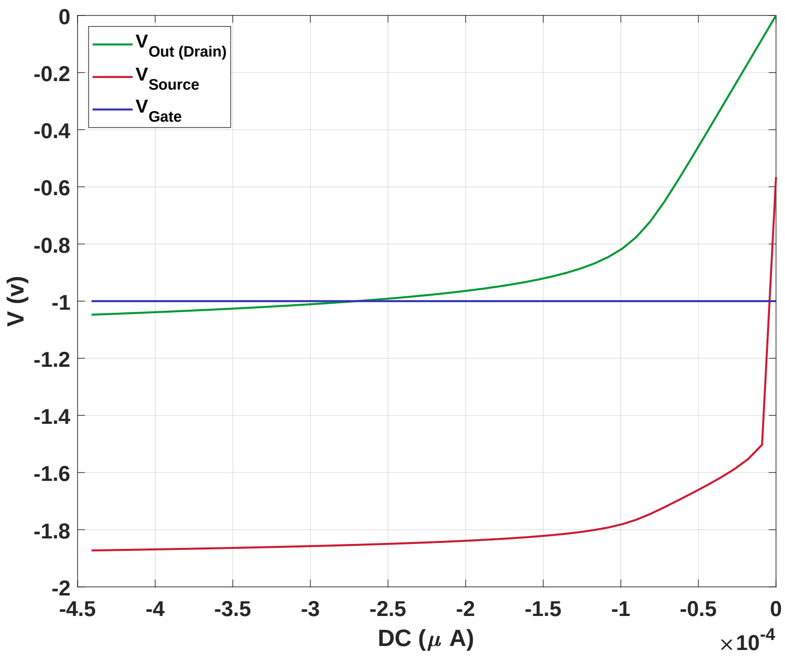


Figure 6.11: Source, gate and drain voltage variation of the negative half.

Table 6.7: Operation Regions of NMOS in Negative Side Tanh

Input Current (μA) / Voltage (mV)	**-450 – -84 (Saturated V)**	**-84 – -9 (Linear Section)**	**-9 – 0 (Cutoff)**
V_g	-1000	-1000	-1000
V_s	[-1870, -1500)	[-1810, -1500)	[-1500, -570)
V_d	[-1049, -84)	[-908, -84]	[-84, 0)
V_{gs}	[810, 870)	(500, 810]	(430, 500]
V_{gd}	[-92, 49)	[-916, -92)	(-1000, -916)
V_{ds}	[821, 902)	[902, 1416]	(570, 1416]
Operation region	$V_{ds} > V_{gs} - V_{th}$ $\Rightarrow$ Saturation	$V_{ds} > V_{gs} - V_{th}$ $\Rightarrow$ Saturation	$V_{gs} < V_{th}$ $\Rightarrow$ Cutoff

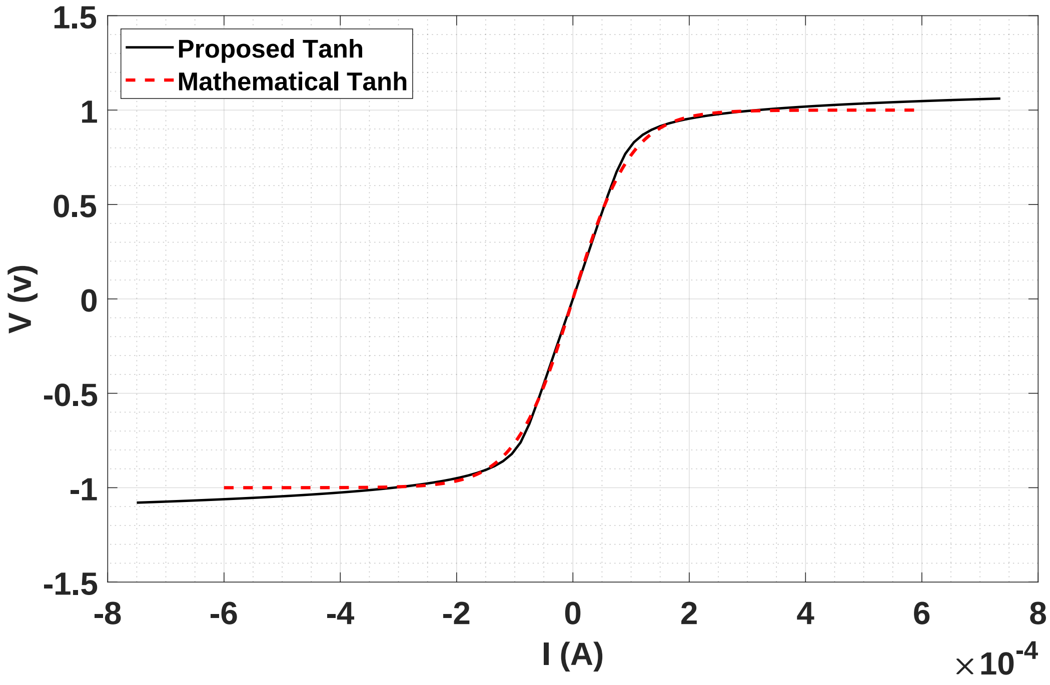


Figure 6.12: Comparison of our proposed *Tanh* with the mathematical form of *Tanh*.

Table 6.8: Accuracy Results of the Proposed Design for MNIST and Fashion-MNIST

	Training Loss		**Training Accuracy (%)**		**Validation Loss**		**Validation Accuracy (%)**	
	Tanh	Proposed *Tanh*	*Tanh*	Proposed *Tanh*	*Tanh*	Proposed *Tanh*	*Tanh*	Proposed *Tanh*
MNIST	0.10	0.10	96.92	96.95	0.14	0.14	95.51	95.58
Fashion-MNIST	0.38	0.38	86.19	86.13	0.39	0.39	85.83	85.76

Table 6.9: Noise Tolerance (Validation accuracy) of the Proposed Design for MNIST and Fashion-MNIST

	White Noise		**Salt&pepper (20%)**		**Salt&pepper (50%)**	
	Ideal *Tanh*	Proposed *Tanh*	Ideal *Tanh*	Proposed *Tanh*	Ideal *Tanh*	Proposed *Tanh*
MNIST	92.17	92.01	94.53	94.98	90.07	89.14
Fashion-MNIST	75.29	74.96	82.88	82.53	75.61	75.21

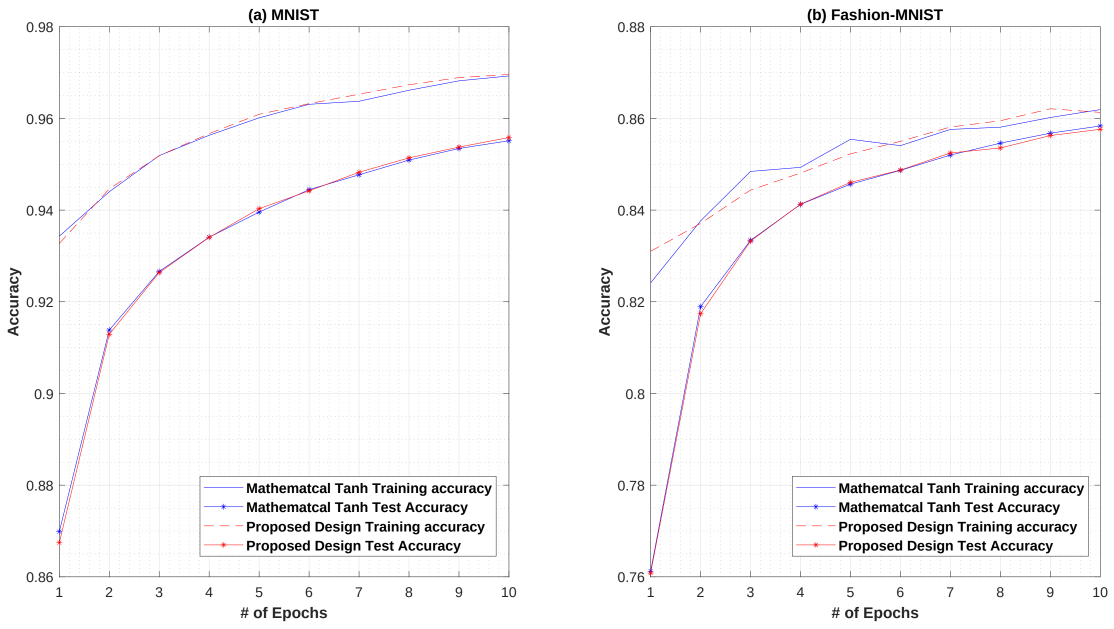


Figure 6.13: Training and test accuracy for MNIST and Fashion-MNIST.

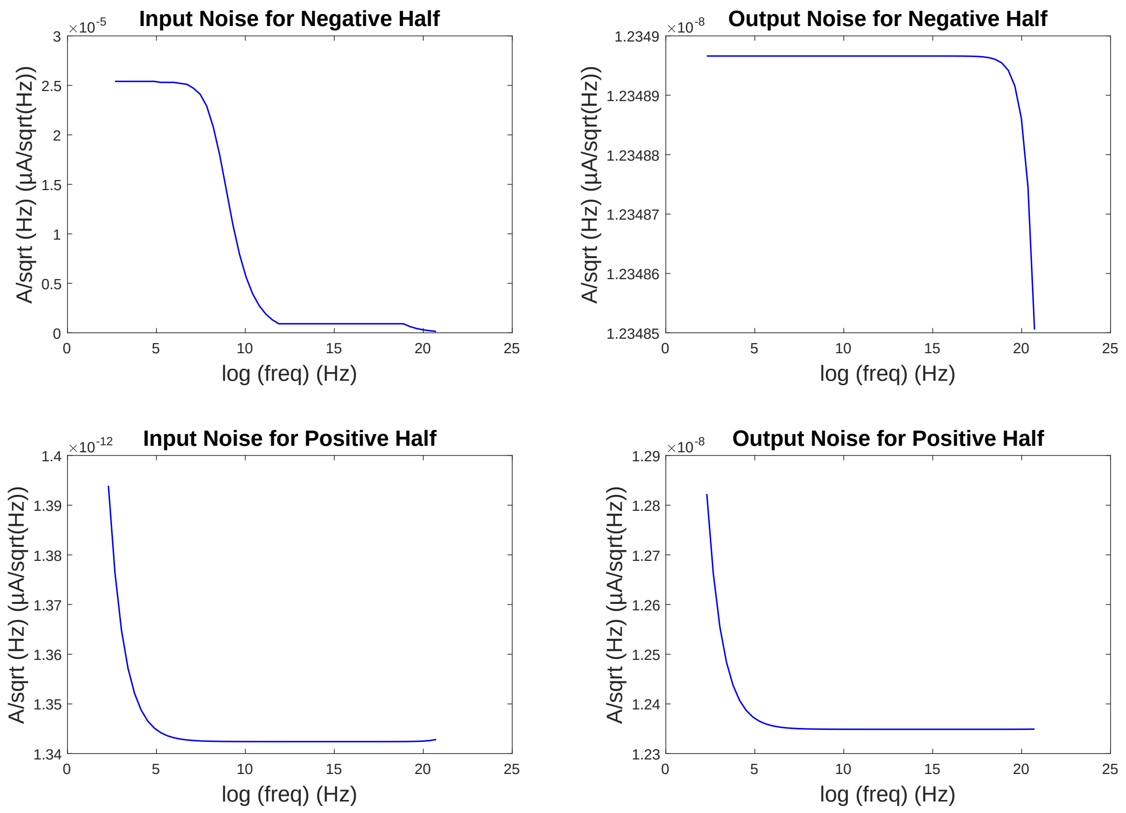


Figure 6.14: Noise analysis of the proposed design for *Tanh*.

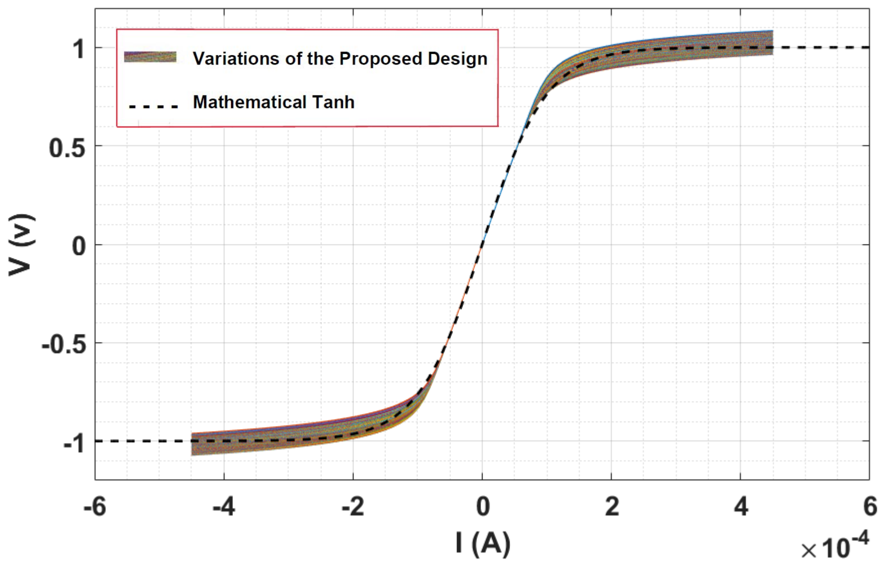


Figure 6.15: Monte Carlo results for mismatch and process variation effects.

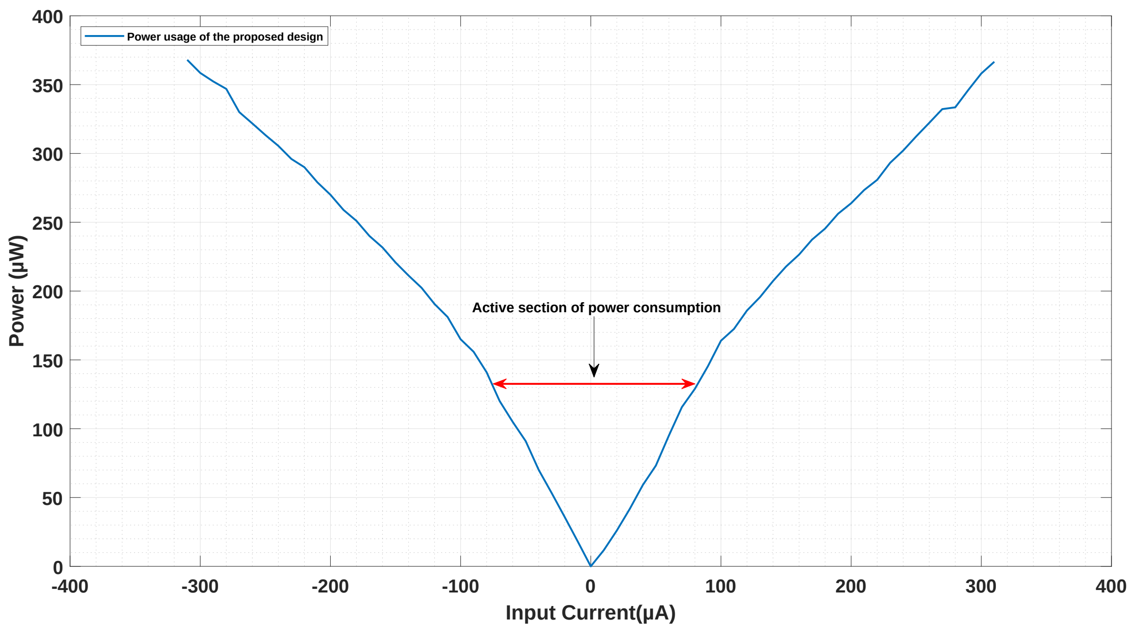


Figure 6.16: Power consumption of the proposed design.

Table 6.10: Comparison of Different Designs for *Tanh*

Architecture	**Pros**	**Cons**	**Area** (μm^2)	**Delay** (ns)	**Area×Delay** (μm^2×ns)
Scheme-1 [257]	High speed Low error rate	Very high area usage Digtal-based domain Not compatible with recent MNNs	32069.83	903	2.895×10^{-11}
LUT [262]	Low error rate	Very low speed High area usage Digtal domain Not compatible with recent MNNs	9045.94	2.15	1.944×10^{-17}
RALUT [262]	Low error rate	Low speed High area usage Digtal-based domain Not compatible with recent MNNs	7090.40	**1.85**	1.311×10^{-17}
LUT-Sub [264]	Small area usage	Low speed Digital-based domain Not compatible with recent MNNs	3646.83	2.31	$\mathbf{0.842} \times 10^{-17}$
CR Spline [293]	Low error rate	Low speed Very high area usage Digital-based domain Not compatible with recent MNNs	5840 Gates No Memory	N/P	N/P
3in1 Activation Function [294]	Very low area usage	Low speed High error rate Digital-based domain Not compatible with recent MNNs	**1947.69**	N/P	N/P
DCTIF [295]	Low error rate Small area usage	Very Low speed High area usage Digita-based domain Not compatible with recent MNNs	2451.0	7.632	18706.032
MCA-based Analog NN [40]	High speed Analog-based domain Compatible with recent MNNs	High area usage High power consumption	N/P	N/P	N/P
Proposed *Tanh*	High speed Analog-based domain Low error rate Low power consumption Compatible with recent MNNs	Moderate area usage Moderate speed	7058.94	**1.85**	$\mathbf{1.305 \times 10^{-17}}$

CHAPTER 7

CONCLUSION AND FUTURE DIRECTIONS

7.1 Contributions

In this study, various ANN methods are discussed and their applications in power systems are investigated. The first chapter of this thesis discusses the preliminaries which are utilized in the subsequent chapters. Also, transmission lines as a system with imbalanced datasets are described.

In the second chapter, a comprehensive review of ML methods for TL fault detection, classification, and location estimation is presented. The existing methodologies are divided into three main categories called generic ML, ANNs, and hybrid methods. Generic ML approaches are categorized into five subsections, namely, Naive Bayes Classifier, Decision Tree, Random Forest, Support Vector Machine, and k-Nearest Neighbour. ANNs are divided into four main categories based on their various structures, namely, Feed forward Neural Network, Convolutional Neural Network, Adaptive Neuro Fuzzy Inference System, and other small groups such as Extreme Learning Machine and Probabilistic Neural Network. Finally, the third category is assigned to hybrid methods. The basic idea, fundamental equations, and relevant publications since 2015 are included and summarized for each method. The significance of this survey is highlighted as compared to the existing review studies in this paper. Moreover, the advantages and disadvantages of the ML approaches are discussed in details and summarized to provide a clear pathway to future research directions.

In the third chapter, the problem of fault detection, identification, and location estimation of transmission lines using two NNs, namely, GRNN and CNN is studied. The data used in this study are the phase current and voltage measured from one end of the transmission line. An FFT is used to extract the amplitude of the fundamental frequency components from the current and voltage waveforms, and feed them to both of the NNs for the detection, identification, and location estimation of faults. The main contribution of this chapter is that it analyzes the robustness of the proposed detection, identification, and location estimation techniques against the parameter changes in a transmission line, namely fault resistance, fault inception angle, source inductance, phase difference between the two buses, bus voltage amplitude variation, and measurement noise. In addition, a time delay analysis is performed to guarantee that these modules can successfully complete their tasks within the desired time window based on the $IEEE39$ standard model before the tripping relays disconnect the transmission line. In overall, CNN has a better performance in both identification and localization of the faults in comparison with GRNN. It should also be noted that both GRNN and CNN have better robust performance compared to other techniques in the literature.

In the fourth chapter, a deep learning methodology is proposed based on the transfer learning technique to improve the results of insulator image classification problem. One contribution of this chapter is that the original CPLID dataset has only 248 broken insulator images among 3,808 images which puts this dataset in an imbalanced dataset category. Using the data augmentation approach with different portions, a tremendous balanced dataset with 16,720 images is produced, which is a more reliable dataset comparing to the dataset with only 3,808 images. This chapter presents a transfer learning methodology. First a VGG-19 CNN is implemented as the base model for transfer learning, which is trained using the ImageNet dataset. Next, the weights of VGG-19 layers, except the two fully connected final layers, are kept frozen to perform the feature extraction task. The weights of the fully connected final layers are updated by using the insulator image dataset for the fine tuning. This transferred VGG-19 CNN generates better accuracy results than its benchmarks. The results show that the proposed transfer learning technique is able to distinguish the intact and broken insulator images with more than 99.9% accuracy, and the required time for insulator image classification in the proposed technique is about half of the reported time in existing studies.

The fifth chapter proposes a novel architecture for hyperbolic tangent activation function in neural networks. Implementing ANNs in hardware and new generation MNN provides us with high parallelism to run a large application faster than software methods that face von Neuman bottleneck limits [34, 273, 281]. Memristive architectures as in [48, 275, 297, 40] are attracting a lot of attention because of the vital and time-consuming arithmetic operations like multiplications and additions that can be performed by simple components in hardware that by taking advantage of the recent designed resistive devices. In these methods, power dissipation and execution time are much lower in comparison with multiprocessor-based systems that utilize neuromorphic calculations [270, 298], or GPGPU-based architectures [299].

Considering the above-mentioned methods, a novel design for the *Tanh* activation function is proposed to help the neuromorphic architectures overcome the heavily demanded computations through the layers in the training step. These computations make them the most time-consuming section among all. The proposed architecture provides scientists with a pure hardware realization opportunity. It is a low-power analog design for the hyperbolic tangent activation function which has the simplest architecture up to now including only two MOSFETs and one resistor after a sign unit. Accuracy results on MNIST and Fashion-MNIST datasets, power consumption computation, reliable noise frequency analysis, and easy implementation of this design outperforms most of the similar architectures in literature. The other important advantage of the proposed design is its compatibility with CMOS-based architectures especially the recent MNNs [29, 300]. It is experimentally proven that the power dissipation of the proposed analog design for *Tanh*, and its small area usage are the two significant improvements as compared to the existing digital designs for this important activation function, while the accuracy is not affected.

7.2 Limitations

After conducting all the research and investigation in the deep learning methodologies and their applications in industries, numerous problems are raised for which there is still no solution, such as the high computational load of deep neural networks, training the neural networks while having no or limited amount of labels, the usage area of deep neural networks, finding the compatible neural network while facing a new dataset, and etc. There are also some novel ideas to build neural networks using LEDs which makes them more efficient with less area and power usage [301]. The above mentioned problems and novel ideas should be further studied and analyzed in deep learning area.

In this section, some of the existing limitations in deep learning and power system fields are discussed.

1. Although the proposed intelligent techniques in this work result in a great promise in the areas of power system and smart grid, applying these AI-based method to industry and taking advantage of them in reality remain an open research and development field.

2. Deep learning methodologies are usually computationally expensive, and complex architectures such as VGG-19 should be used with extra attention in reality. There are always some solutions for reduction of the computational load in deep learning method, such as different optimization ideas, which can be applied to the existing studies.

3. Using transfer learning technique to generalize fault diagnosis of transmission lines is dependent on the firstly given length and might not be applicable to more complex scenarios. To generate more robust techniques we always need to consider possible variations.

4. Considering the two generator model of a transmission line is a special case and there are much more complex networks in power systems. Having more than two generator models in these systems can affect the performance and the robustness of the proposed fault diagnosis method.

5. There are always outliers, or mis-recorded or missing data in real datasets, which are not encountered in the data only based on simulations. Hence, being cautious about data related issues in real world is vital.

7.3 Future Work Directions

Considering the above-mentioned limitations, we can take advantage of some studies which help us to handle the outliers or newly emerged data points such as those in [302, 303]. Despite the fact that the approaches in this dissertation are inclusive and accurate for fault diagnosis in TLs, majority of the reviewed papers fail to propose solutions for some remarkable problems such as:

- Variations of voltage amplitude and phase changes of generators
- Additive noise
- False detection of high-impedance TL faults [304]
- Shortage of fault data in the real world for the training step

Besides the above-mentioned issues, the innovations and advancements, which are continuously happening in power system monitoring and control, increase the need for higher level ML methods such as hybrid methods [305, 306], transfer learning [307, 175], federated learning [308], edge computing [309], and edge ML [309].

Not only the datasets for TL fault detection and location estimation are time series and unbalanced, but also there exist numerous different types of power distribution networks. Therefore, new hybrid approaches can help with the reduction of training and testing times, and the probability of the over/underfitting [310, 311]. Besides, enormous TL networks in large cities cause huge datasets that are high-cost and unsafe to be transferred. Consequently, the emerging approaches based on "federated learning" can develop methodologies that provide the advanced learning ability without transferring the actual datasets [312, 313].

Edge ML is a revolutionary technique, which resolves security concerns pertaining to storing high-risk information from power systems in the cloud, and reduces strain on cloud networks by processing data locally. This methodology enables the processing of data in real-time, which is a required feature in solutions for TL fault detection problems [148]. It is also used to solve the issues with the low capability of finding fault features for line-to-ground or line-to-line faults, because these faults have complex models according to their numerous modes [314]. Consequently, edge ML assists the complex implementation of recently emerging power TL technologies to become efficient by using cloud computing [315].

Some recent studies such as [314] have used edge computing methods to overcome the TL fault detection issues, such as the low capability of finding fault features for line-to-ground or line-to-line faults. These faults require multiple mode components for detection of their types which complicate the model. Therefore, implementation of computationally intensive power TL technologies can be done efficiently by using edge computing techniques [315]. Moreover, the emerging "transfer learning" methods can reduce the training and testing times of NNs by using different parts of other already trained NNs based on existing labeled datasets [316, 205, 317, 318, 319, 203, 202, 320, 206, 321].

Although in the field of TL protection, finding the locations of faults is as important as detecting and identifying the types of faults, the papers that focus on all 3 tasks (detection, classification, and location estimation) are fewer than those that only solve the fault type classification problem. As well as finding solutions for these three problems of TL faults, an all-embracing analysis of the robustness for risky parameters of generators such as variations of power flow angle, voltage amplitudes, and phase changes is a matter of serious concern and a future research direction.

Advanced techniques in meta-learning, continual learning, and incremental learning can be considered in intelligent fault diagnosis of power systems. Applying deep transfer learning imagery datasets to TLs for various purposes is another interesting path to follow. How to perform fault detection for extended power systems with more than 2 generators remains an open problem. Multiple optimization methods, e.g., should be adopted [322, 323, 324, 325, 326, 327]. Taking advantage of edge ML method to achieve higher speed in TL fault diagnosis might be helpful. Some recent methods, e.g., those in [325, 326, 323, 327, 322, 324], can be adapted for this purpose. Hybrid methods such as Convolutional Neural Network (CNN)-k-Nearest Neighbour (k-NN) to solve the problem of TL fault location estimation problem that is a challenge nowadays, can be a new direction to study. Light-weight CNN-based methods, e.g., [328, 329], can provide a reliable solution to this problem.

REFERENCES

[1] Convolutional neural networks for visual recognition. http://cs231n.github.io/neural-networks-1/.

[2] Krishna Prasad Gnawali, Bijay Raj Paudel, Seyed Nima Mozaffari, and Spyros Tragoudas. Reliability enhancements in memristive neural network architectures. *IEEE Transactions on Nanotechnology*, 18:866–878, 2019 Aug 13.

[3] Hui Pang Yunfeng Li Longlong Chen Xiao Ding Guangfu Tang Pei, Xiangyu. A novel ultra-high-speed traveling-wave protection principle for vsc-based dc grids. *IEEE Access*, 7:119765–119773, 2019.

[4] Ilhan Kocar Jean Mahseredjian Abner Ramirez Cervantes, Miguel. A traveling wave based fault location method using unsynchronized current measurements. In *IEEE Power and Energy Society General Meeting (PESGM)*, pages 1–1, 2019.

[5] Xin Wang Yihui Zheng Lixue Li Ding, Jiali. Distributed traveling-wave-based fault-location algorithm embedded in multiterminal transmission lines. *IEEE Transactions on Power Delivery*, 33(6):3045–3054, 2018.

[6] Yanjun Ma, Haifeng Li, Gang Wang, and Jiyang Wu. Fault analysis and traveling-wave-based protection scheme for double-circuit lcc-hvdc transmission lines with shared towers. *IEEE Transactions on Power Delivery*, 33(3):1479–1488, 2018.

[7] Felipe V Lopes, Paulo Lima, João Paulo G Ribeiro, Tiago Rocha Honorato, Kleber Melo Silva, Eduardo Jorge S Leite, Washington Luiz Araujo Neves, and Geraldo Rocha. Practical methodology for two-terminal traveling wave-based fault location eliminating the need for line parameters and time synchronization. *IEEE Transactions on Power Delivery*, 34(6):2123–2134, 2019.

[8] Chenhao Zhang, Guobing Song, Ting Wang, and Liming Yang. Single-ended traveling wave fault location method in dc transmission line based on wave front information. *IEEE Transactions on Power Delivery*, 34(5):2028–2038, 2019.

[9] Felipe V Lopes and Eduardo JS Leite. Traveling wave-based solutions for transmission line two-terminal data time synchronization. *IEEE Transactions on Power Delivery*, 33(6):3240–3241, 2018.

[10] L De Andrade and T Ponce de Leão. Impedance-based fault location analysis for transmission lines. In *IEEE PES TandD*, pages 1–6, 2012.

[11] Sophi Shilpa Gururajapathy, Hazlie Mokhlis, and Hazlee Azil Illias. Fault location and detection techniques in power distribution systems with distributed generation: A review. *Renewable and sustainable energy reviews*, 74:949–958, 2017.

[12] Ebha Koley, Raunak Kumar, and Subhojit Ghosh. Low cost microcontroller based fault detector, classifier, zone identifier and locator for transmission lines using wavelet transform and artificial neural network: A hardware co-simulation approach. *International Journal of Electrical Power and Energy Systems*, 81:346–360, 2016.

[13] Düzgün Akmaz, Mehmet Salih Mamiş, Müslüm Arkan, and Mehmet Emin Tağluk. Transmission line fault location using traveling wave frequencies and extreme learning machine. *Electric power systems research*, 155:1–7, 2018.

[14] Office of Nuclear Energy. Department of energy report explores u.s. advanced small modular reactors to boost grid resiliency, 2018.

[15] Introduction to artificial neural networks - part 1. http://www.theprojectspot.com/tutorial-post/introduction-to-artificial-neural-networks-part-1/7.

[16] Neural network history. https://stevenmiller888.github.io/mind-how-to-build-a-neural-network/.

[17] Neural network history. https://cs.stanford.edu/people/eroberts/courses/soco/projects / neural-networks/History/history2.html.

[18] Zhe Li, Ao Ren, Ji Li, Qinru Qiu, Bo Yuan, Jeffrey Draper, and Yanzhi Wang. Structural design optimization for deep convolutional neural networks using stochastic computing. pages 250–253, 2017.

[19] Hoon Chung, Sung Joo Lee, and Jeon Gue Park. Deep neural network using trainable activation functions. pages 348–352, 2016.

[20] Ligang Gao, Fabien Alibart, and Dmitri B Strukov. Analog-input analog-weight dot-product operation with ag/a-si/pt memristive devices. In *VLSI and System-on-Chip, 2012 (VLSI-SoC), IEEE/IFIP 20th International Conference on*, pages 88–93. IEEE, 2012.

[21] Mrigank Sharad, Deliang Fan, Kyle Aitken, and Kaushik Roy. Energy-efficient non-boolean computing with spin neurons and resistive memory. *IEEE Transactions on Nanotechnology*, 13(1):23–34, 2014.

[22] Hyongsuk Kim, Maheshwar Pd Sah, Changju Yang, Tamas Roska, and Leon O Chua. Memristor bridge synapses. *Proceedings of the IEEE*, 100(6):2061–2070, 2012.

[23] Fatemeh Mohammadi Shakiba. *CMOS-Based Implementation of Hyperbolic Tangent Activation Function for Artificial Neural Network*. Southern Illinois University at Carbondale, 2018.

[24] A gentle introduction to neural networks. https://towardsdatascience.com/a-gentle-introduction-to-neural-networks-series-part-1-2b90b87795bc.

[25] Neural networks- chapter 10. http://natureofcode.com/book/chapter-10-neural-networks/.

[26] Overview of artificial neural networks and their applications. https://www.xenonstack.com/blog/data-science/overview-of-artificial-neural-networks-and-its-applications.

[27] Neural networks and machine learning. http://watson.latech.edu/WatsonRebootTest / ch14s3p3.html.

[28] Activation functions. https://towardsdatascience.com/activation-functions-and-its-types-which-is-better-a9a5310cc8f.

[29] Krishna Prasad Gnawali, Seyed Nima Mozaffari, and Spyros Tragoudas. Low power artificial neural network architecture. *arXiv*, pages arXiv–1904, 2019.

[30] Julie Grollier, Damien Querlioz, and Mark D Stiles. Spintronic nanodevices for bioinspired computing. *Proceedings of the IEEE*, 104(10):2024–2039, 2016.

[31] Zhe Li, Ao Ren, Ji Li, Qinru Qiu, Bo Yuan, Jeffrey Draper, and Yanzhi Wang. Structural design optimization for deep convolutional neural networks using stochastic computing. In *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2017*, pages 250–253. IEEE, 2017.

[32] Hoon Chung, Sung Joo Lee, and Jeon Gue Park. Deep neural network using trainable activation functions. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 348–352. IEEE, 2016.

[33] Andrei Dinu, Marcian N Cirstea, and Silvia E Cirstea. Direct neural-network hardware-implementation algorithm. *IEEE Transactions on Industrial Electronics*, 57(5):1845–1848, 2009.

[34] KV Ramanaiah and SIRIPURAPU Sridhar. Hardware implementation of artificial neural networks. *I-Manager's Journal on Embedded Systems*, 3(4):31, 2014.

[35] Guang Feng, Boris Deriy, and Ju Wang. A linear mosfet regulator for improving performance of the booster ramping power supplies at the aps. In *2007 IEEE Particle Accelerator Conference (PAC)*, pages 434–436. IEEE, 2007.

[36] Long Wen, Xinyu Li, Liang Gao, and Yuyan Zhang. A new convolutional neural network-based data-driven fault diagnosis method. *IEEE Transactions on Industrial Electronics*, 65(7):5990–5998, 2017.

[37] Dong Yu and Jinyu Li. Recent progresses in deep learning based acoustic models (updated). *arXiv preprint arXiv:1804.09298*, 2018.

[38] O Krestinskaya, B Choubey, and AP James. Memristive gan in analog. *Scientific Reports*, 10(1):1–14, 2020.

[39] Gang Bao, Yide Zhang, and Zhigang Zeng. Memory analysis for memristors and memristive recurrent neural networks. *IEEE/CAA Journal of Automatica Sinica*, 7(1):96–105, 2019.

[40] Hacer A Yildiz, Mustafa Altun, Ali Dogus Gungordu, and Mircea R Stan. Analog neural network based on memristor crossbar arrays. In *2019 11th International Conference on Electrical and Electronics Engineering (ELECO)*, pages 358–361. IEEE, 2019.

[41] Mrigank Sharad, Deliang Fan, Kyle Aitken, and Kaushik Roy. Energy-efficient non-boolean computing with spin neurons and resistive memory. *IEEE Transactions on Nanotechnology*, 13(1):23–34, 2013.

[42] Catherine D Schuman, Thomas E Potok, Robert M Patton, J Douglas Birdwell, Mark E Dean, Garrett S Rose, and James S Plank. A survey of neuromorphic computing and neural networks in hardware. *arXiv preprint arXiv:1705.06963*, 2017 May 19.

[43] Raqibul Hasan, Tarek M Taha, and Chris Yakopcic. On-chip training of memristor based deep neural networks. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 3527–3534. IEEE, 2017.

[44] Ivan Sanchez Esqueda, Huan Zhao, and Han Wang. Efficient learning and crossbar operations with atomically-thin 2-d material compound synapses. *Journal of Applied Physics*, 124(15):152133, 2018.

[45] Manu V Nair and Piotr Dudek. Gradient-descent-based learning in memristive crossbar arrays. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2015.

[46] Arindam Basu, Jyotibdha Acharya, Tanay Karnik, Huichu Liu, Hai Li, Jae-Sun Seo, and Chang Song. Low-power, adaptive neuromorphic systems: Recent progress and future directions. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 8(1):6–27, 2018.

[47] Chenchen Liu, Fuqiang Liu, and Hai Li. Brain-inspired computing accelerated by memristor technology. In *Proceedings of the 4th ACM International Conference on Nanoscale Computing and Communication*, pages 1–6, 2017.

[48] Hyongsuk Kim, Maheshwar Pd Sah, Changju Yang, Tamás Roska, and Leon O Chua. Memristor bridge synapses. *Proceedings of the IEEE*, 100(6):2061–2070, 2011.

[49] Xiaofan Li, Jian-an Fang, and Huiyuan Li. Exponential stabilization of stochastic memristive recurrent neural networks under periodically intermittent state feedback control. *Asian Journal of Control*, 22(2):897–907, 2020.

[50] Shaghayegh Vahdat, Mehdi Kamal, Ali Afzali-Kusha, and Massoud Pedram. Interstice: Inverter-based memristive neural networks discretization for

function approximation applications. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2020.

[51] Zhongrui Wang, Saumil Joshi, Sergey Savel'ev, Wenhao Song, Rivu Midya, Yunning Li, Mingyi Rao, Peng Yan, Shiva Asapu, Ye Zhuo, et al. Fully memristive neural networks for pattern classification with unsupervised learning. *Nature Electronics*, 1(2):137–145, 2018.

[52] Wen-Qian Pan, Jia Chen, Rui Kuang, Yi Li, Yu-Hui He, Gui-Rong Feng, Nian Duan, Ting-Chang Chang, and Xiang-Shui Miao. Strategies to improve the accuracy of memristor-based convolutional neural networks. *IEEE Transactions on Electron Devices*, 67(3):895–901, 2020.

[53] Indranil Chakraborty, Mustafa Fayez Ali, Dong Eun Kim, Aayush Ankit, and Kaushik Roy. Geniex: A generalized approach to emulating non-ideality in memristive xbars using neural networks. *arXiv preprint arXiv:2003.06902*, 2020 Mar 15.

[54] Irem Boybat, Carmelo di Nolfo, Stefano Ambrogio, Martina Bodini, Nathan CP Farinha, Robert M Shelby, Pritish Narayanan, Severin Sidler, Hsinyu Tsai, Yusuf Leblebici, et al. Improved deep neural network hardware-accelerators based on non-volatile-memory: The local gains technique. In *2017 IEEE International Conference on Rebooting Computing (ICRC)*, pages 1–8. IEEE, 2017.

[55] Nolman Barroso Hartmann, Ricardo Caneloi dos Santos, Ahda Pionkoski Grilo, and José Carlos Melo Vieira. Hardware implementation and real-time evaluation of an ann-based algorithm for anti-islanding protection of distributed generators. *IEEE Transactions on Industrial Electronics*, 65(6):5051–5059, 2017.

[56] Ali Raza, Abdeldjabar Benrabah, Thamer Alquthami, and Muhammad Akmal. A review of fault diagnosing methods in power transmission systems. *Applied Sciences*, 10(4):1312, 2020.

[57] James Hare, Xiaofang Shi, Shalabh Gupta, and Ali Bazzi. Fault diagnostics in smart micro-grids: A survey. *Renewable and Sustainable Energy Reviews*, 60:1114–1124, 2016.

[58] M. Kezunovic. Smart fault location for smart grids. *IEEE Transactions on Smart Grid*, 2(1):11–22, 2011.

[59] M Paolone, E Petrache, F Rachidi, CA Nucci, VA Rakov, MA Uman, D Jordan, K Rambo, J Jerauld, M Nyffeler, et al. Lightning induced disturbances in buried cables-part ii: experiment and model validation. *IEEE transactions on electromagnetic compatibility*, 47(3):509–520, 2005.

[60] Mandy Sidana. Intro to types of classification algorithms in machine learning, Apr 2020.

[61] Mohammad Waseem. Data science with python, 2022.

[62] Javaid Nabi. Machine learning — multiclass classification with imbalanced dataset, 2018.

[63] François Fleuret, Ting Li, Charles Dubout, Emma K Wampler, Steven Yantis, and Donald Geman. Comparing machines and humans on a visual categorization test. *Proceedings of the National Academy of Sciences*, 108(43):17621–17625, 2011.

[64] Baptiste Rocca. Imbalanced ds, 2019.

[65] Fei Wu, Xiao-Yuan Jing, Shiguang Shan, Wangmeng Zuo, and Jing-Yu Yang. Multiset feature learning for highly imbalanced data classification. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[66] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[67] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[68] Nguyen Thai-Nghe, Zeno Gantner, and Lars Schmidt-Thieme. Cost-sensitive learning methods for imbalanced data. In *The 2010 International joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2010.

[69] M Tarafdar Haque and AM Kashtiban. Application of neural networks in power systems; a review. *International Journal of Energy and Power Engineering*, 1(6):897–901, 2007.

[70] Avagaddi Prasad, J Belwin Edward, and K Ravi. A review on fault classification methodologies in power transmission systems: Part II. *Journal of Electrical Systems and Information Technology*, 5(1):61–67, 2018.

[71] Avagaddi Prasad, J Belwin Edward, and K Ravi. A review on fault classification methodologies in power transmission systems: Part I. *Journal of electrical systems and information technology*, 5(1):48–60, 2018.

[72] Debani Prasad Mishra and Papia Ray. Fault detection, location and classification of a transmission line. *Neural Computing and Applications*, 30(5):1377–1424, 2018.

[73] Avagaddi Prasad and J Belwin Edward. Importance of artificial neural networks for location of faults in transmission systems: a survey. *2017 11th International Conference on Intelligent Systems and Control (ISCO)*, pages 357–362, 2017.

[74] Vikramsingh R Parihar and Snehal Warudkar. An overview of transmission line fault detection techniques. *Int. J. of Innovative Res. and Studies (IJIRS)*, 8:64–77, 2017.

[75] Avagaddi Prasad, J. Belwin Edward, and K. Ravi. A review on fault classification methodologies in power transmission systems: Part—i. *Journal of Electrical Systems and Information Technology*, 5(1):48–60, 2018.

[76] Shuang-cheng Wang, Rui Gao, and Li-min Wang. Bayesian network classifiers based on gaussian kernel density. *Expert Systems with Applications*, 51:207–217, 2016.

[77] Paula Renatha Nunes da Silva, Hossam A Gabbar, Petrônio Vieira Junior, and Carlos Tavares da Costa Junior. A new methodology for multiple incipient fault diagnosis in transmission lines using qta and naive bayes classifier. *International Journal of Electrical Power and Energy Systems*, 103:326–346, 2018.

[78] Muhammad Sohail Ibrahim, Wei Dong, and Qiang Yang. Machine learning driven smart electric power systems: Current trends and new perspectives. *Applied Energy*, 272:115237, 2020.

[79] Sylvain Verron, Teodor Tiplica, and Abdessamad Kobi. Fault detection with bayesian network. 2008.

[80] Huan Zhang, Liangxiao Jiang, and Liangjun Yu. Attribute and instance weighted naive bayes. *Pattern Recognition*, 111:107674, 2021.

[81] Aleena Swetapadma and Anamika Yadav. Protection of parallel transmission lines including inter-circuit faults using na

[82] Ahmed R Adly, Ragab A El Sehiemy, Almoataz Youssef Abdelaziz, and Nabil MA Ayad. Critical aspects on wavelet transforms based fault identification procedures in hv transmission line. *IET Generation, Transmission and Distribution*, 10(2):508–517, 2016.

[83] FE Perez, E Orduna, and G Guidi. Adaptive wavelets applied to fault classification on transmission lines. *IET generation, transmission and distribution*, 5(7):694–702, 2011.

[84] FE Perez, Ra Aguilar, E Orduna, J Jäger, and G Guidi. High-speed non-unit transmission line protection using single-phase measurements and an adaptive wavelet: zone detection and fault classification. *IET generation, transmission and distribution*, 6(7):593–604, 2012.

[85] Anamika Yadav and Aleena Swetapadma. Combined dwt and naive bayes based fault classifier for protection of double circuit transmission line. *Int. Conf. on Recent Advances and Innovations in Eng. (ICRAIE-2014)*, pages 1–6, 2014.

[86] Elhadi Aker, Mohammad Lutfi Othman, Veerapandiyan Veerasamy, Ishak bin Aris, Noor Izzri Abdul Wahab, and Hashim Hizam. Fault detection and classification of shunt compensated transmission line using discrete wavelet transform and naive bayes classifier. *Energies*, 13(1):243, 2020.

[87] Elhadi Aker, Mohammad Lutfi Othman, Ishak Aris, Noor Izzri Abdul Wahab, Hashim Hizam, and Osaj Emmanuel. Transmission line fault identification and classification with integrated facts device using multiresolution analysis and naive bayes classifier. *International Journal of Power Electronics and Drive Systems*, 11(2):907, 2020.

[88] Revati Godse and Sunil Bhat. Mathematical morphology-based feature-extraction technique for detection and classification of faults on power transmission line. *IEEE Access*, 8:38459–38471, 2020.

[89] Ehsan Aliyan, Mohammadreza Aghamohammadi, Mohsen Kia, Alireza Heidari, Miadreza Shafie-khah, and João PS Catalão. Decision tree analysis to identify harmful contingencies and estimate blackout indices for predicting system vulnerability. *Electric Power Systems Research*, 178:106036, 2020.

[90] Subodh Kumar Mohanty, Anshudip Karn, and Shobhan Banerjee. Decision tree supported distance relay for fault detection and classification in a series compensated line. In *2020 IEEE international conference on power electronics, smart grid and renewable energy (PESGRE2020)*, pages 1–6. IEEE, 2020.

[91] Iqbal H Sarker, Alan Colman, Jun Han, Asif Irshad Khan, Yoosef B Abushark, and Khaled Salah. Behavdt: a behavioral decision tree learning to build user-centric context-aware predictive model. *Mobile Networks and Applications*, 25(3):1151–1161, 2020.

[92] Wenliang Du and Zhijun Zhan. Building decision tree classifier on private data. *College of Electrical Engineering and Computer Science*, 8, 2002.

[93] Sunil K Shukla, Ebha Koley, and Subhojit Ghosh. Dc offset estimation-based fault detection in transmission line during power swing using ensemble of decision tree. *IET Science, Measurement and Technology*, 13(2):212–222, 2018.

[94] BK Chaitanya and Anamika Yadav. Decision tree aided travelling wave based fault section identification and location scheme for multi-terminal transmission lines. *Measurement*, 135:312–322, 2019.

[95] Praveen Kumar Mishra, Anamika Yadav, and Mohammad Pazoki. A novel fault classification scheme for series capacitor compensated transmission line based on bagged tree ensemble classifier. *IEEE Access*, 6:27373–27382, 2018.

[96] Mir Mohammad Taheri, Heresh Seyedi, and Behnam Mohammadi-ivatloo. Dt-based relaying scheme for fault classification in transmission lines using modp. *IET Generation, Transmission & Distribution*, 11(11):2796–2804, 2017.

[97] Haoyue Liu, MengChu Zhou, and Qing Liu. An embedded feature selection method for imbalanced data classification. *IEEE/CAA J. of Automatica Sinica*, 6(3):703–715, 2019.

[98] Zhang Bingzhen, Qiao Xiaoming, Yang Hemeng, and Zhou Zhubo. A random forest classification model for transmission line image processing. In *2020 15th International Conference on Computer Science & Education (ICCSE)*, pages 613–617. IEEE, 2020.

[99] Z. Hongbin, W. Wei, and S. Wengdong. Safety and damage assessment method of transmission line tower in Goaf based on artificial intelligence. *2020 IEEE/IAS Industrial and Commercial Power System Asia (I CPS Asia)*, pages 1474–1479, 2020.

[100] Pranabesh Bala and Sovan Dalai. Random forest based fault analysis method in IEEE 14 bus system. *2017 3rd International Conference on Condition Assessment Techniques in Electrical Systems (CATCON)*, pages 407–411, 2017.

[101] Mahesh Pal. Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, 26(1):217–222, 2005.

[102] Zhendong Yin, Li Wang, Yaojia Zhang, and Yang Gao. A novel arc fault detection method integrated random forest, improved multi-scale permutation entropy and wavelet packet transform. *Electronics*, 8(4):396, 2019.

[103] Hao Wu, Qiaomei Wang, Kunjian Yu, Xiaotao Hu, and Maoxia Ran. A novel intelligent fault identification method based on random forests for hvdc transmission lines. *PloS one*, 15(3):e0230717, 2020.

[104] Mohammad Hossein Rezaeian Koochi, Saeid Esmaeili, and Roohollah Fadaeinedjad. New phasor-based approach for online and fast prediction of generators grouping using decision tree. *IET Generation, Transmission and Distribution*, 11(6):1566–1574, 2017.

[105] Gabriel A Fonseca, Danton D Ferreira, Flávio B Costa, and Aryfrance R Almeida. Fault classification in transmission lines using random forest and notch filter. *Journal of Control, Automation and Electrical Systems*, pages 1–12, 2021.

[106] Hao Zhang, Yongdan Li, Zhihan Lv, Arun Kumar Sangaiah, and Tao Huang. A real-time and ubiquitous network attack detection based on deep belief network and support vector machine. *IEEE/CAA Journal of Automatica Sinica*, 7(3):790–799, 2020.

[107] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

[108] Dmitry Krupenev, Denis Boyarkin, and Dmitrii Iakubovskii. Improvement in the computational efficiency of a technique for assessing the reliability of electric power systems based on the monte carlo method. *Reliability Engineering and System Safety*, 204:107171, 2020.

[109] Bikash Patel. A new fdost entropy based intelligent digital relaying for detection, classification and localization of faults on the hybrid transmission line. *Electric Power Systems Research*, 157:39–47, 2018.

[110] M Jaya Bharata Reddy, P Gopakumar, and DK Mohanta. A novel transmission line protection using dost and svm. *Engineering Science and Technology, an International Journal*, 19(2):1027–1039, 2016.

[111] Papia Ray and Debani Prasad Mishra. Support vector machine based fault classification and location of a long transmission line. *Eng. science and technology, an Int. journal*, 19(3):1368–1380, 2016.

[112] Zahra Moravej, Mohammad Pazoki, and Mojtaba Khederzadeh. New pattern-recognition method for fault analysis in transmission line with upfc. *IEEE Transactions on Power Delivery*, 30(3):1231–1242, 2014.

[113] Ong Wei Chuan, Nur Fadilah Ab Aziz, Zuhaila Mat Yasin, Nur Ashida Salim, and Norfishah A Wahab. Fault classification in smart distribution network using support vector machine. *Indonesian Journal of Electrical Engineering and Computer Science*, 2020.

[114] Ms Rashmi Singh, Tarun Chopra, R Singh, and T Chopra. Fault classification in electric power transmission lines using support vector machine. *Int. J. Innov. Res. Sci. Technol.*, 1(12):388–400, 2015.

[115] Pathirikkat Gopakumar, Maddikara Jaya Bharata Reddy, and Dusmanta Kumar Mohanta. Transmission line fault detection and localisation methodology using pmu measurements. *IET Generation, Transmission & Distribution*, 9(11):1033–1042, 2015.

[116] Nantian Huang, Jiajin Qi, Fuqing Li, Dongfeng Yang, Guowei Cai, Guilin Huang, Jian Zheng, and Zhenxin Li. Short-circuit fault detection and classification using empirical wavelet transform and local energy for electric transmission line. *Sensors*, 17(9):2133, 2017.

[117] Aleena Swetapadma and Anamika Yadav. Directional relaying using support vector machine for double circuit transmission lines including cross-country and inter-circuit faults. *Int. J. of Electrical Power and Energy Systems*, 81:254–264, 2016.

[118] Jenifer Mariam Johnson and Anamika Yadav. Complete protection scheme for fault detection, classification and location estimation in hvdc transmission lines using support vector machines. *IET Science, Measurement and Technology*, 11(3):279–287, 2016.

[119] Amit Kumar Gangwar, Om Prakash Mahela, Bhuvnesh Rathore, Baseem Khan, Hassan Haes Alhelou, and Pierluigi Siano. A novel k-means clustering and weighted k-nn-regression-based fast transmission line protection. *IEEE Transactions on Industrial Informatics*, 17(9):6034–6043, 2020.

[120] Jiawei Han, Micheline Kamber, and Jian Pei. Data mining concepts and techniques third edition. *The Morgan Kaufmann Series in Data Management Systems*, 5(4):83–124, 2011.

[121] Aleena Swetapadma and Anamika Yadav. A novel single-ended fault location scheme for parallel transmission lines using k-nearest neighbor algorithm. *Computers and Electrical Eng.*, 69:41–53, 2018.

[122] Sahendara Kumar and Ebha Koley. A k-nearest neighbour based protection scheme for upfc compensated double circuit transmission line with wind farms. *2020 IEEE First Intl. Conf. on Smart Technologies for Power, Energy and Control (STPEC)*, pages 1–5, 2020.

[123] M. Farshad and J. Sadeh. Accurate single-phase fault-location method for transmission lines based on k-nearest neighbor algorithm using one-end voltage. *IEEE Transactions on Power Delivery*, 27(4):2360–2367, 2012.

[124] Aida Asadi Majd, Haidar Samet, and Teymoor Ghanbari. k-nn based fault detection and classification methods for power transmission systems. *Protection and Control of Modern Power Systems*, 2(1):1–11, 2017.

[125] Mat Nizam Mahmud, Mohammad Nizam Ibrahim, Muhammad Khusairi Osman, and Zakaria Hussain. Fault classification in transmission line using wavelet features and fuzzy-knn. *Applied Mechanics and Materials*, 850:112–117, 2016.

[126] Aritra Dasgupta, Sudipta Debnath, and Arabinda Das. Transmission line fault detection and classification using cross-correlation and k-nearest neighbor. *International Journal of Knowledge-based and Intelligent Engineering Systems*, 19(3):183–189, 2015.

[127] Sunil Kumar Singh, DN Vishwakarma, and RK Saket. An intelligent scheme for categorising fault events in compensated power network using k-nearest neighbour technique. *Intl. J. of Power and Energy Conversion*, 11(4):352–368, 2020.

[128] Tamer S Abdelgayed, Walid G Morsi, and Tarlochan S Sidhu. Fault detection and classification based on co-training of semisupervised machine learning. *IEEE Transactions on Industrial Electronics*, 65(2):1595–1605, 2017.

[129] Aleena Swetapadma, Praveen Mishra, Anamika Yadav, and Almoataz Y Abdelaziz. A non-unit protection scheme for double circuit series capacitor compensated transmission lines. *Electric Power Systems Research*, 148:311–325, 2017.

[130] Saber Ghashghaei and Mahdi Akhbari. Fault detection and classification of an hvdc transmission line using a heterogenous multi-machine learning algorithm. *IET Generation, Transmission and Distribution*, 2021.

[131] Neha Gupta. Artificial neural network. *Network and Complex Systems*, 3(1):24–28, 2013.

[132] Fatemeh Mohammadi Shakiba and MengChu Zhou. Novel analog implementation of a hyperbolic tangent neuron in artificial neural networks. *IEEE Transactions on Industrial Electronics*, 68(11):10856–10867, 2021.

[133] Ujjaval Patel, Praghnesh Bhatt, and Nilesh Chothani. Transmission line protection philosophy. In *Futuristic Trends in Numerical Relaying for Transmission Line Protections*, pages 1–19. Springer, 2021.

[134] Fatemeh Mohammadi Shakiba and Mengchu Zhou. Novel analog implementation of a hyperbolic tangent neuron in artificial neural networks. *IEEE Transactions on Industrial Electronics*, 2020.

[135] Anamika Yadav and Aleena Swetapadma. A single ended directional fault section identifier and fault locator for double circuit transmission lines using combined wavelet and ANN approach. *Int. J. of Electrical Power and Energy Systems*, 69:27–33, 2015.

[136] Abdullah Asuhaimi Mohd Zin, Makmur Saini, Mohd Wazir Mustafa, Ahmad Rizal Sultan, et al. New algorithm for detection and fault classification on parallel transmission line using dwt and bpnn based on clarke's transformation. *Neurocomputing*, 168:983–993, 2015.

[137] Ei Phyo Thwe and Min Min Oo. Fault detection and classification for transmission line protection system using artificial neural network. *Journal of Electrical and Electronic Engineering*, 4(5):89–96, 2016.

[138] M Gowrishankar, P Nagaveni, and P Balakrishnan. Transmission line fault detection and classification using discrete wavelet transform and artificial neural network. *Middle-East J. Sci. Res*, 24(4):1112–1121, 2016.

[139] JQ James, Yunhe Hou, Albert YS Lam, and Victor OK Li. Intelligent fault detection scheme for microgrids with wavelet-based deep neural networks. *IEEE Transactions on Smart Grid*, 10(2):1694–1703, 2017.

[140] Ebha Koley, Anamika Yadav, and Aniruddha Santosh Thoke. A new single-ended artificial neural network-based protection scheme for shunt faults in six-phase transmission line. *International Transactions on Electrical Energy Systems*, 25(7):1257–1280, 2015.

[141] AM Abdel-Aziz, BM Hasaneen, and AA Dawood. Detection and classification of one conductor open faults in parallel transmission line using artificial neural network. *International Journal of Scientific Research and Engineering Trends*, 2(6):139–146, 2016.

[142] Ebha Koley, Khushaboo Verma, and Subhojit Ghosh. An improved fault detection classification and location scheme based on wavelet transform and artificial neural network for six phase transmission line using single end data only. *SpringerPlus*, 4(1):1–22, 2015.

[143] Mat Nizam Mahmud, Mohammad Nizam Ibrahim, Muhammad Khusairi Osman, and Zakaria Hussain. A robust transmission line fault classification scheme using class-dependent feature and 2-tier multilayer perceptron network. *Electrical Engineering*, 100(2):607–623, 2018.

[144] Majid Jamil, Sanjeev Kumar Sharma, and Rajveer Singh. Fault detection and classification in electrical power transmission system using artificial neural network. *SpringerPlus*, 4(1):1–13, 2015.

[145] M Kiruthika and S Bindu. Classification of electrical power system conditions with convolutional neural networks. *Engineering, Technology & Applied Science Research*, 10(3):5759–5768, 2020.

[146] Liu Wuneng, Liu Lilong, Luo Changbing, and Li Bin. Research on transmission line defect detection based on deep learning. *CONVERTER*, 2021(6):854–861, 2021.

[147] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing Systems*, 25:1097–1105, 2012.

[148] Yanpeng Guo, Zhenjiang Pang, Jun Du, Fan Jiang, and Qilong Hu. An improved alexnet for power edge transmission line anomaly detection. *IEEE Access*, 8:97830–97838, 2020.

[149] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[150] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR, abs / 1409.1556*, 2014.

[151] Xusheng Lei and Zhehao Sui. Intelligent fault detection of high voltage line based on the faster r-cnn. *Measurement*, 138:379–385, 2019.

[152] Yanhai Wang, Qingquan Li, and Bo Chen. Image classification towards transmission line fault detection via learning deep quality-aware fine-grained categorization. *Journal of Visual Communication and Image Representation*, 64:102647, 2019.

[153] Zhiyong Dai, Jianjun Yi, Yajun Zhang, Bo Zhou, and Liang He. Fast and accurate cable detection using cnn. *Applied Intelligence*, 50(12):4688–4707, 2020.

[154] Shahriar Rahman Fahim, Yeahia Sarker, Subrata K Sarker, Md Rafiqul Islam Sheikh, and Sajal K Das. Self attention convolutional neural network with time series imaging based feature extraction for transmission line fault detection and classification. *Electric Power Systems Research*, 187:106437, 2020.

[155] Praveen Rai, Narendra D Londhe, and Ritesh Raj. Fault classification in power system distribution network integrated with distributed generators using cnn. *Electric Power Systems Research*, page 106914, 2020.

[156] Syifaul Fuada, Hasbi Ash Shiddieqy, and Trio Adiono. A high-accuracy of transmission line faults (tlfs) classification based on convolutional neural network. *International Journal of Electronics and Telecommunications*, 66(4):655–664, 2020.

[157] Sunil K Shukla, Ebha Koley, and Subhojit Ghosh. Grey wolf optimization-tuned convolutional neural network for transmission line protection with immunity against symmetrical and asymmetrical power swing. *Neural Computing and Applications*, 32(22):17059–17076, 2020.

[158] Kunjin Chen, Jun Hu, and Jinliang He. Detection and classification of transmission line faults based on unsupervised feature learning and convolutional sparse autoencoder. *IEEE Transactions on Smart Grid*, 9(3):1748–1758, 2016.

[159] Ali Yadollahpour, Jamshid Nourozi, Seyed Ahmad Mirbagheri, Eric Simancas-Acevedo, and Francisco R Trejo-Macotela. Designing and implementing an anfis based medical decision support system to predict chronic kidney disease progression. *Frontiers in physiology*, 9:1753, 2018.

[160] Muhammad Budi Rahayu Widodo, Adi Soeprijanto, and Ontoseno Penangsang. Analysis of fault location on distribution system using impulse injection learned by anfis. pages 38–43, 2020.

[161] Yusuf Kurtgoz and Emrah Deniz. Comparison of ANN, regression analysis, and ANFIS models in estimation of global solar radiation for different climatological locations. *Exergetic, Energetic and Environmental Dimensions*, pages 133–148, 2018.

[162] Mohammad Amin Jarrahi, Haidar Samet, Hossein Raayatpisheh, Ahmad Jafari, and Mohsen Rakhshan. An anfis-based fault classification approach in double-circuit transmission line using current samples. *International Work-Conference on Artificial Neural Networks*, pages 225–236, 2015.

[163] Amin Ghaghishpour and Amangaldi Koochaki. An intelligent method for online voltage stability margin assessment using optimized anfis and associated rules technique. *ISA transactions*, 102:91–104, 2020.

[164] Jie-Sheng Wang and Chen-Xu Ning. Anfis based time series prediction method of bank cash flow optimized by adaptive population activity pso algorithm. *Information*, 6(3):300–313, 2015.

[165] Ahmed Al-Hmouz, Jun Shen, Rami Al-Hmouz, and Jun Yan. Modeling and simulation of an adaptive neuro-fuzzy inference system (anfis) for mobile learning. *IEEE Transactions on Learning Technologies*, 5(3):226–237, 2011.

[166] Mohsen Hesami, Roohangiz Naderi, Masoud Tohidfar, and Mohsen Yoosefzadeh-Najafabadi. Application of adaptive neuro-fuzzy inference system-non-dominated sorting genetic algorithm-ii (anfis-nsgaii) for modeling and optimizing somatic embryogenesis of chrysanthemum. *Frontiers in plant science*, page 869, 2019.

[167] Vu Phan Huan and Nguyen Hoang Viet Le Kim Hung. Fault classification and location on 220 kv transmission line hoa khanh–hue using anfis net. *Journal of Automation and Control Engineering*, 3(2), 2015.

[168] Kingsley Monday UDOFIA and Chidinma NnekwuKALU. Fault detection, classification and location on 132 kV transmission line based on DWT and ANFIS. *surge (VS)*, 7(6), 2020.

[169] Ali Khaleghi and Mahmoud Oukati Sadegh. Single-phase fault location in four-circuit transmission lines based on wavelet analysis using anfis. *Journal of Electrical Engineering and Technology*, 14(4):1577–1584, 2019.

[170] Ibrahim Ismael. The distance relay by using anfis to detect faults in transmission line. *International Journal of Advancements in Research and Technology*, 5(9):14–21, 2016.

[171] Veerapandiyan Veerasamy, Noor Izzri Abdul Wahab, Rajeswari Ramachandran, Mariammal Thirumeni, Chitra Subramanian, Mohammad Lutfi Othman, and Hashim Hizam. High-impedance fault detection in medium-voltage distribution network using computational intelligence-based classifiers. *Neural Computing and Applications*, 31(12):9127–9143, 2019.

[172] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006.

[173] Düzgün Akmaz, Mehmet Salih Mamiş, Müslüm Arkan, and Mehmet Emin Tağluk. Transmission line fault location using traveling wave frequencies and extreme learning machine. *Electric power systems research*, 155:1–7, 2018.

[174] Yann Qi Chen, Olga Fink, and Giovanni Sansavini. Combined fault location and classification for power transmission lines fault diagnosis with integrated feature extraction. *IEEE Transactions on Industrial Electronics*, 65(1):561–569, 2017.

[175] Alok Mukherjee, Kingshuk Chatterjee, Palash Kumar Kundu, and Arabinda Das. Probabilistic neural network-aided fast classification of transmission line faults using differencing of current signal. *J. of The Institution of Engineers: Series B*, pages 1–14, 2021.

[176] Haishan Ye, Yujun Li, Cheng Chen, and Zhihua Zhang. Fast fisher discriminant analysis with randomized algorithms. *Pattern Recognition*, 72:82–92, 2017.

[177] Behshad Mohebali, Amirhessam Tahmassebi, Anke Meyer-Baese, and Amir H Gandomi. Probabilistic neural networks: a brief overview of theory, implementation, and application. *Handbook of Probabilistic Models*, pages 347–367, 2020.

[178] Yasha Zeinali and Brett A Story. Competitive probabilistic neural network. *Integrated Computer-Aided Eng.*, 24(2):105–118, 2017.

[179] Vincent Cheung and Kevin Cannons. An introduction to probabilistic neural networks. *Signal and Data Compression Laboratory Electrical and Computer Engineering University of Manitoba*, 2002.

[180] D. F. Specht. Generation of polynomial discriminant functions for pattern recognition. *IEEE Transactions on Electronic Computers*, EC-16(3):308–319, 1967.

[181] Vitor H Ferreira, Rainer Zanghi, Márcio Z Fortes, Sergio Gomes Jr, and Alexandre P Alves da Silva. Probabilistic transmission line fault diagnosis using autonomous neural models. *Electric Power Systems Research*, 185:106360, 2020.

[182] Nabamita Roy and Kesab Bhattacharya. Detection, classification, and estimation of fault location on an overhead transmission line using s-transform and neural network. *Electric Power Components and Systems*, 43(4):461–472, 2015.

[183] Alok Mukherjee, Palash Kumar Kundu, and Arabinda Das. Application of principal component analysis for fault classification in transmission line with ratio-based method and probabilistic neural network: a comparative analysis. *Journal of The Institution of Engineers (India): Series B*, 101(4):321–333, 2020.

[184] Hassan Fathabadi. Novel filter based ann approach for short-circuit faults detection, classification and location in power transmission lines. *International Journal of Electrical Power and Energy Systems*, 74:374–383, 2016.

[185] Arash Moradzadeh, Hamid Teimourzadeh, Behnam Mohammadi-Ivatloo, and Kazem Pourhossein. Hybrid cnn-lstm approaches for identification of type and locations of transmission line faults. *International Journal of Electrical Power and Energy Systems*, 135:107563, 2022.

[186] Pappan Balakrishnan and Singaram Gopinath. A new intelligent scheme for power system faults detection and classification: A hybrid technique. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, 33(5):e2728, 2020.

[187] Sunil Singh, DN Vishwakarma, Amit Kumar, and Shashank. A novel methodology for fault detection, classification and location in transmission system based on dwt & anfis. *Journal of Information and Optimization Sciences*, 38(6):791–801, 2017.

[188] K Gayathri and Narayanan Kumarappan. Double circuit ehv transmission lines fault location with rbf based support vector machine and reconstructed input scaled conjugate gradient based neural network. *Int. J. of Computational Intelligence Systems*, 8(1):95–105, 2015.

[189] H. Livani and C. Y. Evrenosoglu. A machine learning and wavelet-based fault location method for hybrid transmission lines. *IEEE Trans. on Smart Grid*, 5(1):51–59, 2014.

[190] A. Swetapadma and A. Yadav. A novel decision tree regression-based fault distance estimation scheme for transmission lines. *IEEE Trans. on Power Delivery*, 32(1):234–245, 2017.

[191] Mohammad Amin Jarrahi, Haidar Samet, and Teymoor Ghanbari. Fast current-only based fault detection method in transmission line. *IEEE Systems Journal*, 13(2):1725–1736, 2018.

[192] Peng Xi, Pan Feilai, Liang Yongchao, Luo Zhiping, and Li Long. Fault detection algorithm for power distribution network based on sparse self-encoding neural network. *2017 International Conference on Smart Grid and Electrical Automation (ICSGEA)*, pages 9–12, 2017.

[193] D. Lukach and R. Taylor. Transmission line applications of directional ground over current relays. *IEEE Power and Energy Society*, page 10, 2014.

[194] Shifei Ding, Xinzheng Xu, and Ru Nie. Extreme learning machine and its applications. *Neural Computing and Applications*, 25(3):549–556, 2014.

[195] E. Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33:1065–1076, 1962.

[196] Jingjing Dong, Wei Chen, and Chen Xu. Transmission line detection using deep convolutional neural network. In *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, pages 977–980. IEEE, 2019.

[197] IEEE guide for identifying and improving voltage quality in power systems. *IEEE Std 1250 (Revision of IEEE Std 1250-2011)*, page 27, 2018.

[198] V. Malathi, N.S. Marimuthu, S. Baskar, and K. Ramar. Application of extreme learning machine for series compensated transmission line protection. *Engineering Applications of Artificial Intelligence*, 24(5):880–887, 2011.

[199] Ruifan Li, Haoyu Liang, Yihui Shi, Fangxiang Feng, and Xiaojie Wang. Dual-CNN: A convolutional language decoder for paragraph image captioning. *Neurocomputing*, 396:92–101, 2020.

[200] Peishu Wu, Han Li, Nianyin Zeng, and Fengping Li. FMD-Yolo: An efficient face mask detection method for covid-19 prevention and control in public. *Image and Vision Computing*, 117:104341, 2022.

[201] Hasbi Ash Shiddieqy, Farkhad Ihsan Hariadi, and Trio Adiono. Power line transmission fault modeling and dataset generation for ai based automatic detection. *2019 International Symposium on Electronics and Smart Devices (ISESD)*, pages 1–5, 2019.

[202] Elene Firmeza Ohata, Gabriel Maia Bezerra, Joao Victor Souza das Chagas, Aloisio Vieira Lira Neto, Adriano Bessa Albuquerque, Victor Hugo C de Albuquerque, and Pedro Pedrosa Reboucas Filho. Automatic detection of covid-19 infection using chest X-ray images through transfer learning. *IEEE/CAA Journal of Automatica Sinica*, 8(1):239–248, 2020.

[203] Qi Kang, SiYa Yao, MengChu Zhou, Kai Zhang, and Abdullah Abusorrah. Effective visual domain adaptation via generative adversarial distribution matching. *IEEE Transactions on neural networks and learning systems*, 32(9):3919–3929, 2020.

[204] Guanyu Cai, Yuqin Wang, Lianghua He, and MengChu Zhou. Unsupervised domain adaptation with adversarial residual transform networks. *IEEE transactions on neural networks and learning systems*, 31(8):3073–3086, 2019.

[205] Guanyu Cai, Lianghua He, MengChu Zhou, Hesham Alhumade, and Die Hu. Learning smooth representation for unsupervised domain adaptation. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[206] GongMing Wang, JunFei Qiao, Jing Bi, WenJing Li, and MengChu Zhou. Tl-gdbn: Growing deep belief network with transfer learning. *IEEE Transactions on Automation Science and Engineering*, 16(2):874–885, 2018.

[207] Min Zeng, Min Li, Zhihui Fei, Ying Yu, Yi Pan, and Jianxin Wang. Automatic ICD-9 coding via deep transfer learning. *Neurocomputing*, 324:43–50, 2019.

[208] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Transfer learning for time series classification. *IEEE international conference on big data (Big Data)*, pages 1367–1376, 2018.

[209] Liang Guo, Yaguo Lei, Saibo Xing, Tao Yan, and Naipeng Li. Deep convolutional transfer learning network: A new method for intelligent fault diagnosis of machines with unlabeled data. *IEEE Transactions on Industrial Electronics*, 66(9):7316–7325, 2018.

[210] Haidong Shao, Min Xia, Guangjie Han, Yu Zhang, and Jiafu Wan. Intelligent fault diagnosis of rotor-bearing system under varying working conditions with modified transfer convolutional neural network and thermal images. *IEEE Transactions on Industrial Informatics*, 17(5):3488–3496, 2020.

[211] Jipu Li, Ruyi Huang, Guolin He, Shuhua Wang, Guanghui Li, and Weihua Li. A deep adversarial transfer learning network for machinery emerging fault detection. *IEEE Sensors J.*, 20(15):8413–8422, 2020.

[212] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106–154, 1962.

[213] Siyu Shao, Stephen McAleer, Ruqiang Yan, and Pierre Baldi. Highly accurate machine fault diagnosis using deep transfer learning. *IEEE Transactions on Industrial Informatics*, 15(4):2446–2455, 2018.

[214] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[215] Huailiang Zheng, Rixin Wang, Yuantao Yang, Jiancheng Yin, Yongbo Li, Yuqing Li, and Minqiang Xu. Cross-domain fault diagnosis using knowledge transfer strategy: A review. *Ieee Access*, 7:129260–129290, 2019.

[216] Gabriela Csurka. Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*, 2017.

[217] Yann LeCun et al. Lenet-5, convolutional neural networks. *URL: http://yann. lecun. com/exdb/lenet*, 20(5):14, 2015.

[218] Francois Chollet et al. Keras, 2015.

[219] Amin Fallahi and Vir V Phoha. Adversarial activity detection using keystroke acoustics. In *European Symposium on Research in Computer Security*, pages 626–648. Springer, 2021.

[220] Mart

[221] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[222] Matan Haroush, Tzivel Frostig, Ruth Heller, and Daniel Soudry. Statistical testing for efficient out of distribution detection in deep neural networks. *arXiv preprint arXiv:2102.12967*, 2021.

[223] Binbin Zhao, Yunfeng Ge, and Hongzhi Chen. Landslide susceptibility assessment for a transmission line in Gansu province, china by using a hybrid approach of fractal theory, information value, and random forest models. *Environmental Earth Sciences*, 80(12):1–23, 2021.

[224] Chuanyang Liu, Yiquan Wu, Jingjing Liu, Zuo Sun, and Huajie Xu. Insulator faults detection in aerial images from high-voltage transmission lines based on deep learning model. *Applied Sciences*, 11(10):4647, 2021.

[225] Yunpeng Liu, Shaotong Pei, Weiping Fu, Kaiyuan Zhang, Xinxin Ji, and Zihui Yin. The discrimination method as applied to a deteriorated porcelain insulator used in transmission lines on the basis of a convolution neural network. *IEEE Transactions on Dielectrics and Electrical Insulation*, 24(6):3559–3566, 2017.

[226] Zhenbing Zhao, Zhen Zhen, Lei Zhang, Yincheng Qi, Yinghui Kong, and Ke Zhang. Insulator detection method in inspection image based on improved faster r-cnn. *Energies*, 12(7):1204, 2019.

[227] Yongjie Zhai, Di Wang, Muliu Zhang, Jiarong Wang, and Feng Guo. Fault detection of insulator based on saliency and adaptive morphology. *Multimedia Tools and Applications*, 76(9):12051–12064, 2017.

[228] Yincheng Li, Wenbin Zhang, Peng Li, Youhuan Ning, and Chunguang Suo. A method for autonomous navigation and positioning of uav based on electric field array detection. *Sensors*, 21(4):1146, 2021.

[229] Yunpeng Ma, Qingwu Li, Lulu Chu, Yaqin Zhou, and Chang Xu. Real-time detection and spatial localization of insulators for uav inspection based on binocular stereo vision. *Remote Sensing*, 13(2):230, 2021.

[230] Haiyan Cheng, Yongjie Zhai, Rui Chen, Di Wang, Ze Dong, and Yutao Wang. Self-shattering defect detection of glass insulators based on spatial features. *Energies*, 12(3):543, 2019.

[231] Yongjie Zhai, Rui Chen, Qiang Yang, Xiaoxia Li, and Zhenbing Zhao. Insulator fault detection based on spatial morphological features of aerial images. *IEEE Access*, 6:35316–35326, 2018.

[232] Shenglong Liao and Jubai An. A robust insulator detection algorithm based on local features and spatial orders for aerial images. *IEEE Geoscience and Remote Sensing Letters*, 12(5):963–967, 2014.

[233] Wanguo Wang, Yufu Wang, Jun Han, and Yue Liu. Recognition and drop-off detection of insulator based on aerial image. In *2016 9th IEEE international symposium on computational intelligence and design (ISCID)*, volume 1, pages 162–167, 2016.

[234] Yifan Wang, Zhongxu Li, Xuecheng Yang, Ning Luo, Yu Zhao, and Gang Zhou. Insulator defect recognition based on faster r-cnn. In *2020 IEEE International Conference on Computer, Information and Telecommunication Systems (CITS)*, pages 1–4, 2020.

[235] Mahbub Hussain, Jordan J Bird, and Diego R Faria. A study on cnn transfer learning for image classification. In *UK Workshop on computational Intelligence*, pages 191–202. Springer, 2018.

[236] Lichao Huang, Yi Yang, Yafeng Deng, and Yinan Yu. Densebox: Unifying landmark localization with end to end object detection. *arXiv preprint arXiv:1509.04874*, 2015.

[237] Yang Li, Yafei Zhang, Yulong Xu, Jiabao Wang, and Zhuang Miao. Robust scale adaptive kernel correlation filter tracker with hierarchical convolutional features. *IEEE Signal Processing Letters*, 23(8):1136–1140, 2016.

[238] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. *Advances in neural information processing systems*, 28, 2015.

[239] Cesare Alippi, Simone Disabato, and Manuel Roveri. Moving convolutional neural networks to embedded systems: The alexnet and vgg-16 case. In *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 212–223, 2018.

[240] Mohammad Amin Morid, Alireza Borjali, and Guilherme Del Fiol. A scoping review of transfer learning research on medical image analysis using imagenet. *Computers in biology and medicine*, 128:104115, 2021.

[241] Christiane Fellbaum. Wordnet. In *Theory and applications of ontology: computer applications*, pages 231–243. Springer, 2010.

[242] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[243] Asifullah Khan, Anabia Sohail, Umme Zahoora, and Aqsa Saeed Qureshi. A survey of the recent architectures of deep convolutional neural networks. *Artificial intelligence review*, 53(8):5455–5516, 2020.

[244] Veronika Cheplygina, Marleen de Bruijne, and Josien PW Pluim. Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *Medical image analysis*, 54:280–296, 2019.

[245] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[246] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

[247] António Raimundo. Insulator data set-chinese power line insulator dataset (cplid). *IEEE Dataport*, 2020.

[248] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

[249] Luyao Du, Jun Ji, Zhonghui Pei, Hongjiang Zheng, Shuaizhi Fu, Haiyang Kong, and Wei Chen. Improved detection method for traffic signs in real scenes applied in intelligent and connected vehicles. *IET Intelligent Transport Systems*, 14(12):1555–1564, 2020.

[250] Qilai Chen, Tingting Han, Minghua Tang, Zhang Zhang, Xuejun Zheng, and Gang Liu. Improving the recognition accuracy of memristive neural networks via homogenized analog type conductance quantization. *Micromachines*, 11(4):427, 2020.

[251] Chuteng Zhou, Prad Kadambi, Matthew Mattina, and Paul N Whatmough. Noisy machines: Understanding noisy neural networks and enhancing robustness to analog hardware errors using distillation. *arXiv preprint arXiv:2001.04974*, 2020 Jan 14.

[252] Vipin Tiwari and Nilay Khare. Hardware implementation of neural network with sigmoidal activation functions using cordic. *Microprocessors and Microsystems*, 39(6):373–381, 2015.

[253] Z Hajduk. Hardware implementation of hyperbolic tangent and sigmoid activation functions. *Bulletin of the Polish Academy of Sciences. Technical Sciences*, 66(5):563–577, 2018.

[254] Koldo Basterretxea, Jos Manuel Tarela, Ins del Campo, and Guillermo Bosque. An experimental study on nonlinear function computation for neural/fuzzy hardware design. *IEEE transactions on neural networks*, 18(1):266–283, 2007.

[255] Koldo Basterretxea. Recursive sigmoidal neurons for adaptive accuracy neural network implementations. In *2012 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, pages 152–158. IEEE, 2012.

[256] Antonio Armato, Luca Fanucci, Enzo Pasquale Scilingo, and Danilo De Rossi. Low-error digital hardware implementation of artificial neuron activation functions and their derivative. *Microprocessors and Microsystems*, 35(6):557–567, 2011.

[257] Che-Wei Lin and Jeen-Shing Wang. A digital circuit design of hyperbolic tangent sigmoid function for neural networks. In *2008 IEEE International Symposium on Circuits and Systems*, pages 856–859. IEEE, 2008.

[258] Stamatis Vassiliadis, Ming Zhang, and José G Delgado-Frias. Elementary function generators for neural-network emulators. *IEEE transactions on neural networks*, 11(6):1438–1449, 2000.

[259] K Basterretxea, JM Tarela, and I Del Campo. Digital design of sigmoid approximator for artificial neural networks. *Electronics Letters*, 38(1):35–37, 2002.

[260] K Basterretxea, Jose Manuel Tarela, and I Del Campo. Approximation of sigmoid function and the derivative for hardware implementation of artificial neurons. *IEE Proceedings-Circuits, Devices and Systems*, 151(1):18–24, 2004.

[261] Ming Zhang, Stamatis Vassiliadis, and Jose G. Delgado-Frias. Sigmoid generators for neural computing using piecewise approximations. *IEEE transactions on Computers*, 45(9):1045–1049, 1996.

[262] Karl Leboeuf, Ashkan Hosseinzadeh Namin, Roberto Muscedere, Huapeng Wu, and Majid Ahmadi. High speed vlsi implementation of the hyperbolic tangent sigmoid function. In *2008 Third International Conference on Convergence and Hybrid Information Technology*, volume 1, pages 1070–1073. IEEE, 2008.

[263] MT Tommiska. Efficient digital implementation of the sigmoid function for reprogrammable logic. *IEE Proceedings-Computers and Digital Techniques*, 150(6):403–411, 2003.

[264] Ashkan Hosseinzadeh Namin, Karl Leboeuf, Roberto Muscedere, Huapeng Wu, and Majid Ahmadi. Efficient hardware implementation of the hyperbolic tangent sigmoid function. In *2009 IEEE International Symposium on Circuits and Systems*, pages 2117–2120. IEEE, 2009.

[265] Pramod Kumar Meher. An optimized lookup-table for the evaluation of sigmoid function for artificial neural networks. In *2010 18th IEEE/IFIP International Conference on VLSI and System-on-Chip*, pages 91–95. IEEE, 2010.

[266] Zhao Ren, Kun Qian, Yebin Wang, Zixing Zhang, Vedhas Pandit, Alice Baird, and Bjorn Schuller. Deep scalogram representations for acoustic scene classification. *IEEE/CAA Journal of Automatica Sinica*, 5(3):662–669, 2018.

[267] Hadis Takalo, Arash Ahmadi, Mitra Mirhassani, and Majid Ahmadi. Analog cellular neural network for application in physical unclonable functions. pages 2635–2638, 2016.

[268] Babak Zamanlooy and Mitra Mirhassani. Efficient vlsi implementation of neural networks with hyperbolic tangent activation function. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 22(1):39–48, 2013.

[269] Manuel Carrasco-Robles and Luis Serrano. A novel cmos current mode fully differential tanh (x) implementation. pages 2158–2161, 2008.

[270] Filipp Akopyan, Jun Sawada, Andrew Cassidy, Rodrigo Alvarez-Icaza, John Arthur, Paul Merolla, Nabil Imam, Yutaka Nakamura, Pallab Datta, Gi-Joon Nam, et al. Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *IEEE transactions on computer-aided design of integrated circuits and systems*, 34(10):1537–1557, 2015.

[271] Dan Hammerstrom. A highly parallel digital architecture for neural network emulation. pages 357–366, 1991.

[272] Mohsen Imani, Daniel Peroni, Yeseong Kim, Abbas Rahimi, and Tajana Rosing. Efficient neural network acceleration on gpgpu using content addressable

memory. In *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2017*, pages 1026–1031. IEEE, 2017.

[273] Yihua Liao. Neural networks in hardware: A survey. *Department of Computer Science, University of California*, 2001.

[274] S Mahapatra, AE Islam, S Deora, VD Maheta, K Joshi, and M A Alam. Characterization and modeling of nbti stress, recovery, material dependence and ac degradation using rd framework. In *18th IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*, pages 1–7. IEEE, 2011.

[275] Mirko Prezioso, Farnood Merrikh-Bayat, BD Hoskins, Gina C Adam, Konstantin K Likharev, and Dmitri B Strukov. Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature*, 521(7550):61–64, 2015.

[276] Seyab Khan, Said Hamdioui, Halil Kukner, Praveen Raghavan, and Francky Catthoor. Bti impact on logical gates in nano-scale cmos technology. pages 348–353, 2012.

[277] Harika Manem, Jeyavijayan Rajendran, and Garrett S Rose. Design considerations for multilevel cmos/nano memristive memory. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 8(1):1–22, 2012.

[278] Seyed Nima Mozaffari and Ali Afzali-Kusha. Statistical model for subthreshold current considering process variations. In *2nd Asia Symposium on Quality Electronic Design (ASQED)*, pages 356–360. IEEE, 2010.

[279] Shuai Li, MengChu Zhou, and Xin Luo. Modified primal-dual neural networks for motion control of redundant manipulators with dynamic rejection of harmonic noises. *IEEE transactions on neural networks and learning systems*, 29(10):4791–4801, 2017.

[280] Abhronil Sengupta and Kaushik Roy. A vision for all-spin neural networks: A device to system perspective. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 63(12):2267–2277, 2016.

[281] Golnar Khodabandehloo, Mitra Mirhassani, and Majid Ahmadi. Analog implementation of a novel resistive-type sigmoidal neuron. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 20(4):750–754, 2011.

[282] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[283] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017 Aug 25.

[284] Seyed Nima Mozaffari, Krishna Prasad Gnawali, and Spyros Tragoudas. An aging resilient neural network architecture. In *Proceedings of the 14th IEEE/ACM International Symposium on Nanoscale Architectures*, pages 25–30, 2018.

[285] SBS Younus, S Shajun Nisha, and M Mohamed Sathik. Comparative analysis of activation functions in neural network for handwritten digits. *Studies in Indian Place Names*, 40(71):793–799, 2020.

[286] Chi-Chun Zhou, Hai-Long Tu, Yi Liu, and Jian Hua. Activation functions are not needed: the ratio net. *arXiv preprint arXiv:2005.06678*, 2020 May 14.

[287] Marina Adriana Mercioni and Stefan Holban. The most used activation functions: Classic versus current. In *2020 International Conference on Development and Application Systems (DAS)*, pages 141–145. IEEE, 2020.

[288] Keyang Cheng, Rabia Tahir, Lubamba Kasangu Eric, and Maozhen Li. An analysis of generative adversarial networks and variants for image synthesis on mnist dataset. *Multimedia Tools and Applications*, 79:1–28, 2020.

[289] Indranil Chakraborty, Deboleena Roy, and Kaushik Roy. Technology aware training in memristive neuromorphic systems for nonideal synaptic crossbars. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(5):335–344, 2018.

[290] Raqibul Hasan, Chris Yakopcic, and Tarek M Taha. Ex-situ training of dense memristor crossbar for neuromorphic applications. In *Proceedings of the 2015 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH´ 15)*, pages 75–81. IEEE, 2015.

[291] Krishna Prasad Gnawali, Seyed Nima Mozaffari, and Spyros Tragoudas. Low power spintronic ternary content addressable memory. *IEEE Transactions on Nanotechnology*, 17(6):1206–1216, 2018.

[292] Gražvydas Žiemys, Andrew Giebfried, Markus Becherer, Irina Eichwald, Doris Schmitt-Landsiedel, and Stephan Breitkreutz-v Gamm. Modeling and simulation of nanomagnetic logic with cadence virtuoso using verilog-a. *Solid-State Electronics*, 125:247–253, 2016.

[293] Mahesh Chandra. Hardware implementation of hyperbolic tangent function using catmull-rom spline interpolation. *arXiv preprint arXiv:2007.13516*, 2020 Jul 13.

[294] Chih-Hsiang Chang, Hsu-Yu Kao, and Shih-Hsu Huang. Hardware implementation for multiple activation functions. In *2019 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, pages 1–2, 2019.

[295] Ahmed M Abdelsalam, JM Pierre Langlois, and Farida Cheriet. A configurable fpga implementation of the tanh function using dct interpolation. In *2017*

IEEE 25th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), pages 168–171, 2017.

[296] Aaron Stillmaker, Zhibin Xiao, and Bevan Baas. Toward more accurate scaling estimates of cmos circuits from 180 nm to 22 nm. *VLSI Computation Lab, ECE Department, University of California, Davis, Tech. Rep. ECE-VCL-2011-4*, 4:m8, 2011.

[297] Shyam Prasad Adhikari, Hyongsuk Kim, Ram Kaji Budhathoki, Changju Yang, and Leon O Chua. A circuit-based learning architecture for multilayer neural networks with memristor bridge synapses. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 62(1):215–223, 2014.

[298] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1):82–99, 2018.

[299] Erik Lindholm and Stuart Oberman. The nvidia geforce 8800 gpu. In *2007 IEEE Hot Chips 19 Symposium (HCS)*, pages 1–17. IEEE, 2007.

[300] Shangce Gao, MengChu Zhou, Yirui Wang, Jiujun Cheng, Hanaki Yachi, and Jiahai Wang. Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction. *IEEE transactions on neural networks and learning systems*, 30(2):601–614, 2018.

[301] Ravi Teja Velpula, Barsha Jain, Ha Quoc Thang Bui, Fatemeh Mohammadi Shakiba, Jeffrey Jude, Moses Tumuna, Hoang-Duy Nguyen, Trupti Ranjan Lenka, and Hieu Pham Trung Nguyen. Improving carrier transport in algan deep-ultraviolet light-emitting diodes using a strip-in-a-barrier structure. *Appl. Opt.*, 59(17):5276–5281, Jun 2020.

[302] Lei Zou, Zidong Wang, Qing-Long Han, and Dong Yue. Tracking control under round-robin scheduling: Handling impulsive transmission outliers. *IEEE Transactions on Cybernetics*, 2021.

[303] Nguyen Viet Dung, Nguyen Linh Trung, Karim Abed-Meraim, et al. Robust subspace tracking with missing data and outliers: Novel algorithm with convergence guarantee. *IEEE Transactions on Signal Processing*, 69:2070–2085, 2021.

[304] Abdulaziz Aljohani and Ibrahim Habiballah. High-impedance fault diagnosis: A review. *Energies*, 13(23):6447, 2020.

[305] MS Coutinho, LRGS Lourenço Novo, MT de Melo, LHA de Medeiros, DCP Barbosa, MM Alves, VL Tarragô, RGM dos Santos, HBTD Lott Neto, and PHRP Gama. Machine learning-based system for fault detection on anchor rods of cable-stayed power transmission towers. *Electric Power Systems Research*, 194:107106, 2021.

[306] Jian Wu, Shuang Shao, Dongping Jiang, Hongfang Yao, Caiguo Ma, and Kekui Xu. Research on adaptive power transmission line fault inspection. In *IOP Conference Series: Earth and Environmental Science*, volume 787, page 012050. IOP Publishing, 2021.

[307] Lisa Torrey and Jude Shavlik. Transfer learning. *Handbook of Res. on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264, 2010.

[308] Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu. Federated learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 13(3):1–207, 2019.

[309] M. Satyanarayanan. The emergence of edge computing. *Computer*, 50(1):30–39, 2017.

[310] Anjan Kumar Sahoo et al. Comparative analysis of classification techniques used in machine learning as applied on a three phase long transmission line system for fault prediction using python. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(7):2097–2109, 2021.

[311] Ani Harish and MV Jayan. Classification of power transmission line faults using an ensemble feature extraction and classifier method. In *Inventive Communication and Computational Technologies*, pages 417–427. Springer, 2021.

[312] Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. A review of applications in federated learning. *Computers & Industrial Engineering*, 149:106854, 2020.

[313] Xiao-Wei Liu. Research on transmission line fault location based on the fusion of machine learning and artificial intelligence. *Security and Communication Networks*, 2021.

[314] Xiaofei Wang, Yiwen Han, Victor CM Leung, Dusit Niyato, Xueqiang Yan, and Xu Chen. *Edge AI: Convergence of edge computing and artificial intelligence.* Springer, 2020.

[315] Yuwen Qian, Long Shi, Jun Li, Xiangwei Zhou, Feng Shu, and Jiangzhou Wang. An edge-computing paradigm for internet of things over power line communication networks. *IEEE Network*, 34(2):262–269, 2020.

[316] Mohammad Ahmad Ansari, Poonam Agarwal, and Krishnan Rajkumar. Artificial neural network (ann) to design microstrip transmission line. In *Proceedings of Intl. Conf. on Artificial Intelligence and Applications*, pages 25–33. Springer, 2021.

[317] R. Godse and S. Bhat. Mathematical morphology-based feature-extraction technique for detection and classification of faults on power transmission line. *IEEE Access*, 8:38459–38471, 2020.

[318] Pathirikkat Gopakumar, Maddikara Jaya Bharata Reddy, and Dusmanta Kumar Mohanta. Adaptive fault identification and classification methodology for smart power grids using synchronous phasor angle measurements. *IET Generation, Transmission & Distribution*, 9(2):133–145, 2015.

[319] Xiurui Hou, Kai Wang, Cheng Zhong, and Zhi Wei. St-trader: A spatial-temporal deep neural network for modeling stock market movement. *IEEE/CAA Journal of Automatica Sinica*, 8(5):1015–1024, 2021.

[320] Hamid Teimourzadeh, Arash Moradzadeh, Maryam Shoaran, Behnam Mohammadi-Ivatloo, and Reza Razzaghi. High impedance single-phase faults diagnosis in transmission lines via deep reinforcement learning of transfer functions. *IEEE Access*, 9:15796–15809, 2021.

[321] Wenjin Zhang, Jiacun Wang, and Fangping Lan. Dynamic hand gesture recognition based on short-term sampling neural networks. *IEEE/CAA Journal of Automatica Sinica*, 8(1):110–120, 2020.

[322] Jun Tang, Gang Liu, and Qingtao Pan. A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends. *IEEE/CAA Journal of Automatica Sinica*, 8(10):1627–1643, 2021.

[323] Yicun Hua, Qiqi Liu, Kuangrong Hao, and Yaochu Jin. A survey of evolutionary algorithms for multi-objective optimization problems with irregular pareto fronts. *IEEE/CAA Journal of Automatica Sinica*, 8(2):303–318, 2021.

[324] Jiahai Wang, Yuyan Sun, Zizhen Zhang, and Shangce Gao. Solving multitrip pickup and delivery problem with time windows and manpower planning using multiobjective algorithms. *IEEE/CAA Journal of Automatica Sinica*, 7(4):1134–1153, 2020.

[325] Xiwang Guo, Shixin Liu, MengChu Zhou, and Guangdong Tian. Dual-objective program and scatter search for the optimization of disassembly sequences subject to multiresource constraints. *IEEE Transactions on Automation Science and Engineering*, 15(3):1091–1103, 2017.

[326] Xiwang Guo, Mengchu Zhou, Shixin Liu, and Liang Qi. Lexicographic multiobjective scatter search for the optimization of sequence-dependent selective disassembly subject to multiresource constraints. *IEEE Transactions on Cybernetics*, 50(7):3307–3317, 2019.

[327] Qi Kang, Xinyao Song, MengChu Zhou, and Li Li. A collaborative resource allocation strategy for decomposition-based multiobjective evolutionary algorithms. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(12):2416–2423, 2018.

[328] Zhenhua Huang, Shunzhi Yang, MengChu Zhou, Zheng Gong, Abdullah Abusorrah, Chen Lin, and Zheng Huang. Making accurate object detection at the edge: Review and new approach. *Artificial Intelligence Review*, pages 1–30, 2021.

[329] Xudong Luo, Xiaohao Wen, MengChu Zhou, Abdullah Abusorrah, and Lukui Huang. Decision-tree-initialized dendritic neuron model for fast and accurate data classification. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.