Fault Detection, Classification and Location in Transmission Line Systems Using Neural Networks

Ibrahim Farhat

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfilment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montreal, Quebec, Canada

March 2003

0-612-77684-0

ABSTRACT

Fault Detection, Classification and Location in Transmission Line Systems using Neural Networks

Ibrahim Farhat

Transmission lines, among the other electrical power system components, suffer from unexpected failures due to various random causes. These failures interrupt the reliability of the operation of the power system. When unpredicted faults occur protective systems are required to prevent the propagation of these faults and safeguard the system against the abnormal operation resulting from them. The functions of these protective systems are to detect and classify faults as well as to determine the location of the faulty line when a fault is detected in the voltage and/or current line magnitudes. Once the fault is detected and classified the protective relay sends a trip signal to a circuit breaker(s) in order to disconnect (isolate) the faulted line.

Successful applications of neural networks in electrical power systems have demonstrated that this powerful tool can be employed as an alternative method for solving problems accurately and efficiently. The features of neural networks, such as their ability to learn, generalize and parallel processing, among others, have made their applications to many

systems ideal. The use of neural networks as pattern classifiers is among their most common and powerful applications.

This thesis presents an artificial neural network approach to detection, classification and isolation (location) of faults in transmission line systems. The objective is to implement a complete scheme for distance protection of a transmission line system. In order to perform this goal, the distance protection task is subdivided into different neural networks for fault detection, fault identification (classification) as well as fault location in different zones. The other purpose of this work is to study and compare the application of three different neural network architectures for the protection of the transmission lines. The considered three approaches are back-propagation, radial basis functions and support vector machines. Simulation results are provided to demonstrate the advantages and disadvantages of these structures when applied to the problem of transmission line protection.

ACKNOWLEDGEMENTS

I sincerely wish to express my gratitude to my advisor Dr. Khorasani for his full support, expert guidance, understanding and patience throughout the course of this research. Also, I am greatly indebted to him for his critical review of the manuscript of my thesis.

I would also like to thank the members of the Electrical and Computer Engineering Department at Concordia University for their help.

I also wish to express a sincere gratefulness to Mr. A. Amari, Chairman of the Board of Waha Oil Company, WOC, and the staff of the Training Department in WOC as well as the team work of the Training Department of Umm Al-Jawaby Oil Company for their mental and financial support.

Finally, I would like to take this opportunity to thank the many friends and colleagues for their support and encouragement. A very special note of thanks is extended to my fellow countryman Dr. Khalil Al-Aroudi for his continued encouragement and support throughout this endeavor.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

General Background and Research Goals

1.1 Introduction

The greatest threat to the continuity of electricity supply is system faults. Faults on electric power systems are an unavoidable problem. Hence, a well-coordinated protection system must be provided to detect and isolate faults rapidly so that the damage and disruption caused to the power system is minimized. The clearing of faults is usually accomplished by devices that can sense the fault and quickly react to disconnect the faulty section from healthy ones.

In the control centers of the electrical power systems a large number of alarms are received as a result of different types of faults. To protect these systems, the faults must be detected and isolated accurately. The operators in the control centers have to deal with a large amount of data to get the required information about the faults. This, indeed, takes a long time when time is the critical issue. In addition, the protection system may, itself, fail, let alone disruption in the communication networks and corruption of transferred data. All these are challenges to the protection of the electrical power systems. The challenges are to detect, classify and locate the faults as fast as possible when they occur.

Conventional schemes set thresholds according to the fault currents and voltages [1]. When a fault occurs, the fault currents and voltages develop a transient DC offset component and high-frequency transient components in addition to the power frequency components. The fault currents and voltages vary with fault type, location, size, and fault inception angle and system condition. These variations cause the space to be non-linearly separable and none of the thresholds can be found so as to satisfy for various system and fault conditions [2]. Furthermore, when faults take place the faulted phase(s) have an effect on the healthy phase(s) due to mutual coupling between these phase(s). This problem is compounded by the fact that this coupling is highly non-linear in nature and is dependent on a complex interplay amongst a number of variables [2].

Intelligent systems have been in use for fault diagnosis in power systems for some time [3-8]. Among the intelligent systems, *Artificial Neural Networks* (ANN) have been applied to several power system operation and protection [10-20]. The ANN superiority over conventional schemes, (details are presented in Chapter 2), is demonstrated widely [11, 21-29].

ANN employed as pattern classifiers, has been used in the area of transmission line fault diagnosis [3, 11, 30-34]. Due to the neural networks ability to acquire information and to learn through training, many neural network models are proposed in the literature [35]. The capability of neural networks to generalize as well as their fault tolerance makes them a reliable tool to be used to handle unseen fault patterns.

1.2 Research Motivation

Transmission line is the most likely element in the power system to be exposed to faults especially when their physical dimension is taken into consideration [36, 37].

This thesis has concentrated on understanding the behavior of the transmission line phase voltages and currents as a consequence of faults. The objective of this work is to study and employ neural network techniques as a reliable tool to detect, classify and isolate faults in a transmission line system. Artificial neural networks are a powerful tool to use in transmission line fault detection, classification and isolation. The parallelism inherent in neural networks enables them with faster computational time than traditional techniques. Hence, using this technology in transmission line fault diagnosis does validate its usefulness and encourage engineers to use this technique in other power systems applications [3-11, 30-34].

1.3 Literature Review

Due to the possibility of training neural networks with off-line data, they are found useful for power system applications [3, 10-35]. The history of applying neural network techniques to power system is not that long and can be traced to some recent work [39]. Neural network techniques have extensively been of a great interest to the power system community since Sobajic and Pao applied it to transient stability for prediction of the critical clearing time [40]. By the late eighties and early nineties the application of neural networks in this area has become quite well established [41].

The neural network applications in transmission line protection are mainly concerned with improvements in achieving more effective and efficient fault diagnosis and distance relaying. A wide variety of published work in the fault diagnosis of transmission lines can be obtained in the literature. For instance Dalstein and Kulicke [21] employed neural networks to detect transmission line faults and distinguish between arcing and non arcing faults. Kezunovic, Rikalo and Sobajic [42], proposed a neural network scheme to detect and identify high speed faults under changing system conditions. In the area of fault location and selection of the faulted line in a network a large number of published work is available. For instance, Chen and Maun [43] used neural networks to improve single-ended lines fault location while Song [44] applied neural networks to improve fault location for series compensated lines. Other published work in the area may be found in references [24, 26-28, 30, 34, 36, 45-48]. It should be noted that most of the implementations in the above references are based on the feed-forward multilayer perceptron. Although other architectures have also been investigated in the literature by some of the references mentioned [22, 28, 45, 65], they are by all means few in numbers. More discussion about the applications of the various neural network archetictures in the area of fault diagnosis to distance relaying is presented in Chapter 3. A survey of some of the typical neural network approaches in transmission line fault detection, classification and isolation is provided in [16, 25].

1.4 Research Objectives and Contributions of the Thesis

In view of the above discussion, the objectives of this research are to utilize neural network techniques for development and evaluation to the fault detection, classification and isolation of transmission line systems. Specifically, the goal is to detect and to identify the type of fault in the line and to determine which zone (segment) of the line has become faulty. In order to achieve these objectives, three network approaches are studied, implemented and modified to perform the above three tasks. The neural network architectures considered here are; (i) back-propagation network (BP), (ii) radial basis function network (RBF), and (iii) support vector machines (SVM). This thesis presents the results of applying the back-propagation network as a pattern recognition machine to detect faults in a typical transmission line system. The use of the three architectures to identify and classify the type of fault and to specify the faulty zone in a transmission line system is then investigated, and a comparative study is conducted on the performance of the three networks. The outcome of this work would serve as a guideline on the selection of a suitable network from the set of the considered three alternatives as may be appropriate for a specific application. The possibility of using support vector machines, for the first time in the literature, in the protection of transmission line was explored which resulted in improved performance as compared to the other architectures studied.

1.5 Research Methodology

By observing the voltage and current signals of a line, one is then able to identify the existence of faults in the system. These signals are also used to specify the fault type and location. Figure 1.1 shows the stages conducted in this thesis.

As shown in Figure 1.1, the algorithm investigated in this thesis consists of three stages:

1. Fault Detection: A neural network is employed to determine if a fault has occurred or not. The inputs to this network are the three phase currents and voltages of the system, that is $I = [I_a \quad I_b \quad I_c]^T$ and $V = [V_a \quad V_b \quad V_c]^T$, respectively. The output of the network is a binary decision to indicate occurrence of a fault.

2. Fault type identification: A second neural network is used to classify the fault. It receives as inputs the line currents and voltages, that is $I = [I_a \quad I_b \quad I_c]^T$ and $V = [V_a \quad V_b \quad V_c]^T$, respectively. The output of the network has four signals to distinguish the four fault types that are phase-to-earth fault, phase-to-phase fault, double-phase-to-earth fault and three-phase fault.

3. Fault isolation: The third neural network is used to localize faults in the transmission line system according to following three protection zones:

   - The main protection zone Z_1: can be 90%-95% of the protected line.
   - The second protection zone Z_2: 110%-140%.
   - The third protection zone Z_3: 150%-200%.

The network has the line currents and voltages as inputs and the output signals identify the faulty zone, Z1, Z2 or Z3.

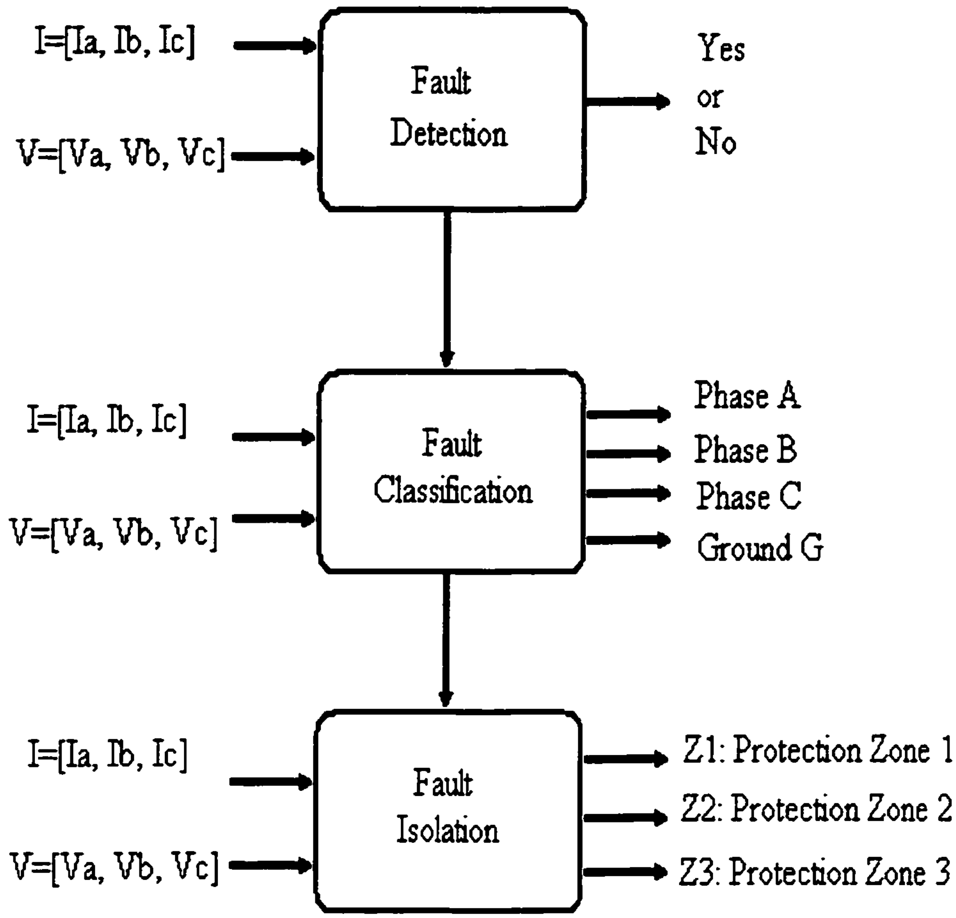


Figure 1.1: The research methodology for fault detection, classification and isolation in typical transmission line system using neural networks

In order to perform the above three tasks, simulation models of the transmission line system are constructed and the generated information are then channeled using the software MATLAB (Version 6.1) and accompanying Neural Network Toolbox (Version 5.0) and SVM Toolbox (Version 3.00) as well as Power System Block Set (Version 2.1). MATLAB, and its associated Toolboxes, provides the means to conveniently simulate and model the studied transmission line and the proposed neural networks.

1.6 Outline of the Thesis

In Chapter 2, the issues and problems related to the protection of a typical transmission line system, the fault types and protection schemes available in the literature are briefly presented. The state of art research results for transmission lines protection and relaying are described in this chapter. In Chapter 3, a brief introduction to neural networks and a few architectures are presented. A literature review of the application of various neural network topologies to power system transmission lines is presented. Chapter 4 presents the development and application of the proposed neural network architectures for detection, identification and isolation of the faults as applied to a typical transmission line system. Extensive simulations are also provided in this chapter. In Chapter 5, conclusions of the thesis are stated and recommendations for future use of the proposed neural networks in the area of faulty transmission line systems are discussed.

Chapter 2

Protection of Power Transmission Line Systems

2.1 Protection Systems

A fault, which is an abnormal system condition, occurs as a random event. The faulty system component (line, bus, transformer, etc.) must be isolated from the system quickly to prevent power system instability or break-up of the system through the action of other protective devices. Therefore, a protection system must be designed to disconnect the faulted component from the system as quickly as possible.

The protection system consists of many other subsystems, which contribute to the fault removal process. These subsystems are presented in Figure 2.1. Besides relays, the protection system consists of transducers (voltage transformers (CVT) and current transformers (CT)). The circuit breaker, which interrupts the current and isolates the faulty section, is operated by energizing its trip coil from the station battery, and the relays do this job by closing contacts between the battery and the trip coil.

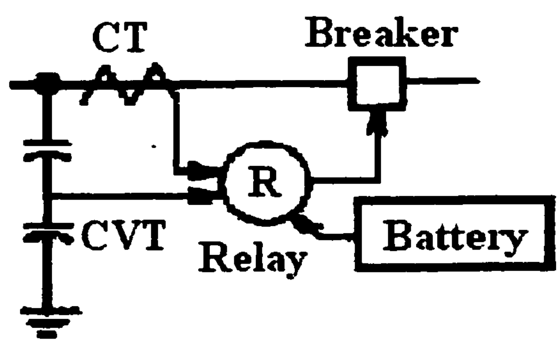


Figure: 2.1 Subsystems of a protection system. Besides relays, the protection system consists of transducers, circuit breakers and station battery.

The most important component of the protection system is the relay. This is a device which responds to the condition of its inputs (voltages, currents, or contact status) in such manner that it provides appropriate output signals to trip circuit breakers when input conditions correspond to faults for which the relay is designed to operate [49].

A relay must be designed so that it produces a trip output only for the faults it is responsible for, while it doesn't for any other faults. This is the concept of reliability as understood in relaying literature. Reliability of the relay is the degree of its certainty to perform as intended. The relays have two alternative ways in which they can be unreliable: they may fail to operate when they are expected to or, they may operate when they are not expected to [50]. A reliable relay has to be dependable and secure [51]. Dependability implies that the relay always operates for conditions for which it is designed to operate. A secure relay does not operate for any other power system disturbance.

2.2 Classification of Relays

In general, the relays that are in use on power systems may be categorized as follows [52,53]:

- *Magnitude Relays*: These relays respond to the magnitude of the input quantity as described in later sections. Overcurrent relays belong to this class.
- *Directional Relays*: These relays respond to the phase angle between two inputs.

- *Ratio Relays*: These relays respond to the ratio of two input signals expressed as phasors. The most common ratio relays are distance relays.
- *Differential Relays*: These relays respond to the magnitude of the algebraic sum of two or more inputs.
- *Pilot Relays*: These relays employ communicated information from remote locations as input signals.

The candidates under which one uses these relays are explained in later sections. The advantages and disadvantages of these relays will also be described.

2.3 Zones of Protection

A zone of protection is a portion of the power system for which a specific protection system is designated. The idea of an area of responsibility of a protection system is formalized by assigning zones of protection to various protection systems [54].

Figure 2.2, which is a one-line diagram of a portion of a power system containing a transformer and four buses, illustrates the concept of zones of protection.

The closed dashed lines indicate the five zones of protection by which the various power system components are covered. Each zone contains one or more of the system components in addition to two or more circuit breakers [49, 54]. The boundary of each zone defines a portion of power system such that when a fault occurs in any part of that zone, the protection system for that zone takes an action to isolate everything within that zone from the rest of the system. In order to isolate (de-energize) the faulty zone (or component), the relays in that zone activate trip coils of certain circuit breakers.

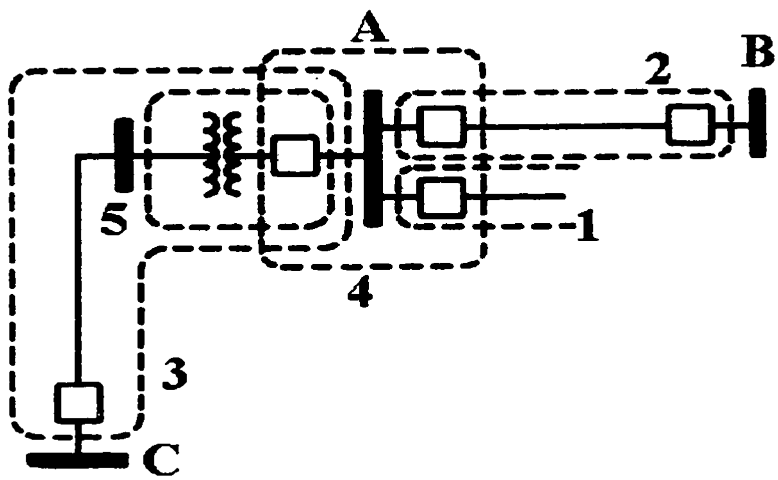


Figure: 2.2 Zones of protection.
Zone 2 defines the boundary for protection of transmission line *A-B*. Zone 4 defines bus-*A* protection. Zone 5 is a transformer protection.

It should be noted that a circuit breaker should be inserted at each point where the equipment inside a zone and the rest of the system such that the circuit breaker defines the boundaries of the zones of protection [54].

In order to make sure that there is no portion of the system that is left without protection, each segment of the system must be covered by a primary high speed protection. In addition, the various zones always overlap as shown in Figure 2.2 to ensure that there are no blind spots in the protection system. In other words, to reinforce the dependability of the overall protection system, a zone will be protected by several protection systems. This will ensure that even when a failure occurs in the protection system itself, the power system will not be unprotected. On the other hand, it should be clear that if a fault occurs within the overlapped areas, a larger portion of the power system will be isolated. In other words, the smaller the region of overlap, the less the segment of the system that would be lost from service [54].

2.4 Protection of Transmission Lines

As mentioned above, among the faults of a power system, transmission line faults occur more frequently. About two thirds of the faults in power systems occur in the transmission line network [36, 37]. Faults in transmission line can be:

- Single-line-to-ground faults (conductor-to-ground)
- Phase-to-phase-to-ground faults (two conductors-to-ground)
- Phase-to-phase faults (two conductors)
- Three-phase faults (three conductors)

Statistically, three-phase faults only form a 5% of faults while 70%-80% of faults are single-line-to-ground faults [31]. Transmission line fault diagnosis has been a subject of study for a long time. Traditional protective relaying methods are based on preset values of the parameters and conditions of the system considering only the common situations. These parameters and conditions vary widely when faults take place. This influences the relays, which are set to perform well only during predetermined fault conditions [30].

2.4.1 Overcurrent Relays

As a result of the occurrence of a fault, the current becomes greater than the pre-fault current in any power system element. Therefore, the current magnitude is used as an indicator of a fault. Applying this principle, overcurrent relays are used to protect the transmission lines as explained below.

When a fault occurs in the zone of protection, the fault current is smallest at the far end of the line and greatest at the breaker end [49, 50]. In Figure 2.3 (a), if the minimum fault current possible within the zone of protection is greater than the maximum fault load current, then the operating principle of a relay can be defined as follows:

$$\begin{cases} |I| \geq I_P & \textit{fault in zone, trip.} \\ |I| < I_P & \textit{no fault in zone, do not trip.} \end{cases} \tag{2.1}$$

where I is the current in the relay and I_P is known as the pickup setting of the relay. Figure 2.3 (b) depicts the variation in steady state ac fault current with fault location. The relay characteristic given by Equation (2.1) is defined in terms of the symmetrical fault current.

Figure 2.4 (a) shows the operating characteristic of an ideal relay, which is described by Equation (2.1). As shown in the figure, the relay only operates when the current magnitude exceeds I_P taking a time T_{min} to close its contacts. This type of relay is called an instantaneous relay. Figure 2.4 (b) depicts the traditional method of representing the operating characteristic of the overcurrent relay. The pickup current I_P of a relay is adjustable through the taps on its input winding. For a given current the operating time of the relay can be shifted up or down by adjusting the time-dial setting to produce slow or fast operation time

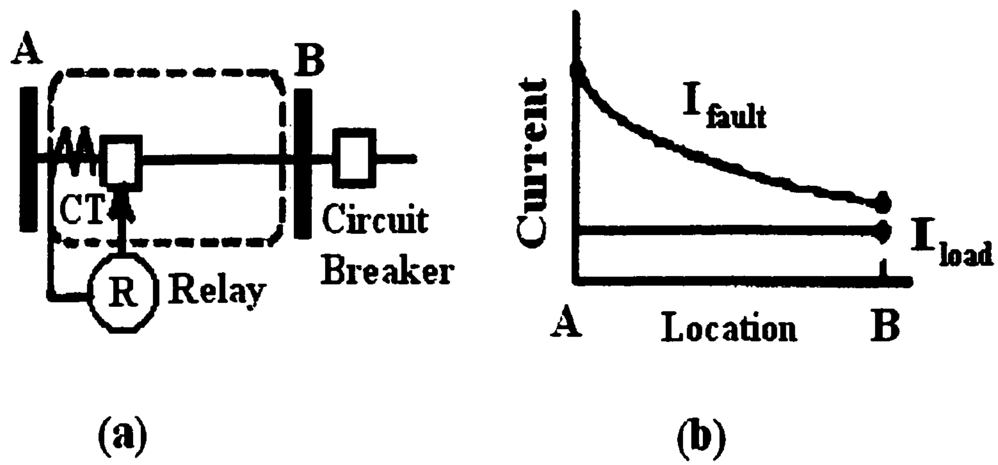


Figure 2.3: Overcurrent protection of a transmission line. (a) Radial system protection, (b) Fault current magnitude as a function of fault location.

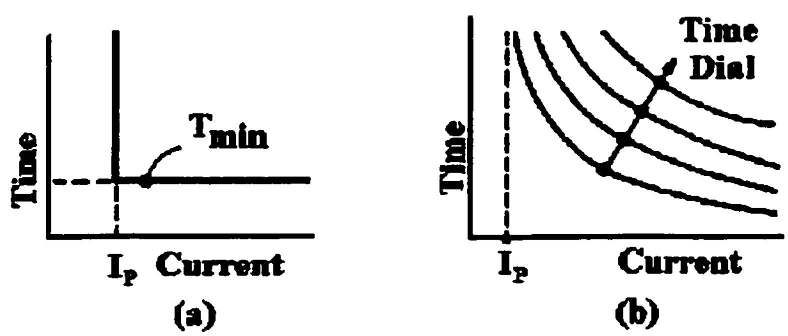


Figure 2.4: Overcurrent relay operating time, (a) Instantaneous relay,

(b) Time overcurrent relay.

2.4.2 Directional Relays

For some power systems, an overcurrent relay may not be able to provide adequate protection. Figure 2.5 (a) illustrates a system that has sources on both ends of the line.

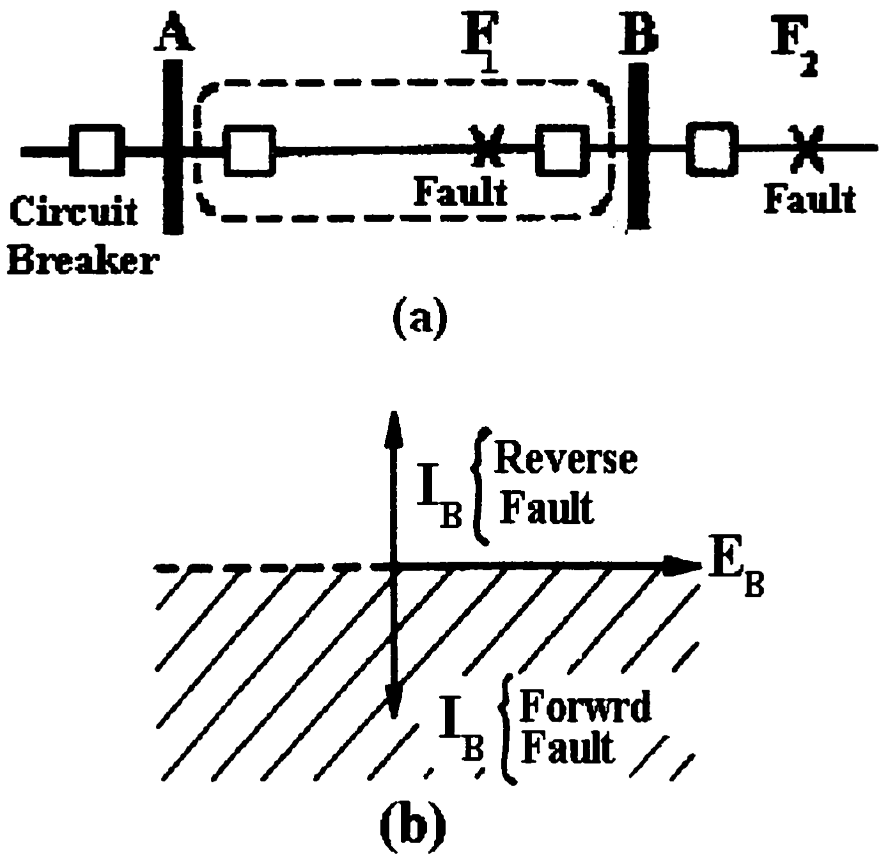


Figure 2.5: Line protection for a loop system, (a) System diagram, (b) Voltage and current phasors.

In this figure F_1 is a fault within the zone of protection of the line and F_2 is outside this zone. Depending on the relative strength of the source on the two sides, the current flowing through the relay at B for a fault at F_1 might be less than the current that would flow through the same relay for a fault at F_2. In such a case, an overcurrent relay set to trip for a fault at F_1 would also trip for a fault at F_2. This is unacceptable from a security point of view. To avoid this situation a directional relay is provided at B. This relay is insensitive to the faults that occur outside the zone of protection [49]. It uses a polarizing voltage where its operation depends upon the direction of the current with respect to the voltage. In view of these descriptions, this type of relay is called directional. Figure 2.5 shows that the directional relay operates when the fault is in the forward direction, i.e. towards the zone of protection.

2.4.3 Distance Relays

Distance relays are widely used to protect transmission lines [55]. These relays respond to the impedance between the relay location and the fault location. They, in fact, respond to the distance to a fault on the transmission line, as the impedance per unit length is fairly constant. In view of the drawbacks of the overcurrent relays, distance relays provide excellent protection under all circumstances.

Figure 2.6 (a) shows a fault at a fractional distance k from the relay location. If a phase-to-phase fault occurs between phases x and y such that $x \neq y$, and $x, y = a, b, c$, then it can be shown that [55, 56]:

$$\frac{E_X - E_Y}{I_X - I_Y} = kZ_1 \tag{2.2}$$

where Ex *and* Ix are the phase x voltage and current, Ey and Iy are the phase y voltage and current and Z_l is the positive sequence impedance for the entire line, which is the impedance to positive-sequence current [54]. Similarly, for a phase to ground fault on phase x we have:

$$\frac{E_x}{I_x + mI_0} = kZ_1 \tag{2.3}$$

where $m = (Z_0 - Z_1)/Z_0$ and Z_0 is the zero sequence impedance of the line, which is the impedance to zero-sequence current [54]. The ratio of the appropriate voltages and currents represent the fraction of line positive sequence impedance at which a fault occurs. The computed ratio can be compared with the total positive sequence impedance of the zone being protected, and if smaller, trip output is produced. The comparison is made in the complex impedance plane as shown in Figure 2.6 (b). For faults on the transmission line, the ratio is a complex number lying on line segment *A-B*. To define the fault region in the complex plane, a rectangular, circle or a segment of a circle can be used to define the zone of protection in the X-R plane as described in [49]. The circular zone shown in Figure 2.6 (b) belongs to the class of relays known as "offset impedance relay", the center of the circle being offset from the origin.

The performance of distance relay near its zone boundaries is not predictable due to various error types such as the transducers accuracies and fault arc resistance [57]. Consequently, it is necessary to form multiple zones of protection in order to achieve the required dependability and security for protecting the entire line. For the line *A-B* in

Figure 2.7 (a), the dotted line represents the zone of protection for the line. However, to be sure of covering it in presence of input errors, two zones (zone 1 and zone 2) are used.

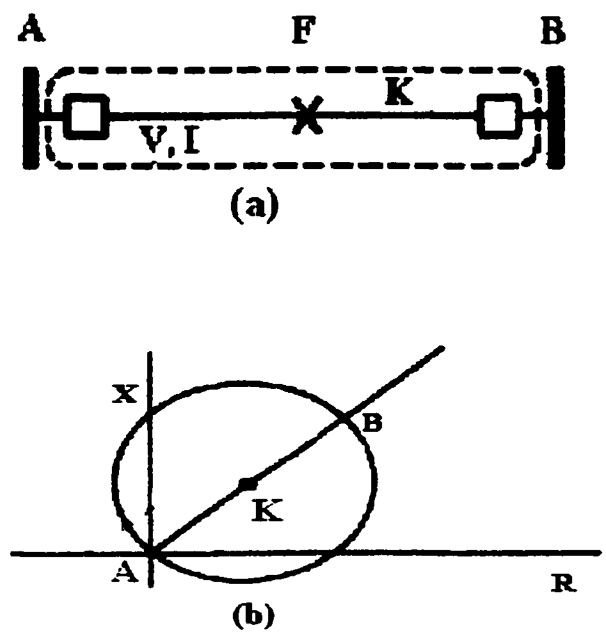
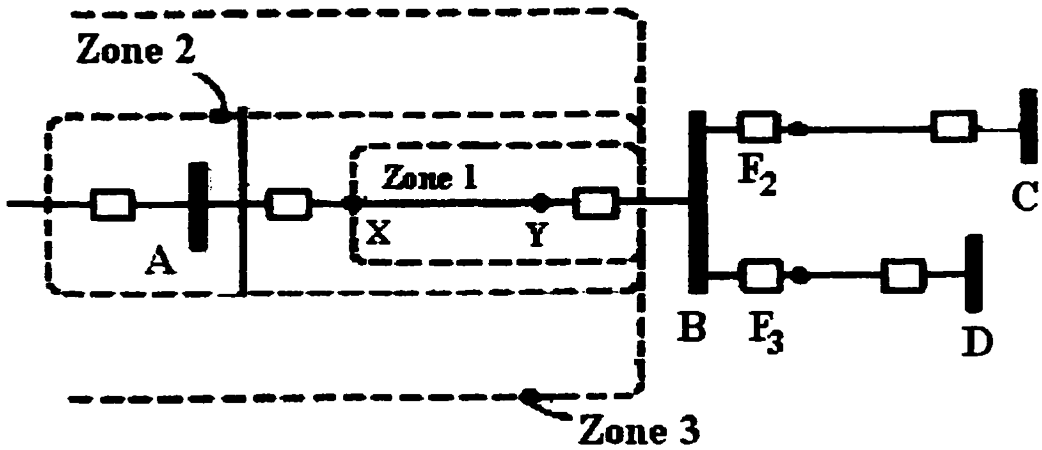
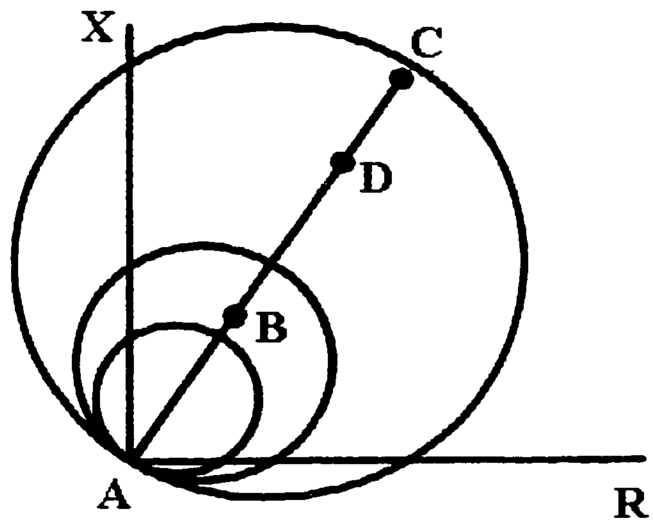


Figure 2.6: Distance relaying of transmission line, (a) Line with a fault in zone of protection, (b) Distance relay characteristics.

The relay in zone 1 is set to operate instantaneously (without delay, i.e. in about one or two cycles) while a fault in zone 2 causes the relay to operate with an added delay (about 20-30 cycles). The zone 2 operating delay is to permit other relays such as those belonging to lines *B-C* and *B-D* to operate for faults within their respective first zones, such as F_2 or F_3, which may lie in zone 2 of the relay protecting line *AB*. It should be noted, hereby, that a similar protection system exists at the *B* terminal looking towards *A*. It is clear that such a line protection scheme would provide high speed protection from both ends against faults in the middle portion of the line, in the *XY* region.



(a)



(b)

Figure 2.7: Distance relaying of a transmission line, (a) Transmission lines protected by distance relays, (b) Three zones of protection.

On the other hand, faults on the line, but outside the *XY* region, are cleared instantaneously by near relay and with zone 2 time delay by the distance relay. In addition to these two zones, a third zone with an additional time delay of the order of one second is provided at each end to provide remote backup for the protection of neighboring circuits. Zone 3 overlaps the longest line connected to the same bus as the line being protected [49, 50, 57]. The three zones of protection are shown in Figure 2.7(b).

The distance relays can be classified according to the shape of their zones of protection. Traditionally, all zone shapes have been circular, because an electromechanical relay produces a circular boundary for the zones of protection. Relay types are, generally, recognized according to the shapes of their operating zones [55, 57, 58]:

- Impedance relays, which have a circular shape, centered at the origin of the R-X diagram.
- Admittance or mho relays have a circular shape, which passes through the origin.
- Reactance relays have a zone boundary defined by a line parallel to the R-axis.
- Quadrilateral relays, which have their characteristic, as the name implies, defined by four straight lines. This characteristic is only available in solid-state or computer relays.

2.4.4 Pilot Relaying

When an entire transmission line is to be covered with high speed protection, pilot relaying technique is employed. Pilot relaying principles are based upon information obtained by the relay from a remote location. The information, which is usually in the form of open-or-close contact status, is sent over a communication channel using power line carrier, microwave, telephone circuits, etc.

Two classes of pilot relaying systems are in use: the directional comparison system and the phase comparison system. In practice, both of these systems could be sub-classified into more sub-categories such as permissive or non-permissive, over or under-reaching [58]. It should be noted that the phase comparison function becomes totally inoperative in case of communication system failure, while the directional comparison system can provide an additional distance relaying protective function.

To summarize, faults in transmission line systems need to be classified accurately and as fast as possible. This chapter reviews some of the conventional methods available in the literature corresponding to protection set thresholds based on the increase of the fault currents and the decrease of the fault voltages [1]. However, because fault currents and voltages vary with fault types, fault location and the power system condition, the fault classification space is not linearly separable and there exists no specific threshold that can satisfy for all various systems and fault conditions.

In the next chapter it is demonstrated that neural networks may be used as an alternative method to the conventional protection methods discussed in this chapter. Subsequently, in Chapter 4, practical and simple neural networks are designed to learn the complex decision functions from several sample patterns, in addition to performing the task of transmission line protection associated by fault detection, classification and isolation.

Chapter 3

Neural Networks and their Applications in Power Transmission Line Systems

3.1 A Brief Introduction to Neural Networks

A common engineering problem is that of estimating a function or a map based on the knowledge of some examples of input-output pairs. This process is known as supervised learning by the neural network community. Other designations ued are function approximation (numerical analysis), regression analysis (statistics) or system identification (control theory). The training set (examples) is composed of pairs of values for the independent (input) and dependent (output) variables. In general the neural network will be playing the role of representing the map $\phi(.)$ in [35, 38]:

$$y = \phi(x) \tag{3.1}$$

where x is the vector of inputs and y is the vector of outputs.

The supervised learning problem can be subdivided into parametric and nonparametric models. In parametric estimation, the form of the functional relationship is known but it may contain free parameters that are determined during the learning process. Usually the free parameters of a parametric model have meaningful interpretations in terms of the physical parameters of the system. An example of a parametric model is polynomial regression.

Nonparametric models are different in the sense that there is no a priori knowledge of the form of the function being estimated. The function is still modeled using an expression with many free parameters but in a way which allows the class of functions which the model can represent to be very broad. Neural networks, as well as Fourier series, spline functions, and wavelets are nonparametric models.

A neural network is a massively parallel distributed processor that has the ability of storing knowledge and making it available for use. It resembles the brain in three aspects:

- The knowledge is acquired by the network through a learning process.
- Interneuron connection strengths, known as synaptic weights, are used to store the knowledge.
- The network is capable of generalization.

The learning process is performed by a learning algorithm. The objective of the algorithm is to change the synaptic weights of the network to attain a desired design objective. Once the network is trained it is capable of generalization. Generalization refers to the capability of the neural network producing "reasonable" outputs for inputs not encountered during the training process.

The characteristics of neural networks that are relevant to this thesis are as follows:

- Input-output mapping. One popular class of training algorithms is called supervised learning. The network is presented with input samples and the network weights are modified so as to minimize the difference between the network output and the desired output. The training proceeds until the network reaches a state where there are no further significant changes in weights.

- Nonlinearity. A neuron basically represents a nonlinear element. Therefore, a neural network made up of a collection of neurons is itself nonlinear.
- Adaptivity. A neural network trained to perform a specific task in a specific environment (input-output pairs) can be easily retrained to deal with minor changes in this environment.

The most popular type of neural networks for supervised learning is the multi-layer perceptron (MLP) which became prevalent in 1986 with the development of the back-propagation algorithm [59]. MLP's have feed-forward connections with adaptable weights, i.e., the free parameters. Training a MLP implies estimating the best set of weights so that the tracking error between network output and desired response is minimized. This requires the solution of a nonlinear optimization problem which is usually performed by means of the gradient descent in the weight-space [59]. The weights are incrementally adjusted to decrease the error, and this process is iterated until the error can no longer be minimized. Nonlinear optimization of this nature has typically slow convergence properties and can also get stuck in local minima.

3.2 The Neuron

A biological neurons, as shown in Figure 3.1, behaves as function representative. It transforms an input signal x into an output signal $\phi(x)$. The function $\phi(.)$ can assume many forms. It can model a simple static activation function such as the sigmoid, the threshold, the radial basis, or linear functions. Figure 3.2 represents some of these activation functions [35, 38].

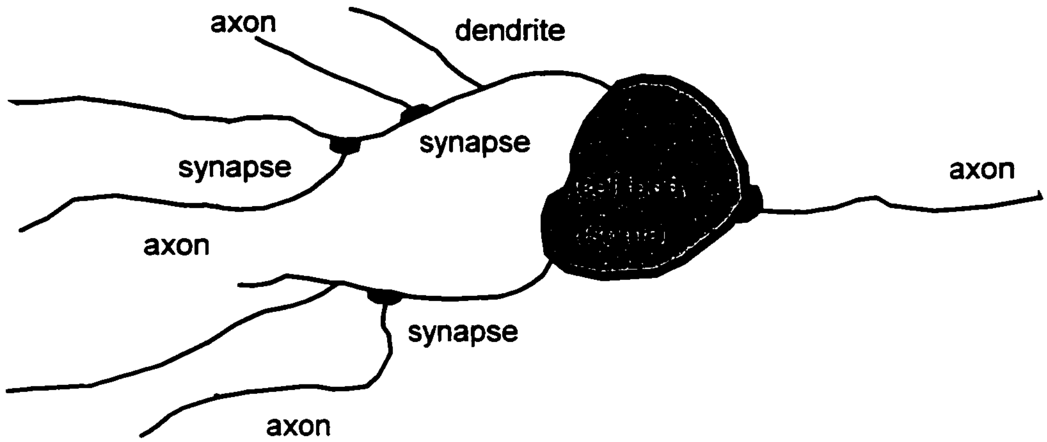


Figure 3.1: Biological neuron.

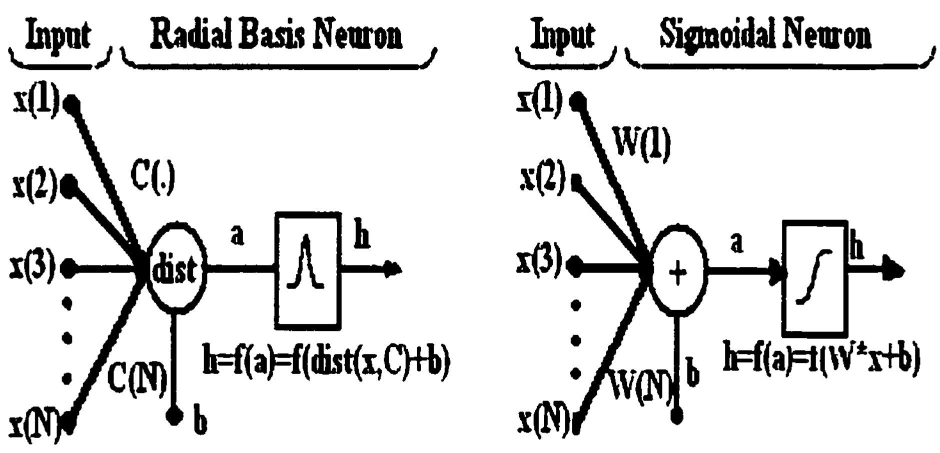


Figure 3.2: Examples of nonlinear activation functions for a neuron

The physiological interpretation of the input signal x and the output signal y involves electrical impulses of potential difference and their temporal summation. Input activations involve small membrane pulses while output signals involve large axonal pulses. Mathematically, the activation input x represents the membrane potential or voltage difference across the neuron's surface membrane. The activation can be positive or negative. The output signal h represents the output induced by the activation input x. In general, signal functions are monotonic nondecreasing. Increasing activation values can only increase the output signal or leave it unchanged. They can never decrease the signal. In practice this means signal functions have an upper bounded saturation value. An important exception to signal monotonicity is the Gaussian functions. Generalized Gaussian functions define potential basis functions as defined later in this chapter.

3.3 Network Architectures

The way in which the neurons of a network are arranged and connected together strongly influences the learning algorithm used to train the network. The various existing architectures can be divided into four main categories [35]:

- Single-layer feed-forward networks
- Multilayer feed-forward networks
- Recurrent networks
- Lattice networks

Multilayer feed-forward networks are only used in this thesis as they are the most widely used architecture in function approximation, classification and pattern recognition problems. Among the many existing variations in both architecture and training algorithms, three specific ones that are implemented in this thesis are described in details below:

- Multilayer perceptron networks trained by the error back-propagation algorithm (BP).
- Radial basis function (RBF).
- Support Vector Machines (SVM).

3.3.1 Multilayer Perceptron Networks

Multilayer perceptron networks have been applied successfully to many different problems since the advent of the error back-propagation learning algorithm [59]. This network architecture and the corresponding learning algorithm can be viewed as a generalization of the popular least-mean-square (LMS) algorithm.

A multilayer perceptron network consists of an input layer, one or more hidden layers of computation nodes, and an output layer of computation nodes. The input signal propagates through the network in a forward direction, layer by layer. The error back-propagation learning algorithm consists of two phases. The first is usually referred to as the presentation phase or forward pass and the second as the back- propagation phase or backward pass. In the presentation phase an input vector x is presented to the network resulting in an output y at the output layer. During this phase the synaptic weights are all fixed. In the back-propagation phase the weights are adjusted based on the error between the actual and desired outputs. The details regarding the above discussion are presented below.

- **The Presentation Phase**

For simplicity a network with just one hidden layer is analyzed. The following is a list of symbols used in the remainder of this chapter.

- NI: number of neurons in the input layer.
- NH: number of neurons in the hidden layer.
- NO: number of neurons in the output layer.
- x: input vector.
- h^H: input for the hidden layer.
- h^O: input for the output layer.
- y^H : output of the hidden layer.
- y: output of the network.
- w_{ji} : matrix $NH \times NI$ of synaptic weights connecting the input and hidden layers.
- w_{kj}: matrix $NO \times NH$ of synaptic weights connecting the hidden and output layers.
- b: bias, or threshold vector.
- $\phi(.)$: the nonlinear function performed by the neuron.
- $i = [1: NI]$: a neuron in the input layer.
- $j = [1: NH]$: a neuron in the hidden layer.
- $k = [1: NO]$: a neuron in the output layer.

Once an input vector is presented to the input layer one can calculate the input to the hidden layer as follows:

$$h_j^H = b_j + \sum_{i=1}^{NI} w_{ji} x_i . \tag{3.2}$$

Each neuron of the hidden layer takes its input h_j^H and uses it as the argument for a function and produces an output given by:

$$y_j^H = \phi(h_j^H). \tag{3.3}$$

Now the inputs to the neurons of the output layer are calculated as:

$$h_k^O = b_k + \sum_{j=1}^{NH} w_{kj} y_j^H . \tag{3.4}$$

and consequently, the network output is then given by:

$$y_k = \phi(h_k^O). \tag{3.5}$$

- **The Error Back-Propagation Learning Algorithm**

An output error is defined as the difference between the network output and the desired output value, that is for the kth output neuron we have [35, 38],

$$e_k = d_k - y_k . \tag{3.6}$$

Based on the output error one can calculate the sum of squared errors as:

$$\varepsilon = \frac{1}{2} \sum_{k=1}^{NO} e_k^2 . \tag{3.7}$$

This is the cost function that is to be minimized during the learning process. The sum-squared-error ε is a function of all the variables of the network. Using the chain rule one can calculate the gradient of the error with respect to the weight matrix connecting the hidden layer to the output layer as follows:

$$\frac{\partial \varepsilon}{\partial w_{kj}} = \frac{\partial \varepsilon}{\partial e_k}\frac{\partial e_k}{\partial y_k}\frac{\partial y_k}{\partial h_k^O}\frac{\partial h_k^O}{\partial w_{kj}}. \tag{3.8}$$

Computing each term of this expression yields:

$$\frac{\partial \varepsilon}{\partial e_k} = e_k$$

$$\frac{\partial e_k}{\partial y_k} = -1$$

$$\frac{\partial y_k}{\partial h_k^O} = \phi_k^{'}(h_k^O)$$

$$\frac{\partial h_k^O}{\partial w_{kj}} = y_j^H$$

Combining the expressions above results in:

$$\frac{\partial \varepsilon}{\partial w_{kj}} = -e_k \phi_k^{'}(h_k^O) y_j^H. \tag{3.9}$$

The correction Δw_{kj} applied to the weight matrix connecting the hidden layer to the output layer is:

$$\Delta w_{kj} = -\eta \frac{\partial \varepsilon}{w_{kj}} = \eta e_k \phi_k^{'}(h_k^O) y_j^H. \tag{3.10}$$

where η is a constant known as the step-size or the learning rate. The equation above can be rewritten as:

$$\Delta w_{kj} = \eta \delta_k y_j^H . \tag{3.11}$$

where $\delta_k = e_k \phi_k^{'}(h_k^O)$ is the local gradient term. To update the weights connecting the input layer to the hidden layer we need to repeat the procedure above according to

$$\frac{\partial \varepsilon}{\partial w_{ji}} = \frac{\partial \varepsilon}{\partial e_k} \frac{\partial e_k}{\partial y_k} \frac{\partial y_k}{\partial h_k^O} \frac{\partial h_k^O}{\partial y_j^H} \frac{\partial y_j^H}{\partial h_j^H} \frac{\partial h_j^H}{\partial w_{ji}} . \tag{3.12}$$

After calculating each of the terms above, the correction to the weight matrix is written as:

$$\Delta w_{ij} = -\eta \delta_j x_i . \tag{3.13}$$

where $\delta_j = \phi_j^{'}(h_j^H) \sum_{k=1}^{NO} \delta_k w_{kj}$. In general, the correction term is calculated by:

$$\Delta w_{lm} = \eta \delta_m x_l = \text{learning rate} \times \text{local gradient} \times \text{input to the layer}. \tag{3.14}$$

- **Neuronal Activation Functions**

In general the functions $\phi(.)$ of the hidden layer are different from the ones in the output layer. There are many possible choices for the activation functions used to shape the weighted input sum and to produce an output. The selection of the activation function depends on the task of the neuron. Fig. 3.3 shows some activation functions that are commonly used are further described below [35, 38]:

- Hard Limit Transfer Function: This function sets the neuron output at unity if its net input reaches some threshold, otherwise the output is zero.

- The Symmetrical Hard Limit: This function sets the neuron output to unity if its net input reaches a pre-specified threshold, otherwise, the output is set to -1.
- The Linear Transfer Function: It passes the neuron's input signal after multiplying it by some scaling constant (slope) and adding a neuron bias to its output port.
- The Saturated linear Transfer Function: The saturated linear transfer function output is -1 if the input is less than -1 and has an output +1 if the input is larger than +1. In between the interval [-1, +1] the neuron acts as a linear neuron with slope=1.
- The Log-Sigmoid: This function is used to produce an output that varies from 0 to +1 as the input varies from $-\infty$ to $+\infty$. The log-sigmoid is a differentiable function and that makes it suitable for networks that are trained with error back propagation algorithm.
- The Tan-Sigmoid Function: This function is used to map neuron input in the interval $(-\infty,\infty)$ to obtain an output varying from -1 to +1

The linear function is normally used in the output layer while in the hidden layer the sigmoidal class of functions is the preferred choice. If the functions in the hidden and the output layers are selected to be linear one obtains a LMS network.

The log-sigmoidal function is formally defined as:

$$y = \frac{1}{1+e^{-x}}, \qquad -\infty < x < +\infty. \tag{3.15}$$

with

$$\frac{\partial y}{\partial x} = \frac{e^{-x}}{(1+e^{-x})^2} = y(1-y). \tag{3.16}$$

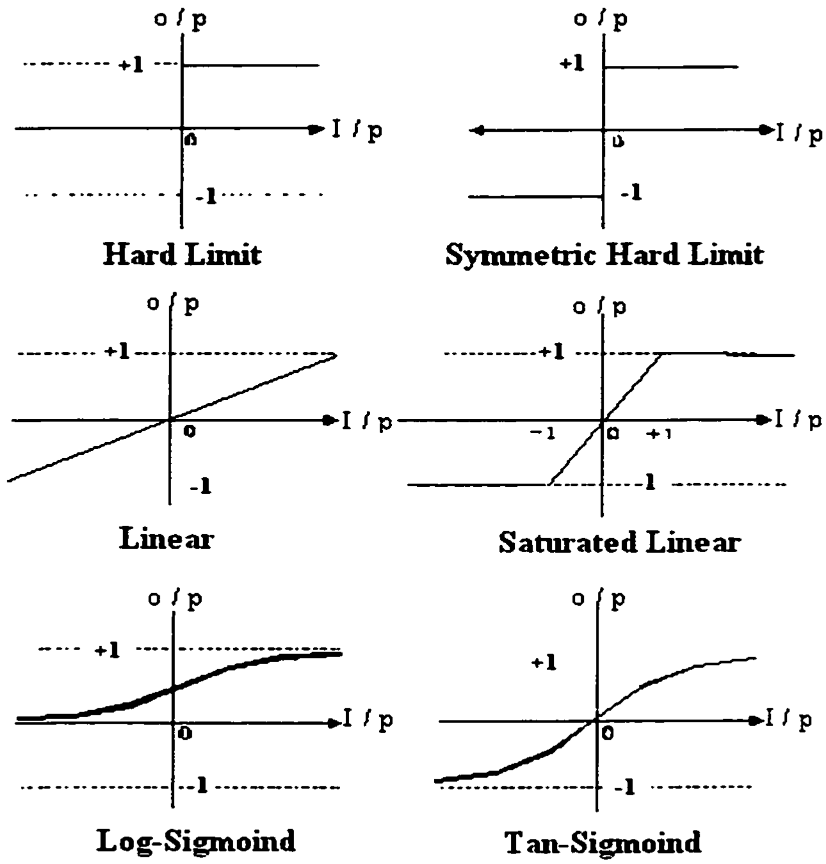


Figure 3.3: Commonly used activation functions for a neuron, where o/p denotes output of the neuron and I/p denotes the input.

The special form of the derivative makes this function very attractive for practical implementation considerations. A variation to the log-sigmoidal function is the tan-sigmoidal function that is formally given by:

$$y = a\tanh(bx) = a\frac{1-e^{-bx}}{1+e^{-bx}} = \frac{2a}{1+e^{-bx}} - a. \tag{3.17}$$

The common choices for a and b are $a = 1.716$, and $b = 2/3$.

The multilayer perceptron network has been proven to be a universal approximator, implying that it can approximate any continuous multivariate function to any degree of accuracy provided that enough hidden neurons are selected [60, 61].

As with any steepest-descent gradient algorithm, the error back-propagation algorithm suffers from slow convergence and high probability of converging to a local minima. There are ways to improve the algorithm by using adaptive learning rates and momentum terms as described in [35] .

Other training algorithms have also been developed for multilayer perceptron networks. Among these the most popular algorithms are the ones that are based on conjugate-gradient techniques and on Newton's method [35].

3.3.2 Radial Basis Function Networks

Radial basis functions (RBF) were first used in the context of neural networks by Broomhead and Lowe [62] in 1988. These networks belong to a single layer perceptron category where the output of the neurons in the hidden layer is expressed as [35, 38]:

$$y(z) = \sum_{j}^{NH} w_j \phi_j(z) \tag{3.18}$$

where the argument z has the form:

$$z = (x - c_j)^T R_j^{-1} (x - c_j) \tag{3.19}$$

and where y is the vector denoting the output of the network, x is the input vector to the network, R is the weighting matrix that depends on a metric selected, and ϕ is the radial basis function.

By using the Euclidean metric, the argument z is written as:

$$z = \frac{\|x - c_j\|^2}{\sigma_j^2} \tag{3.20}$$

where c_j σ_j denote the "center" and "variance" of the functions of the neurons. For the network to be a linear function of the free parameters *wij*, the number of hidden neurons (p), their position (*cj*), and the metric R or σ_j) have to be constant. Often a constant term, called bias or threshold, is added to the weighted sum as in Equation (3.18) so that the function now becomes:

$$y(z) = w_0 + \sum_{j=1}^{NH} w_j \phi_j(z). \tag{3.21}$$

Park and Sandberg [63] showed that the RBF networks are suitable for nonparametric regression and that under certain conditions they are universal approximators. The conditions are:

- The hidden layer must contain enough neurons, and
- The function $\phi(.)$ must be continuous, bounded and monotonic in $[0, \infty)$.

Common choices for $\phi(.)$ are the Gaussian functions of the form:

$$\phi(z) = e^{-z}, \tag{3.22}$$

and the Cauchy functions of the form:

$$\phi(z) = \frac{1}{1+z} \tag{3.23}$$

The Gaussian and Cauchy functions may then be written as follows;

$$\phi_i(x) = e^{-\frac{\|x-c_i\|^2}{\sigma_i^2}} \tag{3.24}$$

and

$$\phi_i(x) = \frac{1}{1 - \frac{\|x - c_i\|^2}{\sigma_i^2}} \tag{3.25}$$

The parameter σ (the spread constant) determines how the basis function selectively responds to an input. For large values of σ the range of input activation of the basis function is broader. The center c determines where the maximum output will occur.

A Radial Basis Function (RBF) network has the architecture shown in Figure 3.4. When the input vector x is presented to the network its distance to each of the centers cj is measured and each neuron in the hidden layer will output a value between 0 and 1 according to the proximity of the input vector to the neuron's center. This output is weighted by the connections between the hidden and output layers to yield the network output y. Neurons with centers far from the input vector will have output close to zero.

This small output will have only a small effect on the output neurons. In contrast, neurons with centers close to the input vector x will output values close to one, and will influence the final output, y.

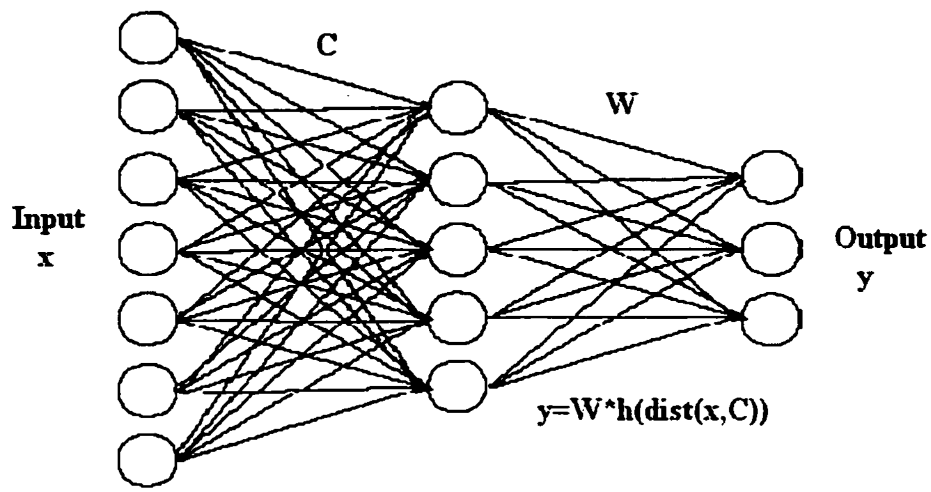


Figure 3.4: RBF network architecture.

- **The Learning Process**

RBF network is completely defined by Equations (3.18), (3.21), and (3.22) or (3.23) and Figure 3.4 is its schematic representation. Each radial basis function defines a hyper-spherical receptive field. The i^{th} neuron outputs a signal that is close to one for sample activation vectors x that fall within its receptive field, and smaller values otherwise. The training of RBF network comprises the following steps [35]:

- Selection of the centers: There are different ways of selecting the centers. The most commonly used approaches are: (i) to use the input vectors as centers; (ii) to randomly choose vectors from the input set as centers; and (iii) to use the K-means algorithm [64, 65] to cluster the input vectors and select the centers based on the number of clusters and their variance.

- Choice of σ: σ can also be chosen in a variety of ways. It is a measure of the variance of the input set. One way of calculating σ is to use the P-nearest neighbor algorithm according to:

$$\sigma_j = \frac{1}{P}\sum_{k=1}^{P}(\|c_j - c_k\|^2)^{1/2}. \tag{3.26}$$

where c_k is the P-nearest neighbor of σ_j and P is determined heuristically.

- Calculation of the weights : The output of the network is given by:

$$y = W\phi(x,c,\sigma) \Rightarrow W = y\phi^T(\phi\phi^T)^{-1}. \tag{3.27}$$

The training of the weights are continued so that the error of the network is close to zero for the training input vectors. The quality and performance of the generalization capability of the network would then depend on how well the training set represents the solution space.

3.3.3 Support Vector Machines

Support Vector Machines (SVM) was invented by Vladimir Vapnik [35]. They are a method for creating functions from a set of labeled training data. The function can be a classification function (the output is binary) or it can be a general regression function.

Support Vector Machines, like multilayer perceptrons and radial basis function networks, are used for pattern classification tasks [35].

The theory of Support Vector Machines (SVM) is based on Statistical Learning Theory [66, 67]. SVM use linear separating hyper-planes to create a classifier. For the problems that cannot be linearly separated in the original input space, SVM would non-linearly transform the original input space into a higher dimensional feature space. In this feature space it is then possible to find linear optimal separating hyper-planes in the sense of maximizing the margin of the classifier with respect to training data [35]. Thus, for classification problems, SVM operate by finding a hyper-surface in the space of possible inputs. This hyper-surface will attempt to split the positive examples from the negative examples. The split is chosen to have the largest distance from the hyper-surface to the nearest of the positive and negative examples. Intuitively, this makes the classification robust for testing data that is near, but not identical to the training data [35, 67, 68].

- **SVM for the Classification Problem**

SVM is a pattern recognition technique whose foundations stem from statistical learning theory. The basic training principle that is used in SVM is to find the optimal linear hyper-plane such that the expected classification error is minimized.

For classification problem, first let us consider the case of classifying the set of linear separating samples

$$(x_i, y_i),\ i = 1, \dots l.\ x_i \in R^n,\ y_i \in \{-1, +1\}. \tag{3.28}$$

The general form of a linear decision surface function in n-dimensional space is $g(x) = w^T x + b$, where x is an input vector, w is an adjustable weight vector and b is the bias. Let w_O and b_O be the optimum values of the weight vector and bias, respectively. Correspondingly, the classification optimal hyper-plane is defined by:

$$w_o^T x + b_o = 0$$

and

$$g(x) = w_o^T x + b_o \tag{3.29}$$

A set of vectors is said to be optimally separated by the hyper-plane if this set is separated without error and the minimal distance between the vector x_i and the hyper-plane is maximal. Without loss of generality it is appropriate to consider a canonical hyper-plane, i.e. to normalize the function $g(x)$. In other words, all samples would satisfy:

$$|g(x)| = |w_o^T x_i + b_o| \geq 1 \tag{3.30}$$

Such constrained condition is favorable to simplify the formula developed later. It can be shown that the margin is $2/||w_o||$. A separating hyper-plane in canonical form must then satisfy the following constraints:

$$y_i[(w^T x_i) + b] \geq 1, \qquad i = 1,..l \tag{3.31}$$

Hence, the hyper-plane that optimally separates the data is the one that satisfies the above equation and minimizes:

$$\Phi(w) = \frac{1}{2} w^T w \tag{3.32}$$

The optimization solution of Equation (3.32) under the constraints of Equation (3.31) is given by the Lagrangian:

$$L(w,b,\alpha) = \frac{1}{2}(w^T w) - \sum_{i=1}^{l} \alpha_i \{y_i[(w^T x_i) + b] - 1\} \tag{3.33}$$

where a_i's are the Lagrange multipliers. The Lagrangian *L (w, b, a)* has to be minimized with respect to *w, b* and maximized with respect to $a_i \geq 0$. Classical Lagrangian duality principle enables the primal problem, Equation (3.33), to be transformed to its dual problem, which is easier to solve. The dual problem is to maximize:

$$Q(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j (x_i - x_j) \tag{3.34}$$

If the solution to the problem is $\alpha_{o,i}$, then:

$$w_o = \sum_{i=1}^{l} \alpha_{o,i} y_i x_i \tag{3.35}$$

Because most $\alpha_{o,i}$'s are equal to zero, samples corresponding to non-zero $\alpha_{o,i}$'s are support vectors. They are often part of all the samples.

According to the support vectors, the optimal classification function is

$$f(x) = \mathrm{sgn}\{(w_o^T x) + b_o\} = \mathrm{sgn}\{\sum_{i=1}^{l} \alpha_{o,i} y_i (x_i^T x) + b_o\} \tag{3.36}$$

If the two classes are not linearly separable, the input vectors are then non-linearly mapped to a higher dimensional feature space by an inner product kernel function $K(x, x_i)$. If the inner product kernel function is used instead of the dot product, the optimal decision function becomes:

$$f(x) = \mathrm{sgn}\{(w_o^T x) + b_o\} = \mathrm{sgn}\{\sum_{i=1}^{l} \alpha_{o,i} y_i K(x_i^T x) + b_o\} \tag{3.37}$$

More detailed information on support vector machines may be found in [66, 69- 72].

- **Classification using SVM**

As discussed in the previous section, after support vectors are obtained, the optimal classification function is determined, which can be applied to data classification. The detailed steps are as follows:

For the kernel functions with restricted domain, data normalization is required. This normalization is also advantageous for non-restricted kernels. Therefore, before classification is performed, the input data needs to be pre-processed first. This normalization is performed by computing the inner production of the kernel functions. The classifier is then trained using training samples so that the support vectors required for classification are obtained. The support vector set is a subset of the training data. Finally, the performance of the classifier is tested using the test samples.

For implementation purposes, since one is not aware a priori whether the data is linearly separable or not, therefore, one needs to introduce an additional cost function associated with the misclassification of data. This can be accomplished by introducing positive slack variables ζ_i, $i=1, 2, \ldots, l$. The generalized optimal separating hyper-plane is then determined by vector w, which minimizes the function

$$\Phi(w,\zeta) = \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \tag{3.38}$$

where C is a value that controls the magnitude of the penalty added to the errors, and realizes the compromise between the ratio of misclassified samples to the algorithm complexity. Correspondingly, the constraint in Equation (3.35) is $0 \leq \alpha_i \leq C$. It is important to note that the parameter C is to be selected by the user experimentally by trail and error or analytically [72].

3.3.4 Selecting the Proper Network

Neural networks are nonparametric models able to represent any function with arbitrary accuracy, provided enough neurons are selected. Supervised learning is the preferred algorithm to train a neural network for function approximation and from the many existing architectures, feed-forward and radial basis function networks are the most widely used.

The back-propagation learning algorithm is the most commonly used procedure yielding usually good generalization capabilities. However, the algorithm generally requires long training periods and may possibly converge to local minima. Radial basis function networks produce, on the other hand, more localized approximation of the mapping considered and depend linearly on the connection weights. For the SVM approach, the optimization criterion attempts to maximize the classification margin, that is the distance from the closest data point to the separating hyper-plane. The location and slope of this hyper-plane are defined by a group of training points called support vectors. When choosing between one of the above three approaches one should consider the characteristics of the task being performed, the available amount of training data, and the amount of time required to train the network for producing acceptable desired results.

3.4 Neural Network-based Approaches Power System Problems

Due to a combination of increasing energy consumption and impediments of various kinds to extension of existing electric transmission networks, the power systems are operated closer and closer to their limits. This situation requires a significantly less conservative power system operation and control regime which, in turn, is possible only by monitoring the system state in much more detailed form than was necessary previously.

Fortunately, the large quantity of information required can be provided in many cases through recent advances in telecommunications and computing techniques. There is, however, a lack in evaluation techniques required to extract the salient information and to use it for higher order information processing. Whilst the sheer quantity of available information is always a problem, this situation is aggravated in emergency situations when rapid decisions are required. Because of the high dimensionality and nonlinearities of power systems, its monitoring and control involves several hundred variables. Nonlinear load demands and dynamic loads are difficult to model. These problems provide an important motivation to explore novel data processing techniques such as neural networks.

Different power system problems, where neural network approaches have been investigated, are documented in the literature [3, 10-35]. Neural networks have the inherent capacity of modeling functional relationships between input and output data without the explicit knowledge of an analytical model. Neural networks can easily be adapted to several data sets in a short amount of time. This feature, also called "black-box approach", encouraged researchers to apply neural networks to load forecasting where they were already applied as early as 1975 [73].

In the area of on-line transient stability assessments early work on pattern recognition techniques inspired research on simple neural network models with supervised learning [16]. In addition to load forecasting and security assessment, other applications of neural networks to power systems are also explored such as, control, system identification, optimization and alarm processing [16, 41].

3.4.1 Neural Networks for Transmission Line Relaying

As discussed in Chapter 2, one of the most popular transmission line protection principles for high voltage applications is the distance relaying. A distance relay for the protection of transmission line is usually designed on the basis of a fixed setting. The relay either overreaches or underreaches depending on the operating conditions of power system and location of the fault. Since fault detection and classification is traditionally performed on-line, new pattern recognition techniques should be developed and implemented that are quick and flexible. Comprehensive surveys of literature using neural networks for distance relaying are found in [16, 25]. The uniqueness of the neural networks based distance protection is that it does not explicitly use impedance information as the basis of decision making criteria but rather learns it from examples presented to it during the training process. The neural network operates as a pattern classifier and is able to detect the changing power system conditions quickly and accurately. Consequently, resulting in the improvement of performance over conventional digital relays techniques.

Various applications of neural networks were used in the past to improve the recognition capability of the impedance used in the distance relaying of transmission lines [25].

3.4.2 Network Architectures used for Transmission Line Protection

To employ neural network techniques successfully, some fundamental issues should be considered. Among these issues is the selection of the network architecture and the learning algorithm, such as the net size, learning step, number of training patterns and number of iterations. However, almost all the studies in the literature have so far only employed the back-propagation (BP) neural network structure with supervised learning. Despite the advantages of back-propagation networks, they have their drawbacks such as slow learning process, requirement of large training sets, easily getting trapped in local minima and poor robustness [35].

One other alternative is the use of unsupervised learning. A typical unsupervised learning commonly used is the self-organizing maps (SOM) developed by Teuvo Kohonen [35, 45]. A SOM network possesses the advantage of fast learning and small training sets. However, due to the absence of a desired output information, it is not suitable for use by itself for some decision-making processes. Rather, it is used as a front-end to an output layer with desired target information through a supervised training process. That is, a combination of both an unsupervised learning as well as a supervised learning process integrated together. Combined unsupervised and unsupervised networks have the added capability to sort out very complex, highly non-linear pattern recognition problems, such as transmission line fault diagnosis. This type of neural network is insensitive to noise due to the low dimensional internal representation, thus resulting in improved robustness [35].

Among the other class of networks that are used for transmission line protection are radial basis functions (RBF) and counter-propagation (CP) networks [41, 45].

3.4.3 Fault Diagnosis in Transmission Line Systems

In general, a fault diagnostic system consists of three components: detection, isolation and identification. Fault detection and isolation are very often referred to in the literature as fault diagnosis and abbreviated as (FDI) [46, 47, 48].

A comprehensive scheme for fault diagnosis of transmission line systems should accomplish the following three tasks:

- Fault Detection: To establish and determine if a fault has occurred in the line or not.
- Fault Classification: To determine what is the type of the fault detected.
- Fault Location: To determine in which zone the faulty line is located.

Accordingly, a neural network or a bank of networks should be designed to carry out each of the above three tasks for the problem of fault diagnosis in the transmission line system.

In the next chapter, the above three tasks are accomplished by properly designing and developing suitable networks. Different neural network architectures are investigated for these three tasks and comparative study is conducted to determine the advantages and disadvantages of these solutions.

Chapter 4

Protection of Power Transmission Line Systems using Neural Networks

4.1 Introduction

As discussed in the previous chapter, neural networks have been proposed and used to protect transmission lines by many researchers [24, 26-28, 30, 34, 36, 45-48]. The capabilities of neural networks to perform pattern recognition and classification are employed in this part of the thesis to address the fault diagnosis problem in this area.

This chapter presents a neural network-based approach for developing and implementing a complete scheme for distance protection of a transmission line system. In order to perform this goal, the distance protection problem is subdivided into different neural network models for fault detection, fault identification and fault location corresponding to different protection zones.

4.2 Modeling the Transmission Line System

A 440 kV transmission line system is used to develop and implement the proposed architectures and algorithms for this problem. Figure 4.1 shows a one-line diagram of the system used to train and test the neural networks. The system consists of two generators, four buses and three transmission lines. The zones of protection are shown in Figure 4.1, where zone1 is 142.4 Km, zone 2 is 195 Km and zone 3 is 225 Km from bus B respectively. More detailed information regarding the transmission line parameters are presented in Appendix A. The line is modeled with four Π sections. A typical Π section is shown in Figure 4.2.

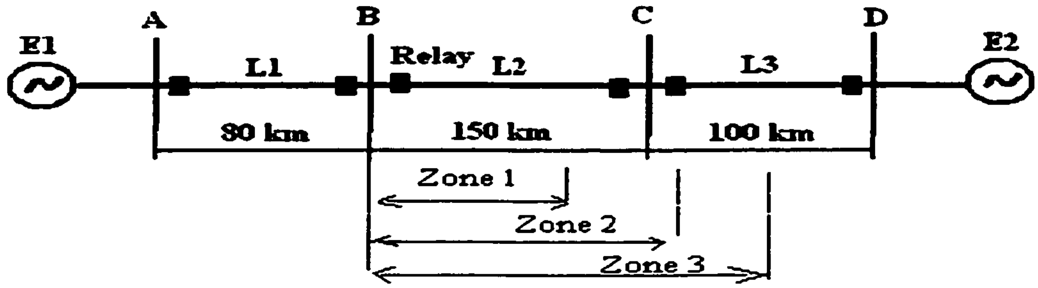


Figure 4.1: One-line diagram of the system studied.

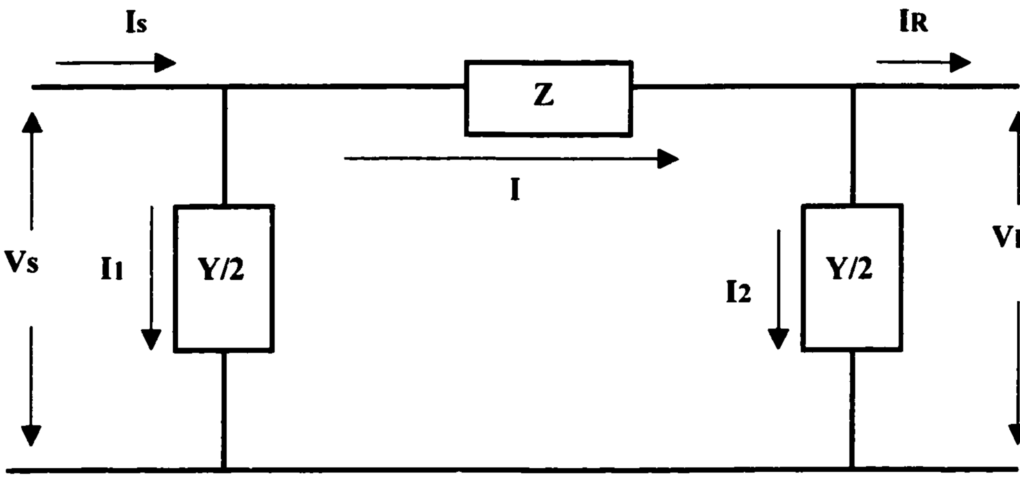


Figure 4.2: A typical *Π* section equivalent model of the transmission line system.

In Figure 4.2, V_S and I_S are the sending end voltage and current and V_R and I_R are the receiving end voltage and current, respectively. In general, using a *Π* section representation of a transmission line, the voltage at sending end can be expressed as:

$$V_S = I \times Z + V_R$$

Using $I = I_2 + I_R$ we now have

$$V_S = (V_R \times \frac{1}{2} Y + I_R) Z + V_R$$

or equivalently,

$$V_S = (1 \times \frac{1}{2}YZ)V_R + ZI_R \tag{4.1}$$

The current at the sending end is represented by:

$$I_S = I_1 + I$$

or equivalently $$I_S = [(1 + \frac{1}{2}YZ)V_R + ZI_R]\frac{1}{2}Y + (V_R + ZI_R)\frac{1}{2}Y$$

that is,

$$I_S = (Y + \frac{1}{4}Y^2Z)V_R + (1 + \frac{1}{2}YZ)I_R \tag{4.2}$$

Equations (4.1) and (4.2) can be rewritten in the following compact matrix form:

$$\begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} 1 + \frac{1}{2}ZY & Z \\ Y + \frac{1}{4}ZY^2 & 1 + \frac{1}{2}ZY \end{bmatrix} \cdot \begin{bmatrix} V_R \\ I_R \end{bmatrix} \tag{4.3}$$

The above power system was simulated using the *Matlab Power System Block Set*. The parameters and the data corresponding to the system are listed in Appendix A.

The three-phase voltages and currents, $V = [V_a \quad V_b \quad V_c]^T$ and $I = [I_a \quad I_b \quad I_c]^T$ are measured at bus *B* in Figure 4.1. These signals will be utilized subsequently as inputs to the proposed neural networks-based schemes. The Matlab Power System Block Set is used to generate the data for the 440 kV transmission line system corresponding to both normal and faulty conditions.

In the subsequent simulation results we consider the following four categories, namely (i) phase to ground faults, (ii) phase to phase faults, (iii) double-phase to ground faults and (iv) three-phase faults. The data set required for training the neural networks developed below is generated from various fault situations considering different fault locations as shown in details in Appendix B.

For training, validating and generalization purposes more than 300 different fault cases were generated for the detection task, about 130 cases for the classification problem and more than 1000 cases for the fault location task.

4.3 Top View of the Proposed Scheme

Although the relay characteristics, in most cases, remain fixed, the trip/no trip decisions using digital technology-based distance relays have experienced some improvements as compared to older electromechanical relays. The objectives of this chapter are to design, test and implement a complete methodology for the problem of transmission line system fault diagnosis. As a pre-processing step the training and the testing data generated from the transmission line system are collected. The first step is that of the detection of a fault situation in the system. Following that, fault classification and fault isolation/ location (zone 1, 2 or 3) tasks are investigated. As stated before, the contribution of this thesis is to propose an integrated methodology by utilizing a bank of neural networks to perform these tasks. A back-propagation neural network is used for the fault detection while three neural network architectures are designed for the classification and isolation tasks, respectively. A comparison between the proposed architectures is also investigated and presented.

4.3.1 Data Pre-Processing and Feature Extraction

Feature extraction is the process by which one intends to reduce the size of the neural network and improves its performance [2]. In order to make use of all the relevant information contained in the waveforms and obtain fast response, instantaneous values of the voltages and currents could be utilized [21]. On the other hand, this could require a large neural network size and a large body of training patterns. In fact, due to the stochastic nature of the time domain waveforms, simply considering a limited window of the instantaneous signals may not be representative of the fault voltages and currents [1].

In order to convert the instantaneous voltage and current signals to practically constant values, they were transformed to RMS values. The per unit values are used to scale the RMS values of the voltages and currents. The use of the per unit scaling is due to the presence of different nominal values for the voltages and currents measured in volts and amps. To select a characteristic feature fault and to reduce the training set and, therefore, the training computation time, the average values of the voltages and currents during the fault period are used.

As an illustration, Table 4.1 shows the voltage and current per unit values used as a training set in the normal condition (no fault) as well as the training set subject to various fault types occurring at bus C, 150 km from bus B.

Table 4.1: Voltage and current values at bus B.
Normal condition as well as faulty cases occurring at bus C, 150 km from bus B.

Case No.	**Input Vector (pu)**						**Fault Type**
	V_a	V_b	V_c	I_a	I_b	I_c	
1	0.6345	0.6345	0.6345	0.1345	0.1345	0.1345	No Fault
2	0.00439	0.5404	0.5618	0.09496	0.1235	0.1143	A to Ground
3	0.5618	0.00439	0.5404	0.1143	0.09496	0.1235	B to Ground
4	0.5404	0.5618	0.00439	0.1235	0.1143	0.09496	C to Ground
5	0.3174	0.3171	0.6345	0.09745	0.09094	0.1345	A to B
6	0.6345	0.3174	0.3171	0.1345	0.09745	0.1345	B to C
7	0.3171	0.6345	0.3174	0.09094	0.1345	0.09094	C to A
8	0.00513	0.006046	0.1527	0.09005	0.1047	0.05796	A to B to Ground
9	0.1527	0.005126	0.00605	0.05796	0.09005	0.1047	B to C to Ground
10	0.00605	0.1527	0.00513	0.1047	0.05796	0.09005	C to A to Ground
11	0.00198	0.00198	0.001982	0.07625	0.07625	0.0763	A to B to C to Ground

In the above table it should be noted that the voltage of the faulty phase is approximately zero as expected. It should also be noted that, since the system is considered to be balanced, the voltages and currents of the faulty phases have the same values when they experience the same fault type.

4.3.2 Training Data

As indicated in Section 4.3.1, for all the networks considered in this chapter, inputs are the average per unit RMS values of the voltages and currents of each phase at the reference point at bus *B* (refer to Section 4.2.1), and the output is the fault condition.

In order to train all the proposed networks, the input and output data is applied sequentially. To obtain enough examples for training, different types of faults are simulated at different fault locations. Considering these factors, 130 different fault cases for each fault category are used to train the proposed networks.

Furthermore, different locations corresponding to various fault categories are also considered. In addition, other system parameters associated with the fault are varied within each simulation, namely

- Fault resistance is varied from 0 Ω, 0.5 Ω, 0.75 Ω, 1.0 Ω, 5 Ω, 10 Ω, 25 Ω, 50 Ω, 75 Ω to 95 Ω.

- Fault inception angle is varied from 0, 30, 45, 90, 150, 180, 210, 270, 315 to 330 electric degrees on the phase voltage waveform.

In the following sections we consider the design, implementation and testing of all the proposed neural networks to perform the three objectives of fault detection, classification and isolation.

4.4 Design of Neural Networks for Fault Detection

In order to design a neural network for addressing the fault detection problem, several different topologies of MLP (Multi Layer Perceptron) neural networks are studied. The criteria used to implement and select an appropriate MLP neural network for the problem of fault detection does take into consideration the factors such as the network size, the suitable learning rule, and the size of the training data.

4.4.1 Training Procedure and Learning Rule

The back-propagation learning rule is used in perhaps in over 80-90% of practical applications [48]. However, the standard back-propagation training algorithm is slow, since it generally requires small learning rates for stable learning process so that changes in the network weights using the steepest descent algorithm remains small.

Some techniques to improve the standard back-propagation method such as the addition of momentum terms and adaptive learning rate as well as alternative methods to the gradient descent such as Levenber-Marquadt optimization routine can also be used.

Through the application of various improvement techniques to different network architectures, it was determined that the most suitable training method for the selected network was the back-propagation method based on the Levenber-Marquadt optimization routine.

During the training process, different learning rates were also tried out in the hidden and the output layers. The momentum factor was also included and adjusted to yield the most suitable results.

4.4.2 Selecting the Right Network Size

Selecting the right structure size of the network reduces not only the training time but also significantly impacts the generalization and representational capabilities of the trained network [35]. The number of the hidden layers and neurons in these layers are an important factor in determining the optimal size and structure of the network. Clearly, the lower the number of hidden layers and neurons, the smaller the network. However, a minimum number of neurons are generally needed to represent a given problem which cannot be determined a priori [35].

As mentioned earlier, the three phase voltage and current values are used as inputs to the network. The network has one output to discriminate between the fault and the normal situations. Figures 4.3 to 4.8 depict the training process performance and the squared sum error of some of the networks studied.

The networks presented here represent only a sample of those that were investigated and correspond to the "best" results that were obtained after extensive trail and error procedure.

RMS error is often used as a criterion to terminate the learning process. Figures 4.4, 4.6, 4.8 and 4.11 illustrate the learning error in terms of the RMS of the network. The time to reach a pre-specified error target (learning speed) is an important factor that one has to consider in the task of network selection. However, it should be pointed out that smaller RMS values do not necessarily imply a better generalization performance. There is generally a trade-off between the learning error and the generalization error.

After extensive simulations, it was decided that the desired selected network would have one hidden layer with four hidden neurons. The selected network is shown in Figure 4.9 and the performance and error plots associated with this architecture are given by Figures 4.10 and 4.11.

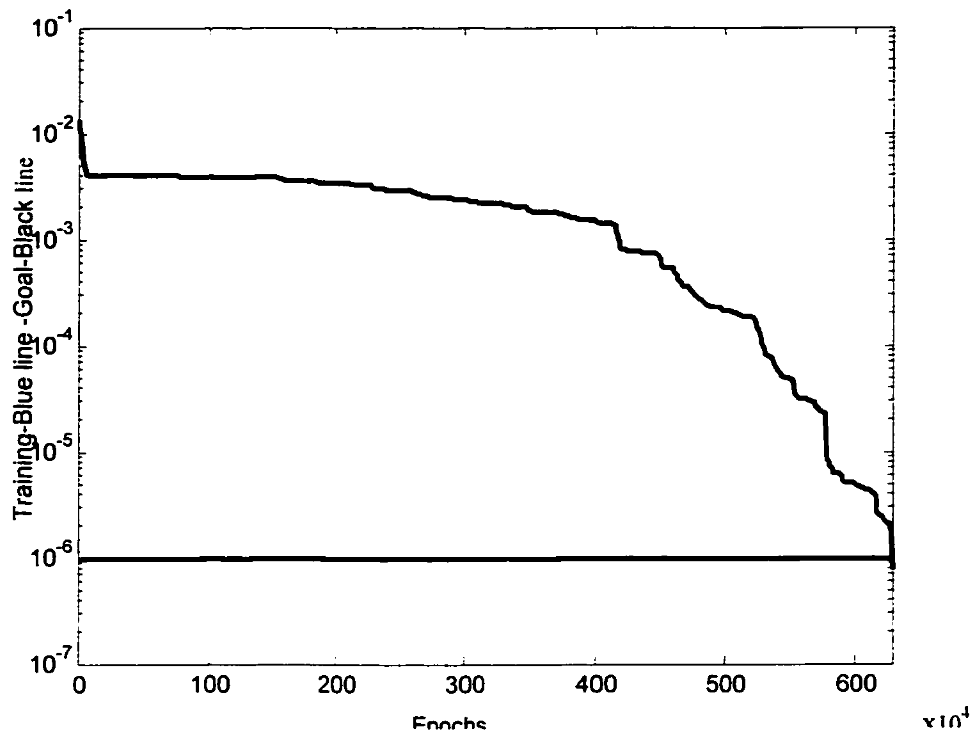


Figure 4.3: Learning process for the BP neural network 6-5-5-1.

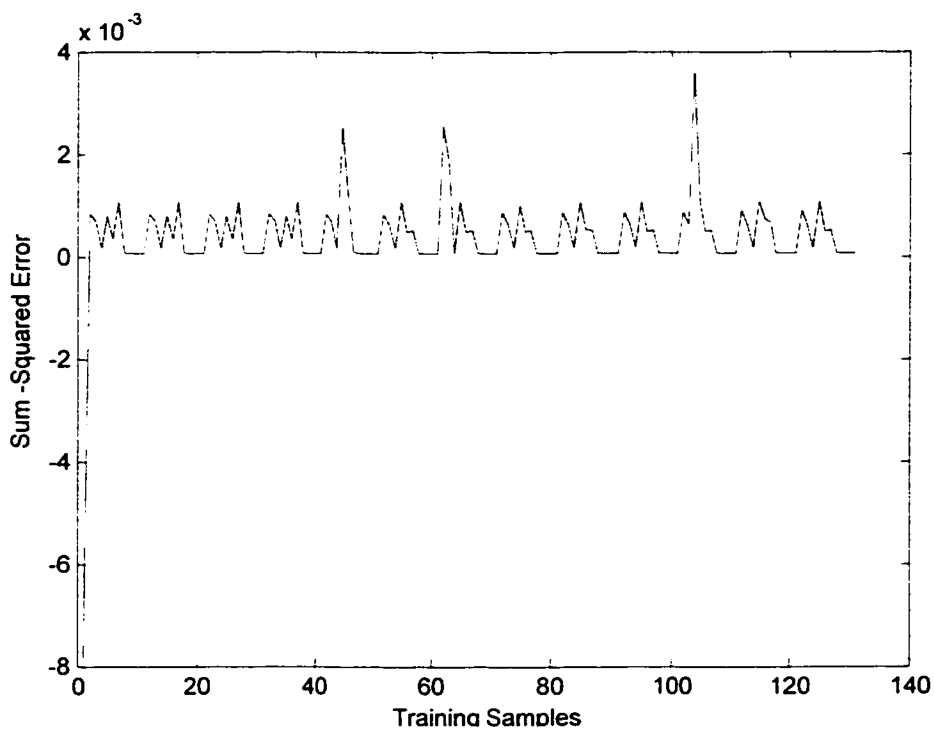


Figure 4.4: Learning process output error for the BP neural network 6-5-5-1.

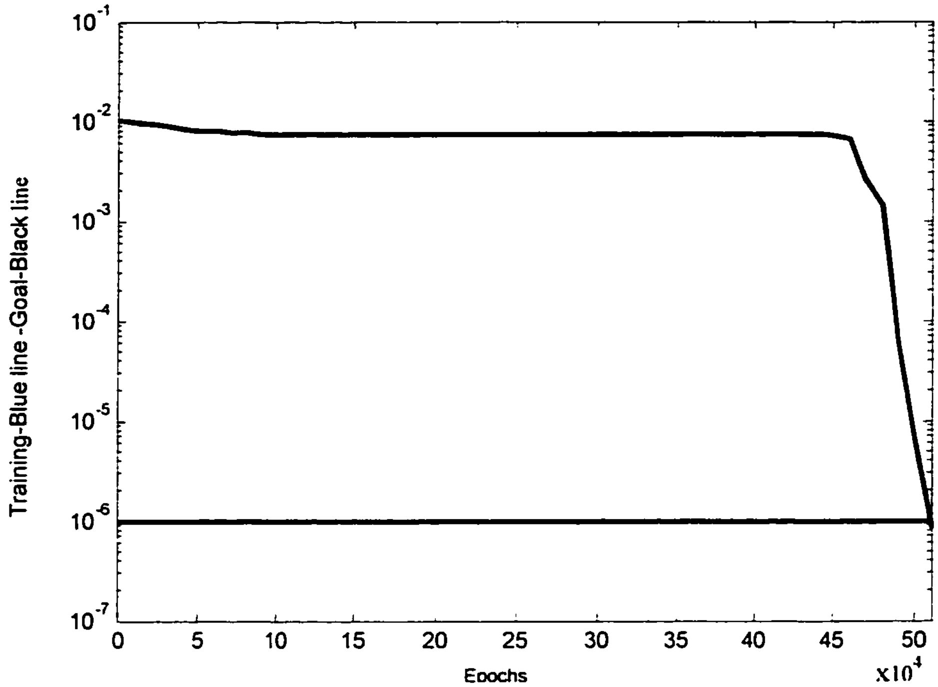


Figure 4.5: Learning process for the BP neural network 6-2-1.

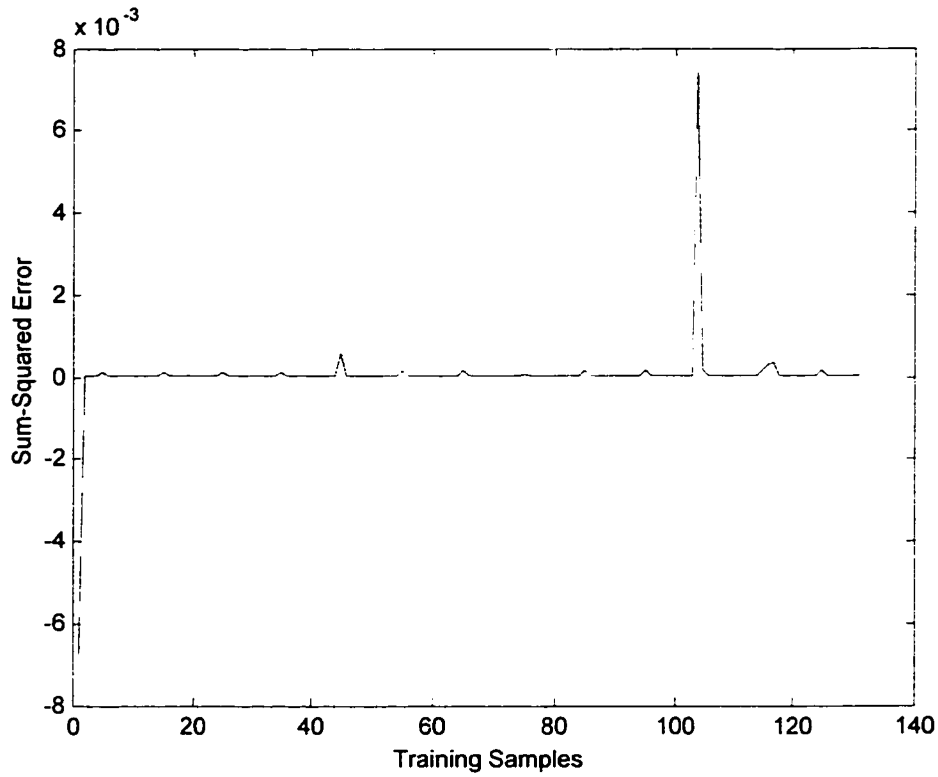


Figure 4.6: Learning process output error for the BP neural network 6-2-1.

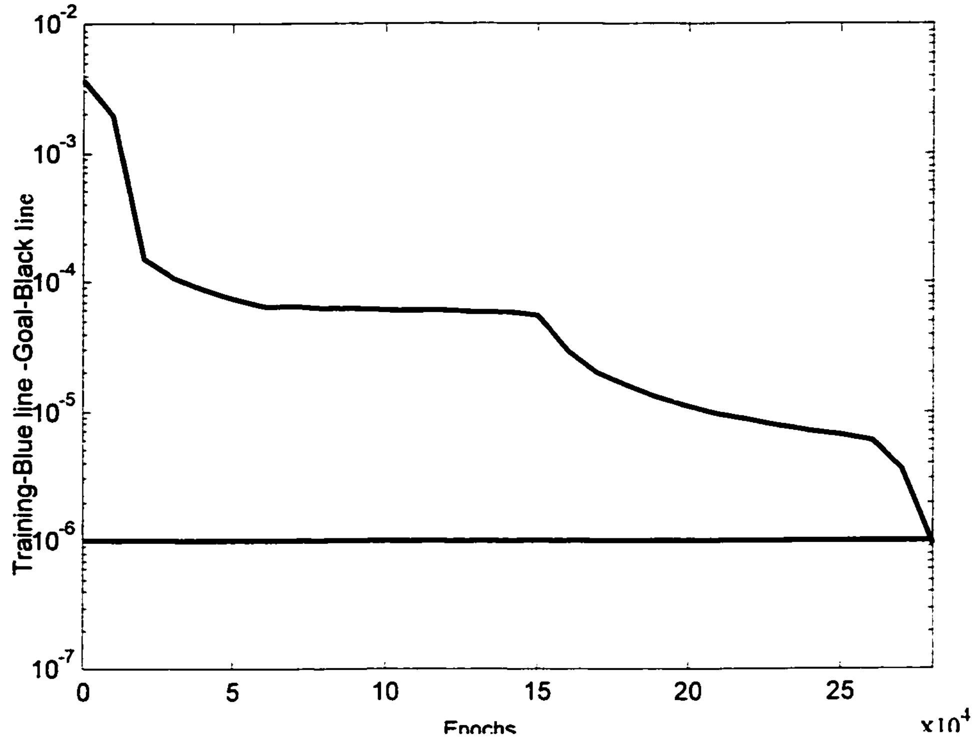


Figure 4.7: Learning process for the BP neural network 6-3-1.

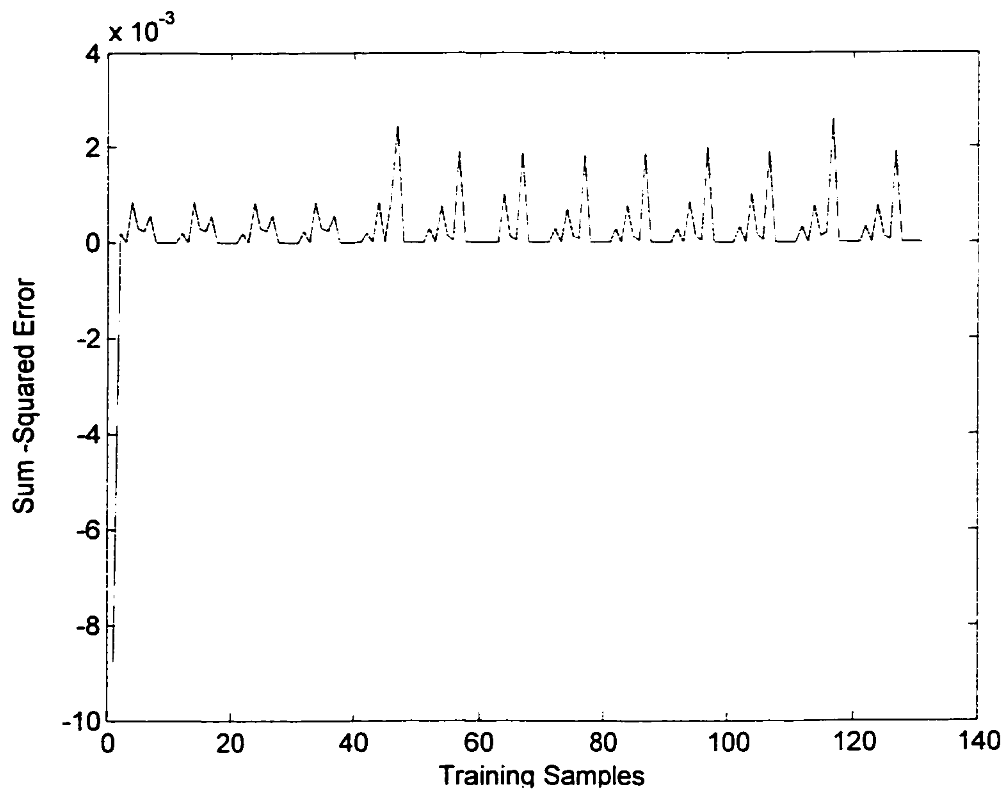


Figure 4.8: Learning process output error for the BP neural network 6-3-1.

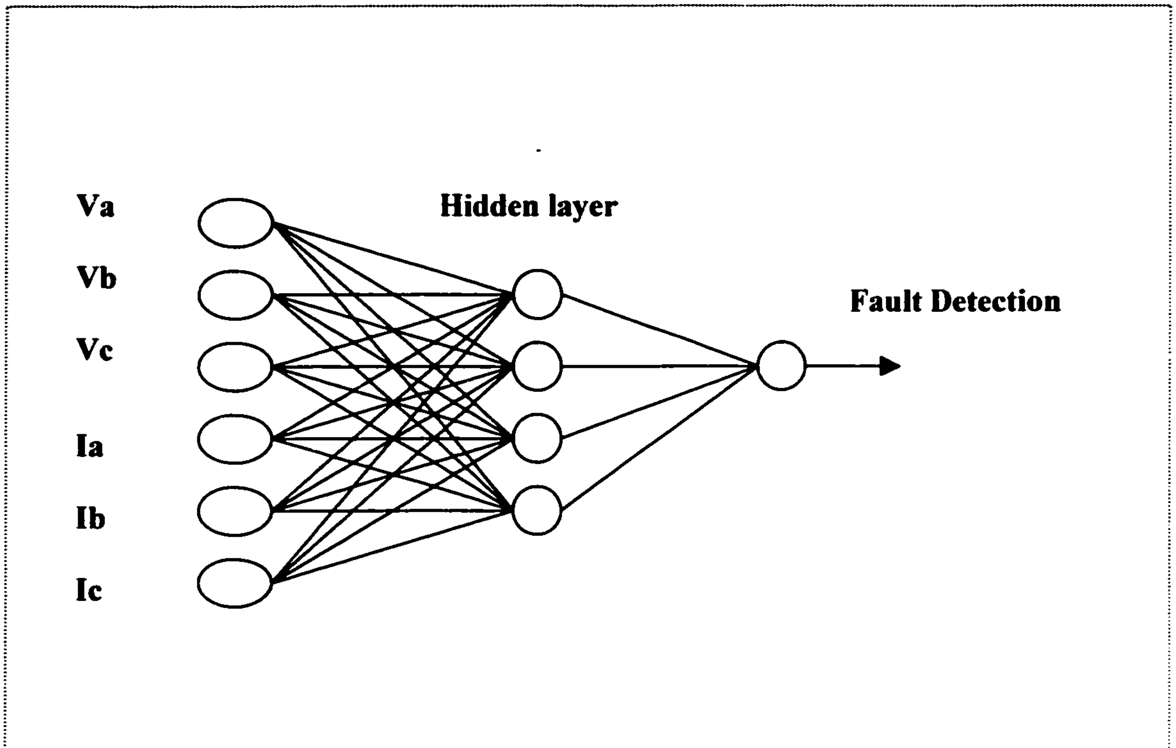


Figure 4.9: Neural network used for fault detection.

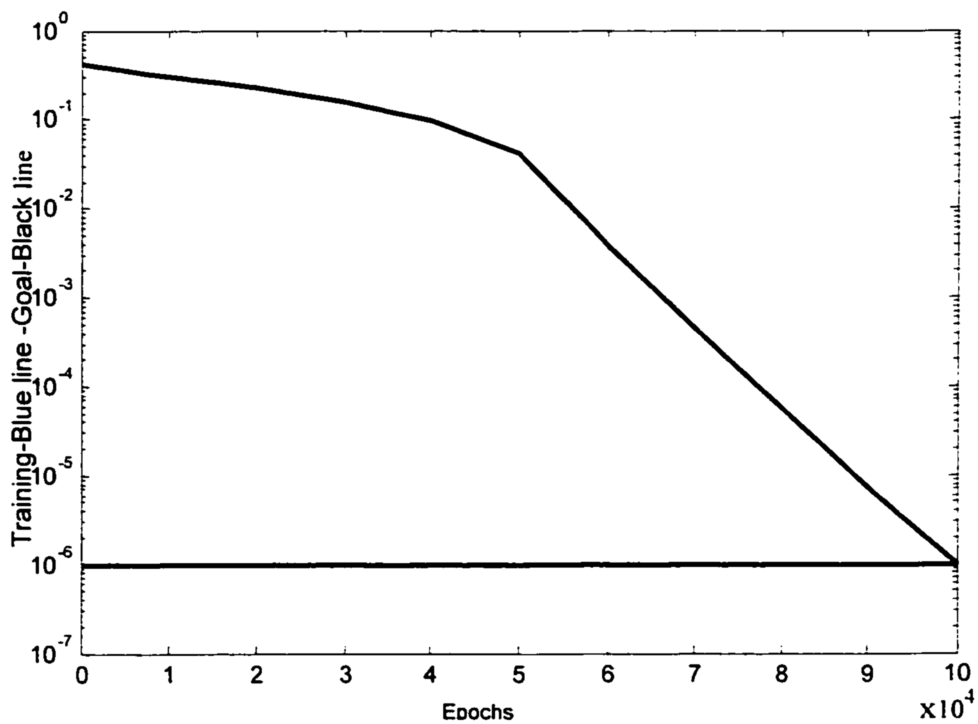


Figure 4.10: Learning process for the BP neural network 6-4-1.

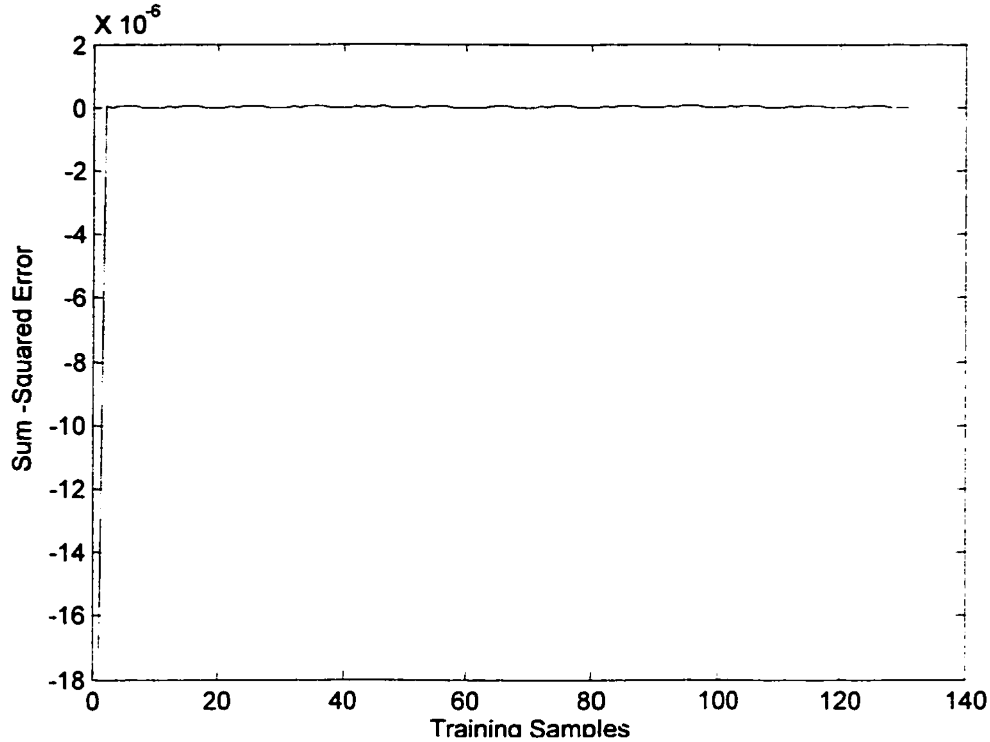


Figure 4.11: Learning process output error for the BP neural network 6-4-1.

4.4.3 Testing (Generalization) of the Network

A test set was created to analyze the performance of the proposed network. A total of 405 fault cases for each category of fault were utilized in the test set. Appendix B includes the variables considered to form the test set.

The selected network from the previous section was able to recognize and classify correctly both the normal condition as well as the fault conditions (i.e. 100% recognition rate) after 4 ms of the occurrence of a fault. Table 4.2 presents the results obtained by this network.

Table 4.2: Correct recognition and processing time required for the proposed BP fault detection neural network. Legend: Ph-Gr (phase to ground), Ph-Ph (phase to phase), Ph-Ph-Gr (phase to phase to ground) and 3-Ph (3-phase).

Percentage (%) Correct Classification and Recognition Time				
Recognition Time	**Ph-Gr faults**	**Ph-Ph faults**	**Ph-Ph-Gr faults**	**3-Ph faults**
2 ms	70.82%	76.88%	82.14%	82.45%
3 ms	95.20%	98%	97.46%	96.57%
4-5 ms	100%	100%	100%	100%

Concerning the time required by the neural network to classify the presence of a fault or a normal operating condition, Table 4.2 illustrates that the network yields 100% correct recognition rate after 4-5 ms. Figure 4.12 depicts the results obtained in Table 4.2.

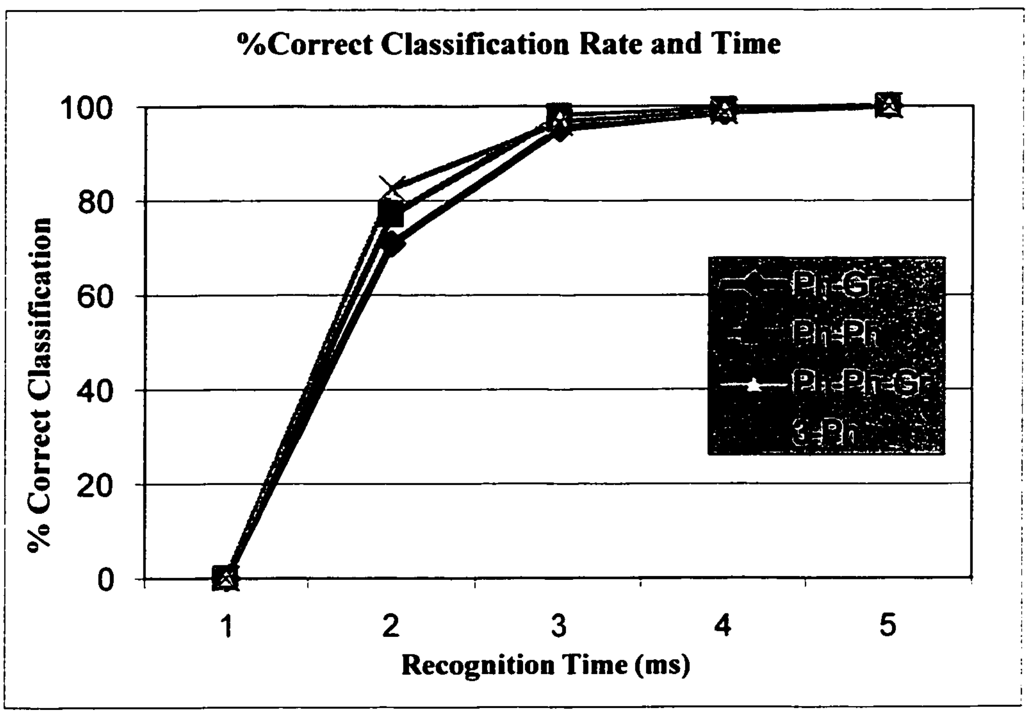


Figure 4.12: The performance of the proposed neural network detection structure.

4.5 Fault Identification (Classification)

When a fault situation is detected, the next step is to identify the type of the detected fault. This section presents the results for the design and implementation of a neural network that successfully performs the task of classification of different types of faults detected.

Neural network-based classifiers for transmission line system faults have been extensively proposed by a number of investigators [29, 31, 33, 45]. However, almost all of the studies reported in the literature have so far employed the multilayer perceptron (MLP) neural network structure with back-propagation (BP) supervised learning algorithm. Although (BP) can provide a very compact distributed representation of complex data sets, it has the following disadvantages:

- Slow learning
- Need for a large training sets
- Getting easily trapped in a local minima and poor robustness
- Difficulty in selecting the optimal size of the network.

In the following sections three neural network paradigms are considered. The first is the back-propagation (BP) algorithm, which is used here as a benchmark. The second architecture is the radial basis function (RBF) network that is designed, implemented and tested for the identification problem. The third architecture is the Support Vector

Machines (SVM) paradigm. The above three approaches are all applied to perform the fault identification task for the considered transmission line system. A comparison between the three approaches is also investigated in terms of the attributes such as the size of the neural networks, the learning speed, and the classification performance and accuracy.

4.5.1 Back-Propagation Network

The same process that was used in Section 4.4 for the design and development of the detection neural network is also followed in this section in order to choose the most suitable BP network as a fault classifier.

The network to be designed here has to have six inputs (the three phase voltages and currents) and four outputs associated with the four fault categories. The outputs contain variables whose values are given as either 0 or 1 corresponding to the three phases and the ground (that is, A, B, C and G) and can be generalized to represent all the practical fault categories permutation involving combinations of phases.

The proposed neural networks here should classify the specific phases involved in the fault scenario. It should be able to distinguish among ten (10) different categories of faults as illustrated in Table 4.3.

Table 4.3: The BP classification network truth table.

Fault Situation	Network Outputs			
	A	B	C	G
A - G	1	0	0	1
B - G	0	1	0	1
C - G	0	0	1	1
A - B	1	1	0	0
B - C	0	1	1	0
C - A	1	0	1	0
A - B - G	1	1	0	1
B - C - G	0	1	1	1
C - A - G	1	0	1	1
A - B - C	1	1	1	0

A large number of BP networks with different structures were studied and analyzed in order to obtain the simplest structure with the fastest training time. The training results for some of the selected networks, namely structures 6-5-5-4, 6 -2-4, 6-5-4 and 6-7-4, are shown in Figures 4.13 to 4.20, respectively.

It should be pointed out that the above specific selected networks correspond to only representative of a large number of networks that were extensively studied.

After an exhaustive search for the most suitable network size, the one with only one hidden layer and eight hidden neurons was chosen to carry out the classification task. The proposed network as before has six inputs (the three phase voltages and currents) and four outputs. This network is illustrated in Figure 4.21.

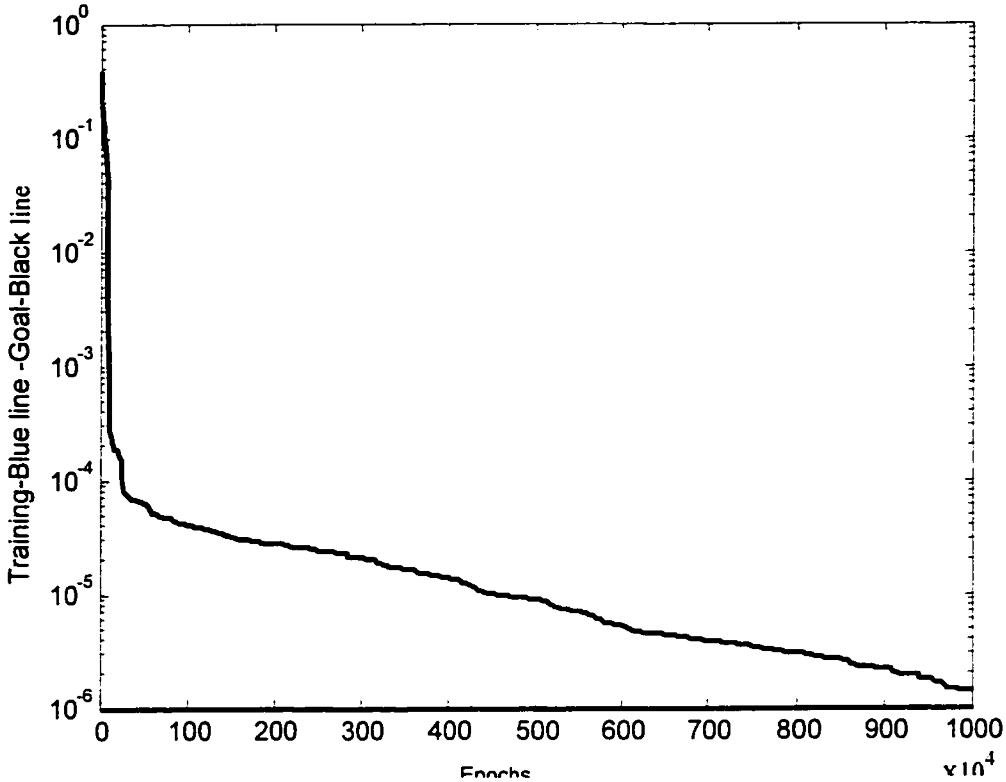


Figure 4.13: Learning process for the BP neural network 6-5-5-4.

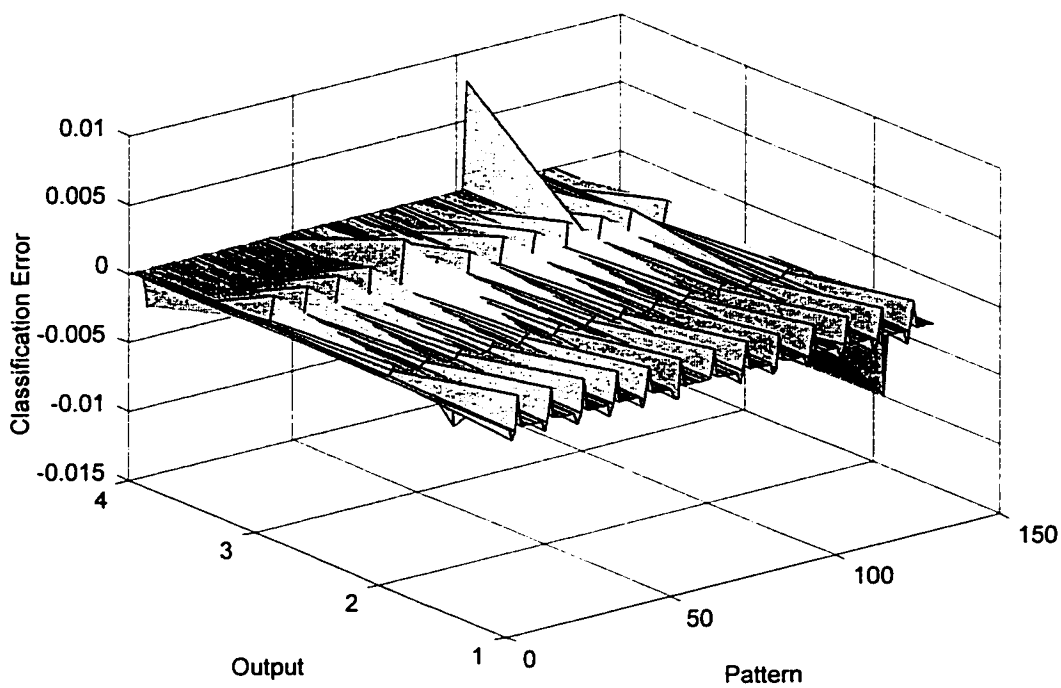


Figure 4.14: Learning process output error for the BP neural network 6-5-5-4.

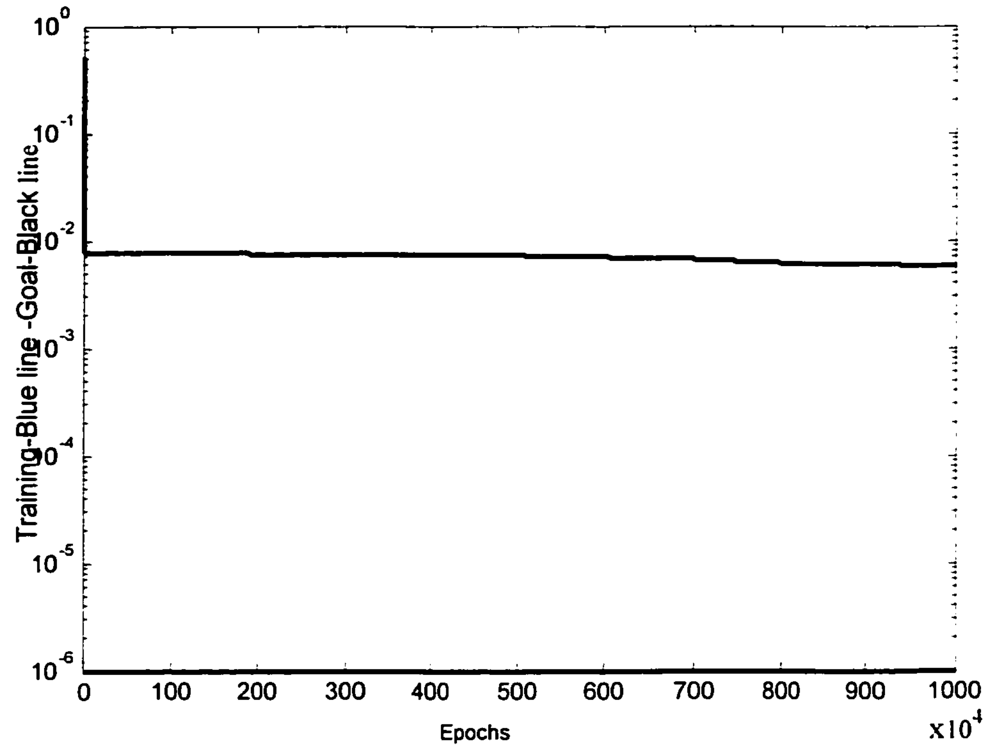


Figure 4.15: Learning process for the BP neural network 6-2-4.

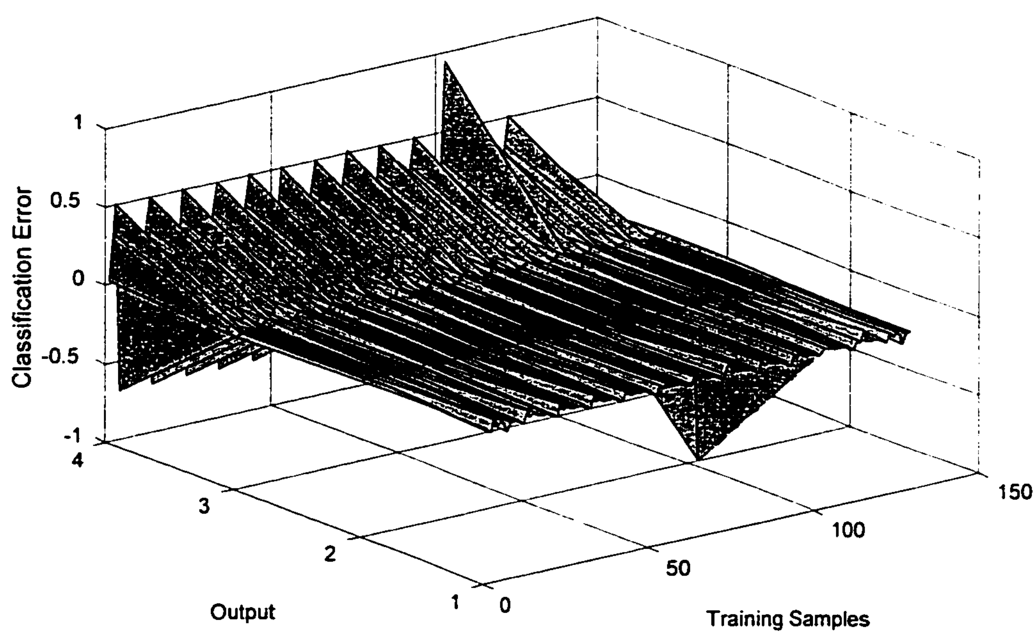


Figure 4.16: Learning process output error for the BP neural network 6-2-4.

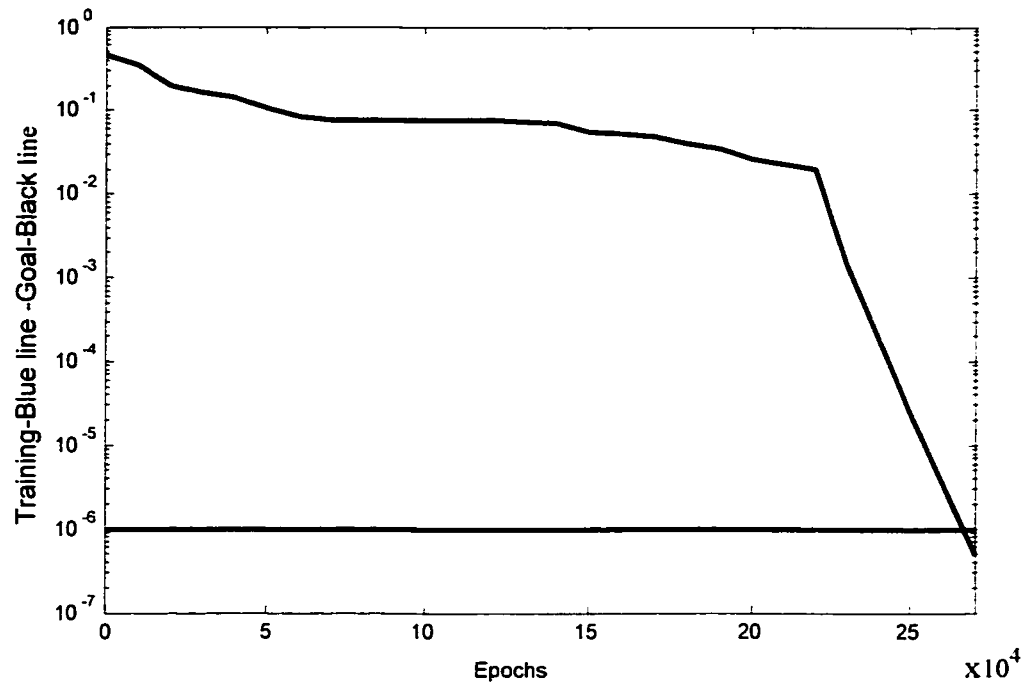
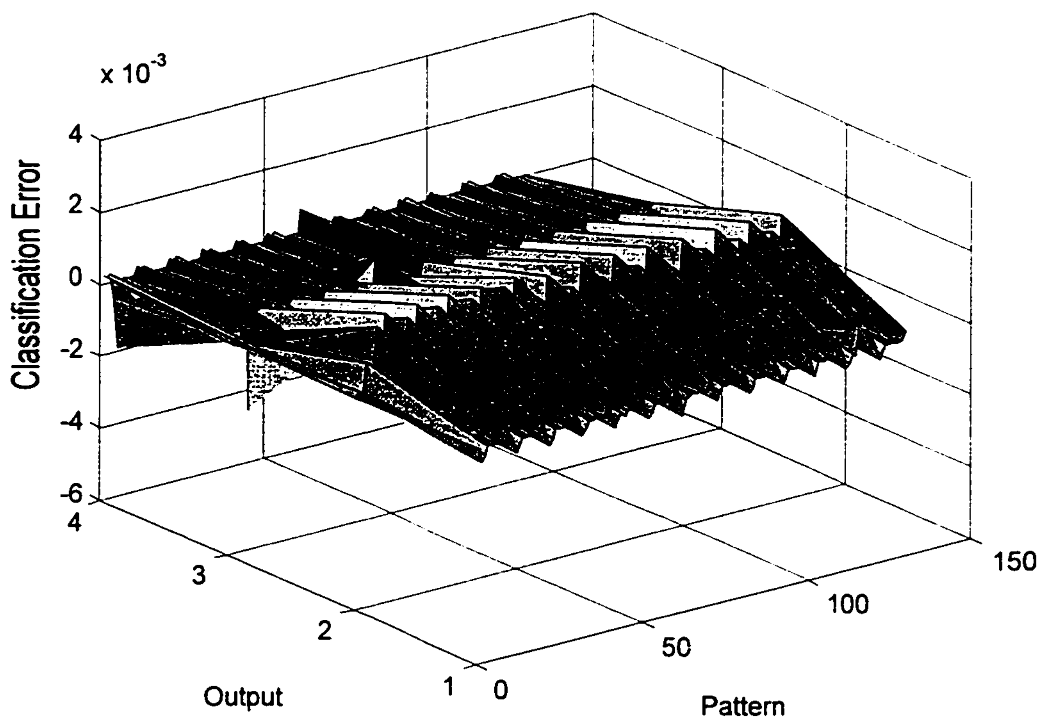


Figure 4.17: Learning process for the BP neural network 6-5-4.



F

igure 4.18: Learning process output error for the BP neural network 6-5-4.

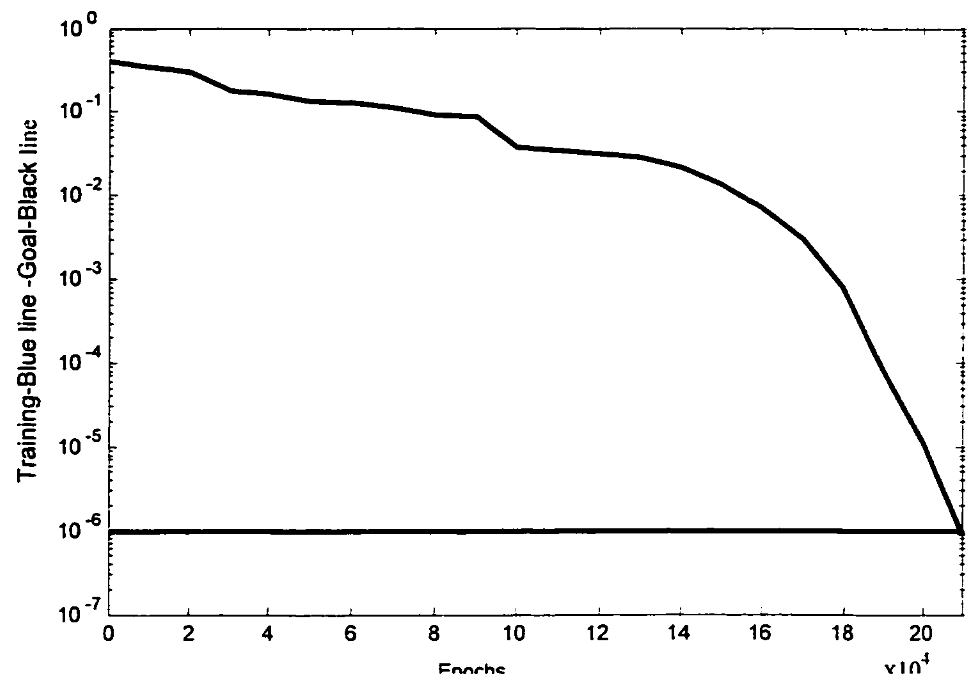


Figure 4.19: Learning process for the BP neural network 6-7-4.

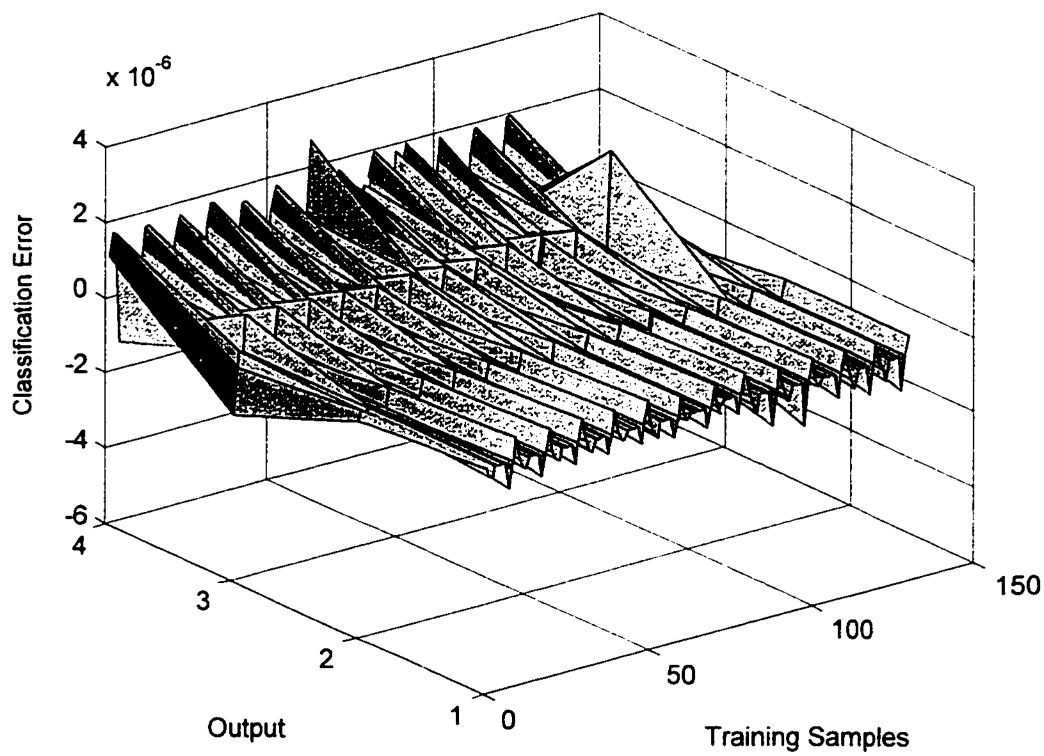


Figure 4.20: Learning process output error for the BP neural network 6-7-4.

The performance and the errors achieved by the selected network during the training process are shown in Figures 4.22 and 4.23.

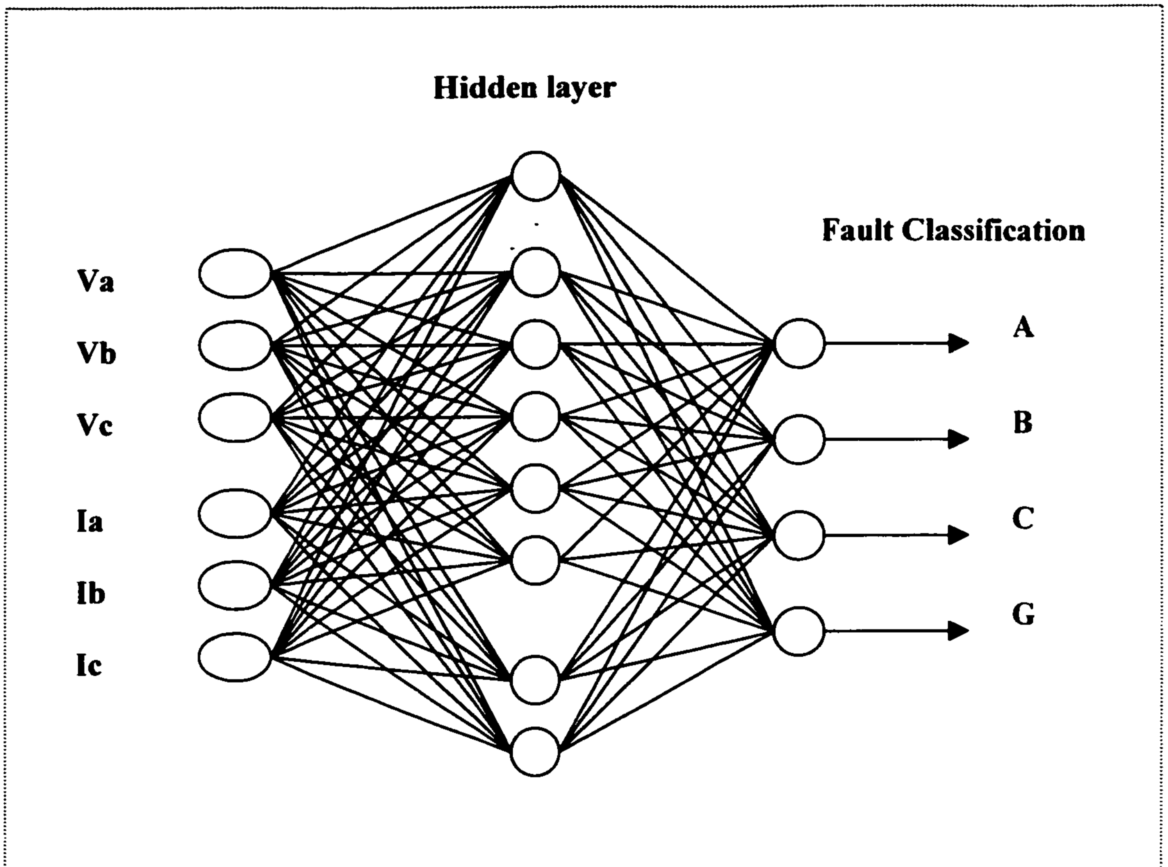


Figure 4.21: BP Neural network chosen for fault classification.

- **Generalization Results of the BP Classification Network**

The BP neural network for fault classification was tested using the same test set which was used to test the detection network by taking into account the different fault conditions given in Appendix B. The results are illustrated in Table 4.4.

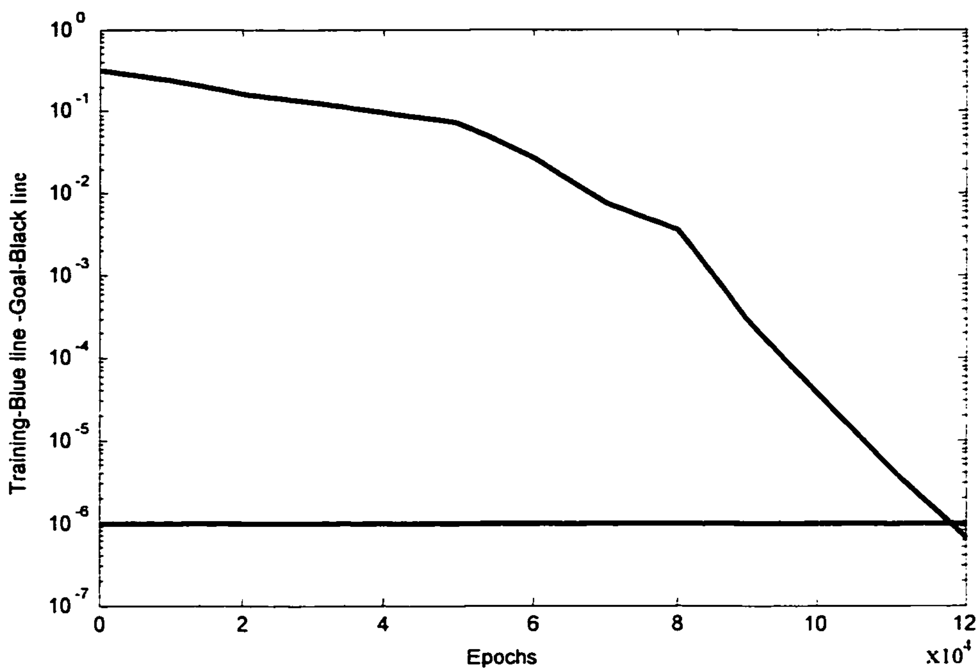


Figure 4.22: Learning process for the BP neural network 6-8-4.

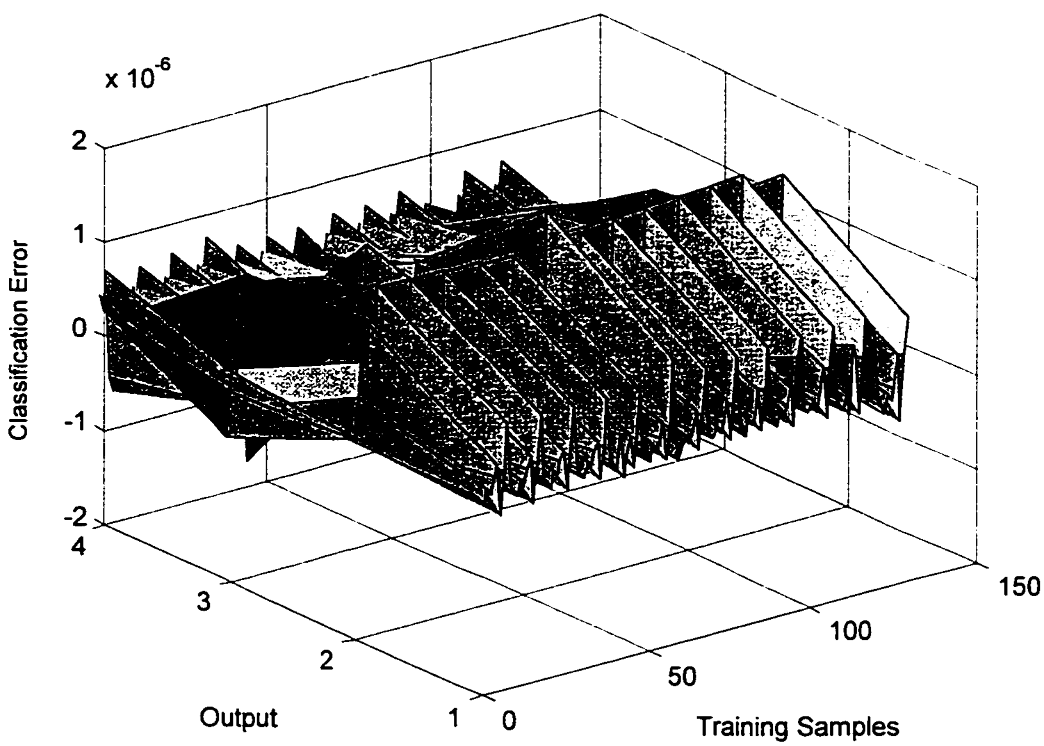


Figure 4.23: Learning process output error for the BP neural network 6-8-4.

It can be seen from Table 4.4 that the BP classification network was able to discriminate correctly all the faulty phases involved on an average of 93.15 % within 4 to 18 ms after the fault has occurred. Figure 4.24 depicts these results.

Table 4.4: Classifications and recognition time for the BP identification neural network.

Percentage (%) Correct Classification and Recognition Time					
Recognition Time	Ph-Gr faults	Ph-Ph faults	Ph-Ph-Gr faults	3-Ph faults	Average
4 ms	20.41%	0%	0%	0%	5.10%
5 ms	57.12%	4.51%	6.25%	0%	16.97%
6 ms	87.23%	18.62%	16.33%	0%	30.55%
7 ms	92.83%	36.14%	42.45%	29.12%	50.14%
8 ms	93.41%	51.22%	65.72%	57.84%	67.05%
9 ms	93.98%	77.08%	84.19%	81.29%	84.17%
10-18 ms	94.11%	93.24%	93.07%	92.17%	93.15%
19 ms	94.11%	93.24%	93.07%	92.17%	93.15%

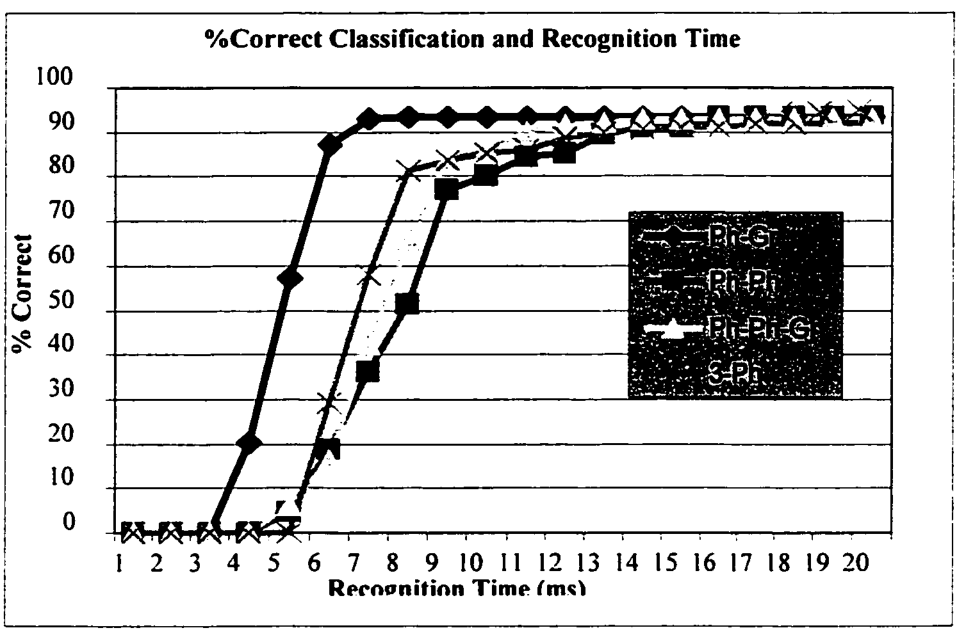


Figure 4.24: The classification BP neural network correct answers and processing time.

4.5.2 RBF Neural Network Fault Classifier

As discussed in Chapter 3, RBF neural networks basically contain three entirely different layers. The input layer, the output layer and the hidden layer, which is composed of the kernel nodes whose functions are different from those of a BP network.

Kernel nodes based on radial basis functions compute symmetric functions that are maximum when the input is near the center of the node. The output nodes contain simple linear activation functions.

- **Learning Process and Network Size**

As discussed in Chapter 3, the training process of the RBF network is carried out by an algorithm that determines the unit centers through adaptive clustering. Further, the widths of the neurons are determined by a nearest-neighbor method. The algorithm employs the Delta-rule with sigmoidal function to calculate the weights between the hidden and output layers. The learning coefficient is set as 0.20 and the learning ratio as 0.50.

The number of the inputs is chosen as six (6) which is the same as that of the BP network classifier. The number of outputs, as in the BP case, is also selected as 4. In order to find the optimal structure of the RBF network, different models were studied. In each model different number of kernel nodes was applied. The number of kernel nodes investigated varied from 4 to 44.

Figures 4.25 to 4.28 illustrate the learning process and convergence rates for a selected number of the RBF networks studied. These figures depict the convergence rate corresponding to different number of neurons selected in the hidden layer (44, 40, 36 and 32, respectively). An RBF neural network with 32 kernel nodes was chosen for this application as it gave the best overall performance with respect to the convergence rate and the RMS classification error.

It should be pointed out that the networks of fewer than 32 neurons in the hidden layer gave larger RMS error. In fact, in these cases the error resulted was higher than the desired value which was 1×10^{-6}.

- **Generalization Results of the RBF Classification Network**

The RBF network that is tested for generalization capabilities of this type of network is the 6-32-4 structure subject to different fault conditions provided in Appendix B. The results obtained are illustrated in Table 4.5.

Table 4.5 shows that the RBF fault classification network was able to categorize the faulty phases and the type of the faults correctly on average of about 95.36% within 3 to 15 ms after the fault has occurred. Figure 4.29 provides a graphical summary of these results.

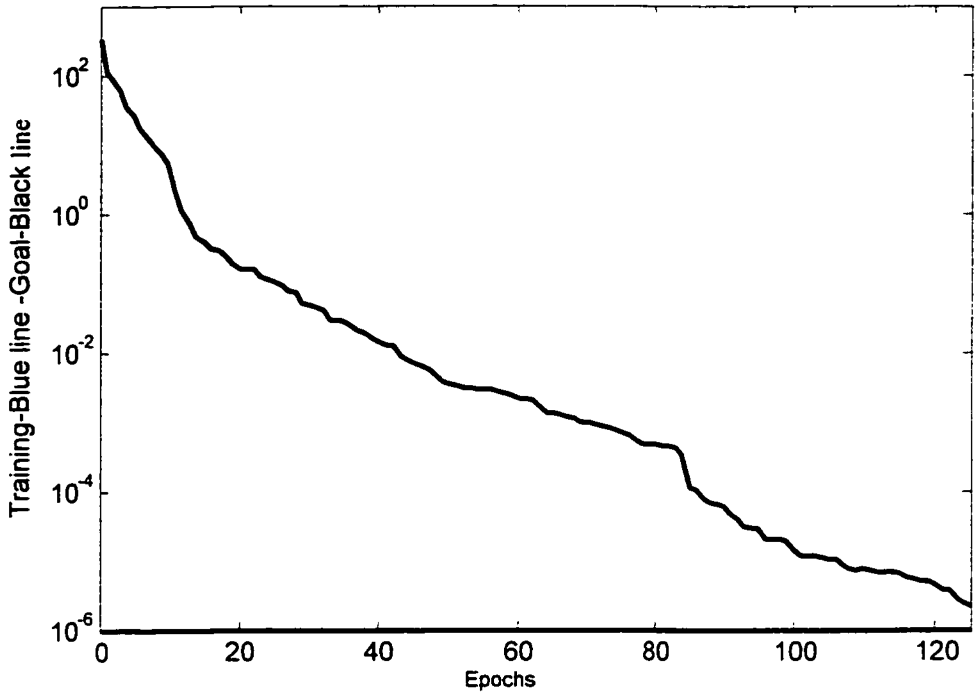


Figure 4.25: Learning process for the RBF neural network 6-44-4.

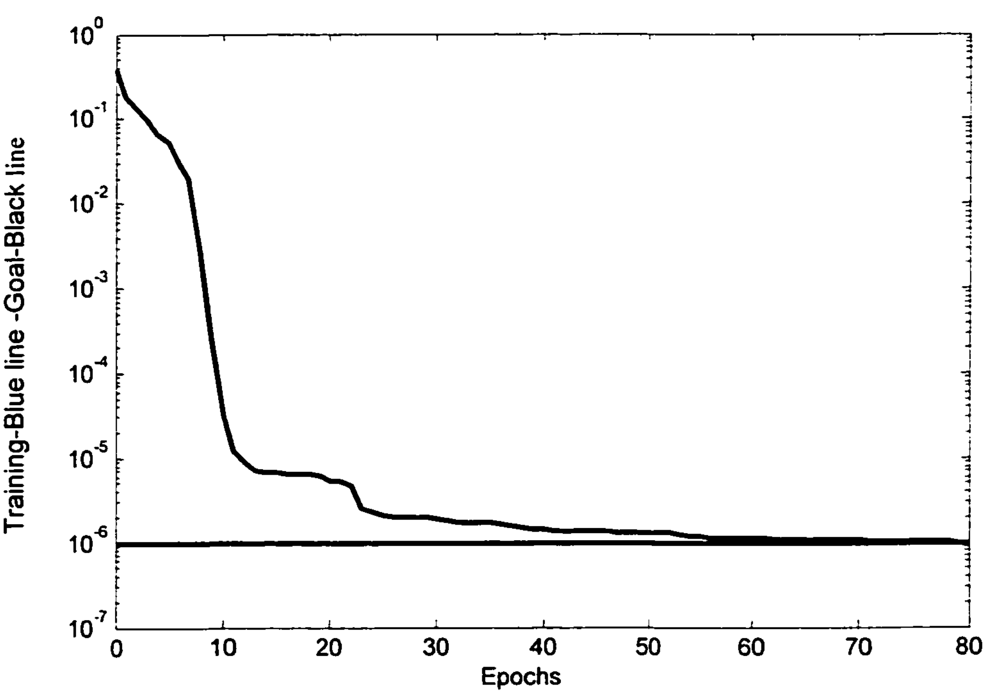


Figure 4.26: Learning process for the RBF neural network 6-40-4.

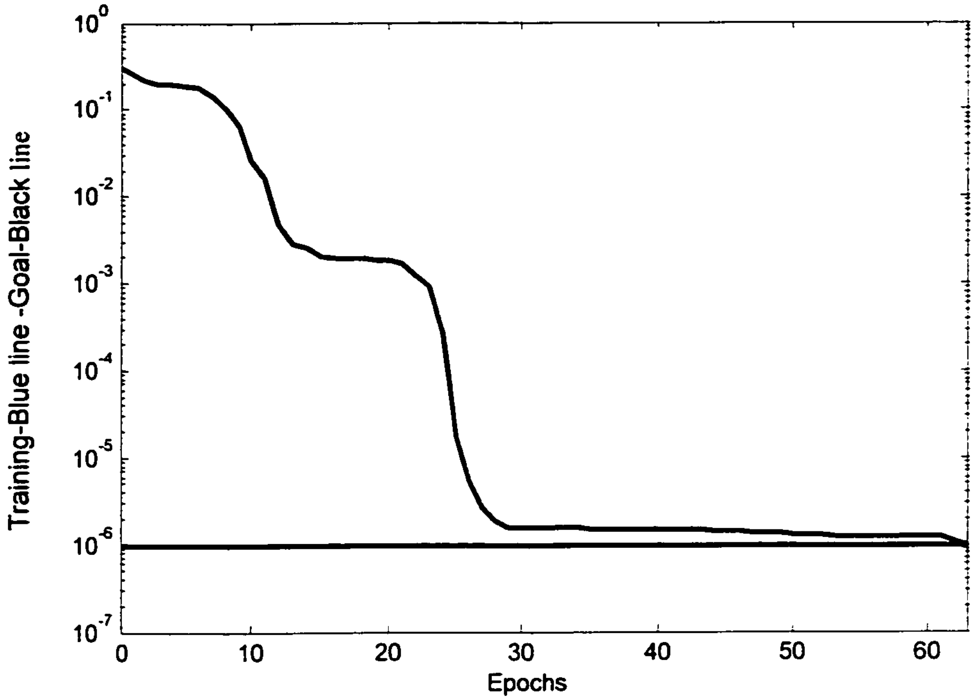


Figure 4.27: Learning process for the RBF neural network 6-36-4.

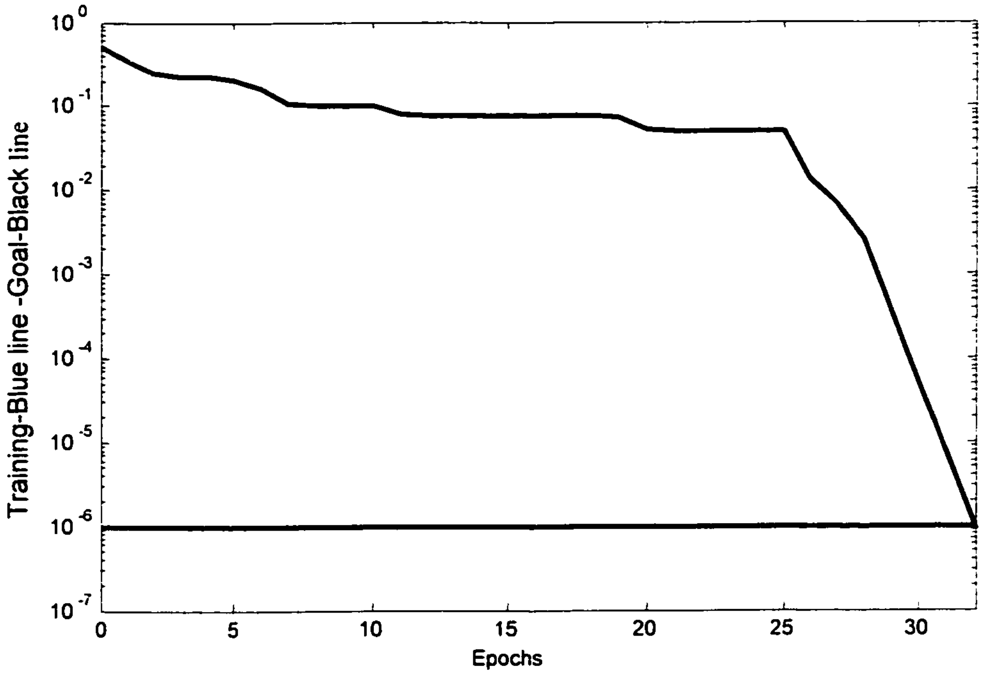


Figure 4.28: Learning process for the RBF neural network 6-32-4.

Table 4.5: Classification and processing time for the RBF identification neural network.

Percentage (%) Correct Classification and Recognition Time				
Recognition Time	**Ph-Gr faults**	**Ph-Ph faults**	**Ph-Ph-Gr faults**	**3-Ph faults**
3 ms	33.74%	6.09%	5.64%	0%
4 ms	64.95%	15.36%	16.12%	5.01%
5 ms	90.34%	22.35%	20.03%	11.25%
6 ms	93.56%	54.25%	49.72%	39.03%
7 ms	93.87%	66.38%	65.13%	66.75%
8 ms	94.75%	84.64%	85.08%	86.31%
9 ms	96.21%	94.97%	93.88%	93.24%
10-15ms	96.36%	95.84%	95.34%	93.89%
16 ms	96.36%	95.84%	95.34%	93.89%

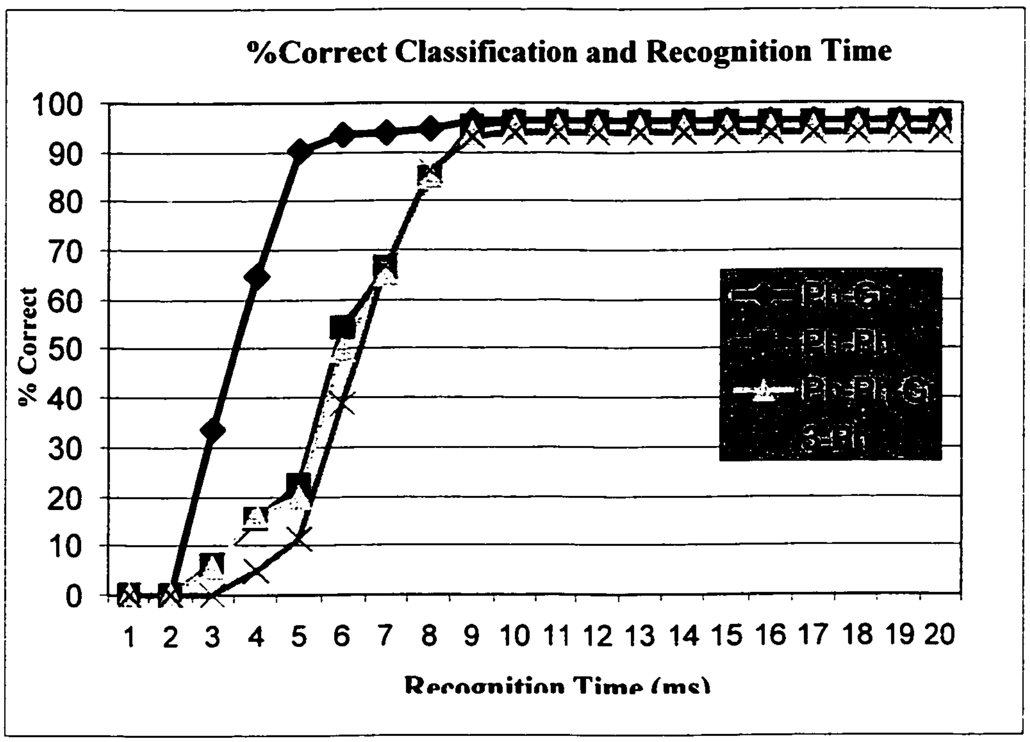


Figure 4.29: The fault classification of the RBF neural network and its recognition time.

4.5.3 Fault Classification using Support Vector Machines

In order to implement the SVM for the fault identification task, the Matlab SVM toolbox (Version 3.00) is used. The SVM toolbox provides routines for support vector classification and support vector regression problems [74]. The toolbox works with the Matlab optimization toolbox.

For training, 130 different fault cases were considered for the classification task taking into account different fault scenarios (fault locations and resistances as discussed in section 4.3.1). As a pre-processing stage, all the data samples are normalized in order to make their " energy " equal to unity followed by scaling all data samples to the interval [-1 1].

In the simulations, three functions, namely Linear function, Polynomial kernel function and radial basis kernel function, were used in order to compare the effects of different kernel functions on the classification results. The explicit mathematical description of these functions is as follows:

- The linear function: $K(x,x_i) = k(x.x_i)$.
- The polynomial function: $K(x,x_i) = [(x.x_i)+1]^q$, where q is the order of the polynomial function.
- The radial basis function: $K(x,x_i) = \exp(-\frac{|x-x_i|^2}{\sigma^2})$, where σ is the basis function width and x_i is its mean (center).

- **Classification Generalization Results**

The SVM classifier was tested first using the polynomial kernel function with *C* set equal to 1 and the order *q* was changed from 1 to 20,000. These results are shown in Table 4.6. It is clear that the results become better as *q* is increased, however the complexity of the algorithm is also increased and the training time required for convergence is significantly longer (as represented by the number of support vectors constructed).

Table 4.6: Fault identification results using polynomial kernel function ($C=1$).

Parameter *q*	The Average Number of SV	Percentage (%) Classification Accuracy			
		Ph-Gr faults	Ph-Ph faults	Ph-Ph-Gr faults	3-Ph faults
1	347	67.33%	66.25%	67.08%	65.13%
10	221	84.67%	80.65%	79.85%	81.42%
100	139	90.34%	88.28%	89.82%	88.51%
1000	103	96.12%	95.45%	95.64%	94.87%
10000	102	96.23%	95.56%	95.66%	94.91%
20000	102	96.23%	95.56%	95.66%	94.91%

Further results obtained from different support vector machines constructed by choosing different functions are given in Tables 4.7, 4.8 and 4.9. In these three cases, only *C* was changed. It should be noted that as a higher error penalty (*C* being larger) is imposed, the number of support vectors is correspondingly reduced and the classification performance is also improved.

Table 4.7: Comparative results for using linear kernel function (*k=1*).

Parameter *C*	**The Average Number of SV**	**Percentage (%) Classification Accuracy**			
		Ph-Gr faults	**Ph-Ph faults**	**Ph-Ph-Gr faults**	**3-Ph faults**
1	347	65.12%	62.51%	66.02%	60.30%
10	281	78.33%	75.25%	77.26%	69.54%
100	193	88.45%	89.32%	87.63%	88.12%
1000	143	93.32%	92.25%	92.97%	90.26%
10000	109	96.98%	95.74%	95.13%	94.55%
20000	109	96.98%	95.74%	95.13%	94.55%

Table 4.8: Comparative results for using polynomial kernel function (*q=2*).

Parameter *C*	**The Average Number of SV**	**Percentage (%) Classification Accuracy**			
		Ph-Gr faults	**Ph-Ph faults**	**Ph-Ph-Gr faults**	**3-Ph faults**
1	221	83.24%	81.22%	80.32%	78.94%
10	163	88.92%	86.32%	88.12%	86.23%
100	118	94.71%	94.00%	94.21%	91.65%
1000	101	96.94%	95.98%	95.88%	95.64%
10000	101	96.94%	95.98%	95.88%	95.64%

Table 4.9: Comparative results for using radial basis kernel function ($\sigma=1$).

Parameter C	**The Average Number of SV**	**Percentage (%) Classification Accuracy**			
		Ph-Gr faults	**Ph-Ph faults**	**Ph-Ph-Gr faults**	**3-Ph faults**
1	313	79.13%	78.12%	77.24%	77.83%
10	215	82.13%	81.52%	83.15%	83.12%
100	143	91.61%	90.84%	90.42%	89.55%
1000	104	96.11%	95.91%	95.74%	94.76%
10000	104	96.11%	95.91%	95.74%	94.76%

The above results show clearly that the polynomial kernel classifier yields the best performance with an average rate of 96.1% correct classification.

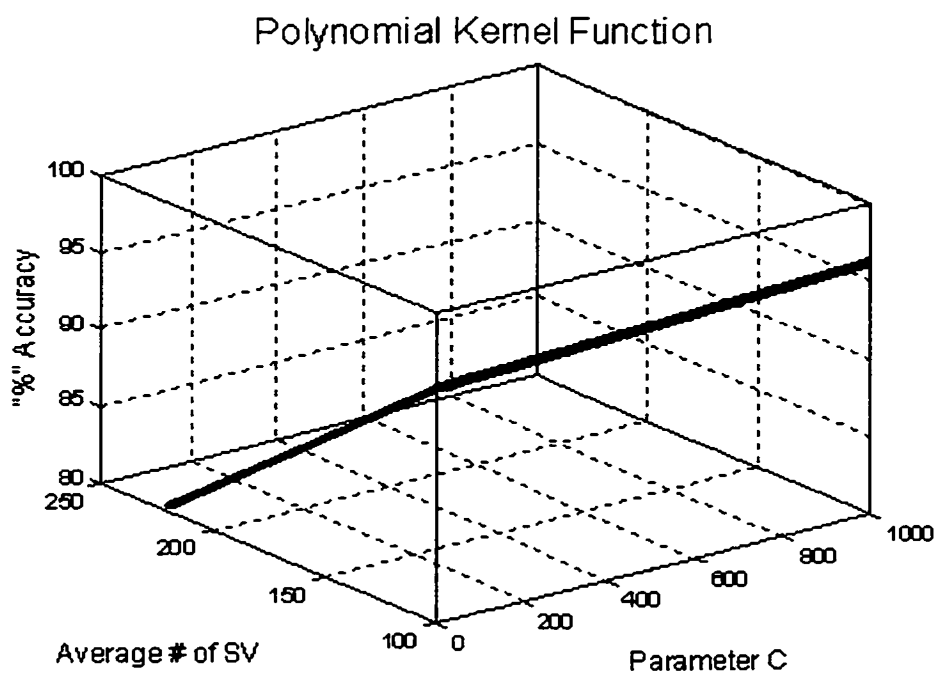


Figure 4.30: SVM with polynomial kernel function to classify faults.

4.5.4 Comparison of the Performance of the Proposed Three Classifiers

A classifier can be evaluated by considering many factors and metrics. Among these are misclassification rate, training time, computation time, adaptation and robustness performance, and real-time implementation requirements. In this section, emphasis is placed on a comparison of the performance of the three considered neural network classifiers in terms of the size of the neural network, the learning process and the classification accuracy.

- **Neural Network Size**

As discussed in the previous sections, the number of the inputs to the three network classifiers is 6 and the number of outputs is 4. In the BP network only one hidden layer with 8 nodes was used and found to be quite adequate for this particular application. The RBF neural network with 32 Kernel nodes performed very well. For the SVM network, the parameter C characterizes the complexity of the network as reflected in the number of support vectors that is about 101. Basing on the above results, it is concluded that the BP network has the least complexity in its structure as compared to the others.

- **Learning Process**

The learning algorithm used with the BP network was the back-propagation rule based on the Levenber-Marquadt optimization technique. The Levenber-Marquadt method is a single-shot method which attempts to find the local fit-statistic minimum nearest to the starting point. Its principal advantage is that it uses information about the first and second order derivatives of the fit-statistic as a function of the thawed parameter values to guess the location of the fit-statistic minimum. This method works quite well (and fast) if the statistic surface is well-behaved. Its principal disadvantage is that there is no guarantee it will find the global fit-statistic minimum [48]. The Hyperbolic tangent transfer function was used due to the fact that it has better convergence performance as compared to the sigmoid function [45]. During the training process the learning parameters were adjusted to obtain the best performance. For instance, the learning factor, which controls the rate of convergence and speed, was chosen to be 0.5 at the beginning of the training process and was gradually reduced to 0.01. The momentum factor, which is normally added to stabilize the training and avoid the local minima, was chosen at 0.45 experimentally through a number of trails and errors.

Three steps were conducted to train the RBF network. First, the adaptive clustering algorithm determined the unit centers. The widths were determined using a nearest-neighbor method. Finally, the other network parameters were chosen by trail and error to yield the best performance. The learning coefficient was set to 0.2.

The statistical learning theory is the basis for the SVM to determine the optimal linear separating hyperplanes and to minimize the classification error. Three functions were investigated. The classifier was tested with the different functions and with different values of the parameters.

- **Fault Type Classification Rates**

Table 4.10 summarizes the correct classification rates for the generalization cases corresponding to the four types of fault. It can be observed that the error rates vary with the type of the fault. Clearly, the SVM offers the best performance among the three networks. It can also be noted that the correct classification rate for the three phase faults is lower than that of the other three fault types.

Table 4.10: Summary of the results for the three networks for fault classification.

Fault Type	**% Correct Classification (Average)**		
	BP	**RBF**	**SVM**
Phase to Ground	94.11%	96.36%	96.94%
Phase to Phase	93.24%	95.84%	95.98%
Two Phase to Ground	93.07%	95.34%	95.88%
Three Phase	92.17%	93.89%	95.64%
Average	93.15%	95.36%	96.11%

4.6 Fault Isolation/ Location (Identifying the Faulty Zone)

Following the detection and classification of the faults, the task of fault isolation/ location is performed. In other words, one has to determine the physical location of the fault. As mentioned in Chapters 1 and 2, the reach for protection zones 1, 2 and 3 are set to 95%, 130% and 150% of the protected transmission line to identify the zones 1, 2 and 3, respectively.

The BP, RBF and SVM architectures are designed and applied to handle the fault location task. The networks are provided with the same number of inputs as in the previous task of fault classification (that is, the three phase voltages and currents) and are now assigned with three outputs corresponding to the three fault zones/ locations (Z1, Z2 and Z3). Different structures for the above three network paradigms were studied as in the previous section on fault classification.

4.6.1 Back-Propagation Neural Network for Fault Location

It is found experimentally through trail and error that a BP network with two hidden layers provides the best training performance. The first hidden layer has 48 neurons and the second hidden layer has 44 neurons. The network is expected to identify the location of the fault by classifying the identified fault into one of the three fault zones, namely Z1, Z2 and Z3. The desired truth table for the network training is shown in Table 4.11.

Table 4.11: The isolation network desired response.

Fault Location	**Network Output**		
	Z1	**Z2**	**Z3**
Zone 1	1	0	0
Zone 2	0	1	0
Zone 3	0	0	1

- **The BP Isolation Network Results**

The BP network was tested using the same test set that was used previously corresponding to different fault conditions as shown in Appendix B. As mentioned before, a total of 405 different fault cases for each type of fault were utilized in the test set. The results are illustrated in Table 4.12.

Table 4.12 shows that the fault isolation BP network is capable of determining the faulty zones correctly on an average of about 92.44 %. It can also be noted that the recognition time required by the network is 18 ms after the occurrence of the fault in order to reach the maximum possible correct recognition rate. These results are also shown in Figure 4.31.

Table 4.12: Correct isolation and recognition time for the fault isolation BP neural network.

Percentage (%) Correct Isolation and Recognition Time												
Recognition Time	**Ph-Gr faults**			**Ph-Ph faults**			**Ph-Ph-Gr faults**			**3-Ph faults**		
	Zone 1	**Zone 2**	**Zone 3**	**Zone 1**	**Zone 2**	**Zone 3**	**Zone 1**	**Zone 2**	**Zone 3**	**Zone 1**	**Zone 2**	**Zone 3**
8 ms	0.8%	0.0%	0.0%	0.0%	0.0%	0.0%	0.6%	0.0%	0.0%	0.0%	0.0%	0.0%
9 ms	6.1%	5.8%	3.7%	4.9%	4.6%	4.1%	8.8%	5..32%	4.6%	0.0%	0.0%	0.0%
10ms	22.3%	18.2%	16.9%	18.2%	18.0%	16.4%	17.3%	16.8%	15.2%	9.0%	5.4%	4.0%
11 ms	39.3%	39.0%	38.2%	32.3%	31.2%	30.0%	44.8%	44.3%	44.1%	20.0%	18.9%	15.7%
12 ms	62.9%	62.5%	61.4%	53.1%	52.2%	51.4%	56.0%	55.3%	54.6%	32.4%	32.1%	31.1%
13 ms	75.0%	74.8%	73.3%	75.0%	74.2%	72.9%	71.9%	71.3%	70.4%	66.1%	65.8%	65.2%
14 ms	86.9%	86.1%	85.5%	87.9%	87.0%	86.5%	86.0%	85.7%	85.2%	78.9%	78.7%	77.0%
15 ms	93.1%	93.0%	92.5%	90.4%	90.1%	89.8%	90.8%	90.0%	89.9%	88.9%	88.2%	87.9%
16-18 ms	93.8%	93.5%	93.3%	92.8%	92.0%	91.2%	92.0%	91.4%	91.9%	92.7%	92.4%	92.2%
19 ms	93.8%	93.5%	93.3%	92.8%	92.0%	91.2%	92.0%	91.4%	91.9%	92.7%	92.4%	92.2%

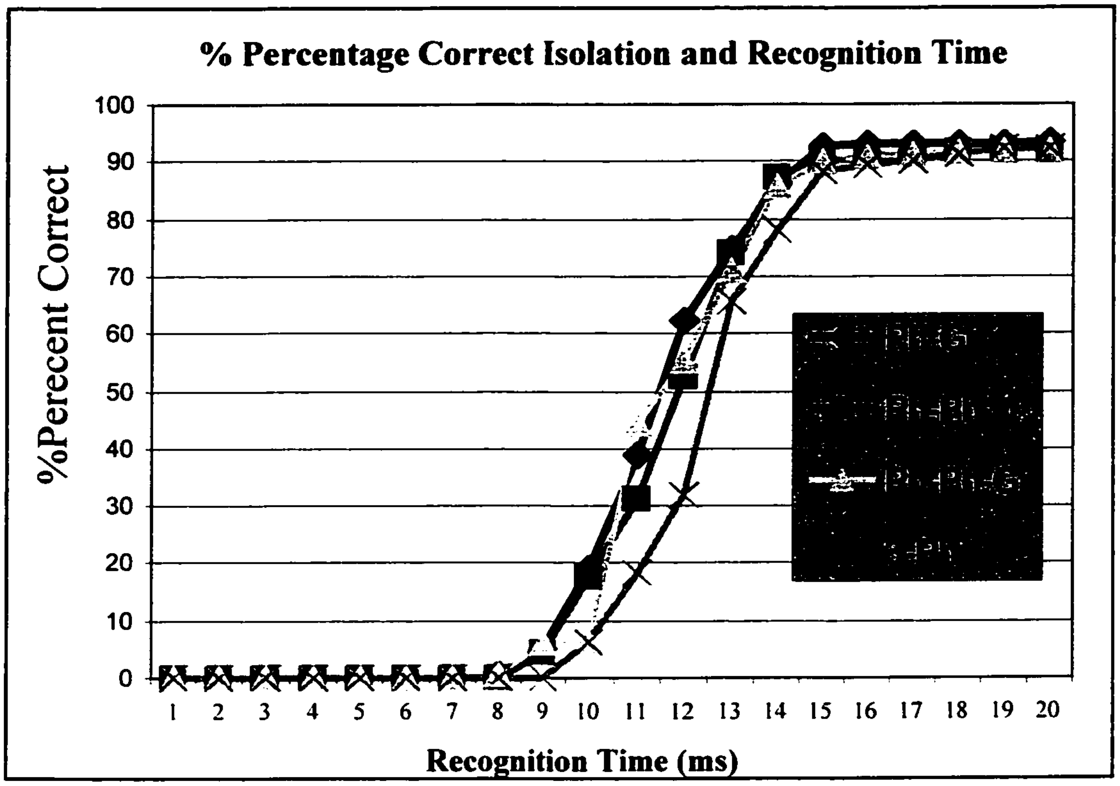


Figure 4.31: The fault location BP NN correct isolation and processing time.

4.6.2 RBF Network for Fault Location

After extensive trails and errors a 6-44-3 RBF network structure was selected. The network was tested on different fault conditions as provided in Appendix B. These results are given in Table 4.13 and Figure 4.32.

The proposed network is capable of isolating the faulty zones correctly on an average of about 92.46% within 9-17 ms after the occurrence of the fault.

Table 4.13: Correct isolation and recognition time for the RBF neural network.

Percentage (%) Correct Isolation and Recognition Time												
Recognition Time	Ph-Gr faults			Ph-Ph faults			Ph-Ph-Gr faults			3-Ph faults		
	Zone 1	Zone 2	Zone 3	Zone 1	Zone 2	Zone 3	Zone 1	Zone 2	Zone 3	Zone 1	Zone 2	Zone 3
9 ms	2.1%	1.0%	0.3%	2.2%	1.0%	0.5%	1.9%	1.1%	0.7%	0.9%	0.5%	0.3%
10 ms	23.2%	21.1%	19.4%	20.6%	19.1%	18.8%	23.7%	22.4%	21.9%	15.1%	11.0%	9.2%
11ms	44.1%	40.2%	39.8%	36.1%	35.6%	32.3%	45.1%	44.7%	39.9%	20.2%	20.1%	19.0%
12 ms	66.0%	66.0%	61.7%	62.1%	60.6%	57.6%	60.3%	59.6%	58.8%	33.8%	33.7%	32.1%
13 ms	86.8%	86.0%	79.9%	72.4%	71.2%	70.2%	69.1%	68.1%	67.7%	52.1%	50.7%	47.9%
14 ms	91.5%	90.2%	89.7%	84.0%	82.8%	80.8%	85.5%	85.1%	84.0%	66.1%	65.1%	65.0%
15 ms	94.1%	92.9%	92.8%	89.1%	88.6%	87.1%	89.7%	89.0%	88.2%	73.9%	73.7%	72.1%
16-17 ms	95.0%	94.1%	93.5%	93.1%	92.2%	91.7%	93.0%	92.1%	91.3%	91.7%	91.2%	90.5%
18 ms	95.0%	94.1%	93.5%	93.1%	92.2%	91.7%	93.0%	92.1%	91.3%	91.7%	91.2%	90.5%

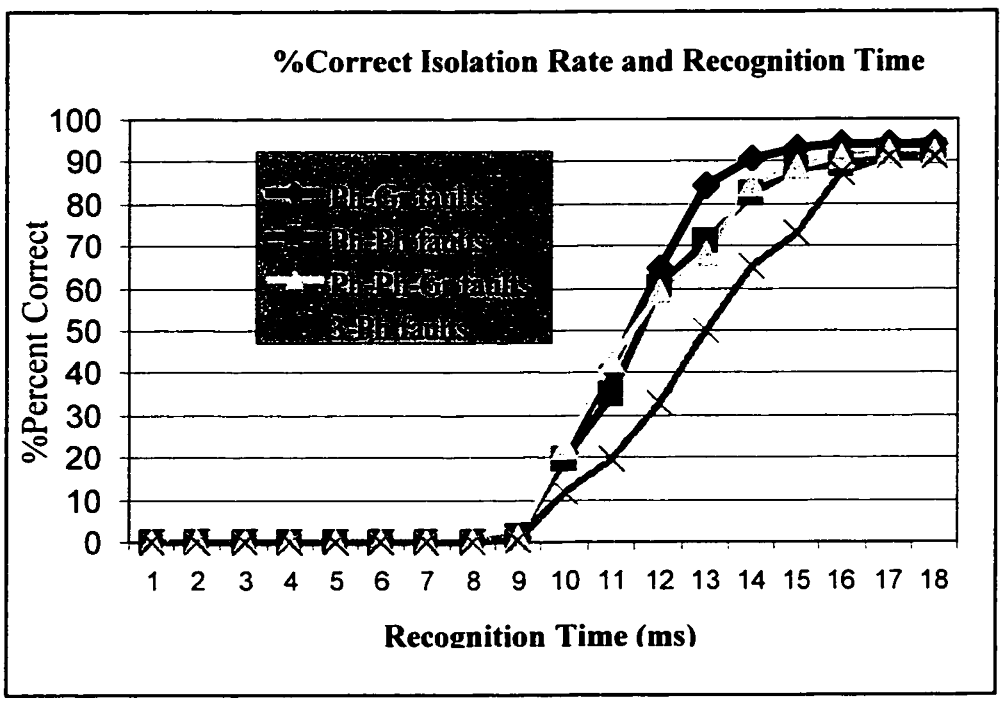


Figure 4.32: The fault isolation RBF neural network correct rates and recognition time.

4.6.3 Fault Location using Support Vector Machines

Using polynomial kernel function, many networks were tested and finally the SVM's structure represented by Table 4.14 is chosen.

In this network, only the parameter C was changed. As expected from previous results the higher the error penalty term (C is larger), the fewer the number of support vectors and correspondingly the more accurate the results.

Table 4.14: Results of using polynomial kernel function SVM for fault location.

C	The Average Number of SV	Percentage (%) Correct Isolation											
		Ph-Gr faults			Ph-Ph faults			Ph-Ph-Gr faults			3-Ph faults		
		Zone 1	Zone 2	Zone 3	Zone 1	Zone 2	Zone 3	Zone 1	Zone 2	Zone 3	Zone 1	Zone 2	Zone 3
1	346	80.9%	80.1%	80.0%	79.7%	79.1%	78.0%	77.1%	76.9%	76.6%	79.0%	78.9%	78.3%
10	263	86.8%	86.2%	85.6%	86.1%	85.6%	85.4%	83.9%	83.5%	82.2%	85.7%	85.1%	84.3%
100	174	92.5%	92.4%	92.1%	91.9%	91.4%	90.5%	91.0%	90.6%	89.7%	91.9%	91.1%	91.0%
1000	121	95.9%	95.2%	94.7%	95.1%	94.9%	94.2%	95.0%	94.7%	94.3%	95.1%	94.6%	94.9%
10000	121	95.9%	95.2%	94.7%	95.1%	94.9%	94.2%	95.0%	94.7%	94.3%	95.1%	94.6%	94.9%

The above table shows that the polynomial kernel SVM's network identifies the faulty zones with an average of 94.86% correct classification rate.

4.6.4 Comparison between the Results of the Three Architectures used for the Fault Location

As in the classification task in section 4.5.4, each neural network architecture proposed in this section for fault isolation is now compared to the others by considering several metrics and factors. The three network architectures are compared by taking into account the of size of the network, the learning process and the classification accuracy.

- **Neural Network Size**

As in the previous cases, each of the three networks has 6 inputs and 4 outputs. In the BP network two hidden layers are employed with the first layer having 48 nodes and the second layer having 44 nodes. The RBF neural network with 44 Kernel nodes provides the best performance. For the SVM network, the parameter C characterizes the complexity of the network. Setting the parameter C to 1000 reduces the number of support vectors to 121. Consequently, in view of the above obtained results, it can be concluded that the RBF network has the smallest size when compared to the others.

- **Learning Process**

As in the classification case, the back-propagation rule was used to train the BP network for the fault isolation. The learning factor was chosen to be 0.6 at the beginning of the training and was gradually reduced to 0.05 to control the training speed and convergence rate. The momentum factor, which is normally added to stabilize the training and avoid the local minima, was set at 0.4 after a number of trails and errors.

For the RBF network, the learning coefficient was set to 0.25 while the other network parameters were adjusted to yield the best performance.

For the fault isolation SVM architecture, as in the case of the fault classification, three functions were investigated. The network was tested with the different functions and with different values of the parameters. The polynomial kernel function was chosen.

- **Correct Isolation Rates**

The results of the three approaches, as far as isolation rates are concerned, are shown in Table 4.15. The table demonstrates the average of correct isolations percentage rate obtained by each network for each type of fault.

From Table 4.15 it can be concluded that the SVM again gives the best performance among the three networks for the task of fault location.

Table 4.15: Summary of the results of the three approaches used for fault location.

Fault Type	**% Correct Isolation (Average)**		
	BP	**RBF**	**SVM**
Phase to Ground	93.54%	94.21%	95.23%
Phase to Phase	92.02%	92.32%	94.71%
Two Phase to Ground	91.78%	92.14%	94.65%
Three Phase	92.42%	91.17%	94.86%
Average	92.44%	92.46%	94.86%

4.7 Overall Discussion of the Results

The results obtained in the previous sections demonstrate that the overall performance of the proposed neural network architectures was highly satisfactory as far as speed and accuracy of the network models are considered.

It should be emphasized that a lengthy process was involved in training and generalization of these networks until satisfactory performance was obtained. This, of course, was a time consuming process. It should also be pointed out that the proposed networks estimated the expected response for the patterns tested corresponding to the changes in the operational conditions of the system as indicated in Appendix B.

For the fault detection objective, the network outputs converged to the correct class after 5 ms of the occurrence of the fault. As shown in Section 4.4, the proposed network recognized the correct scenario with 100% recognition rate.

The classification task was performed using the BP, RBF and the SVM architectures. The SVM paradigm yielded the best results as compared to the structures as shown in Section 4.5.

Finally, considering the problem of three protection zones, the results obtained demonstrate that the three proposed approaches yield a reliable identification of the location of the faults corresponding to all types of faults. As in the previous problem, the SVM approach resulted in the best candidate compared to the other networks as presented in Section 4.6.

Chapter 5

Conclusion and Further Work

5.1 Conclusions

This thesis has empirically investigated the use of three neural network architectures as an alternative method for detection, fault classification and isolation of faults in a transmission line system. The three approaches use phase voltage and phase current RMS values as inputs. This work took into consideration single phase to ground faults, double phase faults, double phase to ground faults and three phase faults.

The neural networks concerned in this thesis are: back-propagation neural network (BP), radial basis neural network (RBF) and support vector machines network (SVM). These three networks were used to simulate a comprehensive protection scheme from detection to isolation stages for a transmission line system. This work also presents a comparison study of the three neural network approaches for fault classification (fault type identification) and isolation (fault location zones).

The results obtained demonstrate that in general the performance of the proposed different neural network structures was highly satisfactory. As further illustrated, the support vector machines, and the radial basis function paradigms yield the best classification/ recognition error rates compared to the back-propagation network. Over all, the support vector machines network, which is employed in this area for the first time in the literature, is shown to be the best candidate for the classification and isolation of faults while the back-propagation yields a very high performance for the fault detection.

To simulate the power transmission system and the different proposed neural network structures used in the study, MATLAB (Version 6.1) with Power System Block Set (Version 2.1), Neural Network Toolbox (Version 5.0) and SVM Toolbox (Version 3.00) were employed.

The following general conclusions can be drawn from this study:

- Neural networks do indeed provide a reliable and an attractive alternative approach for the development of an ideal protection relaying system for the modern complex power transmission systems. The design of a power transmission system protection can be treated as a problem of learning machine pattern classification.
- It is important to investigate various neural network structures and learning algorithms before one selects a particular structure for a specific application. A trade-off between the off-line training and real-time implementation factors should be considered. In the case of transmission line system protection, the accuracy and classification/ recognition rates should be the first priority.

5.2 Future Work

As a further work it would be quite useful to generate a complete and integrated algorithm so as to convert all the neural networks proposed for the three tasks of fault detection, classification and location into a single program code. This algorithm would interface the three networks in a sequential manner so that the process starting with fault detection network to zone protection is fully automated. Consequently, when a fault is detected the last RMS voltage and current values are transferred to the fault classification network and after classifying the fault type, data are directed to the location network.

Another recommendation for future work is to implement the proposed networks as part of a real-time control scheme for the protection of power transmission systems and to verify the practical implications of the online realistic factors.

It is worth stating that this work concentrated on the performance of the proposed three network approaches as applied to fault detection, classification and location. To evaluate the practicality of the proposed networks, a number of realistic factors such as field data test and hardware implementation issues should be taken into consideration.

This work has shown that neural networks utilized effectively as a tool can facilitate and introduce more new dimensions in the problem domain of relaying and distance protection of power transmission line systems.

Bibliography:

[1] IEEE Tutorial Course, "*Microprocessor relays and protection systems*", The Institute of Electrical and Electronics Engineers, Inc. Printed in U.S.A. 1987.

[2] R. Aggarwal and Y. Song, "*Artificial neural networks in power systems; Part 3: Examples of applications in power systems*", IEE Power Engineering Journal, pp. 279-287, December 1998.

[3] Xiao-Ru Wang, Si-Tao Wu and Qing-Quan Qian, "*Neural Network Approach To Power Transmission Line Fault Classification*", International Conference on Information, Communications and Signal Processing ICICS '97, Singapore, 9-12 September 1997.

[4] M. Pfau- Wagenbauer and H. Brugger, "*Model and rule based intelligent alarm processing*", The Third Symposium on Expert Systems Application to Power Systems, Tokyo-Kobe, Japan, April 1-5, 1991, pp. 27-32.

[5] E. Handschin and W. Hoffmann, "*Integration of an expert system for security assessment into an energy management system*", Electrical Power and Energy Systems, vol. 14 no. 2/3, April/June 1992.

[6] D. S. Kirschen and B. F. Wollenberg, "*Intelligent alarm processing in power systems*", IEEE, vol. 80, no. 5, pp. 663–672, May 1992.

[7] T. Dillon, "*Survey on expert systems in alarm handling,*", Electra, CIGRE TF 38-06-02, Convener, 1991.

[8] E. Vásquez, O. Chacón, and H. J. Altuve, *"An on-line expert system for fault section diagnosis in power systems"*, IEEE Trans. on Power Systems, vol. 12, no. 1, pp. 357–362, Feb. 1997.

[9] Y. M. Park, G. Kim, and J. Sohn, *"A logic based expert system (LBES) for fault diagnosis of power systems"*, IEEE Trans. on Power Systems, vol. 12, no. 1, pp. 363–369, Feb. 1997.

[10] K. L. Butler, J. A. Momoh, L. G. Dias and D. J. Sobajic, *"An approach to power distribution fault diagnosis using a neural net based supervised clustering methodology"*, Engineering Intelligent Systems, 1: 51-57, CRL Publishing Ltd.

[11] M. Kezunovic and I. Rikalo, *"Detect and Classify Faults Using Neural Nets"*, IEEE Computer Applications in Power, pp.42-47, October 1996.

[12] D. S. Kirschen and B. F. Wollenberg, *"Intelligent Alarm Processing in Power Systems"*, Processing of the IEEE, vol. 80, no. 5, May 1992.

[13] D. Kirschen and J. Bann, *"AI Applications In Power Systems Operation"*, The Institute of Electrical Engineers, Printed and Published by IEE, Savoy Palace, London, WC2R, UK.

[14] R. Aggarwal and Y. Song, *"Artificial neural networks in power systems; Part 1: General introduction to neural computing"*, IEE Power Engineering Journal, pp. 129-134, June 1997.

[15] R. Aggarwal and Y. Song, *"Artificial neural networks in power systems; Part 2: Types of artificial neural networks"*, IEE Power Engineering Journal, pp. 41-47, February 1998.

[16] Members of CIGRE TF 38-06-06 on Artificial Neural Network Applications for Power Systems, "Artificial *neural networks for power systems: A literature survey"*, Engineering Intelligent Systems, 3: 133-158, CRL Publishing Ltd. December 1993.

[17] M. A. El-Sharkawi, *"Neural Networks and its Ancillary Techniques as Applied to Power Systems"*. The Institute of Electrical Engineers, Printed and Published by IEE, Savoy Palace, London, WC2R, UK. 1995.

[18] *"Neural network applications in power systems"*, Int. Journal of Engineering Intelligent Systems, vol. 1, no. 3, pp. 133–158, Dec. 1993.

[19] J. C. S. Souza, A. M. Leite Silva, and A. P. Alves Silva, *"Data debugging for real-time power system monitoring based on pattern analysis"* , IEEE Trans. on Power Systems, vol. 11, no. 3, pp. 1592–1599, Aug. 1996.

[20] J. C. S. Souza, A. M. Leite Silva, and A. P. Alves Silva, *"Online topology determination and bad data suppression in power system operation using artificial neural networks"* , IEEE Trans. on Power Systems, vol. 13, no. 3, pp. 796–803, Aug. 1998.

[21] T. Dalstein and B. Kulicke, *"Neural network approach to fault classification for high speed protective relaying"*, IEEE Transactions on Power Delivery, no. 4. pp. 1002-1009, 1995.

[22] B. H. Chowdhury and K. Wang, *"Fault Classification Using Kohonen Feature Mapping"*, The conference on intelligent system applications to power system. U.S.A., pp.194-198, 1996

[23] E. A. Mohamed and N. D. Rao, "*Artificial neural network based fault diagnosis system for electric power distribution feeders*", Electric Power Systems Research, 1995(35): 1-10.

[24] N. Kandil, K. Khorasani, R. Patel and V. Sood "*Fault identification in an ac-dc transmission system using neural networks*", IEEE Trans. on Power Systems. vol. 7, no. 2. pp. 812-819. 1991.

[25] T. S. Dillon and D. Niebur, "*Neural Networks applications in power systems*", CRL Publishing, London, England. 1996.

[26] W. M. Al-Hasawi, N. H. Abbasy and M. M. Mansour, " *Using fault induced transients and neural networks for T. L. ultra-speed fault detection and classification*", IPST'97 – International Conference on Power Systems Transients, pp. 412-417. 1997.

[27] W. B. Su, P. L. Mao, Z. Q. Aggarwal and A. T. Johns, " *A fault detection technique for compositive transmission circuits using artificial neural networks*", UPEC'97 – 32nd Universities Power Engineering Conference, 1, pp. 291-294. 1997.

[28] R. K. Aggarwal, Q. Y. Xuan, A. T. Johns and A. Bennett, "*Fault classification technique for double-circuit lines based on a combined unsupervised/supervised neural network*", IEEE Transactions on Power Delivery 14(4), pp.1250-1256. 1999.

[29] W. W. L. Keerthipala, C. T. Wai and W. Huisheng, "*Neural Network based classifier for power system protection*", Electrical Power Systems Research, 42, pp. 109-114, 1997.

[30] S. Vasilic and M. Kezunovic, "*An Improved Neural Network Algorithm for Classifying the Transmission Line Faults*", IEEE Computer Applications in Power Systems, pp. 918-923. 2002.

[31] M. Oleskovics, D. Coury and A. C. P. L. F. Carvalho, "*Artificial Neural Network Applied to Power System Protection*", Third Latin-American Congress: Electricity Generation and Transmission, Campos do Jordao, Brasil, 430-435, 1997.

[32] J. C. S. Souza, M. A. P. Rodrigues, M. T. Schilling and M. B. D. C. Filho, "*Fault Location in Electrical Power Systems Using Intelligent Systems Techniques*", IEEE Transactions on Power Delivery, vol. 16, no. 1 pp. 59-67. January 2001.

[33] V. N. A. L. Silva, L. A. V. Carvalho and G. Zaverucha, "*Artificial Neural Networks for Power Systems Diagnosis*", IEEE Transactions on Power Delivery, no. 94, pp.3738-3743. 1994.

[34] S. Websper, R. W. Dunn, R. K. Aggarwal, A. T. Johns and A. Benett, "*Feature extraction methods for neural network-based transmission line fault discrimination*", IEE Proc.-Gener. Transm. Distrib. no. 3, May 1999.

[35] S. Haykin, "*Neural Networks: A Comprehensive Foundation*", Prentice-Hall, Inc. Simon and Schuster/ A Viacom Company, Upper Saddle River, New Jersey, U.S.A., 1999.

[36] M. Sanaye-Pasand and O. P. Malik, "*Neural Network-Based Fault Direction Discrimination for High-Speed Transmission Line Protection*", Electric Power Components and Systems, vol. 29, no. 8, pp. 757-770, 2001.

[37] W. A. Elmore, *"Protective Relaying Theory and Application"*, Marcel Dekker Inc., New York, 1994.

[38] R. J. Schalkoff, *"Artificial Neural Networks"*, McGraw-Hill, New York , USA, 1997.

[39] M. El-Sharkawi and D. Niebur, *"A tutorial Course on Artificial Neural Networks with Applications to Power Systems"*, IEEE, Publ. No. 96TP 112-0, 1996.

[40] Y. H. Pao and D. J. Sobajic, *"Autonomous Feature Discovery of Critical Cleaning Time Assessment"*, Symposium on Expert Systems Applications to Power Systems, Stockholm-Helsinki, pp. 5.22-5.27, August 22-26, 1988.

[41] M. Kezunovic, *"A survey of Neural Net Applications to Protective Relaying and Fault Analysis"*, Engineering Intelligent Systems, vol.5, no.4, pp. 185-192, Dec.1997.

[42] M. Kezunovic, I. Rikalo and D. J. Sobajic, *"Real-Time and Off-Line Transmission Line Fault Classification Using Neural Networks"*, Engineering Intelligent Systems, vol. 10, pp. 57-63, 1996.

[43] Z. Q. Chen and J-C. Maun, *"Artificial Neural network Approach to Single-Ended Fault Locator for Transmission Lines"*, Power Industry Computer Application Conference, pp. 125-131, May.1997.

[44] Y. H. Song, *"Accurate Fault Location Scheme Based on Neural Networks Applied to EHV Transmission Systems"*, International Conference on Power System Technology, pp. 1028-1031, Oct.1994.

[45] Y. H. Song, Q. X. Xuan and A. T. Johns, "*Comparison of five neural network based classifiers for complex transmission lines*". Electric Power Systems Research, no.43, pp. 125

[46] Chen, J. and Patton, R., "*Robust Model-Based Fault Diagnosis for Dynamic Systems*", Kluwer Academic Publishers, 1999.

[47] Gerter, J., "*Fault Detection and Diagnosis in Engineering Systems*", Marcel Dekker, Inc, 1998.

[48] A. J. Mazon, I. Zamora, J. F. Minambres, M. A. Zorrozua, J. J. Barandiaran and K. Sagastabeitia, "*A New Approach to fault Location in Two-Terminal Transmission Lines Using Artificial Neural Networks*". Electric Power Systems Research, no. 56, pp. 261-266 56, 2000.

[49] A. G. Phadke and J. S. Thorp, "*Computer Relaying for Power Systems*", John Wiley and Sons Inc. 1988.

[50] *S. H. Horowitz and A. G. Phadke, "Power System Relaying"*, John Wiley and Sons Inc, 1992.

[51] *"IEEE Standard Dictionary of Electrical and Electronic Terms"*, ANSI/IEEE 100.

[52] T. Gonen, "*Modern Power System Analysis*", John Willey and Sons Inc., 1988.

[53] T. Gonen, "*Electric Power Distribution System Engineering*", McGraw Hill Book Comp., 1986.

[54] W. D. Stevenson, "Elements *of Power System Analysis*", McGraw-Hill, 1982.

[55] S. H. Horowitz, "*Protective Relaying for Power Systems*", IEEE Press book, John Wiley and Sons, New York, 1980.

[56] W. A. Lewis and L. S. Tippett, *"Fundamental Basis for Distance Relaying on 3-Phase Systems"*, AIEE Transactions, Vol. 66, pp 694-708, 1947.

[57] V. Cook. *"Analysis of Distance Protection"*, John Wiley and Sons Inc. 1985.

[58] *"Applied Protective Relaying"*, Westinghouse Electric Corporation, 1976.

[59] D. E. Rumelhart and J. L. McCleland. *"Parallel Distributed Processing"*, M.I.T. Press, 1986.

[60] K. Funahashi. *"On the approximate realization of continuous mappings by neural networks"*, Neural Networks, vol. 2 pp.183-192, 1989.

[61] K. Hornik, M. Stinchcombe, and H. White. *"Multilayer feed-forward networks are universal approximators"*, Neural Networks, vol. 2 pp.:359-366, 1989.

[62] D. S. Broomhead and D. Lowe. *"Multivariate functional interpolation and adaptive networks"*, Complex Systems, vol. 2 pp. 321-355, 1988.

[63] J. Park and I. W. Sandberg. *"Universal approximation using radial-basis-function networks"*, Neural Computation, vol. 32 pp. 246-257, 1991.

[64] A. Cichocki and R. Unbehauen. *"Neural Networks for Optimization and Signal Processing"*, John Wiley and Sons, 1993.

[65] James A. Leonard and Mark A. Kramer. *"Radial basis function networks for classifying process faults"*. IEEE Control Systems, pp. 31-38, Apr 1991.

[66] V. Vapnik, *"The Nature of Statistical Learning Theory"*, NY: Springer-Verlag. 1995.

[67] V. Vapnik, *"Statistical Learning Theory"*, Wiley-Interscience, New York, (1998).

[68] N. Cristianini, J. Shawe-Taylor, "*An Introduction to Support Vector Machines*", Cambridge University Press, 2000.

[69] S. Schölkopf, C. J. Burges, A. J. Smola, "*Advances in Kernel Methods: Support Vector Learning*", MIT Press, Cambridge, MA, (1999).

[70] J. Platt, N. Cristianini, J. Shawe-Taylor, "*Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods*", Advances in Neural Information Processing Systems 12, pp. 547-553, MIT Press, (2000).

[71] C. J. Burges, "*A Tutorial on Support Vector Machines for Pattern Recognition*", Knowledge Discovery and Data Mining, vol.2, no.2, 1998.

[72] B. Scholkopf, K. Sung, C. Burges, "*Comparing Support Vector Machines with Gaussian Kernels to Radial Basis Function Classifiers*", IEEE Trans. on Signal Processing. Vol. 45, no. 11, pp. 2758-2765, 1997.

[73] T. S. Dillon, K. Morsztyn and K. Phua, "*Short Term Load Forecasting Using Adaptive Pattern Recognition and Self-Organizing Techniques*", Proc. of 5th Power System Computation Conference, Cambridge, paper 2.4/3, pp. 1-16, 1975.

[74] S. R. Gunn, "*Support Vector Machines for Classification and Regression*", Technical Report, Image Speech and Intelligent Systems Research Group, University of Southampton, 1997.

Appendix A:

Power Transmission Line System Data

Bus Phase Voltage:

Vrms = 311 KV

Transmission Line 1:

Line length: 80 km	$R = 2.21\ \Omega$	$L = 0.0936$ H	$C = 19.08\ \mu F$

Transmission Line 2:

Line length: 150 Km	$R = 4.43\ \Omega$	$L = 0.1755$ H	$C = 35.775\ \mu F$

Transmission Line 3:

Line length: 100 Km	$R = 3.225\ \Omega$	$L = 0.117$ H	$C = 23.85\ \mu F$

Generation Units:

$E_1 = 1.0 \angle 0^o$

$E_2 = 0.95 \angle 45^o$

S_{base}=500 MVA (Base Power).

V_{base}=350 KV (Base Voltage).

Z_{1IR}=0.1 (E_1 Impedance Ratio).

Z_{2IR}=0.1 (E_2 Impedance Ratio).

Appendix B:

Variables Considered for the Test Set:

Fault Locations (km):

94, 112, 122, 133, 142, 153, 162, 173, 182, 193, 202, 213, 222, 227, 233, 242, 248, 252, 262, 273, 277, 283, 288, 292, 297, 303, 313, 312, 316, 319.

Fault Resistance (ohm):

- **Phase to Ground:** 30, 60, 90.
- **Phase to Phase** : 0.3, 0.5, 0.6, 0.9.