Rough Neural Fault Classification for HVDC Power Systems

by
Liting Han

A Dissertation
Submitted to the Faculty of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy
in Computer Engineering

Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba
Canada R3T 5V6

ABSTRACT

This Ph.D. thesis proposes an approach to classify faults that commonly occur in a High Voltage Direct Current (HVDC) power system. These faults are distributed throughout the entire HVDC system. The most recently published techniques for power system fault classification are the wavelet analysis, two-dimensional time-frequency representation for feature extraction and conventional artificial neural networks for fault type identification. The main limitation of these systems is that they are commonly designed to focus on a group of faults involved in a specific area of a power system. This thesis introduces a framework for fault classification that covers a wider range of faults.

The proposed fault classification framework has been initiated and developed in the context of the HVDC power system at Manitoba Hydro, which uses what is known as the Transcan^{TM} system to record and archive fault events in files. Each fault file includes the most active signals (there are 23 of them) in the power system. Testing the proposed framework for fault classification is based on fault files collected and classified manually over a period of two years.

The fault classification framework presented in this thesis introduces the use of the rough membership function in the design of a neural fault classification system. A rough membership function makes it possible to distinguish similar feature values and measures the degree of overlap between a set of experimental values and a set of values representing a standard (*e.g.*, set of values typically associated with a known fault). In addition to fault classification using rough neural networks, the proposed framework includes what is known as a linear mean and standard deviation classifier. The proposed framework also includes a classifier fusion technique as a means of increasing the fault classification accuracy.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my esteemed supervisor, Professor J. F. Peters, whose expertise, understanding and patience added considerably to my graduate experience. I appreciate his vast knowledge and skill in the areas of modeling, design and analysis of intelligent systems and his assistance in writing journal articles, conference papers and other reports. I am grateful to my committee members, Professor G. Martens, Professor M. Pawlak, Professor D. Gunderson and the external examiner, Professor Z. Ras, for their suggestions, comments and insights concerning this thesis. I would also like to thank the other members of the Computational Intelligence Laboratory at the University of Manitoba for the assistance they gave me throughout my doctoral studies.

My sincere gratitude also goes to all my family members for their love, support, and patience over the last few years. I must acknowledge my husband, Ruifeng, without whose love, encouragement and editing assistance, I would not have finished this thesis. My lovely boy, Alfred, has been a source of inspiration during my research.

In particular, this completed thesis will be the most valuable and unexpected gift for my mother in her life, for everything she has done for me, and for her success in fighting against cancer.

The financial support received from Manitoba Hydro is gratefully acknowledged. The assistance provided by Mr. L. Crowe and the other engineers and staff at Manitoba Hydro Dorsey Station was also very much appreciated.

Contents

List of Figures

List of Tables

1 Introduction

This thesis introduces a rough membership neural network (rmNN) approach to fault classification for high voltage direct current (HVDC) power systems. In addition to rmNNs, a number of other methods such as autocorrelation, cross-correlation, Wavelets, FFT, IFFT, low pass filter, phase shifting, derivatives and coding have been used in this research to analyze and characterize the 23 most active signals recently recorded at the Dorsey Station, Manitoba, Canada. The capstone of the proposed classification system is a form of classifier fusion that combines rmNNs and a linear mean and standard deviation (LMD) based classification method. Classifier fusion has been introduced as a means of improving fault classification accuracy.

1.1 Motivation

With the rapid increase of electrical power consumption by utilities and industries, more stability and efficiency in power delivery is needed. A report by CEIDS (Consortium for Electric Infrastructure to Support a Digital Society) shows that the U.S. economy is losing between $104 billion and $164 billion a year due to power outages [6]. The analysis and classification of power system disturbances are becoming mandatory in working towards minimizing and even eliminating power outages. Typically, an effort is made to identify the most significant patterns of system faults that provide input to a region-based analysis system for decision support. Operators or engineers make use of the summary reports to operate and maintain a power system.

1.2 Thesis Outline

This thesis is organized as follows:

- Chapter 1 (this chapter) is an introduction for this thesis.
- Chapter 2 briefly introduces power system fundamentals and a brief overview of power system faults.
- Chapter 3 reviews the methods commonly used in power system fault identification.
- Chapter 4 gives an overview of fault identification techniques commonly used in the electrical power industry.
- The main parts of the research completed for this thesis are presented in Chapter 5 to Chapter 9.
- Chapter 10 gives a conclusion.

2 Power System Fundamentals

This section briefly introduces the power system fundamentals [76, 21] required for an understanding of power system faults.

2.1 Power Systems

Electric power transmission was originally developed with direct current (DC). The availability of transformers and the development and improvement of induction motors at the beginning of the 20^{th} century, led to the use of alternating current (AC) transmission. Even so, d.c transmission is generally used for the following reasons:

1. An overhead DC transmission line with its towers can be designed to be less costly per unit of length than an equivalent AC line designed to transmit the same level of electric power. However the DC converter stations at each end are more costly than the terminating stations of an AC line and so there is a break-even distance above which the total cost of DC transmission is less than the cost of AC transmission. In addition, DC transmission line can have a lower visual profile than an equivalent AC line, which contributes to a lower, perceived environmental impact. An environmental advantage to a DC transmission line over an AC line is the presence of lower electromagnetic fields.

2. If transmission is by underground cable, the break-even distance is less than overhead transmission. It is not practical to consider AC cable systems exceeding 50 kilometers but hundreds of kilometers of underground DC cable transmission systems are feasible.

3. Some AC electric power systems are not synchronized with neighboring networks even though the physical distance between them is quite small. This occurs in Japan where half the country has a 60Hz network and the other has a 50Hz system. It is physically impossible to connect the two by direct AC methods for the purpose of exchanging electric power between them. However, if a DC converter station is located in each system with an interconnecting DC link, it is possible to transfer power flow from one system to the other.

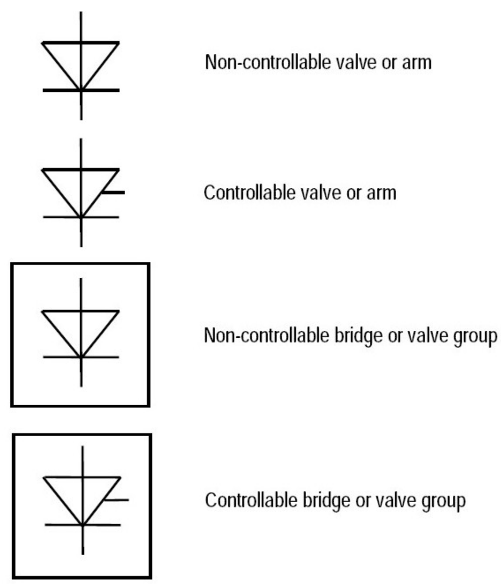


Figure 1: Standard graphical symbols for valves and bridges [76].

The integral part of an HVDC power converter is the valve or valve arm. It may be non-controllable if constructed from one or more power diodes in series or controllable if constructed from one or more thyristors in series. Figure 1 depicts the International Electrotechnical Commission (IEC) graphical symbols for valves and bridges (valve groups). The standard bridge or converter connection is defined as a 2-way connection consisting of six valves or valve arms, which are shown in Figure 2. Electric power flowing between an HVDC valve group and an AC system is three phase. When electric power flows into a DC valve group from an AC system, it flows through a rectifier. If power flows from the DC valve group into the AC system, it flows through an inverter.

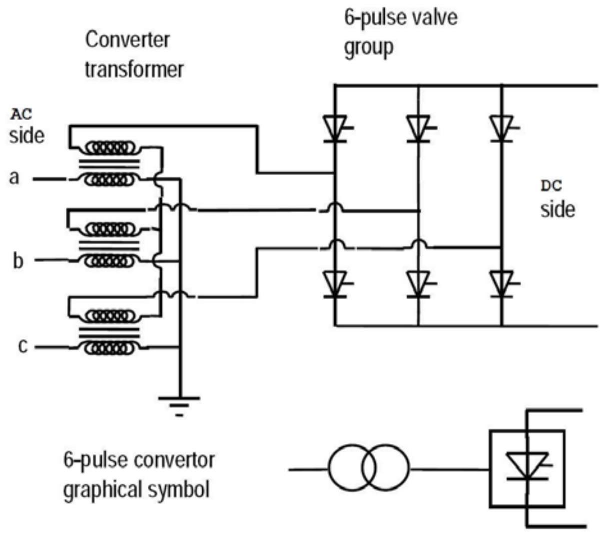


Figure 2: Electric circuit configuration of the basic 6-pulse valve group with its converter transformer in star-star connection [76].

The most common building block for HVDC valves is the thyristor (see Figure 3 for characteristics of a thyristor). In the 'off' state, a thyristor blocks the flow of current as long as the reverse or forward breakdown voltages (V_{br} or V_{bo}) are not exceeded. A thyristor can be made to attain an 'on' state if it is forward biased ($V_{ak} > 0$) and a small positive 'gate' voltage is applied between the gate and the cathode. This 'firing pulse' need not be present once the thyristor is ignited, although in practice, a train of pulses in rapid succession is often maintained over an entire conduction period. Once turned on, a thyristor follows its 'on' characteristic as shown in Figure 3. Note that the forward voltage drop in the on condition is relatively small and an actual thyristor characteristic closely follows that of an ideal switch (horizontal line for the 'off' state, vertical y axis for the 'on' state). The thyristor can also turn on if the voltage across it exceeds the forward break-over voltage V_{bo}. This mechanism is often used to protect a thyristor against excessive voltage.

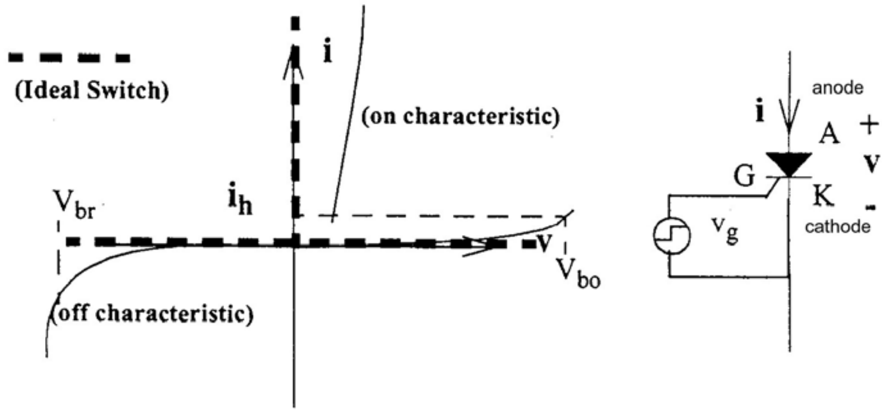


Figure 3: Thyristor characteristic [21].

The normal state transition diagram for a thyristor is shown in Figure 4. The thyristor attains its 'off' state when the current through it attempts to reverse. One other factor that is necessary for a successful turn-off is that a thyristor must not be subject to a forward biasing voltage too soon after the current has extinguished. Otherwise, there is a possibility of re-ignition even in the absence of a pulse. Re-ignition occurs when the charge carriers in the semi-conductor have not had sufficient time to be re-absorbed. This critical time is referred to as the turn-off time t_{off} and often expressed in terms of a so called "extinction angle" $\gamma = \omega t_{off}$, if AC waveforms of angular frequency ω are involved. This phenomenon in which a thyristor fails to attain its forward blocking state, 'off' state, is referred to as commutation failure.

The 6-pulse bridge is the most widely used HVDC converter configuration. Figure 5 shows a typical 6-pulse thyristor bridge with the AC supply, the converter transformer X_c and the DC-side smoothing reactance. A 6-pulse bridge consists of an upper and a lower half as seen in Figure 6(a). It is assumed initially that the converter transformer is ideal so that there is no leakage inductance. It is also assumed that ideal thyristors behave like diodes, *i.e.*, zero voltage drop when the device is on and an ideal open circuit when off.

The device is in a conducting state as soon as the forward biased voltage ($V_{ak} > 0$) causes current to flow in the forward (anode to cathode) direction and no 'firing pulse' is required.

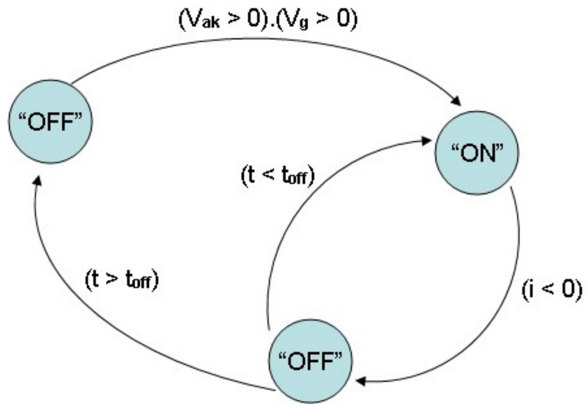


Figure 4: State transition diagram for thyristor switching [21].

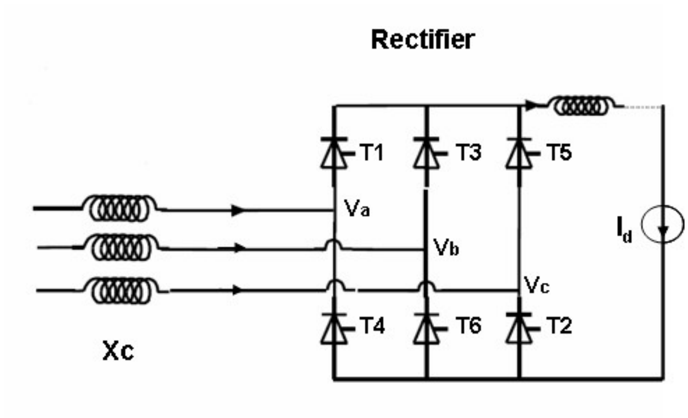
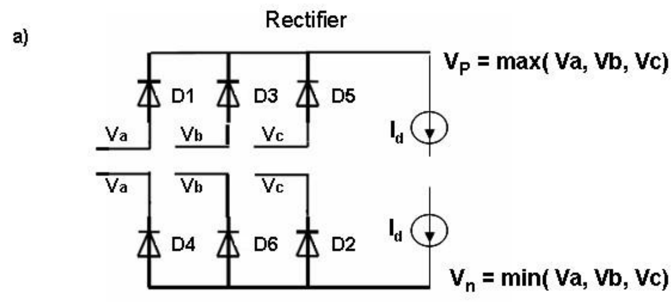


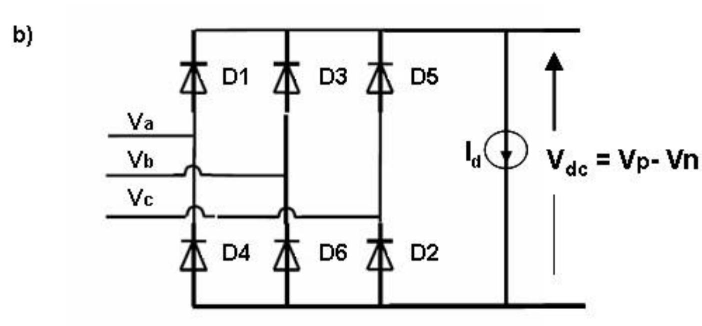
Figure 5: Three phase (6-pulse) bridge.

The upper bridge half is a standard maximum select circuit that selects the largest of the three voltages V_a, V_b and V_c at the common cathode terminal. This can be proved by contradiction. To see this , assume $V_a < V_b$ but that $V_p = V_a$ because D1 is assumed to be conducting. Then D3 should also conduct since it is forward biased because $V_b > V_a$, hence, $V_a = V_b$, which is a contradiction. The only possibility that does not lead to a contradiction is for V_p to be equal to the largest of the three voltages.

Similarly the lower bridge half causes a voltage $V_n = min(V_a, V_b, V_c)$ to appear at the common anode terminal of devices D2, D4 and D6. Thus the total DC side voltage as can be seen from Figure 6(b) must be the difference $V_{dc} = V_p - V_n$. The waveforms for the bridges are shown in Figure 7. The current on the AC side in phase a is I_d when D1 conducts and $-I_d$ when D4 conducts. The conduction period for D1 can be determined from the waveforms as the period in which the voltage V_a of phase a is the largest of the three phase voltages. Similarly, D4 is on when V_a is at its smallest in magnitude.



(a)



(b)

Figure 6: Analysis of three phase (6-pulse) bridge.

The sequence of conduction for the valves in the upper bridges is D1, D3, D5, D1, D3, D5, and so on, since each successive phase dominates over a 120° interval. In the bottom bridge, the sequence is D2, D4, D6, D2, D4, D6, and so on. Considering the two halves together, each valve enters conduction 60° after its predecessor in the sequence D1, D2, D3, D4, D5, D6, D1, D2, D3, D4, D5, D6, and so on.

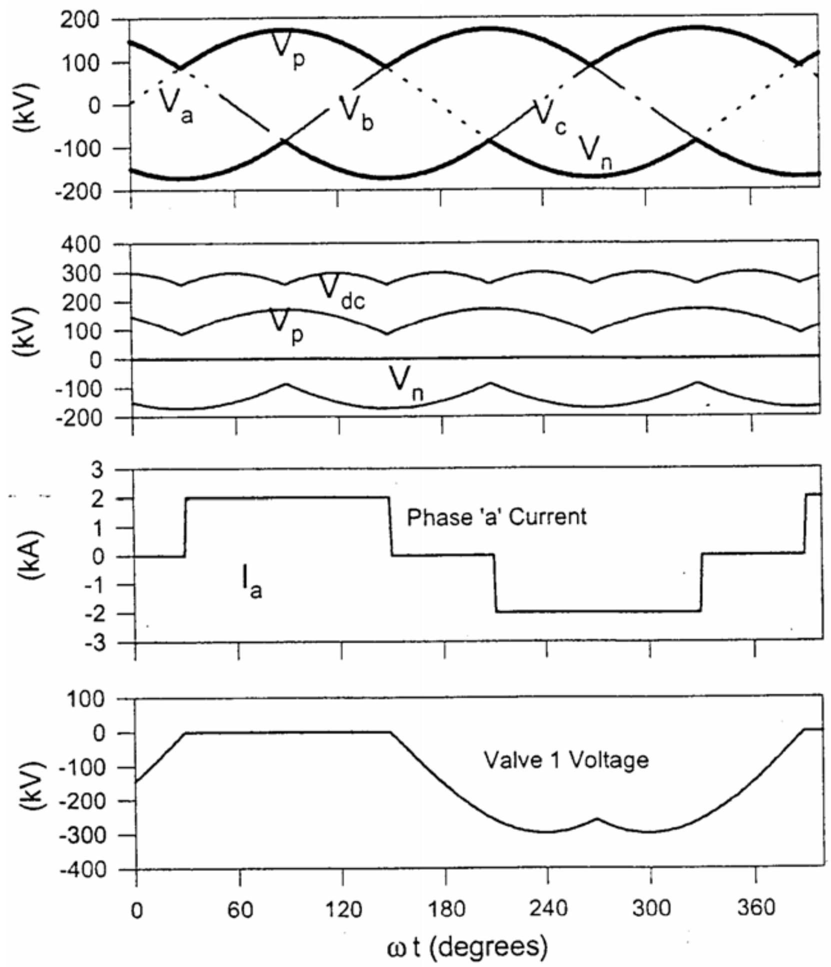


Figure 7: Three phase diode bridge waveforms (no overlap) [21].

Without any series inductance in the circuit, the current instantaneously rises to the value $\pm I_d$ on turn-on and makes an instantaneous transition to zero on turn-off when the current transfers to the next phase. The valve voltage is an important parameter in determining the valve rating. The voltage in the forward direction across valve 1 is determined to be $V_a - V_p$, and while the valve is conducting this voltage is zero.

In practice, transformer leakage inductance must be considered. With the inclusion of transformer reactance X_c shown in Figure 8, the current can no longer make an instantaneous transition from one phase to another because that would require a discontinuous change in inductor current as is evidenced from the waveforms shown in Figure 9. In this case, when valve 1 is turned on, there is an "overlap" between valve 1 and valve 5, *i.e.*, valve 1 is turned on while valve 5 starts to be turned off. The overlap interval is represented by the angle μ. During this interval, the DC-side voltage V_p (similarly V_n) is the average of the two conducting phase voltages, *i.e.*, V_a and V_c. Also note from Figure 9 that the valve voltage waveform now has additional commutation "spikes".

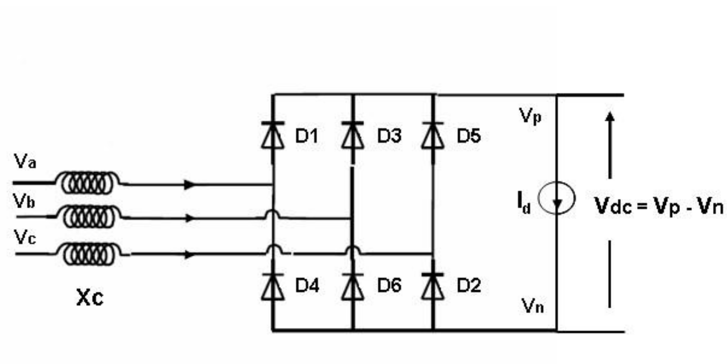


Figure 8: Three phase (6-pulse) bridge: transformer inductance included.

The thyristors in a controlled bridge are idealized, *i.e.*, a thyristor behaves like a diode, except that mere forward bias (positive anode-cathode voltage) is not sufficient to ensure conduction. The additional condition to attain the conducting state is a required gate, 'firing pulse' that must be present in addition to a forward bias. Hence, the main difference in analyzing the operation of a thyristor bridge is that the maximum (or minimum) select action only commences on the issue of a firing pulse. The thyristor valves are fired in the sequence T1, T2, T3, T4, T5, and T6. The elapsed angle from the earliest instant at which a thyristor may conduct (*i.e.*, the point at which forward bias first appears) to the instant at which the firing pulse is issued and the valve commences conduction is called the "firing" or "delay" angle and is denoted by the Greek letter α.

In the waveforms shown in Figure 10, $\alpha = 15^\circ$ has been used. Also note that in Figure 10, the pulse duration is a full 120°. This is not strictly necessary, since a thyristor

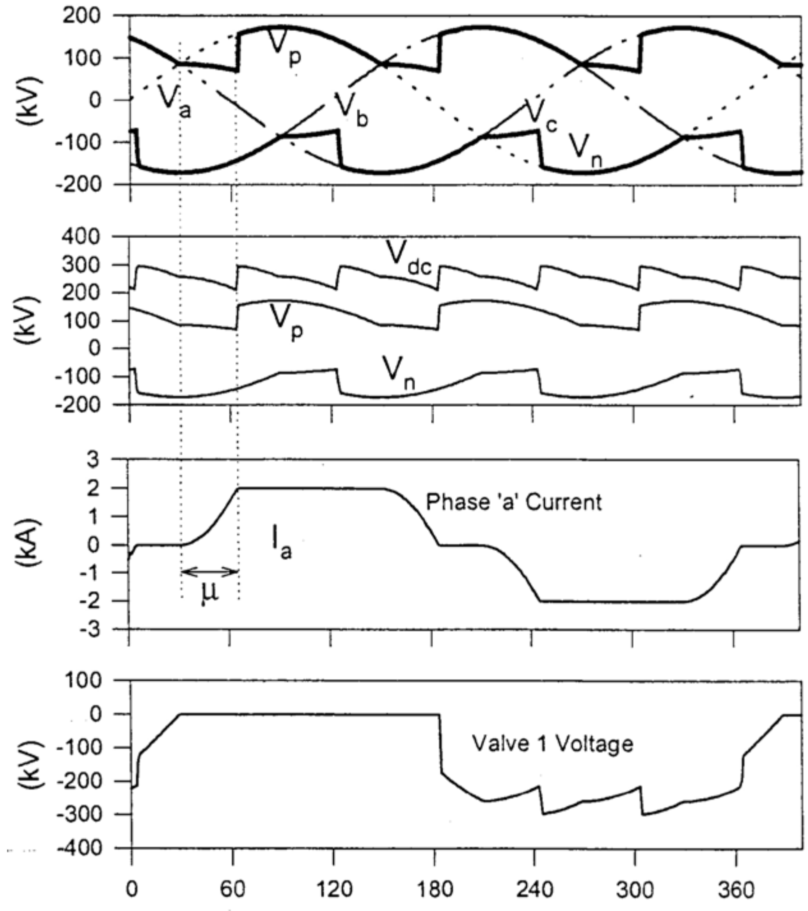


Figure 9: Three phase diode bridge waveforms [21].

valve that has been triggered on continues conducting until the current through it attempts to flow in the reverse direction. However, in HVDC systems, it is common practice to keep pulsing continuous over a valve's nominal conduction interval of 120° (in the form of a train of high-frequency pulses) in case a premature current zero occurs because of waveform distortions. Note that for this value of the firing angle ($\alpha = 15^\circ$), the DC voltage is positive and the power flows from the AC to the DC side. This is the "rectifier" mode of operation. Note that if continuous current is maintained in the circuit by some external device, the firing angle α can be made to have a value in excess of 90°. In this situation, the voltage V_p turns out to be negative and V_n is positive, which causes the DC voltage to be negative. Thus, power transfer is from the DC side to the AC side, although the direction of the DC current remains the same. This is the “inverter” mode of operation.

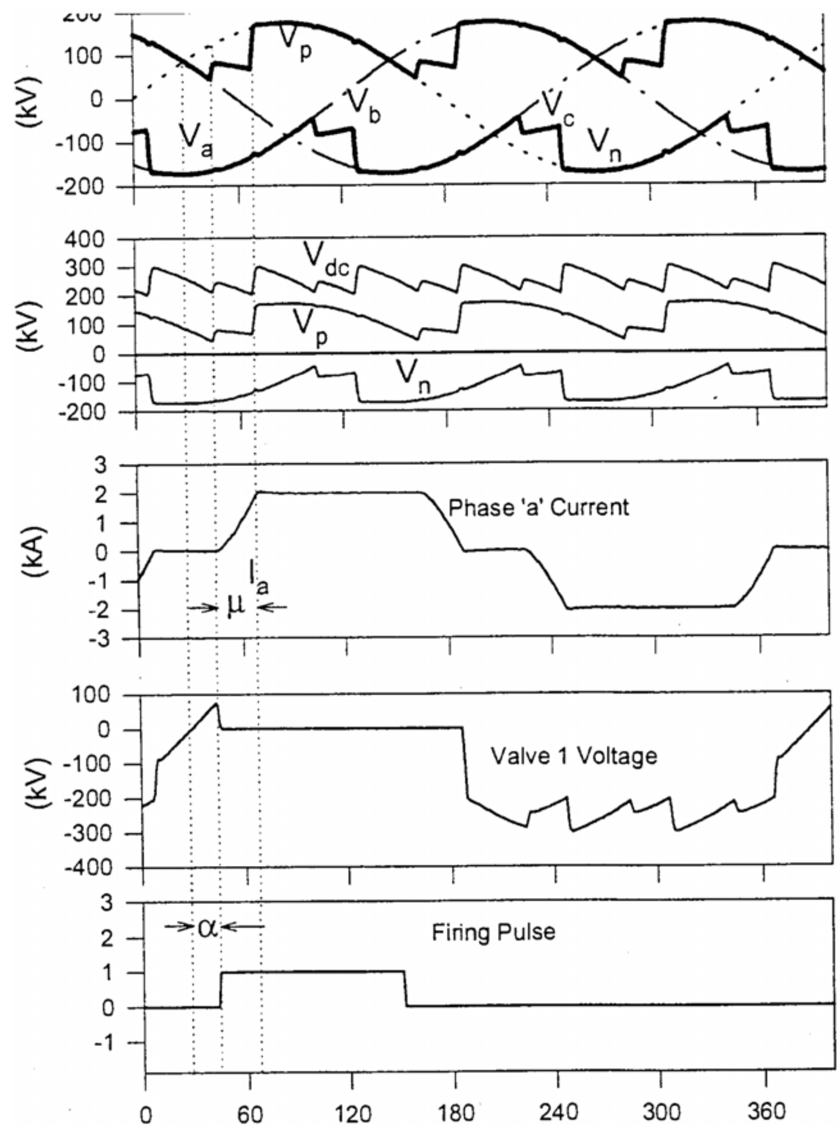


Figure 10: Controlled thyristor bridge waveforms: $\alpha = 15°$ [21].

2.2 Power System Faults

A power system fault is the result of an electrical disturbance. At the Manitoba Hydro Dorsey Station, the Transcan^{TM} recording system is deployed as a power system monitoring tool. It archives 31 power signals in a fault file whenever a power system fault occurs. A typical screen snapshot of 31 signals recorded by Transcan^{TM} is shown in Figure 11. Transcan^{TM} is capable of recording power system faults in a real-time manner. However, this system cannot identify the type and cause of a recorded fault. Engineers at the Dorsey Station must visually assess all the 31 signals then manually log the cause of the fault into the database of the Transcan^{TM} system and consolidate this information into an archived fault file. The graphical user interface (GUI) of the Transcan^{TM} system is shown in Figure 12. The 23 most active and informative signals referenced in the proposed fault classification system are listed in Table 1.

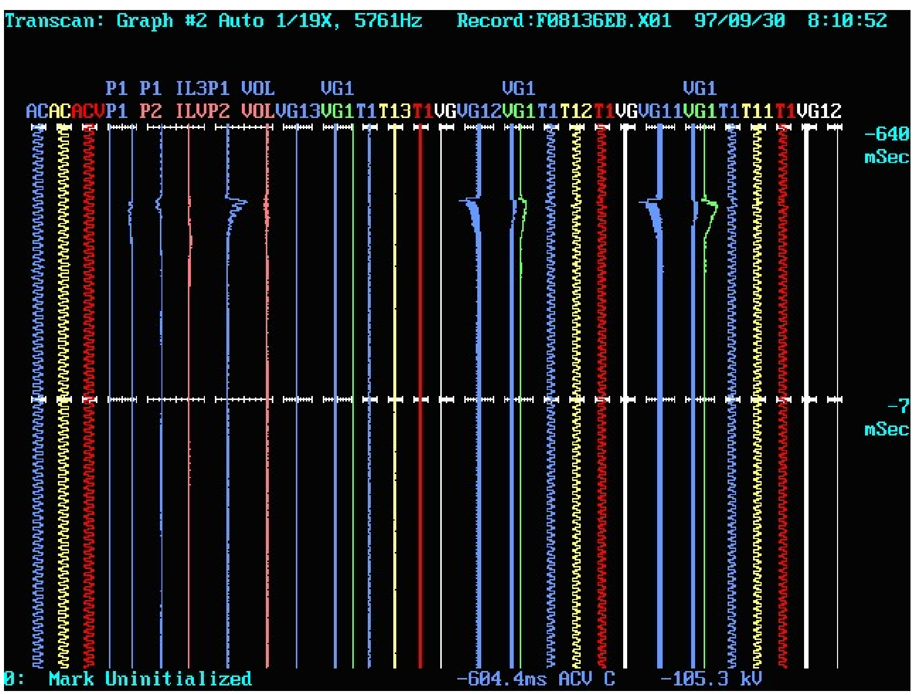


Figure 11: 31 signals in the “Valve Current Commutation Failure” fault.

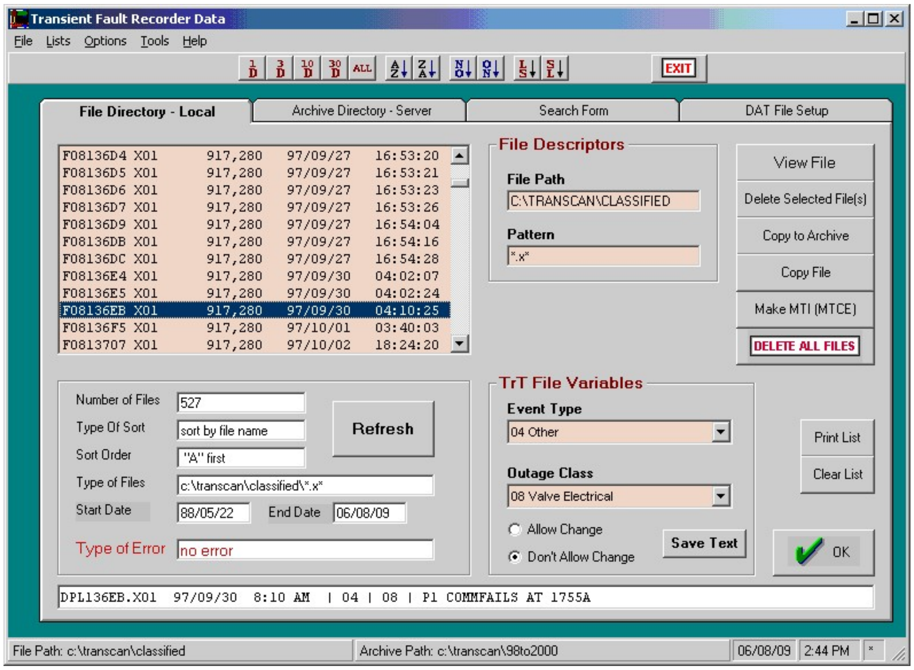


Figure 12: The Transcan^{TM} system GUI.

Table 1: Most active power system signals.

Bus signals	AC Phase A, B, C	Sinusoidal
	Pole voltages and currents	Constant
	Pole current order	Constant
Valve signals (total 3 valve groups)	6-pulse	Periodic
	Current A phase, B phase, C phase	Sinusoidal
	Start pulse	Periodic

Table 2: Common power system faults.

Fault index	Fault name	Number of fault files
fault 1	Minor AC Disturbance	240
fault 2	Severe AC Disturbance	148
fault 3	Valve Current Closed/Blocked/Deblocked	114
fault 4	Line Fault	81
fault 5	Valve Current Commutation Failure	95
fault 6	Pole Voltages/Currents Closed/Blocked/Deblocked	64
fault 7	Phase Current Arc Back	26
fault 8	Parallel Operation	29
fault 9	Pole Current Oscillation	31
fault 10	Normal Affected by Another Pole	18
fault 11	Asymmetric Protection	25
fault 12	Disturbance on DC Voltage	25

The twelve most common power system faults are listed in Table 2. An information table for fault classification cannot be established without a good understanding of the mechanism underlying each fault and the behavior of the signal associated with each fault.

- AC Voltage Disturbance. This is a bus error that will induce some other faults such as valve current commutation failure, line fault and valve current blocked. Normally, three AC phase voltages are sinusoidal signals that have a fixed $120°$ phase delay relative to each other. The AC voltage line will be impacted by different disturbances such as a falling tree hitting a transmission line, heavy snowfall or severe wind, and sometimes radiation or magnetic field interference.

- Valve Current Closed/Blocked/Deblocked. This fault happens in one or two valve groups. There are three valve groups in poles 1 and 2, and two valve groups in poles 3 and 4. Vg11, Vg12, Vg13 designate pole 1; Vg21, Vg22, Vg23, pole 2; Vg31, Vg32, pole 3; Vg41, Vg42, pole 4. A failure of a 6-pulse signal in a valve group will shut down or block the valve currents. An AC voltage disturbance also has the same effect. The restart of the 6-pulse signal will unblock the valve currents.

- Line Fault. This fault is due to the AC voltage disturbance, the pole line short to the ground or the energy of a DC line decreases (line force retard) causing a pole line voltage flashover or shutdown. The power system will restart in a short time if the control system responds quickly.

- Valve Current Commutation Failure. This happens when a valve is not turned off successfully because the valve is subject to a forward biasing voltage too soon after the current has been extinguished. This causes a minor valve current distortion for a very short period of time.

- Pole Voltages/Currents Closed/Blocked/Deblocked. This happens when all the valve groups in one pole are closed, blocked or deblocked.

- Phase Current Arc Back. This happens only in one valve group. The valve current increases sharply for a short period of time and then shuts down. This type of power system fault is caused when valve lines short together or short to ground.

- Parallel Operation. This is not a fault but an indicator that the line maintenance is in progress. When a pole current line needs to be tested, the current will be switched to another pole line. Inside the power station, the current of this pole line goes down to 0; outside the station, the current provided does not decrease, and the pole voltage remains normal.

- Pole Current Oscillation. This fault is caused by oscillation of the pole current order. Usually with this fault, the pole voltage remains relatively constant.

- Normal Affected by Another Pole. This fault happens occasionally. There is a bi-pole power system at the Dorsey Station. Pole 1 and pole 2 compose one active station. Pole 3 and pole 4 are usually for a back-up station. If a fault, especially a line fault, occurs in pole 1, Transcan^{TM} will generate 2 fault files: one for pole 1, and one for pole 2 even in the case where pole 2 is absolutely normal. This occurs because the bus signals, pole voltages and currents are shared and reordered in both pole 1 and pole 2 fault files.

- Asymmetric Protection. If the pulse to open the valve arrives in an abnormal sequence, this fault will cause more than two valves to open at the same time. The circuit control system will then force this valve group to close. The most noticeable event associated with this fault is that the 6-pulse signal will have 7 cycles of severe oscillation and will be closed until the control system restarts the valve group.

- Disturbance on DC Voltage. At the Dorsey Station, the AC voltage is converted from the DC voltage. The long distance transmission of DC voltage is easier and the interference problem is greatly decreased. However, sometimes snow on DC transmission lines or windy weather will cause changes in the DC voltages and a DC voltage disturbance is recorded.

3 Mathematics Underlying Fault Classification and Recognition Techniques

This section gives an overview of the mathematics underlying fault classification and fault recognition techniques.

Nomenclature

Symbol	Brief Explanation
$\bar{f}(t)$	Complex conjugate of $f(t)$
$f(t) \star g(t)$	Cross-correlation of two complex functions $f(t)$ and $g(t)$
$f(t) * g(t)$	Convolution of $f(t)$ and $g(t)$
$R_f(t)$	Autocorrelation function of a continuous real function $f(t)$
$\gamma(s, \tau)$	Continuous wavelet transform
$\Psi_{s,\tau}(t)$	A single basic wavelet, a so-called mother wavelet
$\varphi(\eta, \tau)$	A kernel function in Time-Frequency Representation algorithm (TFR)
$A(\eta, \tau)$	Ambiguity plane of a signal in Time-Frequency Representation algorithm (TFR)
$P(t, f)$	The two-dimensional Fourier transform of the product of the ambiguity plane $A(\eta, \tau)$ of the signal and a kernel function $\varphi(\eta, \tau)$
ρ-correlation	Correlation between multi-classifiers
FFT	Fast Fourier Transform
DFT	Discrete Fourier Transform
CWT	Continuous Wavelet Transform
DWT	Discrete Wavelet Transform
IDWT	Inverse Discrete Wavelet Transform
MRA	(Wavelet) Multi-Resolution Analysis
TFR	Time-Frequency Representation
ANN	Artificial Neural Network
MLF	Multi-Layer Feedforward (ANN)
RBF	Radial Basis Functions
BP	Backprobagation

CG	Conjugate Gradient
LM	Levenberg Marquardt
OLS	Orthogonal Least Square
FP	False Positive (error)
FN	False Negative (error)
TER	Total Error Rate
TSR	Total Success Rate

3.1 Correlation Theory

Correlation is the degree to which two or more quantities are linearly associated. The cross-correlation of two complex functions $f(t)$ and $g(t)$ of a real variable t, denoted $f \star g$, is defined by (1) [46]

$$f \star g = \bar{f}(-t) * g(t), \tag{1}$$

where $*$ denotes convolution and $\bar{f}$ is the complex conjugate of $f(t)$. Since convolution is defined as (2)

$$f(t) * g(t) = \int_{-\infty}^{\infty} f(\tau) g(t-\tau) d\tau, \tag{2}$$

it follows that

$$f(t) \star g(t) = \int_{-\infty}^{\infty} \bar{f}(-\tau) g(t-\tau) d\tau. \tag{3}$$

Let $\tau' \equiv -\tau$, $d\tau' = -d\tau$, then (3) is equivalent to

$$\begin{aligned} f \star g &= \int_{\infty}^{-\infty} \bar{f}(\tau') g(t+\tau')(-d\tau') \\ &= \int_{-\infty}^{\infty} \bar{f}(\tau) g(t+\tau) d\tau. \end{aligned} \tag{4}$$

Similarly, for a complex function $f(t)$, the autocorrelation $\rho_f(t)$ is defined by (5) [46]

$$\begin{aligned} \rho_f(t) &\equiv f \star f \\ &= \bar{f}(-t) * f(t) \\ &= \int_{-\infty}^{\infty} \bar{f}(\tau) f(t+\tau) d\tau, \end{aligned} \tag{5}$$

Let series $\{a_i, i = 0, 1, ..., N-1\}$ be a periodic sequence, then the autocorrelation of the sequences, sometimes called the periodic autocorrelation, is written as (6) [78]

$$\rho_i = \sum_{j=0}^{N-1} a_j a_{j+i}, \tag{6}$$

where the final subscript is understood to be taken modulo N. The cross-correlation and autocorrelation discard phase information, returning only the power, and are therefore irreversible operations.

The most important property of correlation is that $f \star f$ is maximum at the origin ($x = 0$), in other words,

$$\int_{-\infty}^{\infty} f(u) f(u+x) du \leq \int_{-\infty}^{\infty} f^2(u) du. \tag{7}$$

It is efficient to classify the waveforms of fault signals for differentiating one fault from others by applying the cross-correlation and autocorrelation operations.

3.2 Conventional Fast Fourier Transform (FFT)

Fourier methods such as the Fourier series and Fourier integral are used in analyzing continuous time signals. That is, Fourier methods are applicable in systems where there is a characteristic signal $s(t)$ defined for all values of t in the interval [-∞, ∞].

A Fourier transform decomposes a waveform into a sum of sinusoids of different frequencies [3]. The signal $s(t)$ in the time domain is decomposed into the sum of its sinusoids $S(f)$ in the frequency domain by,

$$S(f) = \int_{-\infty}^{\infty} s(t) e^{-j2\pi ft} dt, \tag{8}$$

where $j = \sqrt{-1}$.

In this thesis, the focus is on the application of what is known as the Discrete Fourier Transform (DFT) that is applicable to discrete-time signals. A discrete time signal $s[n]$ is defined for values of n in the interval [-∞, ∞]. A discrete Fourier transform is used in studying finite collections of sampled data $\{s_0, ..., s_{N-1}\}$ relative to the sequence $\{S_0, ..., S_{N-1}\}$. The DFT is given by,

$$S_k = \sum_{n=0}^{N-1} s_n e^{-j\frac{2\pi}{N}nk}, k = 0, 1, ..., N-1. \tag{9}$$

A fast Fourier transform results from the application of a particular algorithm that can compute the DFT more rapidly than other available algorithms [3].

3.3 Wavelet Transform

The big disadvantage of a Fourier expansion is that it has only frequency resolution and no time resolution. This means that although we might be able to determine all the frequencies present in a signal, we do not know when they are present [73]. The wavelet transform provides a means of overcoming the shortcomings of the Fourier transform. In wavelet analysis, the use of a fully scalable modulated window makes it possible to know the exact frequency and the exact time of occurrence of this frequency in a signal. In other words, a signal can simply be represented as a point in the time-frequency space. The window is shifted along the signal and for every position the spectrum is calculated. Then this process is repeated many times with a slightly shorter (or longer) window for every new cycle. In the end, the result will be a collection of time-frequency representations of a signal, all with different resolutions.

Wavelets provide a form of multiresolution analysis resulting from the collection of representations produced by applying a set of functions of different scales to a signal. Large scales are used to paint the big picture, while small scales expose the details. Thus, going from large scale to small scale is analogous to zooming in.

The Continuous Wavelet Transform (CWT) in general is formally defined by (10) [27]:

$$\gamma(s,\tau) = \int f(t)\Psi^{*}_{s,\tau}(t)dt, \tag{10}$$

where $*$ denotes complex conjugation. Equation (10) shows how a function $f(t)$ is decomposed into a set of basis functions called wavelets. The variables s and τ, scale and translation, are the new dimensions after the wavelet transform. The inverse wavelet transform can be written as shown in (11) [27]:

$$f(t) = \int\int \gamma(s,\tau)\Psi_{s,\tau}(t)dsd\tau. \tag{11}$$

The wavelets $\Psi_{s,\tau}(t)$, sometimes called child wavelets, are generated from a single basic wavelet $\Psi(t)$, the so-called mother wavelet, by scaling (parameter s) and translation (parameter τ) [27]. For a wavelet $\Psi_{s,\tau}(t)$, a family of curves with parameters s and τ can be formed as:

$$\Psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\Psi(\frac{t-\tau}{s}), \tag{12}$$

where s is the scale factor, τ is the translation factor and $\frac{1}{\sqrt{s}}$ is the factor for energy normalization across the different scales.

Unlike the Fourier transform or other transforms, the wavelet basis function, $\Psi(t)$ is not specified. The theory of wavelet transforms deals with the general properties of the wavelets and wavelet transforms only. It defines a framework for designing wavelets that satisfy different applications.

When discrete wavelets are used to transform a continuous signal, functions of the form shown in (13) are selected [4].

$$\Psi_{j,k}(t) = \frac{1}{\sqrt{s_0^j}}\Psi(\frac{t - k\tau_0 s_0^j}{s_0^j}), \tag{13}$$

which is normally a piecewise continuous function, where j and k are integers and $s_0 > 1$ is a fixed dilation step. The translation factor τ_0 depends on the dilation step. The effect of discretizing the wavelet is that the time-scale space is now sampled at discrete intervals. We usually choose $s_0 = 2$ so that the sampling of the frequency axis corresponds to dyadic sampling as shown in Figure 13. This is a very natural choice for computers, the human ear and music for instance. For the translation factor, it is usual to choose $\tau_0 = 1$ so that there is also a dyadic sampling of the time axis.

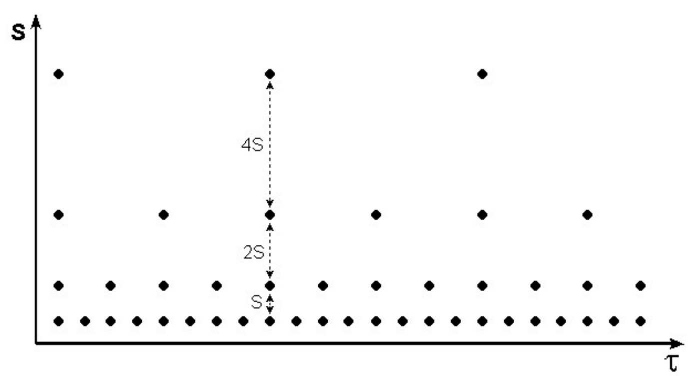


Figure 13: Localization of discrete wavelets in the time-scale space on a dyadic grid [73].

Practical applications require Discrete Wavelet Transforms (DWT). The discrete wavelets can be made orthogonal to their own dilations and translations by special choices of the mother wavelet. There is a large class of wavelet functions for which the set of child wavelets is an orthogonal basis. The simplest of these is the Haar wavelet. An arbitrary signal can be reconstructed by summing the orthogonal wavelet basis functions weighted by wavelet transform coefficients.

The DWT and Inverse DWT (IDWT) of a signal $f(t)$ are defined in (14) and (15), respectively.

$$\gamma_{j,k} = \int_{-\infty}^{\infty} f(t)\Psi_{j,k}(t)dt. \tag{14}$$

$$f(t) = \sum_j \sum_k \gamma_{j,k}\Psi_{j,k}(t), \tag{15}$$

Such wavelets give rise to a Wavelet Multiresolution Analysis (MRA) derived as follows.

Define W_j to be a set of all signals $f(t)$ which can be synthesized from the child wavelets $\Psi_{j,k}(t)$, $-\infty < k < \infty$. These spaces are orthogonal to each other and we can synthesize any signal $f(t)$ using (16)

$$\begin{aligned} f(t) &= \sum_{j=-\infty}^{\infty} f_j(t), \\ f_j(t) &= \sum_{k=-\infty}^{\infty} \gamma_{j,k}\Psi_{j,k}(t), \end{aligned} \tag{16}$$

where $f_j(t)$ is in the space W_j.

There is another way to express this idea. Define V_j to be the set of all signals, $f(t)$, which can be synthesized from the child wavelets $\Psi_{i,k}(t)$ where $i < j$ and $-\infty < k < \infty$ as in (17)

$$f(t) = \sum_{i=-\infty}^{j-1} \sum_k \gamma_{i,k}\Psi_{i,k}(t). \tag{17}$$

The spaces V_j are nested inside each other, as follows:

$$\{0\} \subset \ldots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \ldots \subset L^2. \tag{18}$$

As j goes to ∞, V_j enlarges to become all energy signals (L^2). As j goes to $-\infty$, V_j shrinks down to only the zero signal. It is clear from the definitions that every signal in V_{j+1} is the sum of a signal in V_j and W_j because

$$f(t) = \sum_{i=-\infty}^{j} \sum_k \gamma_{i,k} \Psi_{i,k}(t) = \sum_{i=-\infty}^{j-1} \sum_k \gamma_{i,k} \Psi_{i,k}(t) + \sum_k \gamma_{j,k} \Psi_{j,k}(t). \tag{19}$$

Hence, it can be written:

$$V_{j+1} = V_j + W_j. \tag{20}$$

This shows that the spaces W_j are the differences (in the subspace sense) between adjacent spaces V_{j+1} and V_j. The spaces V_j and W_j can be visualized as shown in Figure 14.

The term *Wavelet Multiresolution Analysis* (MRA) refers to the analysis of signals in relation to a nested sequence of subspaces like the one shown in Figure 14. For example, to decompose a signal, $f(t)$, in space V_0 a few times, use the following decomposition:

$$\begin{aligned} V_0 &= V_{-1} + W_{-1} \\ &= V_{-2} + W_{-2} + W_{-1} \\ &= V_{-3} + W_{-3} + W_{-2} + W_{-1} \\ &= V_{-4} + W_{-4} + W_{-3} + W_{-2} + W_{-1}. \end{aligned} \tag{21}$$

This leads to various decompositions:

$$
\begin{aligned}
f(t) &= A_1(t) + D_1(t) \\
&= A_2(t) + D_2(t) + D_1(t) \\
&= A_3(t) + D_3(t) + D_2(t) + D_1(t) \\
&= A_4(t) + D_4(t) + D_3(t) + D_2(t) + D_1(t),
\end{aligned}
\tag{22}
$$

where $D_i(t)$, in W_{-i}, is called the detail at level i and $A_i(t)$, in V_{-i}, is called the approximation at level i.

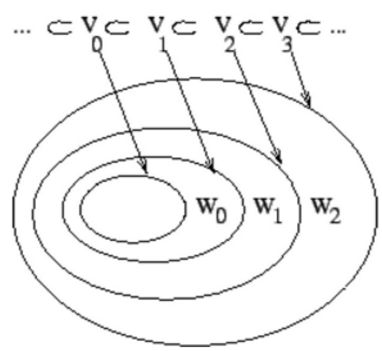
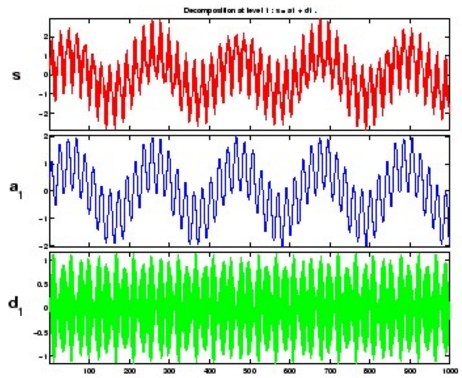


Figure 14: MRA: nested subspaces.

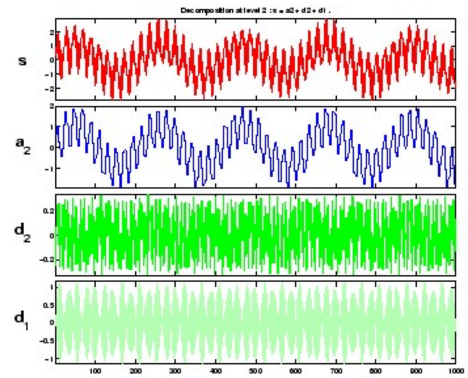
Figure 15 gives an example of how the decomposition can be carried out in Matlab^{TM} using the `wavemenu` interface. There are a number of sample signals, which can be used for a demonstration analysis. The signal `sumsin` is the sum of two sine waves, and is decomposed four times in this example.

Notice that different aspects of the signal appear at different levels of the details and approximations in Figure 15.

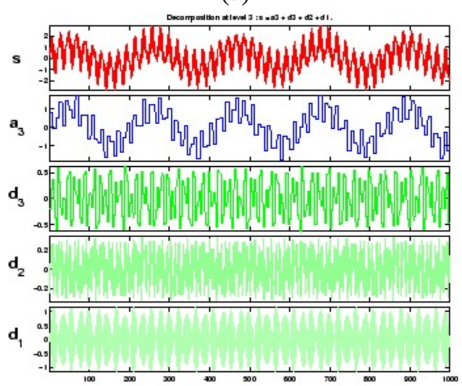
The space V_j has a very important property related to time compression by factors of 2. The MRA Two Scale Property asserts that a signal $f(t)$ is in the space V_j if and only if,



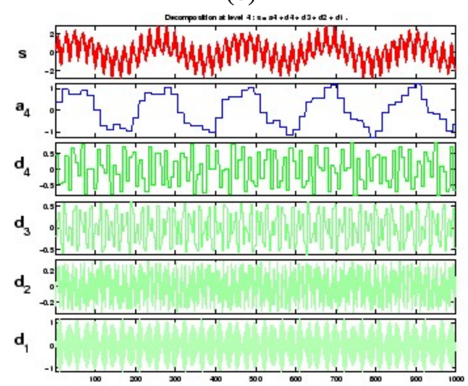
(a)



(b)



(c)



(d)

Figure 15: An example of Wavelet Multiresolution Analysis (MRA) decomposition.

$f(2t)$ is in the next space V_{j+1}. Therefore, investigation of the multiresolution analysis leads to a scaling function, a pair of discrete time filters, and a perfect reconstruction filter bank, which can be used to calculate the DWT quickly. In other words, a wavelet has a band-pass like spectrum. Given that compression in time is equivalent to stretching the spectrum and shifting it upwards, a time compression of the wavelet by a factor of 2 will stretch the frequency spectrum of the wavelet by a factor of 2 and also shift all frequency components up by a factor of 2. Using this insight, the finite spectrum of a signal can then be covered with the spectra of dilated wavelets in the same way that the signal is covered in the time domain with translated wavelets. Alternatively, if one wavelet can be seen as a band-pass filter, then a series of dilated wavelets can be seen as a band-pass filter bank.

The filter bank can be built in several ways. One way is to build many bandpass filters to split the spectrum into frequency bands. Another way is to split the signal spectrum into two (equal) parts, a lowpass and a highpass part. The low-pass part can be split into a lowpass and a highpass part again. This splitting process continues until the details of a signal that has been exposed are satisfied. In this way, an iterated filter bank is created as shown in Figure 16.

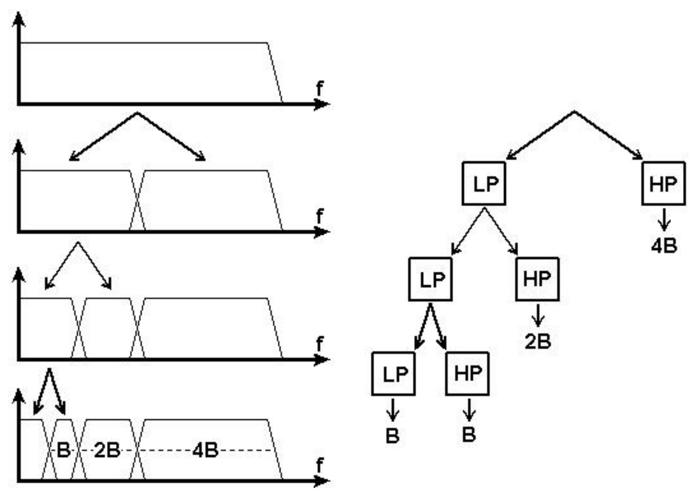


Figure 16: Splitting the signal spectrum with an iterated filter bank [73].

Four mother wavelets often used in wavelet analysis are shown in Figure 17. The difference between these wavelets is mainly due to the different lengths of filters that define the wavelet and scaling functions [36].

The scaled (dilated) and translated (shifted) versions of the Daubechies mother wavelet are shown in Figure 18. Daubechies wavelets belong to a special class of mother wavelets and are actually used most often for detection, localization, identification and classification of power disturbances.

Transient signals in a power system are non-stationary, time-varying voltage and current signals. Wavelet transforms are feasible to provide efficient and localized analysis of non-stationary, fast transient fault signals for power systems. More detailed discussion on the application of wavelets analysis for classification of fault signals for power systems will be addressed in Section 4.2.

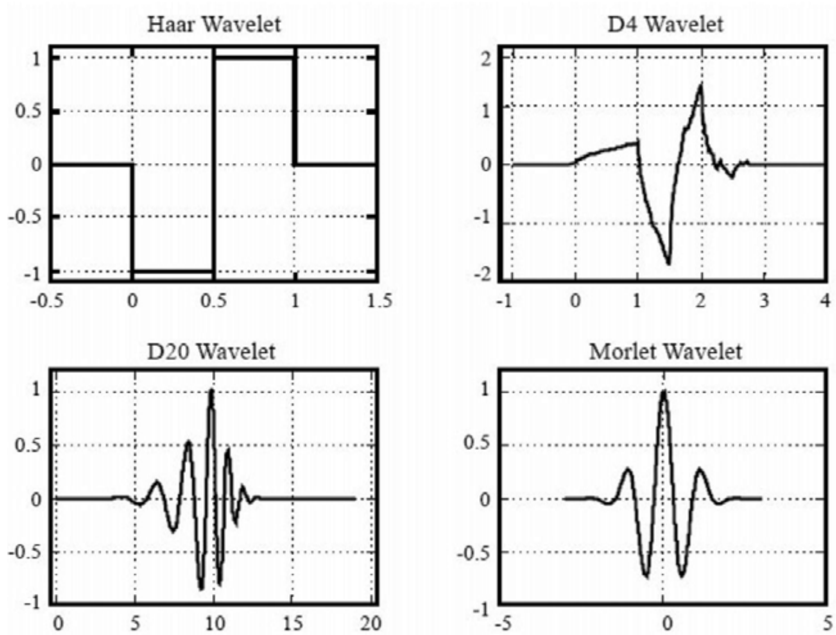


Figure 17: Four mother wavelets often used in wavelet analysis [36].

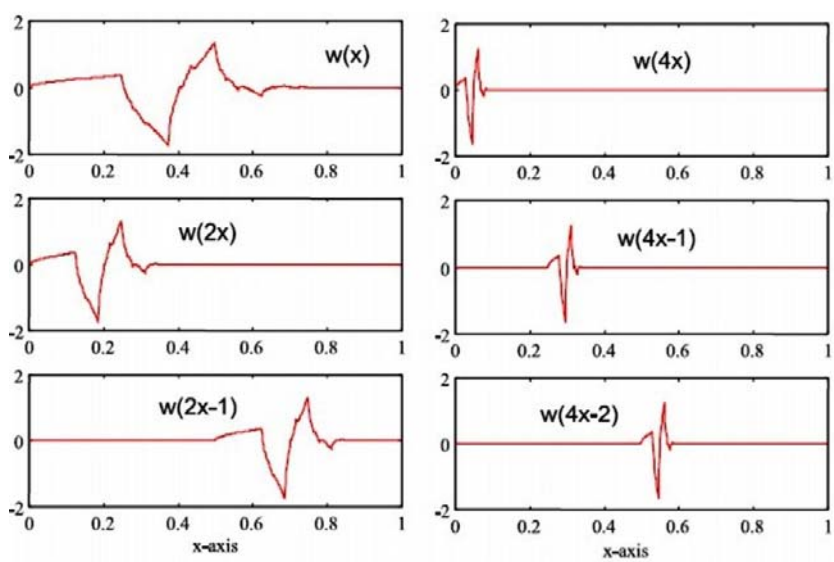


Figure 18: Scaled and translated versions of the D4 wavelet [36].

3.4 Time-Frequency Representation (TFR) Theory

In addition to applying wavelet theory to power system fault classification, the Time-Frequency Representation (TFR) algorithm is becoming attractive to scientists and engineers in the power industry. This section will introduce the basics of TFR theory, and Section 4.3 will present the TFR in classifying power system faults.

TFR $P(t, f)$ can be expressed as a two-dimensional Fourier transform of the product of the ambiguity plane $A(\eta, \tau)$ of the signal and a kernel function $\varphi(\eta, \tau)$ [11]:

$$P(t, f) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} A(\eta, \tau)\varphi(\eta, \tau)e^{j2\pi\eta t}e^{-j2\pi f\tau}d\eta d\tau, \tag{23}$$

where t represents time, f represents frequency, η represents continuous frequency shift, and τ represents continuous time lag. The ambiguity plane $A(\eta, \tau)$ for a given signal $s(t)$ is defined as:

$$A(\eta, \tau) = \int_{-\infty}^{\infty} s(t)s^*(t + \tau)e^{j2\pi\eta t}dt, \tag{24}$$

where $s(t)$ represents the signal at time t, and $s(t+\tau)$ represents the signal at a future time $t + \tau$, and the $s^*(t + \tau)$ means the complex conjugate of $s(t + \tau)$.

The kernel $\varphi_i[\eta, \tau]$ is defined as a binary matrix (each matrix element is either 0 or 1). Feature points are ambiguity plane points of locations (η, τ) where $\varphi_i[\eta, \tau]$ = 1.

3.5 Rough Set Theory

This section briefly presents the basic rough set approach to the approximation of sets [56], which provides a foundation for classifying power system fault signals. The rough set approach introduced by Zdzisław Pawlak [41, 47, 48, 49, 50, 51, 52] and elaborated by others [32, 42, 43, 44, 53, 56, 62, 67] provides the grounds for approximating a set X. Let B denote a set of functions that represent the features (traditionally, also called attributes in rough set theory [50]) of objects in a set U. The basic approach in rough set theory is to use an equivalence relation $\sim_B$ [57]

$$\sim_B = \{(x, x') \in U \times U \mid \forall f \in B,\ f(x) = f(x')\},$$

to define the partition of a set U into non-empty, pairwise disjoint subsets (equivalence classes). An equivalence class in a partition is denoted by $[x]_B$, where

$$[x]_B = \{x' \in U \mid \forall f \in B, f(x) = f(x')\}.$$

The equivalence classes in a partition form a new set, denoted by $U/\sim_B$, where

$$U/\sim_B = \{[x]_B \mid x \in U\},$$

for a given set of objects U. Let $X \subseteq U$ be a set of objects of interest. After the partition of the set U has been defined, the lower and upper approximations of the set X are defined relative to the equivalence classes in the partition.

3.5.1 Preliminaries

The notation and terminology in Table 3 is important for an understanding of basic rough set theory. Let $U, \mathcal{F}$ denote a set of sample objects and a set of functions, respectively. The functions in $\mathcal{F}$ represent the features (attributes) of the objects in U. Assume that $B \subseteq \mathcal{F}$, the notation (U, B) denotes an information system, which is usually represented in table form.

Table 3: Rough Set Theory Symbols.

Symbol	Interpretation
U	Set of sample objects,
$\mathcal{F}$	Set of functions representing object features,
B	$B \subseteq \mathcal{F}$,
X	$X \subseteq U$,
x	$x \in X$,
$\sim_B$	$\sim_B = \{(x, x') \in U \times U \mid \forall x \in U, f(x) = f(x')\}$,
$[x]_B$	$[x]_B = \{x' \in U \vert x' \sim_B x\}$,
$U/\sim_B$	$U/\sim_B = \{[x]_B \mid x \in U\}$, a partition of U,
B_*X	$\bigcup_{x:[x]_B \subseteq X}[x]_B$, B-lower approximation of X,
B^*X	$\bigcup_{x:[x]_B \cap X \neq \emptyset}[x]_B$, B-upper approximation of X,
$Bnd_B X$	$Bnd_B X = B^*X \setminus B_*X = \{x \mid x \in B^*X$ and $x \notin B_*X\}$.

In keeping with current notation for equivalence relations, $\sim$ denotes an equivalence relation on a set U [16]. The $\sim$ symbol is used extensively to express equivalence [5, 14, 16].

The notation $U/\sim$ denotes a partition of U. Let $[x]$ denote a class belonging to $U/\sim$, where

$$[x] = \{x' \in U \mid x \sim x'\}.$$

The classes of a partition are disjoint, ***i.e.***, if $[x], [y] \in U/\sim$, then $[x] \cap [y] = \emptyset$. In addition, every object in U is in only one class in $U/\sim$.

The use of $\sim_B$ drew attention to the role of the set B in partitioning a set U. The basic idea here is that the relation $\sim_B$ provides a classification of objects according to knowledge contained in the system (U, B) [33].

The class $[x]_B$ is called a B-elementary set [47, 50]. If $(x, x') \in \sim_B$ (also written $x \sim_B x'$), then x and x' are said to be *indiscernible* with respect to all functions in B, or simply, B-*indiscernible*. In the case where $B = \{f\}$, $\sim_{\{f\}}$ denotes an equivalence relation defined relative to a set of feature f and $[x]_{\{f\}}$ denotes an equivalence class in $U/\sim_{\{f\}}$. For simplicity, write $\sim_f$ to denote $\sim_{\{f\}}$.

A sample $X \subseteq U$ can be approximated from information contained in B by constructing a B-lower approximation

$$B_*X = \bigcup_{x:[x]_B \subseteq X} [x]_B,$$

and a B-upper approximation

$$B^*X = \bigcup_{x:[x]_B \cap X \neq \emptyset} [x]_B.$$

The B-lower approximation B_*X is a collection of classes of sample elements that can be classified with full certainty as members of X. By contrast, the B-upper approximation B^*X is a collection of classes representing both certain and possibly uncertain knowledge about X because it is possible for B^*X to have one or more classes that are not subsets of X. An approximation boundary Bnd_BX is defined by

$$Bnd_BX = B^*X \setminus B_*X = \{x \mid x \in B^*X \text{ and } x \notin B_*X\}.$$

The set Bnd_BX contains all objects in the upper approximation B^*X that are not in the lower approximation B_*X. Whenever $B_*X \subsetneq B^*X$, the sample X has been classified imperfectly, and is considered a rough set. In other words, a set X is a rough set, if and only if, the boundary Bnd_BX is not empty.

3.5.2 Information Tables

For computational reasons, a syntactic representation of information systems is usually given in the form of tables. Discovering objects in the composition of a class $[x]_B \subseteq U/\sim_B, x \in U$ in the partition $U/\sim_B$ in the system $(U, \mathcal{F})$ is accomplished by gathering together inside the class all of those objects that have matching function values. Identifying the classes in $U/\sim_B$ is greatly aided by a table representation of $(U, \mathcal{F})$.

3.5.3 Decision Systems

Table 4: Decision system notation.

Symbol	Interpretation
d	Decision function,
U	Set of sample objects,
$\mathcal{F}$	Set of functions representing features,
$(U, \mathcal{F}, d)$	Decision system.

Of particular interest is the extension of information systems made possible by including a function d representing what is known as a decision attribute in rough set theory. A decision is defined by a function $d : X \longrightarrow V_d$, where V_d is the range of d. In addition, $(U, \mathcal{F}, d)$ denotes a decision system. It is typical in rough set theory to start with an information system $(U, \mathcal{F})$ and introduce a decision function d as a means of separating sample objects in U into decision classes, *i.e.*, sets of objects representing a particular value of d. Decision systems are also represented by tables.

3.5.4 Rough Membership Function

Because it is important to determine the extent to which a set of sample signals match a class of signals representing a particular power system fault, the rough membership function defined by (25) has been used in this research. The degree of overlap between X and $[x]_B$ containing x can be quantified with the rough membership function (rmf),

$$\mu_X^B : U \to [0,1] \;\; defined\ by \;\; \mu_X^B(x) = \frac{|[x]_B \cap X|}{[x]_B}. \tag{25}$$

The rough membership function has proven to be very useful in measuring the extent that classes of signals for known faults overlap with sets of signals representing power system faults to be classified. This is explained in detail in Section 7, where the rmf is used in the design of a neural network useful in classifying power system faults.

3.6 Artificial Neural Network (ANN) Theory

This section briefly presents the architecture for a convention neural network.

3.6.1 ANN Basic Structures

A neural network is a collection of neurons (or simply nodes) arranged in one or more layers with links that facilitate communication between the neurons [29]. Each node receives the weighted sum of the signals coming either from general input to the network or from other nodes and propagates the weighted output to other nodes as shown in Figure 19.

A node consists of two basic parts, namely, the summing function and the activation function. The weighted input to a node are integrated with a summing function. The resulting sum then provides input to a dedicated activation function that is a source of output for the node. Many activation functions are step change, ramp, sigmoid and Gaussian functions.

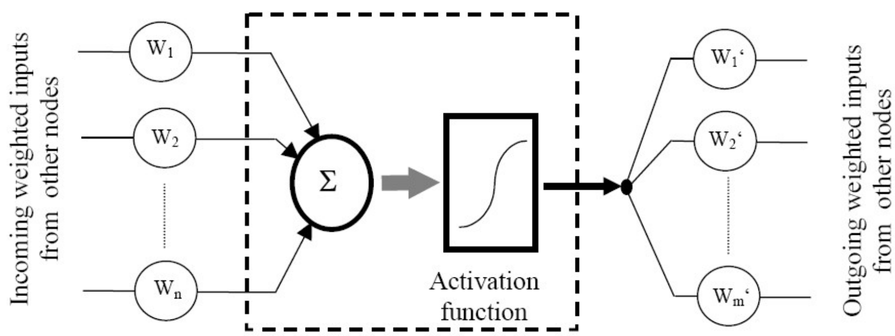


Figure 19: A single node NN basic construction [29].

Node output is usually weighted. The values of the weights are determined via a training process [22]. ANN nodes can be connected in different ways according to the selected ANN architecture. One of the most common ANN structures is realized by arranging the nodes in sequential layers. The first layer is called the input layer. Intermediate layers are called hidden layers. The final layer of nodes is called the output layer. The output of each node is distributed to other nodes in the network. The simplest form of ANN is the multi-layer feedforward network (MLF), where node output from either input layer or a hidden layer is transmitted to the nodes in next layer (see Figure 20(a)). In a radial basis network (RBF), there are three layers, input layer, specially designed hidden layer and output layer (see Figure 20(b)). In a recurrent network, node output not only is transmitted forward to the next layer but also back to connection units connected to nodes in the previous layer (see Figure 20(c)).

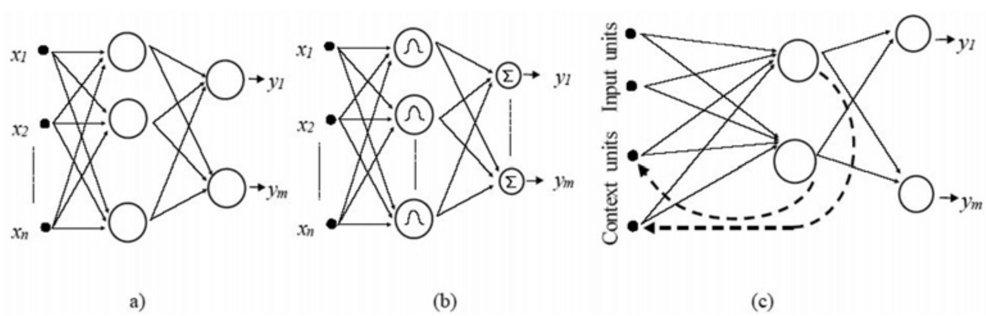


Figure 20: Selected ANN architectures: (a)-MLF-network (b)-RBF-network (c)-Elman network.

The MLF network is the most common type of network because of its easy development, efficient training and direct implementation. For MLF networks, the signals are distributed from the input layers toward the output in one direction.

The RBF network is a special form of feed-forward network utilizing radial basis functions in its hidden layer rather than using sigmoid activation as in normal MLF networks.

The radial basis function ensures each of these hidden neurons has a symmetrical response around its selected center vector. Each of the output nodes accumulates the weighted sum of all hidden neurons. The weights connecting both layers are adjusted through the training process using the Least Square optimization. This simple topological structure and fast training paradigm provide a powerful tool for various applications including function approximation and classification purposes [9, 38].

Recurrent networks, on the other hand, employ one or more feedback output results from a certain layer to the foregoing one(s). Unlike the normal MLF networks, recurrent networks can distinguish temporal patterns due to the memory action of the context input. A popular recurrent network is the Elman network. The special feature of this network is that one or more feedback output results are extracted from the hidden neurons and fed to the input layers [15, 25].

3.6.2 ANN Design

The design of an artificial neural network generally consists of the following four procedures [29]:

1. Feature Extraction.

   The aim of this step is to find out the more relevant variables that can be nominated as the selected input to the desired network. Those variables should have a distinguishable configuration for the required task along the entire range of the dimensional space so that they can help to perform the required convergence through training. This set of input should be selected carefully in order to realize the minimum sufficient number of input training sets, since these redundant input training sets may burden optimization procedure resulting in a weak or non-stable ANN. Also, the output of the neural network is selected according to the required task and its aimed decisions.

2. ANN Preliminary Construction.

   The preliminary configuration of an ANN is arbitrarily begun with a predefined set of input and output. The most critical problem in constructing the ANN is to choose the number of hidden neurons. Using too few hidden neurons may prevent the training process to converge, while using too many hidden neurons may extend the training time and cause overfitting. Thus, it is convenient to start the design with a small set of hidden nodes. Unfortunately, this tends to be a trial-and-error process. The number of hidden neurons can be slightly adjusted upward and the network is retrained in an attempt to improve neural classification with a known set of input training sets. This process is repeated until the lowest acceptable training error is reached.

3. Data Preparation and Training Process.

   Training the ANN can be generally described as a certain mathematical optimization process that leads to an optimum set of network parameters. At this stage, a sufficient number of training cases should be collected covering the entire range of the dimensional space of the required task. The majority of ANN-based applications use supervised training methods, where the output from the collected training patterns are employed to regulate the conversion through the training process. The Delta Rule was introduced in 1960 as the first practical attempt for training the ANN Perceptron (simple ANN structure without hidden layers). However the Perceptron has limited capabilities dealing only with those linearly separable problems. In 1986, Back Propagation (BP) training was introduced for MLF networks. This algorithm has been nominated as the first practical and most common training method for separable or non-separable applications [18, 40].

   The basic BP training algorithm was then enhanced with different mathematical improvements. First of all, the momentum term was added during the convergence process. Momentum makes it possible to perturb the weight changes resulting from the normal BP calculations to avoid falling into a local minimum. Also, considering adaptive learning rates can accelerate the training process. Other figures for training paradigms were developed as well depending upon different mathematical criteria such as the Conjugate Gradient (CG) and the Levenberg Marquardt (LM). Both algorithms employed numerical optimization techniques in order to optimize the initial set of network parameters. The Elman network, as a special MLF network, can be trained similarly with the aforementioned training paradigms. For RBF networks, Orthogonal Least Square (OLS) learning was employed in order to get a compact network structure.

4. Testing of the Trained ANN.

   The final step in the design process is to evaluate the resulting networks thoroughly via a well prepared group of test cases. These test cases should differ from those used cases in the training stage. Also the test cases should cover all circumstances that may affect the system performance for the entire range of the dimensional space.

3.7 Classifier Fusion Theory

Classifier combination has received considerable attention in the past decade and is now an established pattern recognition offspring. It has been recognized for some time that the classical approach to designing a pattern recognition system, which focuses on finding the best classifier has a serious drawback. Any complementary discriminatory information that other classifiers may encapsulate is not tapped. Multiple expert fusion aims to make use of many different designs to improve classification performance. Over the last few years, a myriad of methods for fusing the output of multiple classifiers have been proposed.

Let $\mathcal{D} = \{D_1, D_2, ..., D_L\}$ be a set of classifiers and $\Re^n$ be the feature space. All classifiers produce soft class labels. We assume that $d_{j,i}(x) \in [0, 1]$ is an estimate of the degree of set c_i offered by classifier D_j for an input $x \in \Re^n, i = 1, 2; j = 1, ..., L$. There are two possible classes $C = \{c_1, c_2\}$ and L classifiers $\mathcal{D} = \{D_1, D_2, ..., D_L\}$ [34]. Simple fusion methods are the most obvious choice when constructing a multiple classifier system [30, 35, 68, 69, 2], *i.e.*, the support for class c_i, $d_i(x)$, yielded by the set of classifiers is [34]

$$d_i(x) = \mathcal{F}(d_{1,i}(x), ..., d_{L,i}(x)), i = 1, 2, \tag{26}$$

where $\mathcal{F}$ is the chosen fusion method. Here, it is necessary to study the fusion methods compared in [1]:

- minimum
- maximum
- average
- median
- majority vote
- oracle

For the majority vote, the first step is to harden the individual decisions by assigning class labels $D_j(x) = c_1$ if $d_{j,1}(x) > 0.5$, and $D_j(x) = c_2$ if $d_{j,1}(x) \leq 0.5$, $j = 1, ..., L$. Next, the class label most represented among the L (label) output is chosen.

The oracle model is an abstract fusion model. In this model, if at least one of the classifiers produces the correct class label, then the team produces the correct class label too. Usually, Oracle is used in comparative experiments.

In order to achieve a high overall performance of the classification function, the performance of each individual classifier has to be optimized prior to using it within any fusion schemes. That is, the fusion scheme will be able to improve the overall classification result relative to the performance of the individual, optimized classifiers. If several classifiers with only marginal performance are being used, the results cannot necessarily be expected to reach the high performance sought. On the other hand, if several classifiers are used that work exceptionally well, any further gains will be exceedingly hard to accomplish because the opportunity for diversity will be diminished. Individual classifier optimization can be performed by selecting object features, appropriate parameters, and classifier structure that governs the performance.

After designing a classifier fusion scheme, a confusion matrix ***M*** can be generated for each classifier using labeled training data [20]. The confusion matrix lists the true classes c versus the estimated classes $\widehat{c}$. Because all classes are enumerated, it is possible to obtain information not only about correctly classified states (N^{00} and N^{11}), but also about false positives (N^{01}) and false negatives (N^{10}). A typical two-class confusion matrix ***M*** is shown in Figure 21.

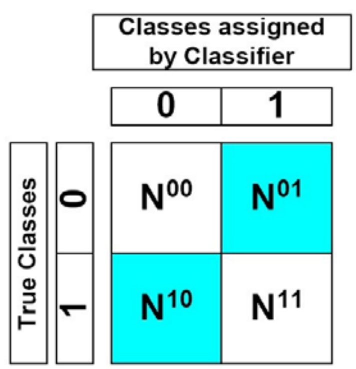


Figure 21: Typical 2-class confusion matrix [20].

From the confusion matrix of each classifier, the false positive (FP) error, the false negative (FN) error, the total error rate (TER), and the total success rate (TSR) can be calculated for the classifier. These error rates are defined as in (27) to (30). The total error rate (TER) or the total success rate (TSR) is typically used as a simple measure for overall performance of a classifier.

$$FP = \frac{N^{01}}{N^{00} + N^{01}}. \tag{27}$$

$$FN = \frac{N^{10}}{N^{10} + N^{11}}. \tag{28}$$

$$TER = \frac{N^{01} + N^{10}}{N^{00} + N^{11} + N^{01} + N^{10}}. \tag{29}$$

$$TSR = 1 - TER. \tag{30}$$

Although each individual classifier's performance is very important to the performance of a classifier fusion, the dependency between the classifiers to be fusioned also affects the fusion results. Some studies [61] have shown that the degree of correlation between the classifiers adversely affects the performance of the subsequent classifier fusion. If two classifiers agree everywhere, the fusion of the two classifiers will not achieve any accuracy improvement no matter what fusion method is used. For classifier fusion design, classifier correlation analysis is, therefore, equally as important as the classifier performance analysis. Based on the classifier output on the labeled training data, a 2x2 matrix ***N*** as shown in Figure 22 can be generated for each classifier pair. The off-diagonal numbers directly indicate the correlation degree of the two classifiers. The smaller the two off-diagonal numbers are, the higher the correlation between the two classifiers will be. The proportion of specific agreement, which here is called the correlation, ρ_2, is defined in [61] as

$$\rho_2 = \frac{2 \times N^{FF}}{N^{TF} + N^{FT} + 2 \times N^{FF}}, \tag{31}$$

where, as further shown in Figure 22, N^{TT} implies that both classifiers classified correctly; N^{FF} means both classifiers classified incorrectly; N^{TF} represents the case of the 1st classifier classified correctly and the 2nd classifier classified incorrectly; and N^{FT} stands for the 2nd classifier classified correctly and the 1st classifier classified incorrectly. In order for classifier fusion to be effective in performance improvement, the correlation, ρ_2, has to be small (low correlation).

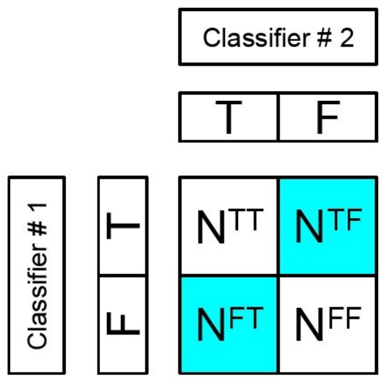


Figure 22: Correlation analysis matrix [20].

Consider the output of two classifiers as enumerated in Table 5. The calculation of ρ_2 yields $\rho_2 = 0.36$. Had classifier 2 been completely redundant to classifier 1, the correlation would have been $\rho_2 = 1$.

The 2-class correlation coefficient can be extended to *n* different classifiers [20]. The notion that redundancy is described by the individual true and false answers of the classifiers is retained from the 2 class correlation analysis. The larger the ρ-correlation, the larger the redundancy. In particular, the ρ-correlation goes to zero if the individual incorrect answers are disjoint for all answers. That implies that there is always at least one correct answer from some classifier for any case available. The ρ-correlation coef-

Table 5: Results from experiment for 2 classifiers [20].

Answer classifier 1	Answer classifier 2
T	T
T	F
F	T
T	F
F	F
F	F
T	F
F	T
T	T
T	T
T	T
T	T
T	F
T	T
T	T
F	T

ficient gets larger as the number of wrong answers are the same for many answers. Let N^f be the number of experiments where all classifiers give a wrong answer; N_i^c be the number of experiments with combinations of correct and incorrect answers; c is the combination of correct and incorrect answers (for 2 classifiers: $c \in \{wr, rw\}$; for 3 classifiers: $c \in \{wwr, wrw, rww, wrr, rwr, rrw\}$ *etc.*); n is the number of classifiers. The ρ-correlation coefficient is then [20]

$$\rho_n = \frac{nN^f}{\sum_{i=1}^{2^n-2} N_i^c + nN^f}. \tag{32}$$

If N is the number of experiments and N^t is the number of experiments for which all classifiers had a right answer, (32) can more conveniently be rewritten as [20]

$$\rho_n = \frac{nN^f}{N - N^f - N^t + nN^f}. \tag{33}$$

Consider a 3-classifier example, which is the same as the previous 2-classifier example except that a third classifier was added that will get answer wrong in 50% of the cases. The calculation of ρ_n yields: $\rho_n = 0.21$.

Although the newly added classifier has poor performance, its addition reduces the overall redundancy of the classifier assembly.

Note that the ρ-correlation does not record redundancy for any particular classifier (for $n > 2$) but for a set of classifiers only. For illustrative purposes, consider two simplistic cases shown in Table 6 and Table 7 [20].

Table 6: Output for 3 classifiers (case 1) [20].

Answer classifier 1	Answer classifier 2	Answer classifier 3
T	F	F
F	T	F
F	T	T
T	T	T
F	F	F

The ρ-correlation is $\rho_n = 0.5$.

Table 7: Output for 3 classifiers (case 2) [20].

Answer classifier 1	Answer classifier 2	Answer classifier 3
T	F	T
F	T	T
F	T	F
T	T	T
F	F	F

The ρ-correlation is $\rho_n = 0.5$.

Obviously the third classifier is different in the two example cases above. However, the degree of correlation is the same because it does not matter whether it is correlated to the first or to the second classifier. Rather, it is only relevant that it is correlated to the combination of the first two classifiers. Note that the calculation of the ρ-correlation factor can be performed on multi-class scenarios as well because the factor is only concerned with the correctness of the outcome.

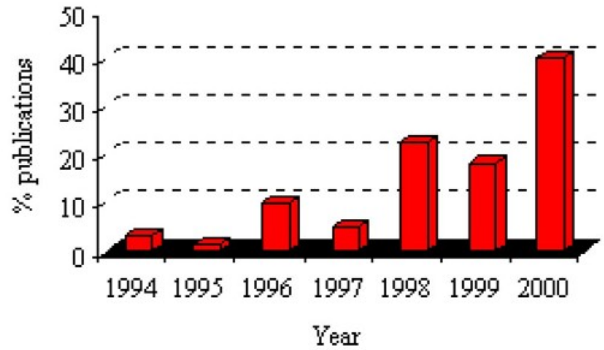
4 Technology Review of Power System Fault Classification (PSFC)

Nomenclature

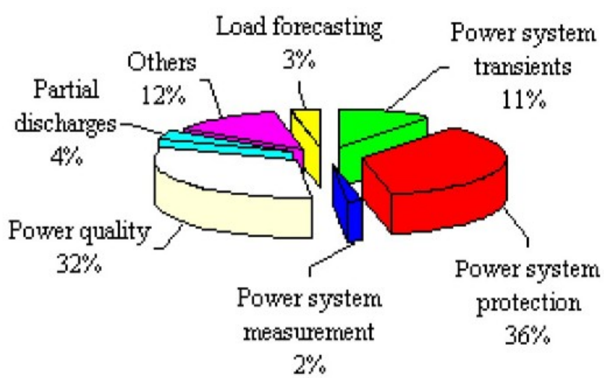
Symbol	Brief Explanation
$J_{Fi}(\eta, \tau)$	Fisher's discriminant score
k_i	In the single-class separator case, kernel k_i is dedicated to discriminate class i from all remaining classes $i+1, ..., N$. In the group-class separator case, kernel k_i is dedicated to discriminate a class group $i, i+1, ..., i+m$ from all remaining classes $i+m+1, i+m+2, ..., i+m+N$
PSFC	Power System Fault Classification
MRA	(Wavelet) Multi-Resolution Analysis
ANN	Artificial Neural Network
DWT	Discrete Wavelet Transform
PNN	Probability Neural Network
PQ	Power Quality
MRD	(Wavelet) Multi-Resolution Decomposition
FDF	Fisher's Discriminant Function

4.1 Wavelet Applications in Power Systems

The main difficulty in dealing with power engineering phenomena is the extreme variability of the signals and the necessity to operate on a case–by–case basis. Another aspect of power disturbance signals is often localized temporally or spatially (*e.g.*, transients in power systems). This requires the efficient use of analysis methods, which are versatile enough to handle signals in terms of their time-frequency localization. Wavelets localize the information in a time-frequency plane. In particular, wavelets are capable of trading one type of resolution for another, which makes them especially suitable for the analysis of non-stationary signals. Considerable work has been done in applying the wavelet transform to power systems in analyzing and processing the voltage-current signals to make a real-time identification of transients in a fast and accurate way [17].



(a) Evolution of wavelet publications in power systems.



(b) Percentage of wavelet publications in different power system areas.

Figure 23: Overview of wavelet applications in power systems [17].

The wavelet transform was first applied to power systems in 1994 by Robertson [64] and Rebeiro [63]. Since then, the number of publications in this area has rapidly increased as Figure 23(a) shows. Figure 23(b) illustrates the most popular wavelet transform applications in power systems:

- Power system protection
- Power quality

- Power system transients
- Partial discharges
- Load forecasting
- Power system measurement

The field of power system transients is the area in which wavelets were first applied to power system applications by Robertson [64]. In this paper, the authors presented a methodology for the development of software for classifying power system disturbances by type from the transient waveform signature. Transients are signals with a finite life, *i.e.*, a transient reduces to zero in a finite time. Electromagnetic transients are caused by sudden changes in system topology or parameters. For instance, short circuit faults are one of the most common causes of transients in a power system. Power system switching causes transients as well. Robertson [65] distinguished single-phase faults from capacitor switching using waveform signatures.

An example of transient analysis using wavelets was given by Ramaswamy [66]. Using the Electromagnetic Transient Package provided in the Power System Simulation Software, MIPOWER, and the wavelet transform toolbox provided in MATLAB *Ver. 5.3*, the authors analyzed a group of simulated transients namely the phase BC-Ground fault, three phase-Ground fault and phase C-Ground fault, in a simple power system network (Figure 24) consisting of a generator, a load, two buses and a transmission line. Figure 25 shows a typical waveform of a certain type of transient disturbance in power systems.

The authors applied different types of wavelets to the transient disturbance signal to perform Multiple Level Decomposition. The Meyer wavelet (Figure 26) was found to work better as the fundamental source signal was restored at the 4^{th} approximation. Other wavelets such as a ‘Haar’ wavelet, added noise to the fundamental wave. The transients

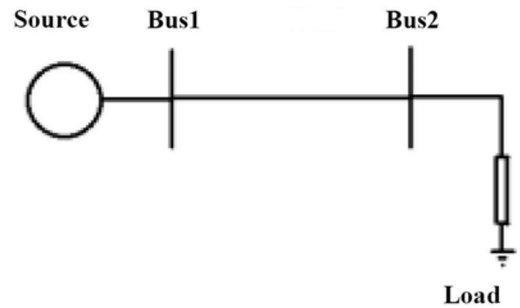


Figure 24: A typical power system network [66].

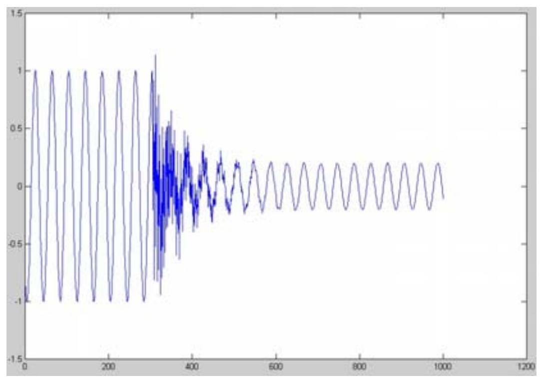


Figure 25: Example of transient disturbance for certain types of faults indistinguishable by the naked eye [66].

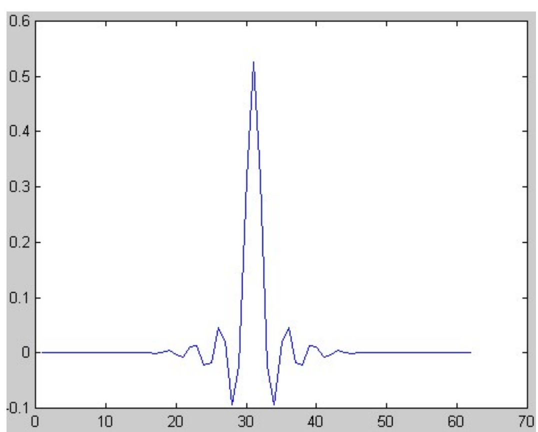


Figure 26: A typical Meyer wavelet [66].

were analyzed by the 'Meyer' mother wavelet and Figure 27 shows Multiple Level Decomposition of the transient disturbance, where s is the source signal, a_4 is the 4^{th} level approximation, d_4 is the 4^{th} level detail coefficient, d_3 is the 3^{rd} level detail coefficient, d_2 is the 2^{nd} level detail coefficient, and d_1 is the 1^{st} level detail coefficient.

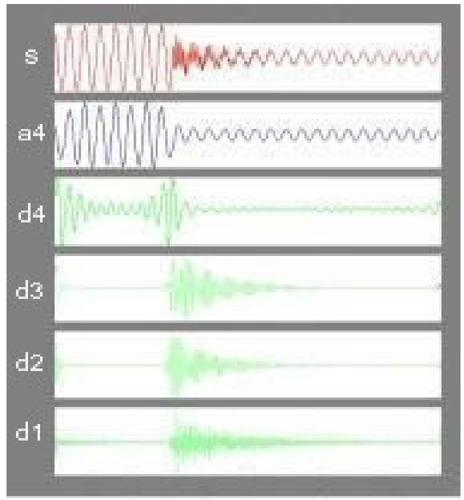
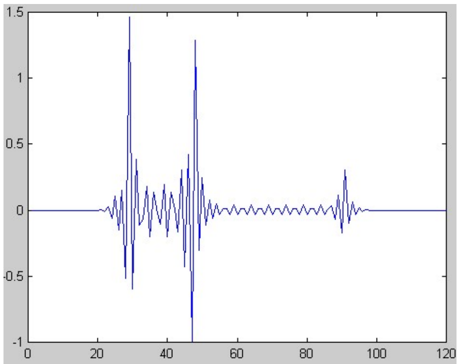


Figure 27: Multiple level decomposition of a transient disturbance [66].

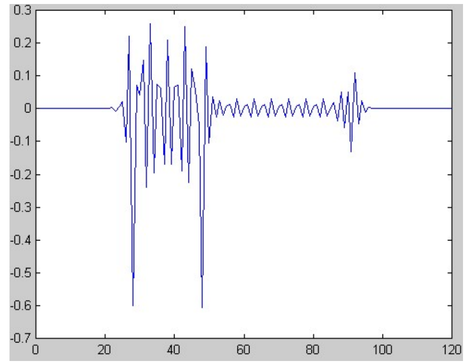
The detail coefficients of faults are given in Figure 28 for the phase BC-Ground fault, three phase-Ground fault and phase C-Ground fault.

In power quality applications, several studies have been carried out to detect and locate disturbances using the wavelet transform to analyze interference, impulses, notches, glitches, interruptions, harmonics, flicker, etc. of non-stationary signals. Drisen [12] analyzed power system harmonics while Santoso [70] analyzed power system interference.

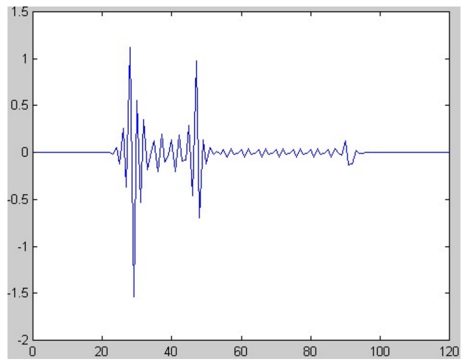
In power system protection applications, the potential benefits of applying the wavelet transform to improve the performance of protection relays and fault classification have been recognized in recent years. Charri [7] analyzed the transient information of a resonant grounded distribution system using the wavelet transform. Imriš [26] presented the analysis of ground fault transients in high voltage networks for earth fault location purposes using the Gaussian mother wavelet method and discussed the main sources of error affecting the



(a) Phase BC-ground fault.



(b) Three phase-ground fault.



(c) Phase C-ground fault.

Figure 28: The detail coefficient of faults [66].

accuracy of the method. Liang [37] proposed an algorithm for fault classification based on Wavelet Multiresolution Analysis (MRA) with Daubechies four (D-4) wavelet measuring and comparing sharp variation in the values of the currents for the three phases in the first stage MRA detail signals extracted from the original signal. Cheng [10] used a B-Spline wavelet transform for fault classification purposes based on threshold values as in [37]. Zhao [77] proposed an algorithm with Daubechies eight (D-8) wavelet for fault detection and classification in an underground cable system using two different levels of MRA detail signals. Chanda [8] presented an algorithm for classification of faults based on MRA with Daubechies eight (D-8) wavelet transforms of the three phase currents on a transmission line fed from both ends.

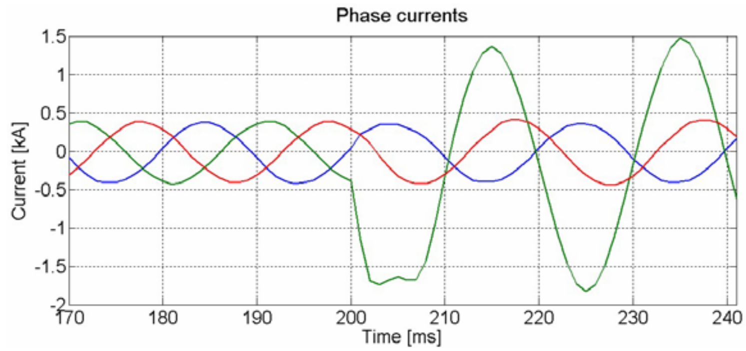


Figure 29: The recorded single phase to ground fault: Phase currents [26].

Imriš [26] and Chanda [8] were both using wavelets for data preprocessing before applying the fault location and classification algorithms to the recorded transients on transmission lines. Imriš analyzed ground fault transients in 110kV networks using low frequency records for fault location purposes. As shown in Figure 29, ground fault signals consist of different frequency components, which result from charging or discharging of the network capacitances. The charge transient is generated by the voltage rise in sound phases during a single-phase to ground fault. This means that a charge transient is always a side effect of the ground fault. Moreover, it is typically of strong amplitude and, therefore, is reasonable to use for single-phase to ground fault location. The fault transients are mixed with the other signals as noise and fundamental frequency components. Sometimes the transient can be short in duration and also small in amplitude. Moreover, the transient can be very close to the fundamental frequency signal in the frequency domain. Therefore, the 50Hz component can negatively affect the fault transient frequency estimation. To enable a more precise analysis of the fault transient, preprocessing is performed with a wavelet filter [26].

The filtering of the signal is performed using a wavelet filter to get the fault transient precisely out of the measured signal. The wavelet filter is set exactly on the frequency of the measured (charge) fault transient estimated by the Fourier transform. The filter's coefficient and its frequency response with an example fault current are shown in Figure 30.

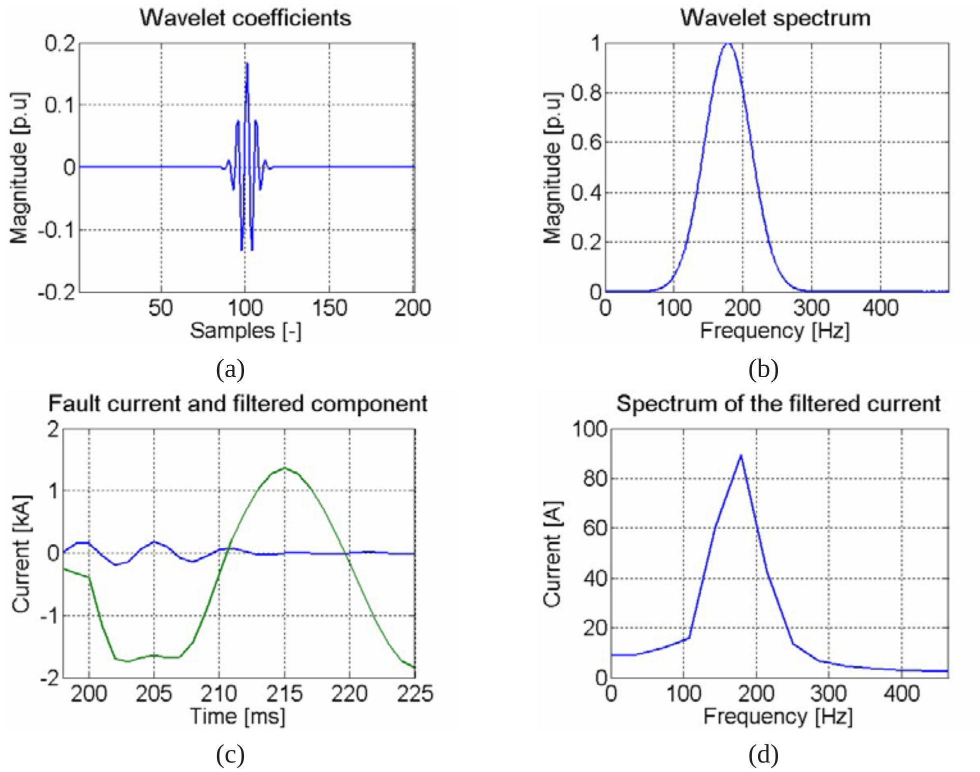


Figure 30: Pre-processing of the fault signals using wavelet [26].

The filter coefficients are represented by a Gaussian mother wavelet. After removing the 50Hz component, the charge transient frequency is detected. In the case of the phase currents shown in Figure 29, the charge transient frequency is detected at 178.57Hz. These transients can then be used for fault location if they are detected. Transient fault location is based on the estimation of the fault path inductance L_f from the detected fault transients. The fault path inductance can be calculated directly from the filtered signal (the charge transient) [26],

$$L_f = \frac{1}{\omega_c} Im \left[\frac{v_c(t,f)}{i_c(t,f)} \right] = \frac{1}{3}(L_0 + L_1 + L_2) \cdot l_f, \tag{34}$$

where ω_c, v_c and i_c are the angular frequency, voltage and current of the charge transient. The fault distance is l_f. The constants L_0, L_1 and L_2 are the zero-, positive- and negative-sequence inductances of the faulty line per km. In (34), t represents time and f the frequency.

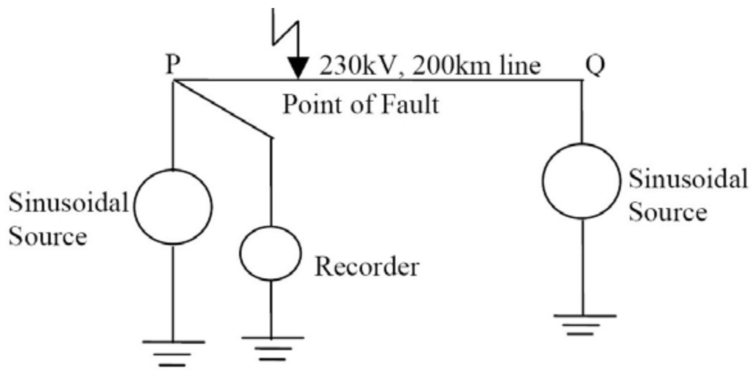


Figure 31: 230kV, 200km transmission line system used for simulation studies [8].

Chanda, on the other hand, simulated the application of Wavelet MRA theory for the classification of faults on a power transmission line as shown in Figure 31. The base values of the voltage and power in the system are taken as 230kV and 100MVA. The frequency of the system is taken to be 50Hz. The phase current signals are recorded at the two ends (P, Q). The generated time domain signals are sampled every 80μs and then used for the analysis using wavelet transform. The data considered in the analysis is assumed to be of finite duration and of length 2^N, where N is an integer. If N is chosen to be 9, the total duration of the analysis comes to 2^9 (=512) $\times$ 80μs = 40.96ms, which is about two cycles and is sufficient for the fault analysis. With N = 9, there are (N + 1) = 9 + 1 = 10 wavelet levels. If these 10 levels are added together, then the original signal is faithfully reproduced at each of the sample points.

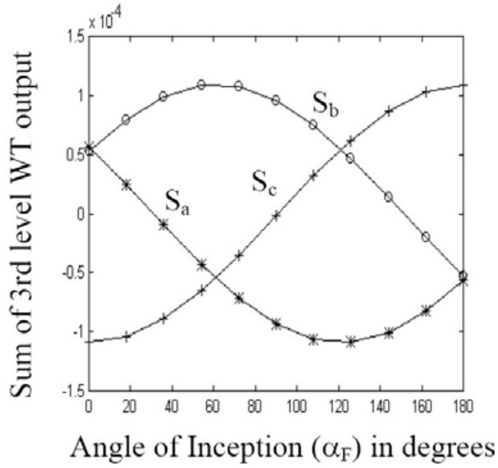
Daubechies Eight (D-8) wavelet is used in this work for the analysis, since it closely matches the signal to be processed (this is of the utmost importance in wavelet applications). Due to the unique feature of providing multiple resolution in both time and frequency by wavelets, the sub-band information can be extracted from the original signal.

When applied to faults, this sub-band information of a faulted power system is seen to provide useful signatures for faults. By randomly shifting the point of fault on the transmission line, a number of simulations can be carried out. The generated time domain signal for each case is analyzed using the wavelet transform. From the different decomposed levels, only 3^{rd} level output is considered for the analysis.

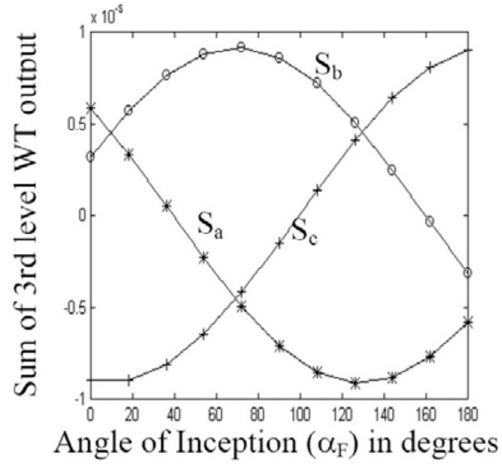
The types of faults considered in the analysis are L-G, L-L-G, L-L, L-L-L. The simulations show that the fault inception angle (α_F) has a considerable effect on the phase current samples and, therefore, also on the wavelet transform output of post-fault signals. Through exhaustive experimentation, the authors have concluded that the parameter identified for classification is the summation of 3^{rd} level output for the three phase currents. The results are shown in Figure 32 and Figure 33, where, S_a = Summation of 3^{rd} level values for current in phase 'a', S_b = Summation of 3^{rd} level values for current in phase 'b', and S_c = Summation of 3^{rd} level values for current in phase 'c'.

If $S_a + S_b + S_c \cong 0$, then the fault is classified as an L-L-L fault, in which the magnitude of all the summation values, S_a, S_b and S_c are comparable to each other. This can be verified from the simulation results shown in Figure 32(a) (an L-L-L fault at 5km) and Figure 32(b) (an L-L-L fault at 195km).

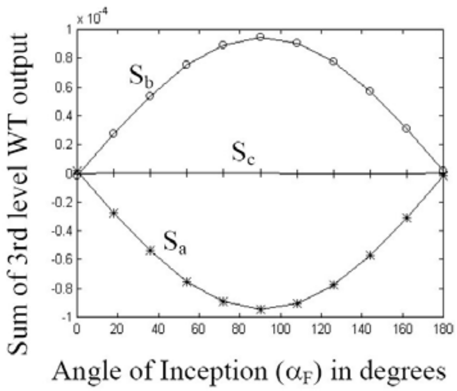
If $S_a + S_b + S_c \cong 0$ and also if the sum of two of the summations S_a, S_b and S_c is equal to zero, *i.e.*, the magnitude of one of the summations is very small and almost negligible in comparison to the equal magnitudes of other two summations, then the fault is classified as an L-L fault, *i.e.*, if $S_a + S_b = 0$, it is a fault involving the a and b phase; $S_a + S_c = 0$, it is a fault involving the a and c phase; and $S_b + S_c = 0$, it is a fault involving the b and c phase. The results of classifying an L-L fault involving the a and b phase are shown in Figure 32(c) (an L-L fault at 5km) and Figure 32(d) (an L-L fault at 195km).



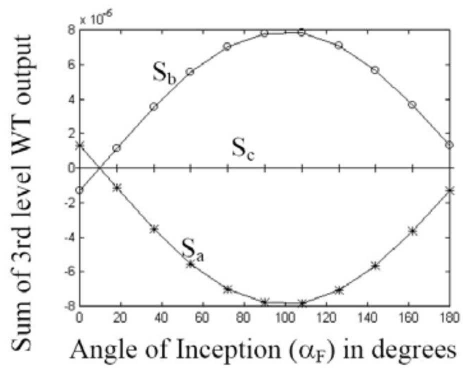
(a) Effect of Inception angle (α_F) for L-L-L Fault at 5km.



(b) Effect of Inception angle (α_F) for L-L-L Fault at 195km.



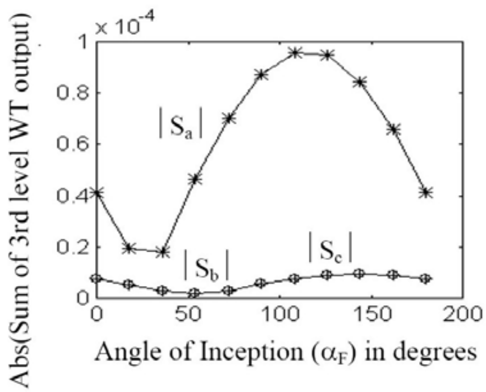
(c) Effect of inception angle (α_F) for L-L fault involving phases ‘a’, ‘b’ at 5km.



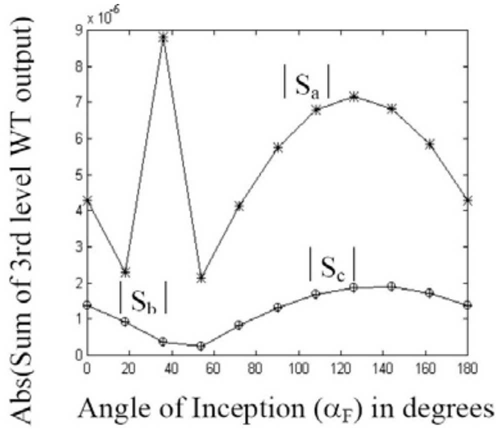
(d) Effect of inception angle (α_F) for L-L fault involving phases ‘a’, ‘b’ at 195km.

Figure 32: Preprocessing of the L-L and L-L-L fault signals using wavelet [8].

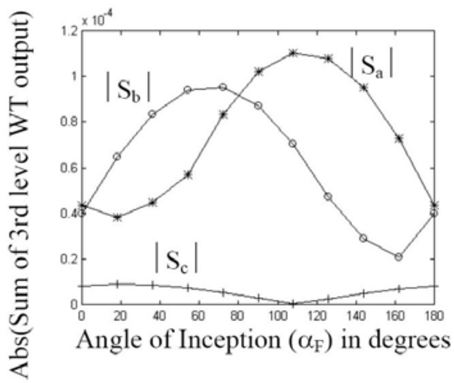
If $S_a + S_b + S_c \neq 0$, then it is either an L-G or L-L-G fault. If the absolute value of any two summations (S_a, S_b, S_c) is equal and is always much smaller than the absolute value of the 3rd summation, then it is an L-G fault. If $|S_b| = |S_c|$ & $<<$ $|S_a|$, it is an L-G fault involving phase a; if $|S_a| = |S_c|$ & $<<$ $|S_b|$, it is an L-G fault involving phase b; and if $|S_a| = |S_b|$ & $<<$ $|S_c|$, it is an L-G fault involving phase c. The results of classifying an L-G fault involving the a phase are shown in Figure 33(a) (an L-G fault involving the a phase at 5km) and Figure 33(b) (an L-G fault involving the a phase at 195km).



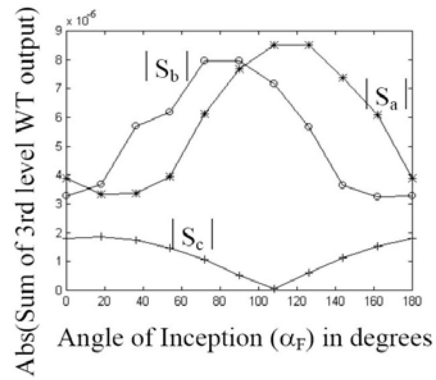
(a) Effect of inception angle (α_F) for L-G fault involving phase 'a' and ground at 5km.



(b) Effect of inception angle (α_F) for L-G fault involving phase 'a' and ground at 195km.



(c) Effect of inception angle (α_F) for L-L-G fault involving phases 'a', 'b' and ground at 5km.



(d) Effect of inception angle (α_F) for L-L-G fault involving phases 'a', 'b' and ground at 195km.

Figure 33: Preprocessing of the L-G and L-L-G fault signals using wavelet [8].

If the absolute value of any two summations (S_a, S_b, S_c) is not equal and is always much higher than the absolute value of the 3rd summation, then it is an L-L-G fault as shown in Figure 33(c) and (d). Furthermore, provided that $S_{min} = min(|S_a|, |S_b|, |S_c|)$, if $S_{min} = |S_c|$ and $<<$ $|S_a|$ or $|S_b|$, then it is an L-L-G fault involving phases a, b and ground; if $S_{min} = |S_b|$ and $<<$ $|S_a|$ or $|S_c|$, then it is an L-L-G fault involving phases a, c and ground; and if $S_{min} = |S_a|$ and $<<$ $|S_b|$ or $|S_c|$, then it is an L-L-G fault involving phases b, c and ground.

4.2 Combination of the Wavelet and Neural Network Techniques for Fault Detection

Recently, research has been focused more on combining the wavelet and neural network algorithms for fault identification in power systems. Wavelet analysis is applied to analyze transient signals, then a neural network algorithm is utilized for the identification of problems.

Ramaswamy [66] and Kashyap [28] proposed a method that incorporates a Probabilistic Neural Network (PNN) for detecting the type of power system fault. The PNN has preference over other Artificial Neural Network (ANN) algorithms in the application of power system fault classification. It combines the merits of statistical theory with that of ANN. Figure 34 shows the entire procedure for fault recognition.

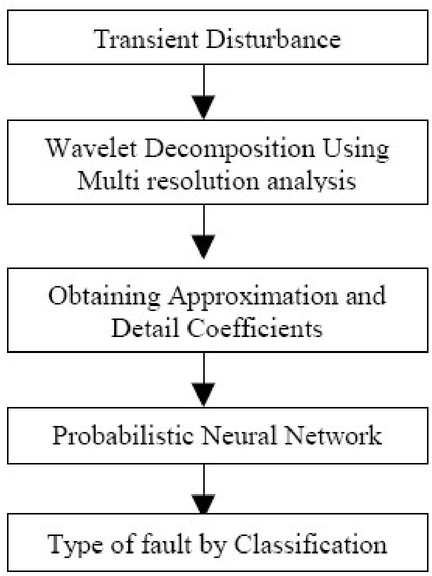


Figure 34: Procedure for fault detection and classification [66].

Three power system faults, *i.e.*, phase A-Ground fault, double phase AB-Ground fault and 3-phase symmetrical fault are simulated and investigated. Transients are analyzed by the Meyer mother wavelet, and Multiple Level Decomposition of the transient disturbance was generated. The final level detail coefficient is considered for the feature detection and used in the Probabilistic Neural Network.

Figure 35 shows the model of a Probabilistic Neural Network, which classifies these three power faults [28]. The PNN Architecture consists of four layers, *i.e.*, the Input Layer: consisting 119 Neurons, number of samples of the detail coefficient; the Exemplar Layer: consisting of 9 Neurons, 3 faults $\times$ 3 sets of data for each fault; the Summation Layer: consisting of 3 Neurons, equal to the number of faults; and the Decision Layer: follows the "Winner take all" mechanism.

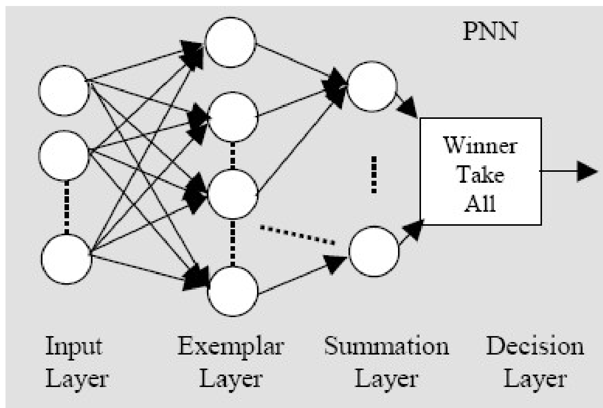


Figure 35: Model of a Probabilistic Neural Network. Detail coefficient is fed to the input layer and the type of fault is obtained at the output [28].

Researchers also proposed solutions for digital relays for transmission line protection. Martin has simulated a system with two generators and three lines (distributed parameters model) [39]. Simulations include 3 different faults at different distances from the beginning of each line, several fault resistances, inception angles, and steady states. The process consists of a preprocessing module based on Discrete Wavelet Transform (DWT) combined with an ANN for detecting and classifying fault events.

Wavelets of length six (N=6) are used for the relay to operate in real time. These wavelets can be expressed as functions of two parameters α and β [4]. By varying parameters α and β, a family of length-6 wavelets can be generated. For a certain range of variation of these parameters, the generated wavelets are classified according to their performance for this particular application. The parameters for the length-6 wavelet with quasi-optimal performance are $\alpha = 0.48\pi$ and $\beta = -0.35\pi$.

Three independent multilayer (two hidden-layers), feed-forward neural networks have been used for detection, classification and location of fault transients. The ANNs are fed with the six detail signals (three currents and three voltages). The input data of the ANN is organized in a sliding-window of a quarter of a cycle, thus a faster response is obtained since only a quarter of a cycle from the occurrence of the fault is required. The input vector has 24 elements. The detection ANN has one output neuron, which indicates the existence of a fault. The location net has one neuron that indicates if the fault has occurred in the protected zone. The classification ANN output layer has four neurons indicating which phases (A, B, C) or ground are involved in the fault event. An error back-propagation algorithm has been used for training the ANN.

4.3 Time-Frequency Representation Technique for Classifying Power Quality Disturbances

Voltage disturbances are the most frequent cause of a broad range of disruption in power supply systems. Power quality (PQ) disturbances cover a broad frequency range and significantly different magnitude variations. Typically, there are five major PQ related waveform events: harmonics, voltage sags, capacitor high frequency switching, capacitor low frequency switching, and normal voltage variations. Harmonics distortion is the most common power quality problem [13].

Approaches for automated detection and classification of PQ disturbances proposed recently are based on wavelet analysis and artificial neural networks [19, 54, 71]. To enhance the sufficiency for supporting a robust PQ monitoring system is one of the most interesting research areas for scientists.

A wavelet transform on a PQ signal produces a multiresolution decomposition (MRD) matrix, which contains time domain information for the signal at different scales. This property has made wavelets a promising tool for detecting and extracting disturbance features for various types of PQ events [19, 54, 71]. However, there are still some issues to be resolved in wavelet-based methods. First, while PQ disturbances cover a wide frequency range, a very small subset of the MRD matrix (*e.g.*, five scales in [71]) may not be a sufficient or optimized selection for capturing features for all different types of PQ events. This feature selection scheme may filter some important information for classification and potentially degrade the recognition rates. Second, the wavelet-based methods relatively require more training examples. They result in greater efforts or difficulties when adapting the algorithm onto a new system.

Wang and Mamishev had been investigating a feature extraction tool, time-frequency ambiguity plane with kernel techniques [74, 75], which is new to the power engineering field. The essence of the feature extraction is to project a PQ signal onto a low-dimension time-frequency representation (TFR), which is deliberately designed for maximizing the separability between classes. A distinct TFR is designed for each class. The classifiers include a Heaviside-function linear classifier and neural networks with feedforward structures.

A set of 860 real world voltage signals from five event classes were collected from industrial databases for the training and testing of the algorithm. Each voltage signal to be identified consists of five cycles of a voltage waveform sampled 128 times per cycle, and has a length of 640 sampling points. In the training stage, four classification-optimal kernels are designed for separating five classes sequentially. The kernel design process selects nine locations from the time-frequency ambiguity plane.

Classification kernels are designed for training according to Fisher's discriminant function. Fisher's discriminant function (FDF), which was developed by R. A. Fisher in the

1930s, is a method that projects high dimensional data onto low-dimensional space for classification. The projection maximizes the distances between the means of the different classes while minimizing the variances within each class.

The kernel $\varphi_i[\eta, \tau]$ is defined as a binary matrix (each matrix element is either 0 or 1). Feature points are ambiguity plane points of locations (η, τ) where $\varphi_i[\eta, \tau] = 1$. Therefore, the process of feature extraction is to select points that are optimal for the classification task from the ambiguity plane.

A total number of $N - 1$ kernels need to be designed for an N-class PQ classification system. A kernel K_s works as either a single-class separator or a group-class separator. In the single-class separator case, kernel K_i is dedicated to discriminate class i from all remaining classes $\{i + 1, ..., N\}$. In the group-class separator case, kernel K_i is dedicated to discriminate a class group $\{i, i+1, ..., i+m\}$ from all remaining classes $\{i+m+1, i+m+2, ..., i+m+N\}$. In the second case, additional kernels are needed in order to uniquely identify class i from the class group $\{i, i + 1, ..., i + m\}$, and the total number of kernels required for an N-class classification is still N-1.

Ambiguity planes for all training signals are calculated before the Fisher's discriminant function is applied for the kernel design. Assume there are n classes and totally N_i training examples for class i. The notation $A_{ij}[\eta, \tau]$ represents the ambiguity plane of the j^{th} training example in the i^{th} class.

With the Fisher's criterion, locations on the ambiguity plane are ranked according to their importance for classification. A certain amount of training data from each class is needed for feature ranking in this statistical method. For example, when designing kernel i, a Fisher's discriminant score is calculated for each location (η, τ) on the ambiguity plane,

$$J_{Fi}(\eta, \tau) = \frac{(m_i[\eta, \tau] - m_{i-remain}[\eta, \tau])^2}{D_i^2[\eta, \tau] - D_{i-remain}^2[\eta, \tau]}, \tag{35}$$

where $m_i[\eta, \tau]$ and $m_{i-remain}[\eta, \tau]$ represent two means of location (η, τ),

$$m_i[\eta, \tau] = \frac{1}{N_i} \sum_{j=1}^{N_i} A_{ij}[\eta, \tau], \tag{36}$$

$$m_{i-remain}[\eta, \tau] = \frac{\sum_{k=i+1}^{5} \sum_{j=1}^{N_k} A_{kj}[\eta, \tau]}{\sum_{k=i+1}^{5} N_k}, \tag{37}$$

and $D_i^2[\eta, \tau]$ and $D_{i-remain}^2[\eta, \tau]$ represent two variances of location (η, τ),

$$D_i^2[\eta, \tau] = \frac{1}{N_i} \sum_{j=1}^{N_i} (A_{ij}[\eta, \tau] - m_i[\eta, \tau])^2, \tag{38}$$

$$D_{i-remain}^2[\eta, \tau] = \frac{\sum_{k=i+1}^{5} \sum_{j=1}^{N_k} (A_{kj}[\eta, \tau] - m_{i-remain}[\eta, \tau])^2}{\sum_{k=i+1}^{5} N_k}. \tag{39}$$

Locations (η, τ) that receive the highest discriminant score $J_{Fi}(\eta, \tau)$ are selected as feature locations.

By examining Fisher's discriminant score $J_{Fi}(\eta, \tau)$, the optimal numbers of feature points for each individual kernel have been found: one for the harmonics kernel; two for the voltage sag kernel; three for the capacitor switching kernel; and three for the capacitor high-frequency switching kernel. Therefore, nine feature locations are selected for these four kernels.

Each classification node consists of a kernel function and a classifier. A Heaviside-function linear classifier is used for the task of separating harmonics that is a great distance apart from other fault cases and is relatively easy to discriminate. Neural networks with small numbers of input nodes are used for all other classification tasks. The structure of the ANN for discriminating sags is 2-12-2 (input layer node number - hidden layer node number - output layer node number); the one for capacitor switching is 3-10-2; and the one for capacitor high-frequency switching is 3-10-2. The transfer and training functions adopted for the ANN include: the hyperbolic tangent sigmoid transfer function as the transfer function for the hidden layer, the linear transfer function as the transfer function for the output layer, backpropagation as the network training function, the gradient descent learning function as the weight learning function, and the mean squared error function as the performance evaluation function.

5 Data Preparation for Manitoba Hydro HVDC PSFC

Nomenclature

Symbol	Brief Explanation
PSFC	Power System Fault Classification
ASCII	American Standard Code for Information Interchange

Prior to feature extraction, data preparation and signal preprocessing are required to define the characteristics of power system signals. The fault data from the Transcan^{TM} is in binary format and non-editable. Data preparation consists of two steps. The first step is to convert the data from binary format to ASCII (American Standard Code for Information Interchange) format. The second step is to separate the signals into different groups according to their physical nature (*i.e.*, Pole voltages/currents, 3 AC phase voltages, valve control signals, valve currents).

5.1 Data Conversion

The data recorded by Transcan^{TM} is in binary format and compressed as *.x01 files [72], which are unreadable by humans. Together with the .x01 files, Transcan^{TM} provides *.scf files. The *.scf file is a configuration file and contains the information for data arrangement. It tells how many channels have been scanned. At the Manitoba Hydro Dorsey Station, a fault file has 48 analog and 4 digital channels, with some of them being spares. The *.scf file indicates the scanning order and the physical name for each channel. The first 52 bytes in a .x01 file are used for recording the file name and date. Every 16 bits that follow are allocated for storing one channel data. In each 16-bit data field, the first 12 bits store one digitalized data for a channel and the last 4 bits indicate the channel number.

For this research, a C++ program has been designed to convert the data to ASCII format (*.dat) from binary format (.x01). Each .x01 file can be converted into 48 *.dat files. Among these files, 23 files are selected to represent the most active and informative signals in the power system for fault classification.

5.2 Signal Grouping

Among the 23 converted signals, some are constant signals and the others are periodic signals. Bus signals, *i.e.*, the 3 AC phase voltages and the pole voltages and currents should be grouped separately from the valve signals. Bus signals will induce more than one fault and usually cause significant problems. Valve signals will affect only one valve group and cause a certain level of decrease or increase of either the pole voltage or current. Table 8 lists the signal groups for a pole 1 file. The number of signal groups will guide the number of the feature sets to be extracted.

Table 8: Signal groups.

group 1 (3 signals)	AC voltage A phase, B phase, C phase
group 2 (5 signals)	Pole 1 and 2 voltages, pole 1 and 2 currents, pole current order
group 3 (3 signals)	Vg11 current A phase, B phase, C phase (first valve group in pole 1)
group 4 (3 signals)	Vg12 current A phase, B phase, C phase (second valve group in pole 1)
group 5 (3 signals)	Vg13 current A phase, B phase, C phase (third valve group in pole 1)
group 6 (3 signals)	6-pulse in 3 valve groups
group 7 (3 signals)	Start pulse in 3 valve groups

6 Signal Preprocessing and Feature Extraction for PSFC

Nomenclature

Symbol	Brief Explanation
V_g11	Valve group 1 in pole 1
V_g12	Valve group 2 in pole 1
V_g13	Valve group 3 in pole 1
V_g21	Valve group 1 in pole 2
V_g22	Valve group 2 in pole 2
V_g23	Valve group 3 in pole 2
$\phi_j(x)$	The granule functions
$r(j)$	Normalized valve current reference signal
$x(j)$	Normalized valve current signal
ρ_0	The autocorrelation of the valve reference signal $r(j)$ at origin point
ρ_0'	The maximum of the cross-correlation of the valve reference signal and every 96-points segment of the input valve current signal $vc(j)$
rmNN	Rough Membership Neural Network

To set up the information table for fault classification, the normal behavior of each signal needs to be clarified and the abnormality of each signal related to each type of fault can then be identified. Signal preprocessing and feature extraction is presented in this section.

6.1 Signal Characteristics in Normal Condition

Standard value or waveform of each signal in normal condition is described in the following two tables.

6.1.1 Constant Signals

In the 23 signals converted from .x01 file, the constant signals are pole-current order, alpha order, pole current, pole voltage. Under normal conditions, their standard values are given in Table 9:

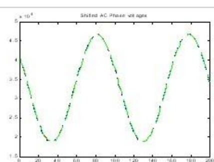
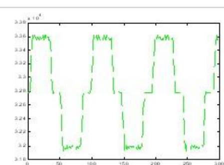
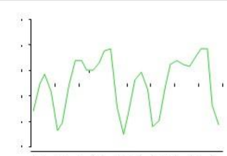
Table 9: Constant signals in the 23 converted signals.

Pole Current Order	Alpha Order	Pole Current	Pole Line Voltage
±1400 amps	150 degrees	±1400 amps	±450 KV, ±300 KV, ±150 KV

6.1.2 Periodic Signals

The periodic signals are AC Phase Voltages, Phase Currents and 6-pulse Voltages. Their normal waveform and standard peak values are shown in Table 10.

Table 10: Periodic signals in the 23 converted signals.

	AC Phase Voltage	Phase Current	6-Pulse Voltage
Amplitude (peak to peak)	200 KV	1400 Amps	27 KV
Waveform			

6.2 Feature Extraction of 12 Types of Faults

Extensive time has been spent in studying 676 fault files provided by the Manitoba Hydro Dorsey Station. The 676 .x01 fault files recorded all the events that happened in two recent years and covered 12 types of faults. Together with .x01 files, 676 .trt files are also provided. A .trt file contains the fault information, *i.e.*, the fault cause and type. This fault information is created manually by power system engineers and provides a reliable basis for the target for PSFC training and testing. Various signal processing techniques are applied to analyze the fault signals. They are auto-correlation, cross-correlation, the FFT and inverse FFT, low pass filter, Wavelet MRD, phase shifting, derivatives and coding techniques. A total of 17 features or attributes in Table 11 are generated for power system fault classification. The 17 functions that represent these 17 features (attributes) are further described in the section 6.2.

Table 11: 17 features/attributes for power system fault classification.

Attr1	Pole voltage sharp dropping
Attr2	AC voltage disturbance severity
Attr3	Pole index
Attr4	Pole 1 or 3 voltage trend
Attr5	Pole 2 or 4 voltage trend
Attr6	Pole 1 or 3 current trend
Attr7	Pole 2 or 4 current trend
Attr8	Valve current trend - valve group 1, vg*1
Attr9	Valve current trend - valve group 2, vg *2
Attr10	Valve current trend - valve group 3, vg *3
Attr11	Valve current minor disturbance
Attr12	Pole current closed with normal pole voltage
Attr13	3 valve groups all closed (True = 1, False = 0)
Attr14	Same current trend in 3 valve groups (True = 1, False = 0)
Attr15	Voltage flashover in 6-pulse signal
Attr16	Valve currents flashover
Attr17	Valve currents flashover happens only in one valve group (True = 1, False = 0)
Notation: * represents the pole index, i.e. for pole 1, the valve groups are Vg11, Vg12, Vg13	

A portion of the information table for power system fault classification training is shown in Table 12. This information table is derived from 508 fault files and consists of 508 lines in total, with each line containing 17 features. This table is further processed to prepare for the training sets to calibrate the rough membership Neural Network (rmNN) for fault classification. Also, a portion of the information table for testing is illustrated in Table 13. The testing table consists of 168 rows generated from an additional 168 fault files. The complete training and testing information tables are attached in appendix A.

Table 12: Partial information table for power system fault classification training.

File index	Fault types	Fault file names	Attr1	Attr2	Attr3	Attr4	Attr5	Attr6	Attr7	Attr8	Attr9	Attr10	Attr11	Attr12	Attr13	Attr14	Attr15	Attr16	Attr17
1	Fault 1	F08101FE.x01	0	1	1	4	43	23	23	2	2	2	0	0	0	1	0	0	0
4	Fault 1	F0813032.x01	0	1	1	4	1	3	1	2	2	2	0	0	0	1	0	0	0
7	Fault 1	F1113009.x01	0	1	1	4	41	23	21	2	2	2	0	0	0	1	0	0	0
31	Fault 1	F224045A.x01	0	1	4	414	4	313	3	2	2	2	0	0	0	1	0	0	0
55	Fault 1, 3	F0420695.x01	0	1	2	3	43	34	34	2	2	21	0	0	0	0	0	0	0
69	Fault 1, 3	F2913FDD.x01	0	1	1	3	4	2	2	2	2	1	0	0	0	0	0	0	0
70	Fault 1, 3, 5	F04102CE.x01	0	1	1	43	4	3	3	21	2	2	11	0	0	0	1	0	0
91	Fault 1, 3, 5, 7	F082016A.x01	0	1	2	4	43	2	2	2	21	212	100	0	0	0	2	1	1
92	Fault 1, 3, 5, 7	F2720494.x01	0	1	2	3	34	2	2	2	2	1212	10	0	0	0	2	1	1
93	Fault 1, 3, 5, 11	F0822697.x01	0	1	2	4	43	3	34	2121	2	2	10	0	0	0	10	0	0
99	**Fault 1, 3, 11**	**F0822405.x01**	**0**	**1**	**2**	**3**	**43**	**34**	**3234**	**2**	**2**	**21**	**0**	**0**	**0**	**0**	**7**	**0**	**0**
103	Fault 1, 3, 11	F0922884.x01	0	1	2	4	42	34	34	2	2	2121	0	0	0	0	28	0	0
104	Fault 1, 3, 11	F0810140.x01	0	1	1	4	43	34	34	2	2	2121	0	0	0	0	7	0	0
128	**Fault 1, 4**	**F1112E8D.x01**	**0**	**1**	**1**	**414**	**4**	**313**	**3**	**212**	**212**	**212**	**0**	**0**	**0**	**1**	**1**	**0**	**0**
131	Fault 1, 4	F2212CD7.x01	1	1	1	431	4	31	34	21	21	21	0	0	1	1	1	0	0
141	Fault 1, 4	F2212E1D.x01	0	1	1	414	4	232	2	212	212	212	0	0	0	1	1	0	0
150	**Fault 1, 4**	**F2213569.x01**	**1**	**1**	**1**	**414**	**4**	**3123**	**3**	**212**	**212**	**212**	**0**	**0**	**0**	**1**	**1**	**0**	**0**
162	Fault 1, 4	F223079B.x01	1	1	3	414	4	3123	3	212	212	212	0	0	0	1	2	0	0
202	**Fault 1, 5**	**F08101CA.x01**	**0**	**1**	**1**	**4**	**4**	**3**	**3**	**2**	**2**	**2**	**101**	**0**	**0**	**1**	**1**	**0**	**0**
209	Fault 1, 5	F0820165.x01	0	1	2	4	434	3	3	2	2	2	10	0	0	1	2	0	0
219	Fault 1, 6	F112267F.x01	0	1	2	4	41	23	21	21	21	21	0	0	1	1	0	0	0
222	Fault 1, 6	F22225C4.x01	0	1	2	41	1	41	31	1	1	1	0	0	1	1	0	0	0
229	Fault 1, 6, 7	F08226DB.x01	0	1	2	4	1	3	1	1	1	1	0	0	1	1	1	1	1
236	Fault 1, 6, 8	F1122605.x01	0	1	2	4	41	1	21	21	21	21	0	1	1	1	1	0	0
239	Fault 1, 6, 8	F11226A5.x01	1	1	2	4	14	32	1	12	12	12	0	1	0	1	1	0	0
252	Fault 2	F20404C1.x01	0	2	4	41	4	341	34	2	2	2	0	0	0	1	0	0	0
253	Fault 2	F20406CC.x01	0	2	4	3	4	34	34	2	2	2	111	0	0	1	1	11	0
254	Fault 2	F2212D83.x01	0	2	1	4	4	3	3	2	2	2	0	0	0	1	0	0	0
265	**Fault 2**	**F2713113.x01**	**0**	**2**	**1**	**4**	**3**	**2**	**2**	**2**	**2**	**2**	**111**	**0**	**0**	**1**	**1**	**0**	**0**
267	Fault 2	F2713116.x01	0	2	1	434	323	343	3	2	2	2	111	0	0	1	1	0	0
305	Fault 2, 3	F041075C.x01	0	2	1	3	32	3213	3	1	2	2	0	0	0	0	0	0	0
349	Fault 2, 3, 7	F0820715.x01	0	2	2	3	313	2	2	212	121	2	0	0	0	0	2	6	1
374	**Fault 2, 6**	**F08226BF.x01**	**0**	**2**	**2**	**4**	**421**	**3**	**1**	**21**	**21**	**21**	**0**	**0**	**1**	**1**	**1**	**32**	**0**
391	Fault 3	F0112939.x01	0	0	1	3	2	2	212	2	2	1	0	0	0	0	0	0	0
392	Fault 3	F0121F8B.x01	0	0	2	3	23	21	21	12	1	2	0	0	0	0	0	0	0
403	**Fault 3**	**F0121F8D.x01**	**0**	**0**	**2**	**3**	**212**	**2**	**212**	**21**	**1**	**12**	**0**	**0**	**0**	**0**	**1**	**0**	**0**
415	Fault 3, 5	F1140866.x01	0	0	4	4	3	3	313	2	1	1	100	0	0	0	0	0	0
420	Fault 4	F2410189.x01	1	0	1	41	41	31	31	21	21	21	0	0	1	1	5	0	0
421	Fault 5	F1122499.x01	0	0	2	4	434	2	2	2	2	2	110	0	0	1	1	0	0
422	Fault 5	F27200E4.x01	0	0	2	4	4	3	3	2	2	2	10	0	0	1	0	0	0
440	Fault 8	F2212F95.x01	0	0	1	4	4	1	2	2	2	2	0	1	0	1	0	0	0
441	Fault 8	F222260C.x01	0	0	2	4	4	1	2	2	2	2	0	1	0	1	0	0	0
447	Fault 9	F1113008.x01	0	0	1	4	14	32	12	2	2	2	0	0	0	1	3	0	0
448	Fault 9	F1121E5D.x01	0	0	2	13	4	12	32	2	2	2	0	0	0	1	0	0	0
465	Fault 9	F2410185.x01	0	0	1	4	14	32	12	2	2	2	0	0	0	1	3	0	0
466	Fault 10	F0813030.x01	0	0	1	4	1	3	1	2	2	2	0	0	0	1	0	0	0
472	Fault 10	F2420133.x01	0	0	2	4	4	3	3	2	2	2	0	0	0	1	0	0	0
475	**Fault 10**	**F272015F.x01**	**0**	**0**	**2**	**41**	**4**	**21**	**2**	**2**	**2**	**2**	**0**	**0**	**0**	**1**	**0**	**0**	**0**
484	Fault 12	F111302A.x01	1	0	1	4	4	3	3	2	2	2	0	0	0	1	0	0	0
490	**Fault 12**	**F224070E.x01**	**1**	**0**	**4**	**3**	**3**	**2**	**2**	**2**	**2**	**2**	**0**	**0**	**0**	**1**	**0**	**0**	**0**
504	Fault 12	F2510DC5.x01	1	0	1	4	4	3	3	2	2	2	0	0	0	1	0	0	0
507	Fault 12	F2240701.x01	1	0	4	3	3	2	2	2	2	2	0	0	0	1	0	0	0
508	Fault 12	F2522498.x01	1	0	2	4	4	3	3	2	2	2	0	0	0	1	0	0	0

Table 13: Partial information table for power system fault classification testing.

File index	Fault types	Fault file names	Attr1	Attr2	Attr3	Attr4	Attr5	Attr6	Attr7	Attr8	Attr9	Attr10	Attr11	Attr12	Attr13	Attr14	Attr15	Attr16	Attr17
1	Fault 1	**F0812D94.x01**	0	1	1	4	43	3	3	2	2	2	0	0	0	1	0	0	0
19	Fault 1	**F222020C.x01**	0	1	2	4	4	3	3	2	2	2	0	0	0	1	0	0	0
30	Fault 1	**F2240598.x01**	0	1	4	414	4	313	3	2	2	2	0	0	0	1	0	0	0
46	Fault 1, 3	**F2513620.x01**	0	1	1	3	4	3	3	2	2	1	0	0	0	0	0	0	0
53	Fault 1, 3, 5, 7	F2720944.x01	0	1	2	3	34	2	2	2	2	1212	10	0	0	0	2	1	1
54	Fault 1, 3, 5	**F08136DC.x01**	0	1	1	3	4	3	3	2	2	1	110	0	0	0	1	0	0
56	Fault 1, 3, 8	F08228BE.x01	0	1	2	4	434	3	1	2	2	212	0	1	0	0	2	0	0
57	Fault 1, 3, 11	**F08224D9.x01**	0	1	2	4	43	3	3	21	2	2	0	0	0	0	7	0	0
63	Fault 1, 4	F2222943.x01	0	1	2	414	4	313	3	2	2	2	0	0	0	1	0	0	0
82	Fault 1, 4	**F2512DE2.x01**	0	1	1	414	4	2312	2	212	212	212	0	0	0	1	1	0	0
86	Fault 1, 5	**F2710116.x01**	0	1	1	4	4	3	3	2	2	2	1	0	0	1	1	0	0
87	Fault 1, 5	**F27101E6.x01**	0	1	1	4	4	3	3	2	2	2	1	0	0	1	1	0	0
90	Fault 1, 6	F2420314.x01	0	1	2	41	41	431	431	21	21	21	0	0	1	1	7	0	0
91	Fault 1, 6, 7	F082D265.x01	0	1	2	4	1	3	1	1	1	1	0	0	1	1	0	1	1
93	Fault 1, 6, 8	F1122405.x01	0	1	2	4	41	1	21	21	21	21	0	1	1	1	1	0	0
94	Fault 1, 8	F1213907.x01	0	1	1	4	43	1	343	2	2	2	0	1	0	1	0	0	0
111	Fault 2	**F22124E5.x01**	1	2	1	4	4	3	3	2	2	2	0	0	0	1	0	0	0
112	Fault 2	**F2213568.x01**	0	2	1	4	4	3	3	2	2	2	0	0	0	1	0	0	0
116	Fault 2, 3	**F22135CA.x01**	0	2	1	3	4	3	3	2	2	1	0	0	0	0	0	0	0
120	Fault 2, 3	**F2921F2B.x01**	0	2	2	3	3	2	2	1	2	2	0	0	0	0	0	0	0
121	Fault 2, 3, 5	F08163D5.x01	0	2	1	3	4	3	3	2	2	1	110	0	0	0	1	0	0
127	Fault 2, 3, 5	F0413DCF.x01	0	2	1	3	4	3	3	2	2	1	100	0	0	0	1	0	0
128	Fault 2, 3, 5, 7	F27208F3.x01	0	2	2	3	3	2	2	12	21	2	1	0	0	0	3	1	1
129	Fault 2, 4	F22124FD.x01	1	2	1	41	1	321	31	21	21	21	0	0	1	1	1	0	0
130	Fault 2, 5	F0814371.x01	0	2	1	4	1	343	1	2	2	2	1	0	0	1	1	0	0
131	Fault 2, 5	F27123EC.x01	0	2	1	4	4	3	3	2	2	2	10	0	0	1	1	0	0
132	Fault 2, 6	F113044A.x01	0	2	3	431	431	31	321	21	21	21	0	0	1	1	8	0	0
134	Fault 2, 6	F242042B.x01	0	2	2	4	41	34	321	21	21	21	0	0	1	1	1	0	0
135	Fault 2, 6, 7	F0827040.x01	0	2	2	3	1	3	1	1	1	1	0	0	1	1	0	2	1
136	Fault 2, 4, 7	F2923565.x01	0	2	2	3	1	23	1	1	1	1	0	0	1	1	2	4	1
137	Fault 2, 8	F1213908.x01	0	2	1	4	31	1	31	2	2	2	0	0	0	1	0	0	0
138	Fault 2, 9	F1123FF6.x01	0	2	2	3	4	3	34	2	2	2	0	0	0	1	0	0	0
139	Fault 3	**F0121F8B.x01**	0	0	2	3	23	21	21	12	1	2	0	0	0	0	0	0	0
141	Fault 3	**F01127C3.x01**	0	0	1	13	4	12	2	12	12	1	0	0	0	0	2	0	0
144	Fault 3, 5	**F0121F8E.x01**	0	0	2	3	3	2	2	12	1	2	1	0	0	0	1	0	0
145	Fault 3, 5, 9	F1112BAE.x01	0	0	1	4	3	3	323	2	1	1	100	0	0	0	0	0	0
146	Fault 4	F2410819.x01	1	0	1	41	41	31	31	21	21	21	0	0	1	1	5	0	0
147	Fault 6	F1130FA4.x01	0	0	3	1	1	1	1	1	1	1	0	0	1	1	0	0	0
148	Fault 6, 8	F082B62B.x01	1	0	2	4	14	32	1	12	12	12	0	1	0	1	2	0	0
150	Fault 6, 8	F2210A17.x01	0	0	1	4	1	1	1	1	1	1	0	1	1	1	0	0	0
151	Fault 8	F22129F5.x01	0	0	1	4	4	1	2	2	2	2	0	1	0	1	0	0	0
152	Fault 8	F2210A1C.x01	0	0	1	4	4	1	2	2	2	2	0	1	0	1	0	0	0
153	Fault 9	F1122E5D.x01	0	0	2	13	4	12	32	2	2	2	0	0	0	1	0	0	0
156	Fault 9	F2410810.x01	0	0	1	4	4	32	32	2	2	2	0	0	0	1	0	0	0
158	Fault 10	F1121E95.x01	0	0	2	31	4	21	2	2	2	2	0	0	0	1	0	0	0
161	Fault 10	F2710117.x01	0	0	1	4	4	3	2	2	2	2	0	0	0	1	0	0	0
167	Fault 12	F2510CD5.x01	1	0	1	4	4	3	3	2	2	2	0	0	0	1	0	0	0
168	Fault 12	F2522894.x01	1	0	2	4	4	3	3	2	2	2	0	0	0	1	0	0	0

The following sections cover the details of signal processing for feature extraction and the 17 functions that represent these features (attributes).

6.2.1 Feature 1 – Pole Voltage Sharp Dropping

The pole line voltage is a constant signal and the standard values are ±450KV, ±300KV, ±150KV. In fault 12 *i.e.*, "Disturbance on DC Line", the pole voltage is affected by high frequency interference and causes a sharp drop at the tripping edge. The pole voltage sharp dropping sometimes happens in fault 4 as well. As described in Section 2.2, there are two cases in fault 4, "Pole Line Fault" and "Force Retard". The pole voltage in "Force Retard" decreases slowly while the pole voltage in "Pole Line Fault" drops as sharp and quick as in fault 12. Figure 36 shows the pole voltages in fault 4 and 12. F1121E8D.x01 is a "Force Retard Fault", and F2213569.x01 is a "Pole Line Fault".

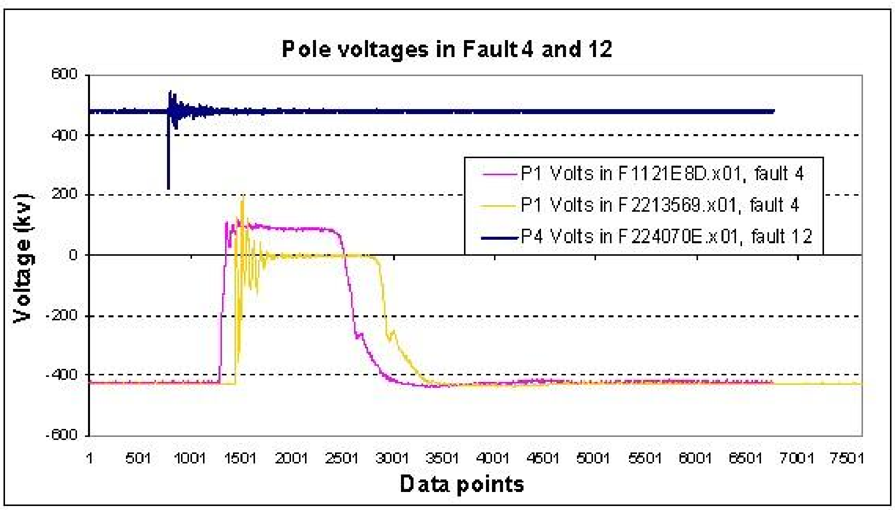


Figure 36: Pole line voltages with sharp dropping.

The derivative of pole line voltages is an efficient method to detect the sharp dropping of the pole line voltage. A 4-point averaging for noise compression is applied before the derivative. The derivative result is shown in Figure 37. It is noticeable that the derivative of P1 in F2213569.x01 and F224070E.x01 both have sharp peaks, while the P1 in F1121E8D.x01 has fairly small output. The threshold to determine a sharp pole voltage drop is 100. The value of feature 1 is 1 for F2213569.x01 and F224070E.x01, and 0 for F1121E8D.x01.

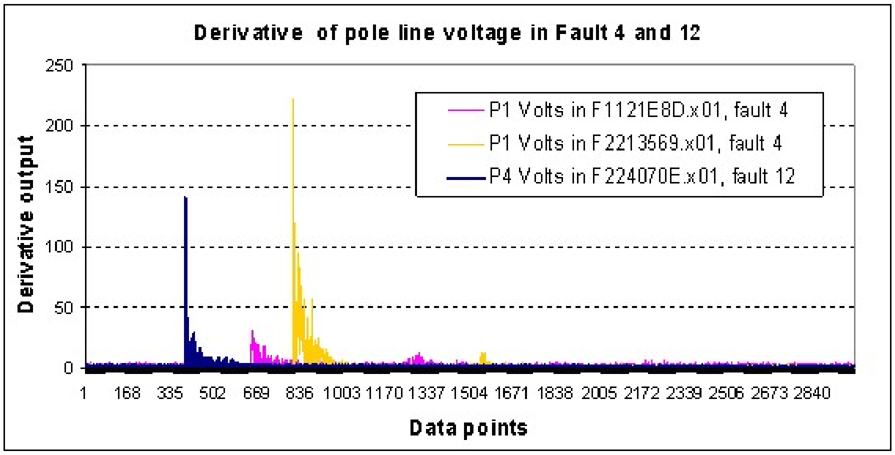


Figure 37: Derivative of pole line voltages.

The function f_1 representing this feature is defined by (40)

$$f_1(x) = \begin{cases} 1 & if \;\; max(derivitive(average(x))) > 100, \\ 0 & otherwise. \end{cases} \tag{40}$$

where x is the discrete pole voltage signal in a fault file.

Figure 36 shows that the pole voltage oscillates at the tripping edge in both “Pole Line Fault” (F2213569.x01) and “Disturbance on DC Line” fault (F224070E.x01). The FFT analysis in Figure 38 shows that “DC Disturbance on DC Line” contains higher frequency components. It has a FFT peak at 60Hz, which indicates that the interference is possibly from the AC line. The FFT peaks for both cases of fault 4 are located lower than 6Hz. This feature is very useful and will be added to improve the accuracy of the fault classification system.

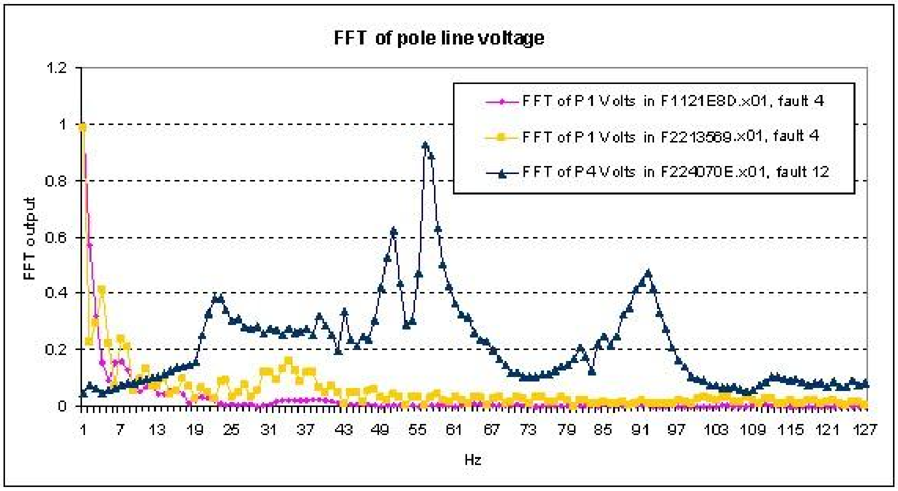


Figure 38: FFT analysis of pole line voltages.

6.2.2 Feature 2 – AC Disturbance

The three AC phase voltages, namely the A-phase, B-phase and C-phase, have a fixed $120°$ phase difference from each other. It is found by studying the data file that one period of AC phase voltage is represented by 96 data points. So if B-phase is shifted 32 points and C-phase is shifted 64 points, the shifted B-phase and C-phase will be exactly the same as the A-phase in normal condition. If the AC voltages have distortion, it can be detected by an error signal, which is calculated by

$$\begin{aligned} err &= \frac{|(A{-}phase) - (shifted\ B{-}phase)|}{3} + \\ & \frac{|(shifted\ B{-}phase) - (shifted\ C{-}phase)|}{3} + \\ & \frac{|(shifted\ C{-}phase) - (A{-}phase)|}{3}. \end{aligned} \tag{41}$$

Taking file F2713113.x01 as an example, the AC phase analysis results are shown in Figure 39. The first graph shows the original three AC phase voltages; the second graph shows the shifted AC phase voltages; and the last one displays the error output of AC voltage signals.

The AC disturbance error can be discretized by granule algorithm. The granule functions are designed based on the Gaussian function and can be written as

$$\phi_j(x) = exp\left(-\frac{\|x-\mu_j\|^2}{2\sigma_j^2}\right), j = 1, 2, 3. \tag{42}$$

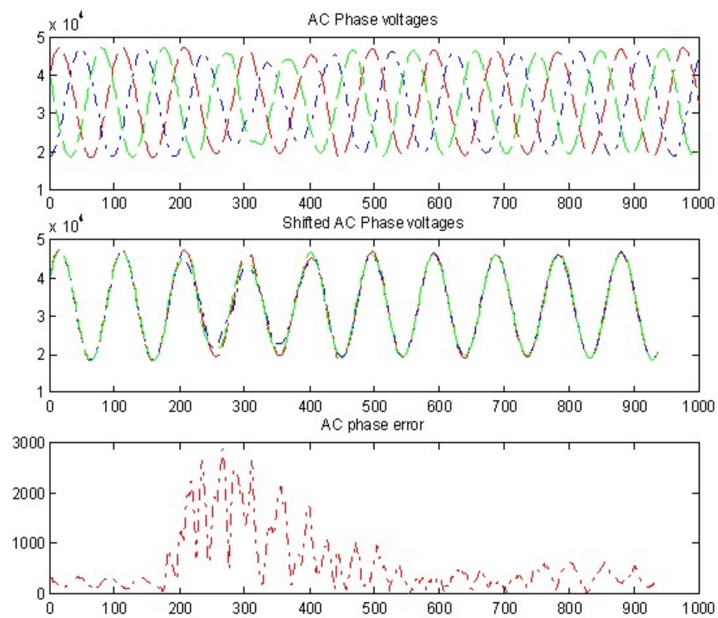


Figure 39: Analysis of AC phase voltages by phase shift method.

Three granule functions need to be designed to discretize the AC disturbance error into three intervals: *low*, *medium* and *high*. The center μ_j and σ_j is estimated based on the 676 files provided by the Manitoba Hydro Dorsey Station. Among those 676 fault files, 240 files are indicated as Minor AC disturbance and 148 files as AC disturbance. The averaged AC phase voltage error calculated from the first 240 files is close to 700 while the averaged AC phase voltage error from the other 148 files is approximately 2100. This leads to a supervised procedure for optimizing the granule function parameters. The Gaussian granule functions for the AC disturbance error discretization are defined as (43) and plotted in Figure 40.

$$\begin{aligned} \phi_1(x) &= exp\left(-\frac{\|x-200\|^2}{2\times 200^2}\right), \\ \phi_2(x) &= exp\left(-\frac{\|x-700\|^2}{2\times 500^2}\right), \\ \phi_3(x) &= exp\left(-\frac{\|x-2100\|^2}{2\times 900^2}\right). \end{aligned} \tag{43}$$

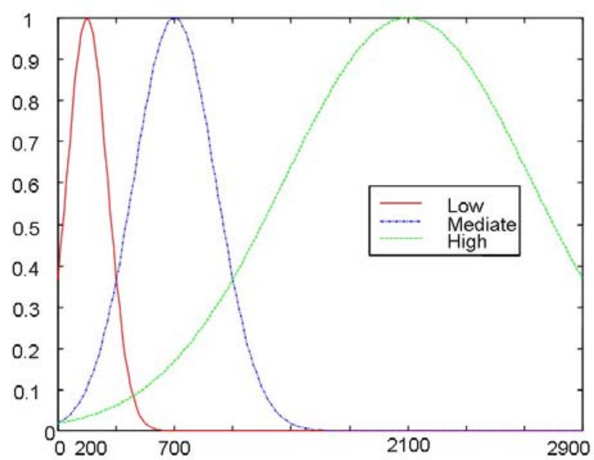


Figure 40: The granule formula.

For an input x, the peak value of the AC voltage error, three granule output $\phi_1(x)$, $\phi_2(x)$ and $\phi_3(x)$ are calculated respectively. If $\phi_1(x)$ is the biggest, the AC error is small enough to be considered as normal and 0 will be assigned. If $\phi_2(x)$ is the biggest, the AC error is moderate implying a minor disturbance and 1 will be assigned. If $\phi_3(x)$ is the biggest, it is a severe AC disturbance and 2 will be assigned.

The function f_2 representing feature 2 can be defined as

$$f_2(x) = \begin{cases} 2 & if \;\; max(\phi_1(x), \phi_2(x), \phi_3(x)) = \phi_3(x), \\ 1 & if \;\; max(\phi_1(x), \phi_2(x), \phi_3(x)) = \phi_2(x), \\ 0 & if \;\; max(\phi_1(x), \phi_2(x), \phi_3(x)) = \phi_1(x), \end{cases} \tag{44}$$

where $x = max(err(V_a, V_b, V_c))$, V_a, V_b and V_c are discrete A-phase, B-phase and C-phase voltages in a fault file.

6.2.3 Feature 3 – Pole Index

The information about the pole index is very easy to retrieve but helpful to identify the fault, “Normal Affected by Another Pole”. According to the *.scf file, it is known that the 4th character of the file name indicates the pole index, *i.e.*, F272015F.x01 file is a pole 2 file. Seventeen features of this fault file are listed in Table 14. It is observed that attributes 4 and 6 notify the pole 1 voltage and the current was blocked. All the other features are for pole 2, and they are normal. The fault file of pole 2 was created due to the effect from pole 1. It is reasonable to classify this file as the fault, “Normal Affected by Another Pole”.

Table 14: Features for F272015F.x01.

File index	Fault types	Fault file names	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17
475	Fault 10	F272015F.x01	0	0	2	41	4	21	2	2	2	2	0	0	0	1	0	0	0

The function f_3 representing this feature (pole index) is defined as

$$f_3(x) = pole\ index, \tag{45}$$

where x is the file name of a fault file.

6.2.4 Features 4, 5, 6 and 7 – Pole Voltage and Current Trend

To derive the trend of the pole line voltages and currents, it is necessary to smooth the waveform by applying a low pass filter. A high order FFT followed by a low order inverse FFT is an alternative to a digital low pass filter. The sampling rate of the Transcan^{TM} system is 6000 points per second. Transcan^{TM} itself is a low pass filter with a cutoff frequency of 3 kHz. An FFT of 8192 points followed by a 32 point inverse FFT is a low pass filter with cutoff frequency around 11.7 Hz. Most interference on the pole line voltage has a frequency of 16 – 90 Hz and is removed by the low pass filter. An example of pole line voltages and currents and their simplified waveforms are shown in Figures 41 and 42, respectively.

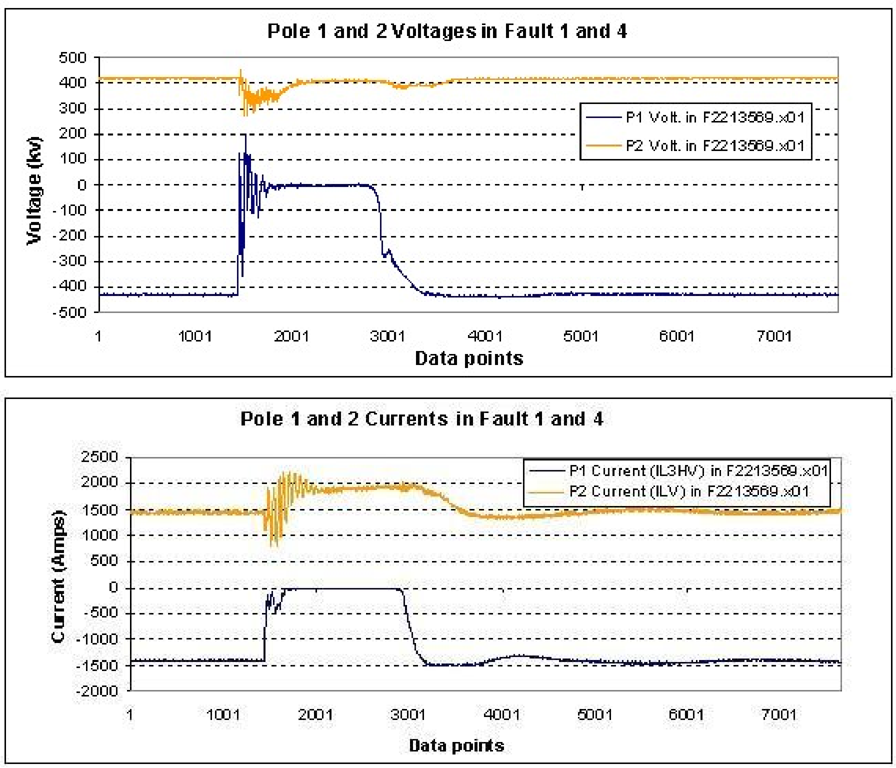
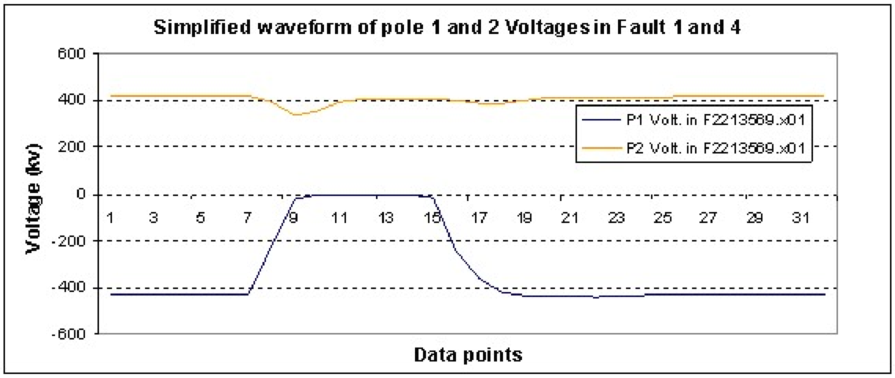


Figure 41: Pole voltages and currents in fault 1 and 4.



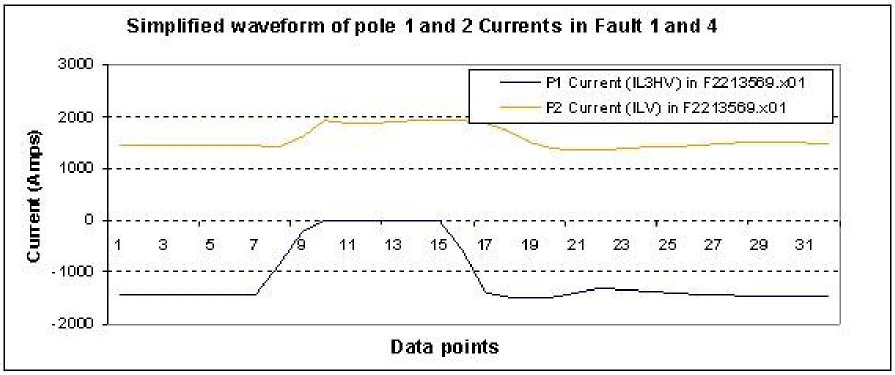


Figure 42: Simplified waveform of pole voltages and currents in faults 1 and 4.

The simplified waveforms are represented by a sequence of numbers (codes) based on (46) and (47).

$$Code_V = \begin{cases} 1 & if \;\; |Pole\ voltage| \leq 100, \\ 2 & if \;\; 100 < |Pole\ voltage| \leq 150, \\ 3 & if \;\; 150 < |Pole\ voltage| \leq 300, \\ 4 & if \;\; 300 < |Pole\ voltage|. \end{cases} \tag{46}$$

$$Code_I = \begin{cases} 1 & if \;\; |Pole\ current| \leq 400, \\ 2 & if \;\; 400 < |Pole\ current| \leq 1000, \\ 3 & if \;\; 1000 < |Pole\ current| \leq 2000. \end{cases} \tag{47}$$

The codes for pole voltages and currents in fault F2213569.x01 are listed in Table 15. The original codes contain 32 numbers; the simplified codes remove all the duplicated numbers and for some special cases, *i.e.*, “43134”, “42124” and “32123”, they are further condensed to “414”, “414” and “313”, respectively.

Table 15: Codes for pole voltage and current trend.

Signal names	Original codes	Simplified codes
Pole 1 volt.	44444443134444444444444444444444	414
Pole 2 volt.	44444444444444444444444444444444	4
Pole 1 current	3333333123333333333333333333333	3123
Pole 2 current	33333333333333333333333333333333	3

Functions f_4, f_5, f_6 and f_7 represent features 4, 5, 6 and 7 respectively and are defined as follows:

$$f_4(x) = Code_V(LF(x)), \tag{48}$$

where x is the discrete pole 1 voltage in a fault file;

$$f_5(x) = Code_V(LF(x)), \tag{49}$$

where x is the discrete pole 2 voltage in a fault file;

$$f_6(x) = Code_I(LF(x)), \tag{50}$$

where x is the discrete pole 1 current in a fault file; and

$$f_7(x) = Code_I(LF(x)), \tag{51}$$

where x is the discrete pole 2 current in a fault file.

In (48) to (51) $Code_V(\cdot)$ and $Code_I(\cdot)$ represent the coding processes and $LF(\cdot)$ represents a lowpass filter.

6.2.5 Features 8, 9 and 10 – Valve Current Trend Vg*1, Vg*2, Vg*3

A normal valve current is a periodic signal with 96 samples per cycle. Reference to A-phase, B-phase and C-phase are 32 and 64 points delayed respectively. The amplitude of the phase current should match the current order in normal condition. A normalized phase current is calculated by (52) and illustrated in Figure 43

$$Normalized\ phase\ current = \frac{phase\ current}{phase\ current\ order}. \tag{52}$$

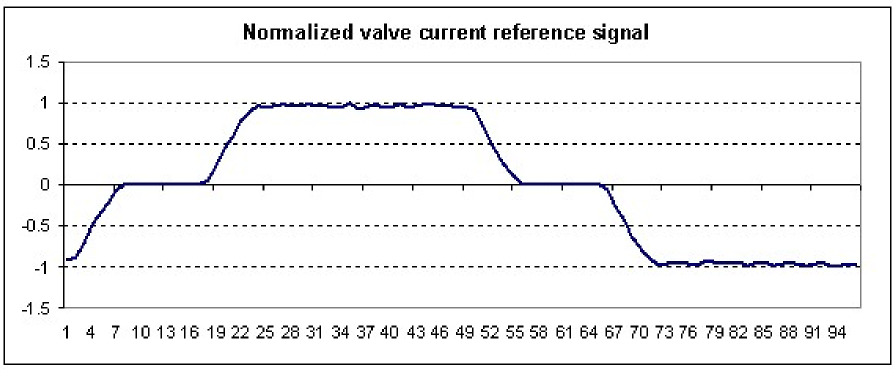


Figure 43: Normalized valve current reference signal.

When a "Valve Current Closed/Blocked/Deblocked" happens, A, B and C phase currents in this group are all closed and/or blocked and/or deblocked. An example of this fault, F0121F8D.x01 is illustrated in Figure 44. In this file, valve group 1 is blocked and valve group 3 is deblocked at a different time. To describe the trend of a valve current, the correlation algorithm plus a coding method is applied. The correlation theory was described in Section 3.1.

The autocorrelation of the normalized valve reference signal $r(j)$ at origin point, denoted ρ_0 is first evaluated by (53).

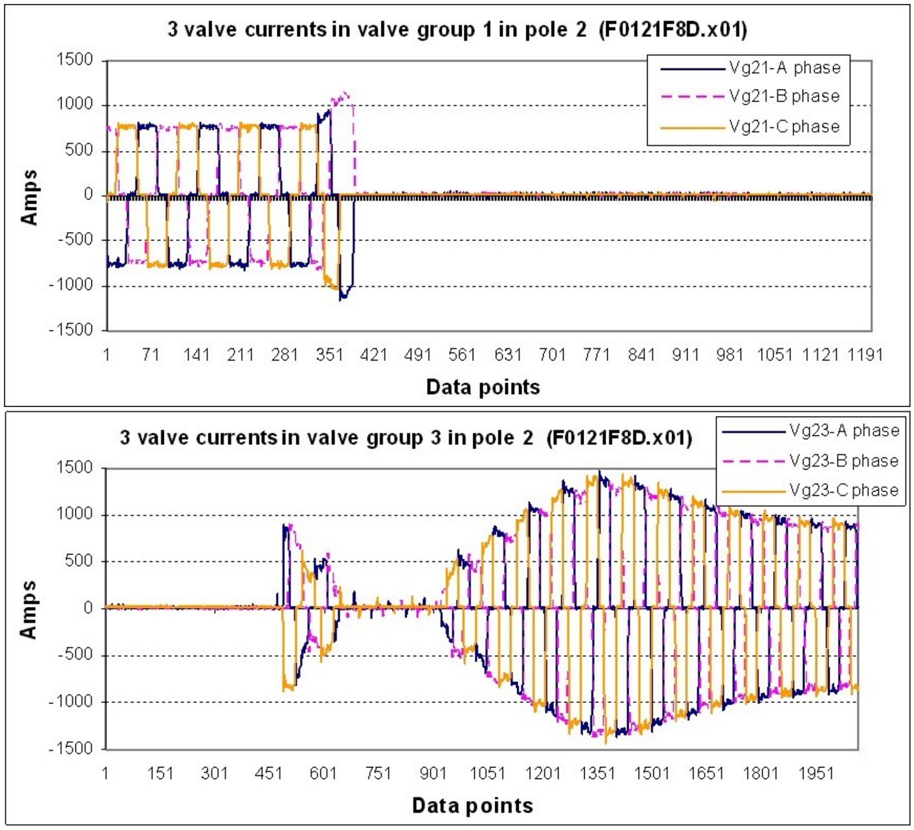


Figure 44: Valve Current Closed/Blocked/Deblocked error.

$$\rho_0 = \sum_{j=0}^{95} r(j)r(j). \tag{53}$$

The Maximum of the cross-correlation of the normalized valve current reference signal and every 96-point segment of the normalized input valve current signal $x(j)$, denoted ρ_0' is calculated by (54).

$$\rho_0' = Max(\rho_0'(i)) = Max(\sum_{j=0}^{95} r(j) \cdot x(j+i)). \tag{54}$$

If ρ_0'/ρ_0 is less than 30%, valve current is considered to be closed and a code 1 will be assigned to this input segment. The valve is considered to be normal or deblocked back to normal if the ratio is bigger than 80% and a code 2 will be assigned. Ratio ρ_0'/ρ_0 for

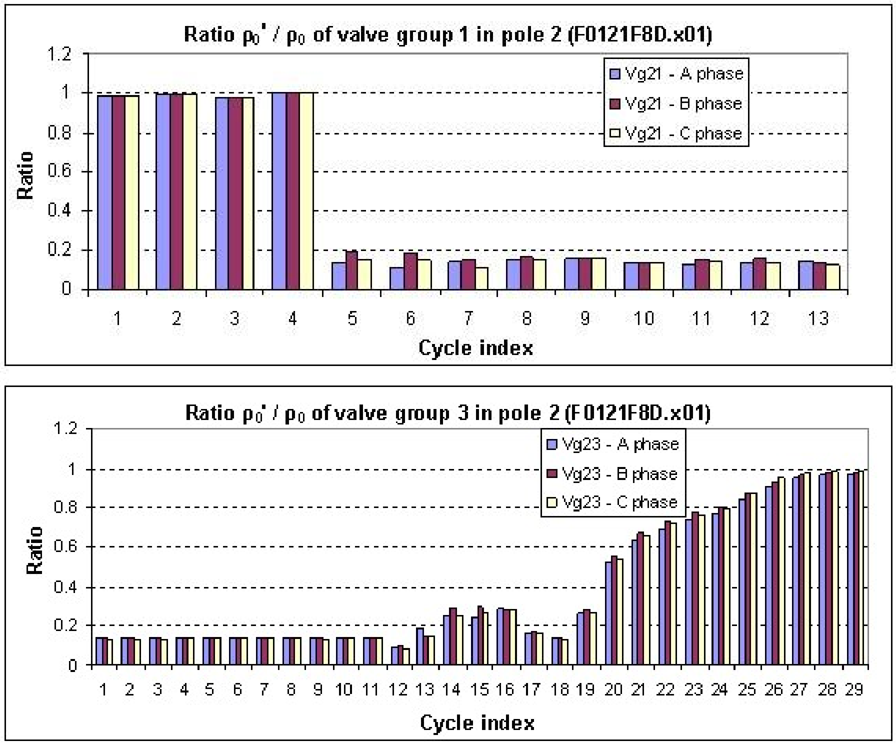


Figure 45: Ratio ρ_0'/ρ_0 for attribute Valve Current Trend.

valve group 1 and 3 in fault file F0121F8D.x01 is illustrated in Figure 45. Usually during the interim from normal to valve closed status and vice versa, various ratio of ρ_0'/ρ_0 will be observed. No code will be assigned to the ratio between 30% and 80%. It is because only the closing and normal stages need to be extracted to describe the trend of valve current. After all the input segments are processed, codes, "2222111111111" for valve group 1 and "11111111111111111111111122222" for valve group 3 are derived. To simplify the codes, only the turn points are maintained. The simplified codes for valve group 1 and 3 are "21" and "12" respectively (see attributes 8 and 10 of fault file 403 in Table 12).

Functions f_8, f_9 and f_{10} representing features 8, 9 and 10 are defined as follows:

$$f_8(x) = Code(max(r \star r)/max(r \star x)), \tag{55}$$

where x is the discrete normalized A, B and C phase valve currents in valve group *1* in a fault file;

$$f_9(x) = Code(max(r \star r)/max(r \star x)), \tag{56}$$

where x is the discrete normalized A, B and C phase valve currents in valve group *2* in a fault file; and

$$f_{10}(x) = Code(max(r \star r)/max(r \star x)), \tag{57}$$

where x is the discrete normalized A, B and C phase valve currents in valve group *3* in a fault file.

In (55) to (57), r denotes the discrete normalized valve current reference signals, $\star$ is the correlation operator, and $Code(\cdot)$ represents the coding processes.

The feature "Valve Current Trend" is very useful for classification of particular faults, *e.g.*, fault 4 ("Line Fault") and fault 6 ("Pole Voltages/Currents Closed/Blocked/Deblocked"). With the occurrence of "Line Fault", all 3 valve groups will have the same trend (the most common pattern is "212"). With this type of fault, pattern "12" or "21" also happens occasionally. With the occurrence "Pole Voltages/Currents Closed/Blocked/Deblocked", all three valve groups behave in the same way. The most commonly observed pattern for this type of fault is "1" (sometimes "12" or "21" can also occur).

6.2.6 Feature 11 – Valve Currents Minor Disturbance

"Valve Currents Minor Disturbance" happens very frequently and usually associated with a fault of "AC Disturbance" or "Valve Current Commutation Failure". The typical waveforms of valve currents minor disturbance is shown in Figure 46, which includes A, B, C, three phases of valve group Vg11 in fault F08101CA.x01. A few cycles present distortions and happen in all three phases. It is considered a valve minor disturbance as long as any one of three phases shows a disturbance.

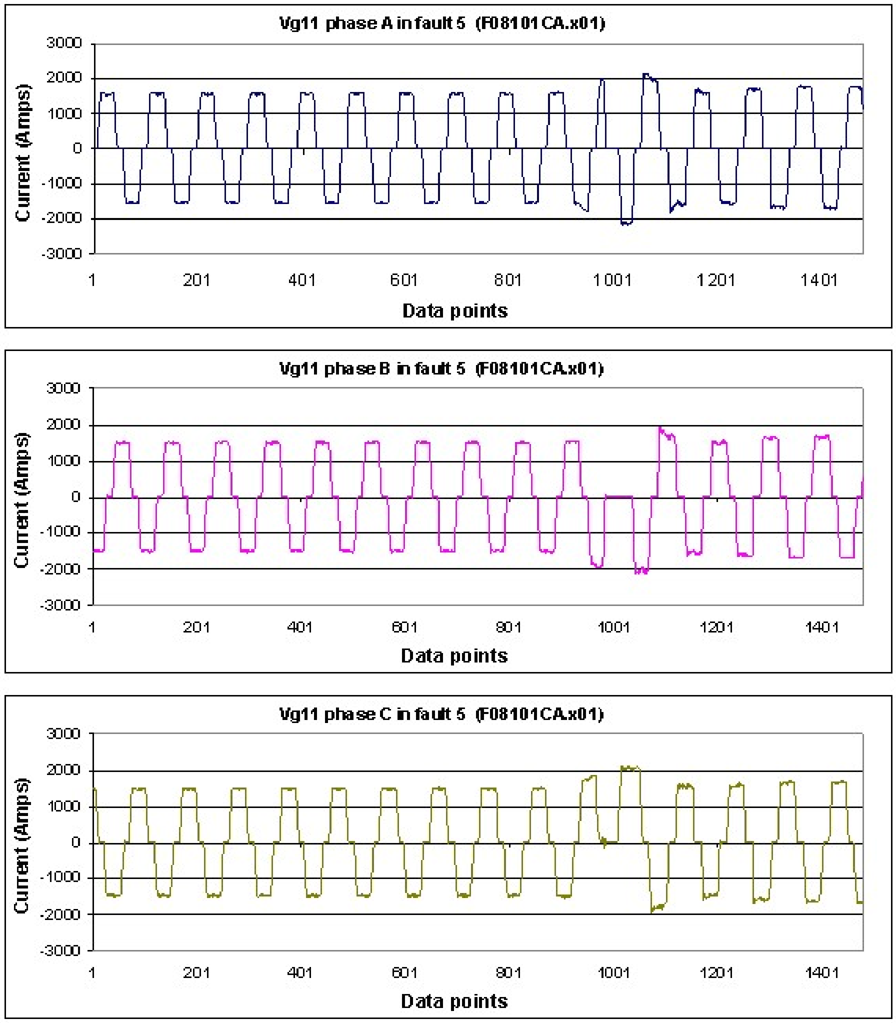


Figure 46: Valve currents in fault 5.

To detect "Valve Current Minor Disturbance", the method applied in the feature extraction for "Valve Current Trend" is adopted here. The ratio ρ_0'/ρ_0 is estimated and displayed in Figure 47. The same threshold is used to assign the code. If the ratio $>$ 80%, code "2" is assigned; ratio $<$ 30%, code "0" is assigned; in addition, between 30% and 80%, code "1" is assigned. The feature patterns indicating a valve current minor distortion are listed in Table 16 and the codes derived for phase A, B and C currents of Vg11 in F08101CA.x01 are listed in Table 17. A minor disturbance is detected in all three phases of Vg11 and a

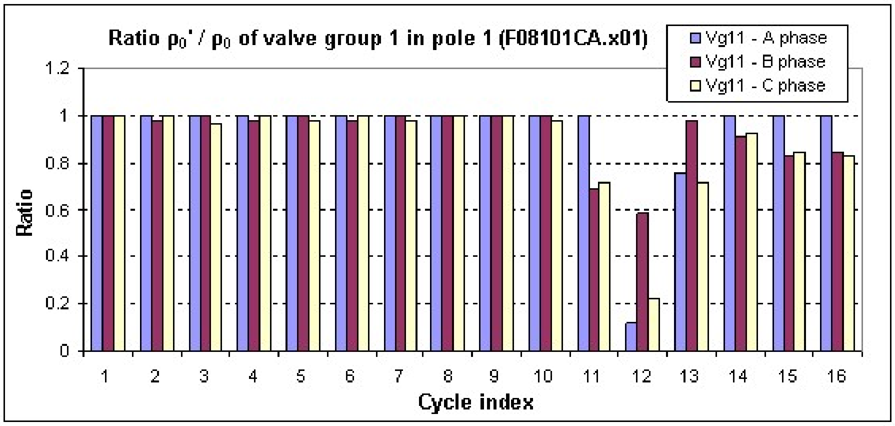


Figure 47: Ratio ρ_0'/ρ_0 for attribute Valve Currents Minor Disturbance.

final code "1" is assigned. Only when all three phases are normal, should a final code "0" be assigned. For each fault file, the same procedure is applied to all three valve groups, *i.e.*, Vg11, Vg12 and Vg13 for pole 1. Three final codes are simply combed together as the value for feature 11. The value of attribute 11 for fault F08101CA.x01 is 101.

Function f_{11} representing feature 11 is similar to the functions for features 8, 9 and 10, except that there is a different coding process.

$$f_{11}(x) = Code(max(r \star r)/max(r \star x)), \tag{58}$$

where x accepts discrete normalized A, B and C phase valve currents from all valve groups.

Table 16: Feature (attribute) codes.

Feature code 1	2,2,2,1,2,2,2
Feature code 2	2,2,2,0,2,2,2
Feature code 3	2,2,2,1,1,2,2,2
Feature code 4	2,2,2,0,0,2,2,2
Feature code 5	2,2,2,1,0,2,2,2
Feature code 6	2,2,2,0,1,2,2,2
Feature code 7	2,2,2,1,0,1,2,2,2

Table 17: Codes for valve currents minor disturbance in Vg11 in F08101CA.x01.

Signal names	Original codes
A phase valve current	2222222222201222
B phase valve current	2222222222112222
C phase valve current	2222222222101222

6.2.7 Feature 12 – Pole Current Closed with Normal Pole Voltage (True = 1, False = 0)

Section 6.2.4 explains how to extract the pole voltage and current trend. In one specific case, the pole current recorded in Transcan^{TM} gives an output of zero while the pole voltage is perfectly normal. This event happens during a parallel operation. For instance, a "pole 1 to pole 3 parallel operation" is to switch the pole 1 current to pole 3 to unload the pole 1 current line for maintenance. In Table 12, the information table for training, file F2212F95.x01 is an example of "pole 1 to pole 3 parallel operation". The 17 attributes of this file are listed in Table 18. Attribute 4 indicates the pole 1 voltage is normal, while attribute 6 shows the pole 1 current is closed, therefore the value of attribute 12 is 1.

Table 18: Features for F2212F95.x01.

File index	Fault types	Fault file names	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17
440	Fault 8	F2212F95.x01	0	0	1	4	4	1	2	2	2	2	0	1	0	1	0	0	0

Function f_{12} representing feature 12 is defined by (59).

$$f_{12}(x,y) = \begin{cases} 1 & if \;\; ((f_4(x) == 4) or (f_4(x) == 3)) AND (f_6(y) == 1), \\ 1 & if \;\; ((f_5(x) == 4) or (f_5(x) == 3)) AND (f_7(y) == 1), \\ 0 & otherwise, \end{cases} \tag{59}$$

where x is the discrete pole voltage and y is the discrete pole current.

6.2.8 Feature 13 – 3 Valve Groups All Closed (True = 1, False = 0)

Features 8, 9 and 10 of valve current trends have been discussed in Section 6.2.5. Based on features 8, 9, 10, feature 13 can be derived. If the codes of valve current trends for three valve groups all end in 1, meaning the three valve groups are all closed in the end, feature 13 yields an output of 1. This usually implies that the whole pole line is closed.

Function f_{13} representing feature 13 is defined as

$$f_{13}(x,y,z) = \begin{cases} 1 & if \; codes \; f_8(x), f_9(y) \; and \; f_{10}(z) \; are \; all \; ended \; in \; 1, \\ 0 & otherwise, \end{cases} \tag{60}$$

where x, y and z are the discrete normalized A, B and C phase currents in valve groups 1, 2 and 3, respectively.

6.2.9 Feature 14 – Same Current Trend in 3 Valve Groups (True = 1, False = 0)

Based on features 8, 9 and 10, feature 14 can also be derived. If the valve current trends of three valve groups are all the same, feature 14 gives an output of 1, which produces a high possibility of the following three faults, fault 4, “Pole Line Fault”; fault 10, “Normal Affected by Another Pole”; and fault 12, “Disturbance on DC Voltage”.

Function f_{14} representing feature 14 is defined by (61).

$$f_{14}(x,y,z) = \begin{cases} 1 & if \quad f_8(x) == f_9(y) == f_{10}(z), \\ 0 & otherwise, \end{cases} \tag{61}$$

where x, y and z are the discrete normalized A, B and C phase currents in valve groups 1, 2 and 3, respectively.

6.2.10 Feature 15 – Voltage Flashover in 6-Pulse

Feature 15 records the number of cycles of voltage flashover that happened in a 6-pulse signal. A normal 6-pulse signal shown in Figure 48 is a periodic signal. When fault 11, “Asymmetric Protection”, happens, the 6-pulse does 7 cycles of voltage flashover and closes the valves for protection. The typical waveform of a 6-pulse signal in fault 11, F0822405.x01, is illustrated in Figure 49.

To detect those 7 cycles of flashover, the Wavelet Multi-resolution Decomposition (MRD) method has been applied to extract different levels of details for the recorded signals. A number of experiments have been done to evaluate the performances of different wavelet functions such as Daubechies wavelets and the Meyer wavelet. The decomposition can be carried out in Matlab^{TM} using functions `wavedec` and `wrcoef`. The MRD with the Daubechies 2 wavelet (DB2) function extracts the 7 cycles of flashover at the 6^{th} level

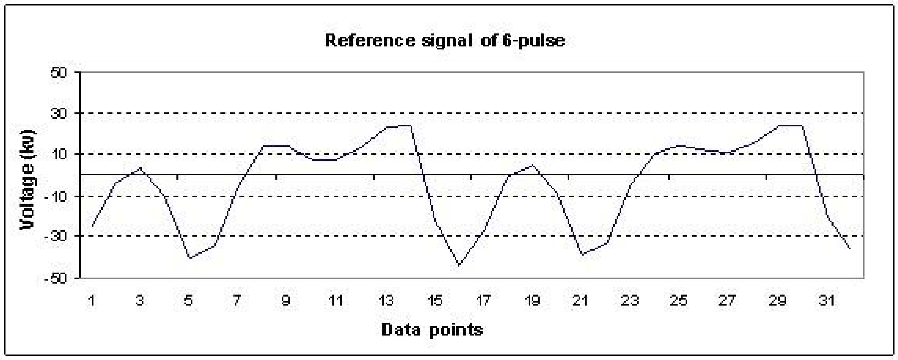


Figure 48: 6-pulse reference signal (2 cycles).

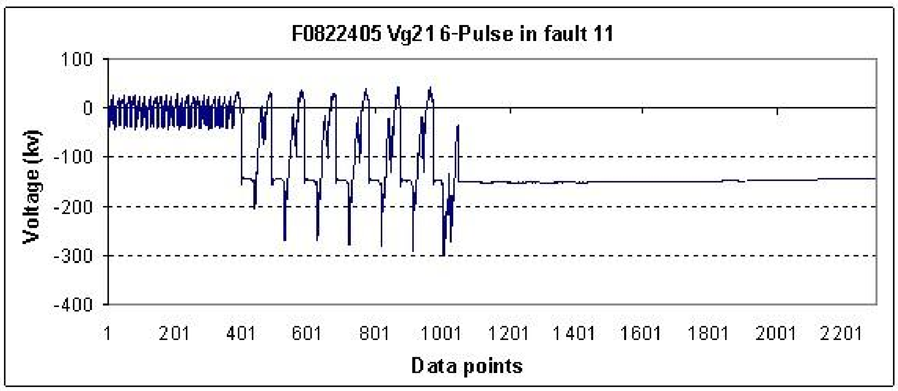


Figure 49: 6-pulse signal in Asymmetric Protection fault.

detail coefficient output. Figure 50 shows the transient signal and 7-level ‘DB2’ MRD details. The experiments with the ‘DB3’, ‘DB4’ and ‘Meyer’ wavelets extract 8 cycles of flashover, which does not agree with the 6-pulse transient signal. The 6^{th} level detail coefficient output from ‘DB2’ MRD is further processed by 32-point averaging. In addition, 7 positive peaks with values greater than 18 are detected and shown in Figure 51. Occasionally, the first 7 cycles of voltage flashover failed to close the valves and the control system continues with another 7 cycles until the valves are closed. An example of this phenomenon is F0922884.x01.

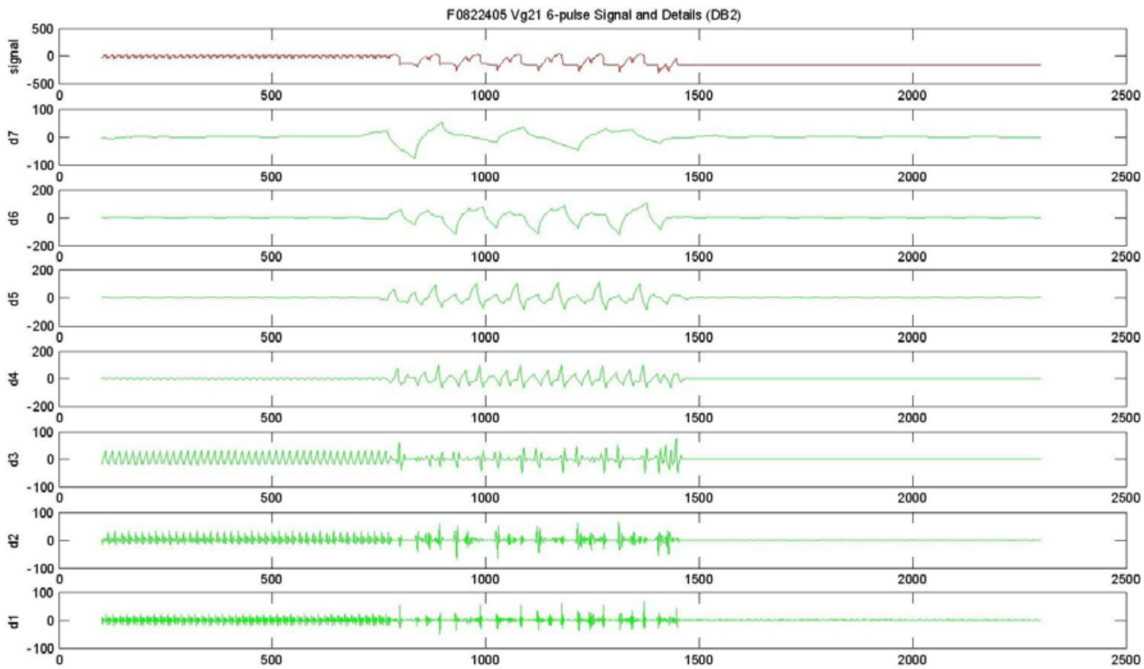


Figure 50: The multi-level details of 'DB2' MRD applied to a 6-pulse signal in Asymmetric Protection fault.

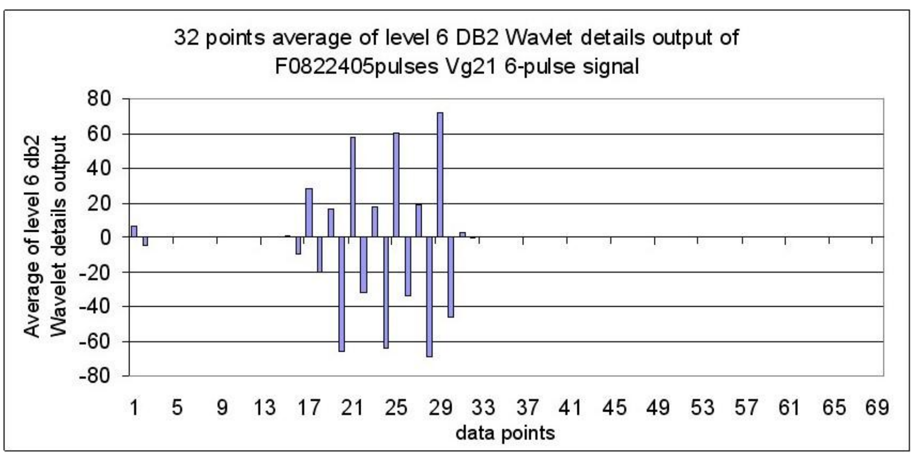


Figure 51: Seven peaks detected in a 6-pulse signal in Asymmetric Protection fault.

Function f_{15} representing feature 15 is given by (62).

$$f_{15}(x,y,z) = max(g_{15}(average(MRD(x))), g_{15}(average(MRD(y))), g_{15}(average(MRD(z)))), \tag{62}$$

where $g_{15}(\cdot)$ picks up the points with values of $average(MRD(\cdot))$ greater than 18. The discrete 6-pulse signals in valve groups 1, 2 and 3 are x, y and z respectively.

6.2.11 Feature 16 – Valve Currents Flashover

Feature 11, the valve current minor disturbance, has been discussed in Section 6.2.6. In this section, a severe fault is addressed that is involved with valve current flashover. A standard peak value for a valve current is 1400 Amps. Occasionally with a severe AC bus error or the valve line shorted together or shorted to ground, valve currents increase dramatically to an excess of 4000 Amps. Usually this happens within a pair of valves in a valve group. Two valve currents increase in opposite directions to prevent the pole current from overshooting. Examples are illustrated in Figures 52 and 53. The first example is fault 7, "Current Arc Back", valve current flashover happens only in one valve group. The second one is fault 2, "AC Disturbance", valve current flashover happens in two valve groups.

To detect the valve current flashover, 96-point averaging is applied to derive a mean value for each cycle. The mean value of a normal cycle is 0 and a flashover cycle is over 1800, which is the threshold used to detect the event of current flashover. The averaged waveforms of B and C phase currents for 3 valve groups in F082016A.x01 and F08226BF.x01 are displayed in Figures 54 and 55, respectively. For F082016A.x01, there is only one point over 1800 in Vg22; for F08226BF.x01, there are three points over 1800 in Vg21 and two points over 1800 in Vg23. The value for feature 16 is 1 for F082016A.x01 and 32 for F08226BF.x01.

Function f_{16} representing feature 16 is defined as

$$f_{16}(x, y, z) = Code(g_{16}(average(x)), g_{16}(average(y)), g_{16}(average(z))), \tag{63}$$

where $g_{16}(\cdot)$ picks up the points with values of $average(\cdot)$ greater than 1800. $Code(\cdot)$ is the coding process used to concatenate the number of points from 3 valve groups and x, y and z are the discrete normalized A, B and C phase currents in valve groups 1, 2 and 3, respectively.

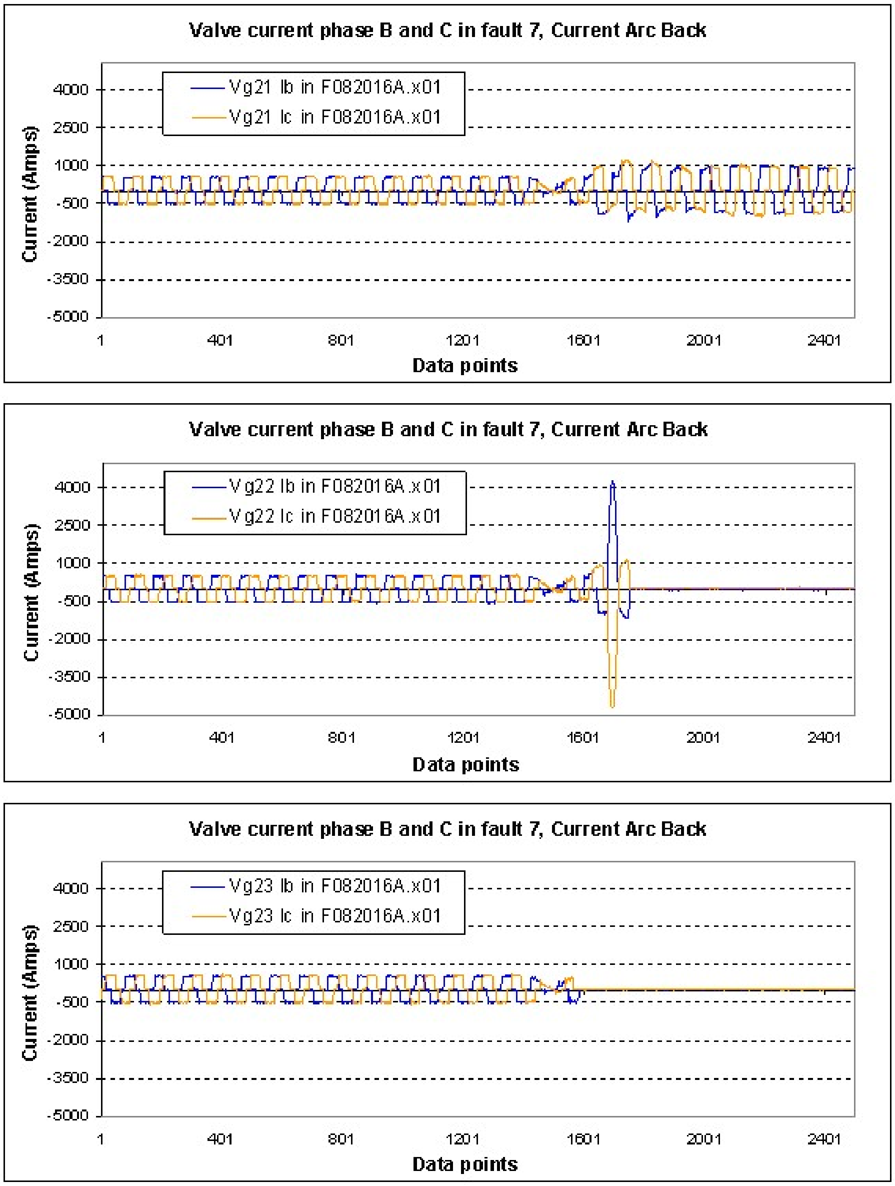


Figure 52: B and C phase currents in F082016A.x01, phase currents flashover in Vg22 valve group.

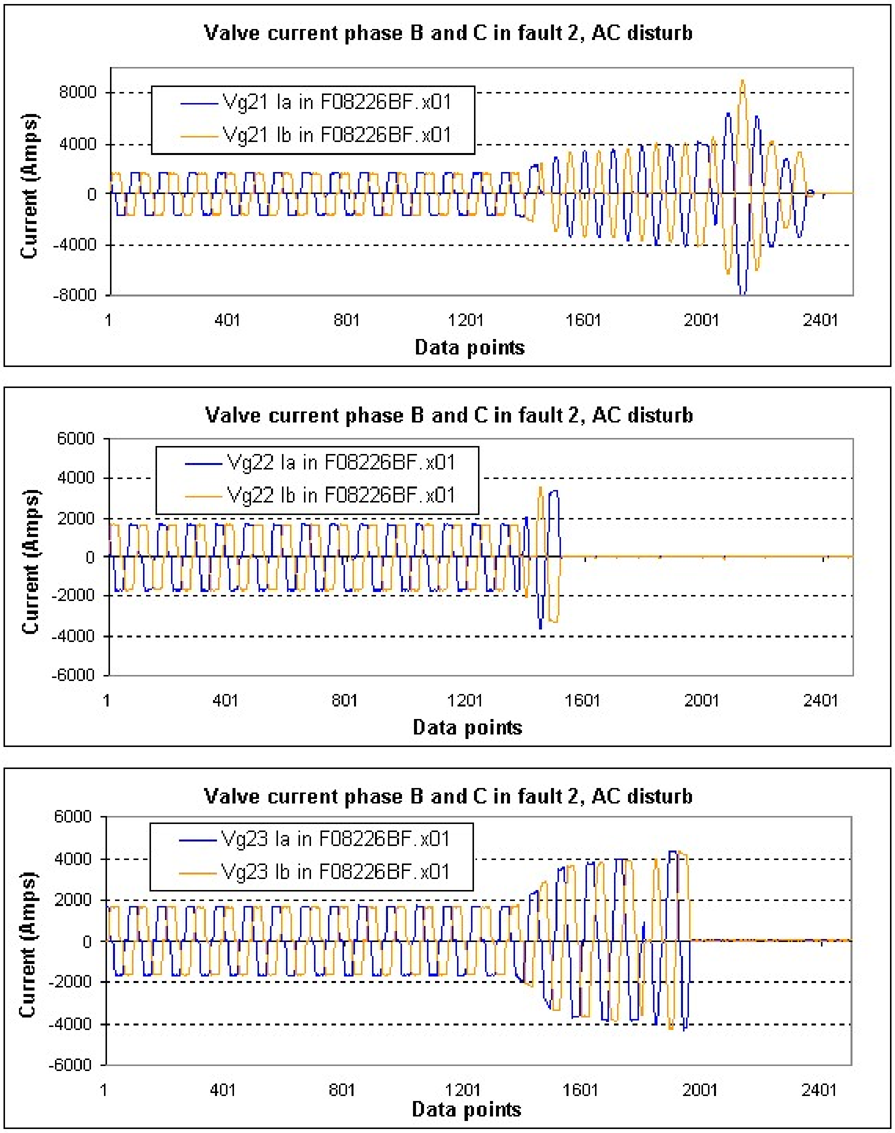


Figure 53: B and C phase currents in F08226BF.x01, phase currents flashover in Vg21 and Vg23 valve group.

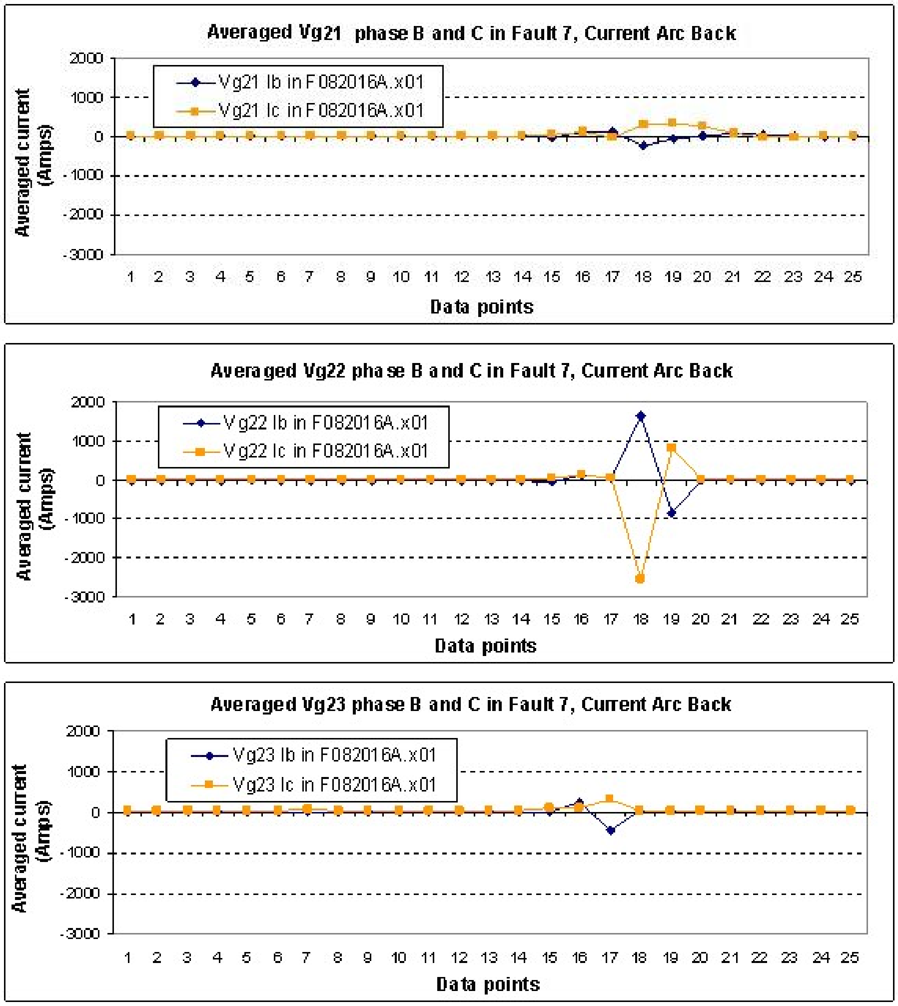


Figure 54: Averaged waveforms for B and C phase currents in F082016A.x01.

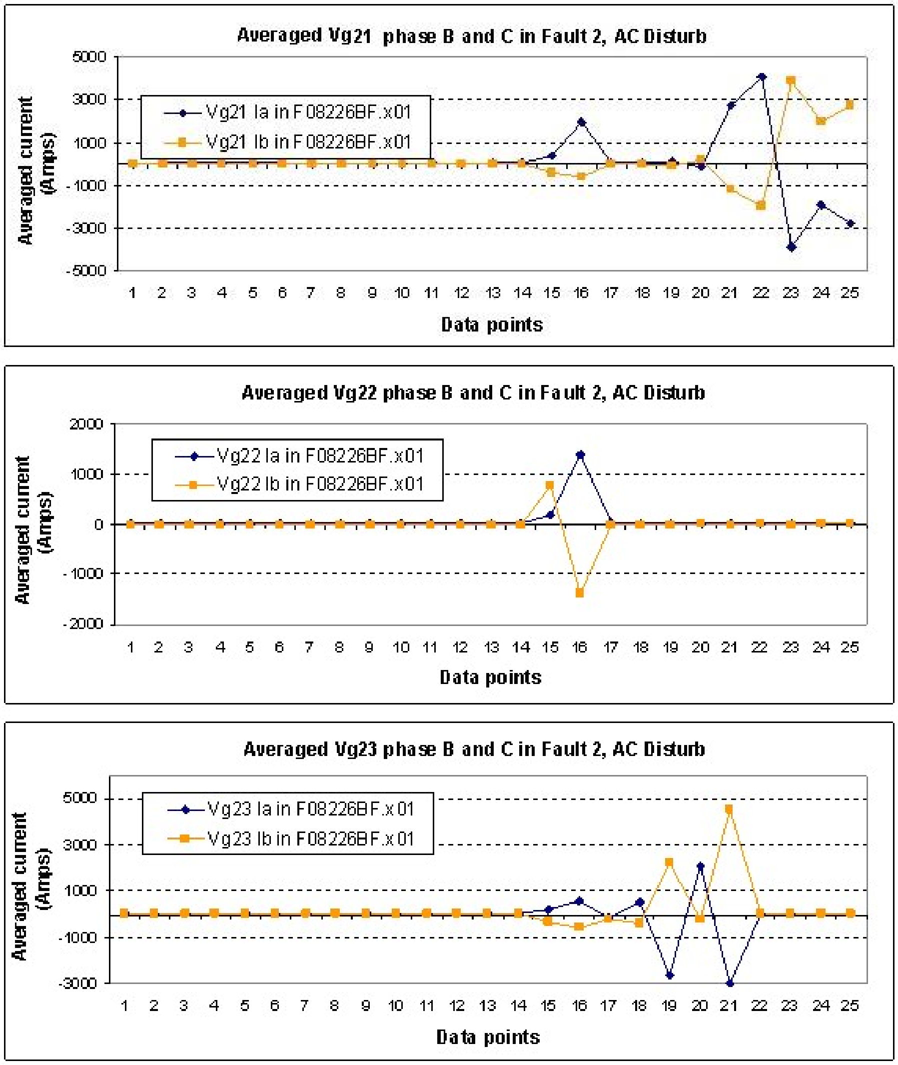


Figure 55: Averaged waveforms for B and C phase currents in F08226BF.x01.

6.2.12 Feature 17 – Valve Current Flashover Happens Only in One Valve Group (True = 1, False = 0)

As seen in the discussion of feature 16, valve current flashover happens in both faults 2 and 7. "Current Arc Back" usually comes with an "AC Disturbance". However, the current flashover in "Current Arc Back" is only due to the electronic faults in the valve group itself. A severe AC disturbance is a bus error and affects all the valve groups. If the severe

AC disturbance induces current flashover, it will affect almost all valve groups. To further separate these two faults, feature 17 is added.

It is simple to obtain feature 17 based on the results of feature 16. Feature 17 will yield an output of 1 if only one valve group gives an output of a non-zero number in feature 16.

Function f_{17} representing feature 17 is defined as

$$f_{17}(x, y, z) = \begin{cases} 1 & if\ only\ g_{16}(average(x)) \neq 0, \\ 1 & if\ only\ g_{16}(average(y)) \neq 0, \\ 1 & if\ only\ g_{16}(average(z)) \neq 0, \\ 0 & otherwise. \end{cases} \quad (64)$$

7 Rough Membership Neural Network (rmNN) for PSFC

Nomenclature

Symbol	Brief Explanation
rmNN	Rough membership function neural network
B_j	Set of attributes associated with neuron j
I	Index set for fault files, $I = \{1, 2, ..., 12\}$
J	Index set for neurons, $J = \{1, 2, ..., 11\}$
$i \in I$	i^{th} fault
RS_{ij}	A set of i^{th} faults represented by their attribute values of B_j
$RS_{ij} \setminus RS_{kj}$	$\{x \mid x \in RS_{ij}$ and $x \notin RS_{kj}\}$, difference set
x	An input fault data file, *.x01
$[x]_{B_j}$	The equivalence class of x based on the attribute set B_j
$\mu^{B_j}_{RS_{ij}}(x)$	Rough membership of i^{th} fault for input x file based on RS_{ij} and $[x]_{B_j}$
$t\text{-}norm(x, y)$	$x \times y$
$s\text{-}norm(x, y)$	$x + y - x \times y$
$r \rightarrow x$	Imply operator, $min(1, \frac{x}{r})$
$rmf^k(B_i(obj_n))$	The rough membership estimation of k^{th} fault based on B_i for fault file n
r_{ij}, w_{ij}	Weights on connection from the i^{th} neuron in the input layer to the j^{th} neuron in hidden layer in rmNN
u_j	Weight on the connection from the j^{th} neuron in the hidden layer to the output neuron in rmNN
h^k_j	Output of the j^{th} neuron in the hidden layer in the k^{th} rmNN
O^k	Output of k^{th} rmNN, estimated degree of k^{th} type of fault
Err	The error between the target and rmNN output
α	Learning rate
LCs	Learning Cycles
LSE	Least Square Error

A form of rough neural computing based on rough sets and rough membership functions [45, 58, 59, 60] is introduced in this section. A rough membership function neural network (rmNN) has been designed and applied to classify power system faults [23, 24, 55].

7.1 Sample Information System For PSFC

The fault files recorded by Transcan^{TM} form the universe of events U. Table 12 and Table 13 in Section 6.2 represent the information system and have 17 features. The 17 features in the information table are sub-grouped into 11 feature sets, $B = \{B_1, B_2, ..., B_{11}\}$ (Table 19). The information system is then represented by (U, B).

Table 19: 11 Feature Sets.

B1	Feature/attribute 1
B2	Feature/attribute 2
B3	Feature/attribute 3, 4, 5, 6, 7
B4	Feature/attribute 8, 9, 10
B5	Feature/attribute 11
B6	Feature/attribute 12
B7	Feature/attribute 13
B8	Feature/attribute 14
B9	Feature/attribute 15
B10	Feature/attribute 16
B11	Feature/attribute 17

A simple information system containing sample fault events and feature set B_4 (Table 20) is discussed in this section to illustrate the rough set basic theory in the application of the power system. In this example, assume that U is a set of sample fault events. By way of approximation of a set of objects, consider $X \subseteq U$ defined as

X = $\{x|x$ is a fault event in the power system$\}$

= {F08101FE.x01, F1113009.x01, F0420695.x01, F2913FDD.x01, F1112E8D.x01, F2212CD7.x01, F223079B.x01, F0820165.x01, F112267F.x01, ...},

F = $\{f_8, f_9, f_{10}\}$, defined in Section 6.2, is a set of functions representing the feature set $B_4 = \{A_8, A_9, A_{10})$.

Table 20: Sample information system.

Events	B_4		
Fault file names	f_8	f_9	f_{10}
F08101FE.x01	2	2	2
F1113009.x01	2	2	2
F0420695.x01	2	2	21
F2913FDD.x01	2	2	1
F1112E8D.x01	212	212	212
F2212CD7.x01	21	21	21
F223079B.x01	212	212	212
F0820165.x01	2	2	2
F112267F.x01	21	21	21
F22225C4.x01	1	1	1
F20406CC.x01	2	2	2
F2713113.x01	2	2	2
F2713116.x01	2	2	2
F041075C.x01	1	2	2
F0820715.x01	1	1	1
F08226BF.x01	21	21	21
F0112939.x01	2	2	1
F1140866.x01	2	1	1
F1112BAE.x01	2	1	1
F2410189.x01	21	21	21
F1122499.x01	2	2	2
F2212F95.x01	2	2	2
F1121E5D.x01	2	2	2
F272015F.x01	2	2	2
F111302A.x01	2	2	2

The fault events and their associated fault types are listed in Table 21.

Table 21: Fault events and associated fault types.

Events	Decision
F08101FE.x01	Fault 1
F1113009.x01	Fault 1
F0420695.x01	Fault 1 and 3
F2913FDD.x01	Fault 1 and 3
F1112E8D.x01	Fault 1 and 4
F2212CD7.x01	Fault 1and 4
F223079B.x01	Fault 1 and 4
F0820165.x01	Fault 1 and 5
F112267F.x01	Fault 1 and 6
F22225C4.x01	Fault 1 and 6
F20406CC.x01	Fault 2
F2713113.x01	Fault 2
F2713116.x01	Fault 2
F041075C.x01	Fault 2 and 3
F0820715.x01	Fault 1 and 6 and 7
F08226BF.x01	Fault 2 and 6
F0112939.x01	Fault 3
F1140866.x01	Fault 3 and 5
F1112BAE.x01	Fault 3 and 5 and 9
F2410189.x01	Fault 4
F1122499.x01	Fault 5
F2212F95.x01	Fault 8
F1121E5D.x01	Fault 9
F272015F.x01	Fault 10
F111302A.x01	Fault 12

Notice that each of the events in class

[F08101FE.x01]$_{B_4}$ = $\{$F08101FE.x01, F1113009.x01, F0820165.x01, F20406CC.x01,
F2713113.x01, F2713116.x01, F1122499.x01, F2212F95.x01,
F1121E5D.x01, F272015F.x01, F111302A.x01$\}$,

has exactly the same B_4 output, namely, $\{2, 2, 2\}$ (Table 20).

The partition of U defined by the relation $\sim_{B_4}$ is as follows:

[F08101FE.x01]$_{B_4}$ = $\{$F08101FE.x01, F1113009.x01, F0820165.x01, F20406CC.x01,
F2713113.x01, F2713116.x01, F1122499.x01, F2212F95.x01,
F1121E5D.x01, F272015F.x01, F111302A.x01$\}$,

[F0420695.x01]$_{B_4}$ = $\{$F0420695.x01$\}$,

[F2913FDD.x01]$_{B_4}$ = $\{$F2913FDD.x01, F0112939.x01$\}$,

[F1112E8D.x01]$_{B_4}$ = $\{$F1112E8D.x01, F223079B.x01$\}$,

[F2212CD7.x01]$_{B_4}$ = $\{$F2212CD7.x01, F112267F.x01, F08226BF.x01, F2410189.x01$\}$,

[F22225C4.x01]$_{B_4}$ = $\{$F22225C4.x01, F0820715.x01$\}$,

[F041075C.x01]$_{B_4}$ = $\{$F041075C.x01$\}$,

[F1140866.x01]$_{B_4}$ = $\{$F1140866.x01, F1112BAE.x01$\}$.

Now select a particular set X, which contains all the events of fault 1: (Table 21), *i.e.*,

X = $\{$F08101FE.x01, F1113009.x01, F0420695.x01, F2913FDD.x01,
F1112E8D.x01, F2212CD7.x01, F223079B.x01, F0820165.x01,
F112267F.x01, F22225C4.x01, F0820715.x01$\}$.

This choice leads to the following lower and upper approximations of the set X.

$B_{4*}X$ = [F0420695.x01]$_{B_4}$ $\cup$ [F1112E8D.x01]$_{B_4}$ $\cup$ [F22225C4.x01]$_{B_4}$
= $\{$F0420695.x01, F1112E8D.x01, F223079B.x01, F22225C4.x01,
F0820715.x01$\}$,

B_4^*X = [F0420695.x01]$_{B_4}$ $\cup$ [F1112E8D.x01]$_{B_4}$ $\cup$ [F22225C4.x01]$_{B_4}$
$\cup$ [F08101FE.x01]$_{B_4}$ $\cup$ [F2913FDD.x01]$_{B_4}$ $\cup$ [F2212CD7.x01]$_{B_4}$

= {F0420695.x01, F1112E8D.x01, F223079B.x01, F22225C4.x01, F0820715.x01, F08101FE.x01, F1113009.x01, F0820165.x01, F20406CC.x01, F2713113.x01, F2713116.x01, F1122499.x01, F2212F95.x01, F1121E5D.x01, F272015F.x01, F111302A.x01, F2913FDD.x01, F0112939.x01, F2212CD7.x01, F112267F.x01, F08226BF.x01, F2410189.x01},

$Bnd_{B4}X = B_4{}^{*}X - B_{4*}X$

= [F08101FE.x01]$_{B_4}$ $\cup$ [F2913FDD.x01]$_{B_4}$ $\cup$ [F2212CD7.x01]$_{B_4}$

= {F08101FE.x01, F1113009.x01, F0820165.x01, F20406CC.x01, F2713113.x01, F2713116.x01, F1122499.x01, F2212F95.x01, F1121E5D.x01, F272015F.x01, F111302A.x01, F2913FDD.x01, F0112939.x01, F2212CD7.x01, F112267F.x01, F08226BF.x01, F2410189.x01}.

In effect, the lower approximation $B_{4*}X$ indicates that the events in [F0420695.x01]$_{B_4}$ $\cup$ [F1112E8D.x01]$_{B_4}$ $\cup$ [F22225C4.x01]$_{B_4}$ certainly are the members of set X. $B_{4*}X$ is called the "Yes" set in Section 7.2. Meanwhile, the non-empty boundary $Bnd_{B4}X$ indicates that set X is a rough set and the events in $Bnd_{B4}X$ might belong to set X. $Bnd_{B4}X$ is called the "YesOrNo" set in Section 7.2.

Next, consider the degree of overlap of class [F08101FE.x01]$_{B_4}$ with the set X, *i.e.*,

X = {F08101FE.x01, F1113009.x01, F0420695.x01, F2913FDD.x01, F1112E8D.x01, F2212CD7.x01, F223079B.x01, F0820165.x01, F112267F.x01, F22225C4.x01, F0820715.x01},

and

[F08101FE.x01]$_{B_4}$ = {F08101FE.x01, F1113009.x01, F0820165.x01, F20406CC.x01, F2713113.x01, F2713116.x01, F1122499.x01, F2212F95.x01, F1121E5D.x01, F272015F.x01, F111302A.x01}, where the degree of overlap is calculated using (65)

$$\mu_X^{B_4}(x) = \frac{|[F08101FE.x01]_{B_4} \cap X|}{|[F08101FE.x01]_{B_4}|} = \frac{3}{11} = 0.273. \tag{65}$$

This demonstrates that the degree to which the events in class [F08101FE.x01]$_{B_4}$ belong to X is 27.3%. This shows that fault events in class [F08101FE.x01]$_{B_4}$ and the faults in the set X are partially related.

7.2 Rough Membership Functions

A rough membership function (rm function) makes it possible to measure the degree to which any specified object belongs to a given set X. In the power fault classification system, there are 11 feature sets and 12 types of faults; the universe is divided into 132 Rough Sets. A mapping of these 132 Rough Sets and the information table is illustrated in Table 22.

Table 22: The mapping of 132 Rough Sets and information table.

	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11
Fault1	RS 11	RS 12	RS 13	RS 14	RS 15	RS 16	RS 17	RS 18	RS 19	RS 110	RS 111
Fault2	RS 21	RS 22	RS 23	RS 24	RS 25	RS 26	RS 27	RS 28	RS 29	RS 210	RS 211
Fault3	RS 31	RS 32	RS 33	RS 34	RS 35	RS 36	RS 37	RS 38	RS 39	RS 310	RS 311
Fault4	RS 41	RS 42	RS 43	RS 44	RS 45	RS 46	RS 47	RS 48	RS 49	RS 410	RS 411
Fault5	RS 51	RS 52	RS 53	RS 54	RS 55	RS 56	RS 57	RS 58	RS 59	RS 510	RS 511
Fault6	RS 61	RS 62	RS 63	RS 64	RS 65	RS 66	RS 67	RS 68	RS 69	RS 610	RS 611
Fault7	RS 71	RS 72	RS 73	RS 74	RS 75	RS 76	RS 77	RS 78	RS 79	RS 710	RS 711
Fault8	RS 81	RS 82	RS 83	RS 84	RS 85	RS 86	RS 87	RS 88	RS 89	RS 810	RS 811
Fault9	RS 91	RS 92	RS 93	RS 94	RS 95	RS 96	RS 97	RS 98	RS 99	RS 910	RS 911
Fault10	RS 101	RS 102	RS 103	RS 104	RS 105	RS 106	RS 107	RS 108	RS 109	RS 1010	RS 1011
Fault11	RS 111	RS 112	RS 113	RS 114	RS 115	RS 116	RS 117	RS 118	RS 119	RS 1110	RS 1111
Fault12	RS 121	RS 122	RS 123	RS 124	RS 125	RS 126	RS 127	RS 128	RS 129	RS 1210	RS 1211

RS_{ij} is a set of the i^{th} faults represented by function values for functions representing $a \in B_j$. For simplicity, fault file names will be replaced by sets of values of functions, *e.g.*, $11 of \{2, 2, 2\}$ representing class $[F08101FE.x01]_{B_4}$ as shown in Section 7.1.

Consider two sets, RS_{ij} and RS_{kj}, the intersection $RS_{ij} \cap RS_{kj}$ belongs to the "YesOrNo" set of RS_{ij} in the case where RS_{ij} and RS_{kj} have feature values in common. The elements that exist only in set RS_{ij} constitute what is known as the "Yes" set. In other words, each RS_{ij} set is divided into two sets, the "YesOrNo" set and the "Yes" set. Examples of "YesOrNo" sets of B4 to B11 in faults 1, 2, 3 and 4 are listed in Table 23. The complete tables of the "YesOrNo" set of all feature sets in all 12 faults are attached in Appendix B. The "Yes" sets of all feature sets in all 12 faults are listed in Table 24. The equivalence classes of all feature sets are listed in Tables 25 and 26.

Table 23: "YesOrNo" set of feature sets from B4 to B11 in faults 1, 2, 3 and 4.

YesOrNo SET								
	B4	B5	B6	B7	B8	B9	B10	B11
Fault 1	72 of {2 2 2 }; 10 of {2 2 21 }; 12 of {2 2 1 }; 1 of {212 2 2 }; 1 of {21 1 2 }; 7 of {1 2 2 }; 3 of {12 2 2 }; 6 of {21 2 2 }; 2 of {2 21 21 }; 4 of {2 21 1 }; 2 of {2 2 212 }; 1 of {212 21 2 }; 1 of {2 21 212 }; 1 of {2 2 1212 }; 1 of {2121 2 2 }; 1 of {2 212 12 }; 1 of {2 1 1 }; 5 of {2 2 2121 }; 4 of {2 21212 2 }; 4 of {2 21 2 }; 63 of {212 212 212 }; 21 of {21 21 21 }; 3 of {21212 21212 21212 }; 1 of {21 21 1 }; 1 of {21 21 2121 }; 9 of {1 1 1 }; 1 of {1 21 21 }; 1 of {21 1 21 }; 1 of {12 12 12 };	199 of 0; 3 of 111; 5 of 11; 7 of 110; 9 of 10; 8 of 100; 7 of 1; 2 of 101;	232 of 0; 8 of 1;	206 of 0; 34 of 1;	170 of 1; 70 of 0;	75 of 0; 79 of 1; 27 of 2; 10 of 4; 1 of 10; 22 of 3; 8 of 7; 3 of 6; 6 of 8; 4 of 28; 4 of 29; 1 of 5;	227 of 0; 9 of 1; 3 of 2; 1 of 6;	227 of 0; 13 of 1;
Fault 2	77 of {2 2 2 }; 6 of {1 1 1 }; 10 of {1 2 2 }; 29 of {2 2 1 }; 1 of {21212 21212 1 }; 1 of {21212 1 1 }; 1 of {212 212 1 }; 1 of {21 1 2 }; 1 of {12 21 2 }; 1 of {212 121 2 }; 1 of {2 1 2 }; 1 of {2 21 2 }; 10 of {21 21 21 }; 4 of {21212 21212 21212 }; 1 of {1 21 21 }; 1 of {21 1 1 }; 2 of {21 21 1 };	51 of 111; 57 of 0; 22 of 110; 4 of 10; 4 of 1; 7 of 11; 2 of 100; 1 of 101;	148 of 0;	128 of 0; 20 of 1;	97 of 1; 51 of 0;	80 of 1; 35 of 0; 7 of 3; 14 of 2; 3 of 6; 4 of 4; 2 of 8; 1 of 10; 1 of 5; 1 of 7;	126 of 0; 5 of 11; 2 of 32; 2 of 22; 4 of 2; 3 of 1; 1 of 6; 1 of 7; 1 of 3; 2 of 5; 1 of 4;	127 of 0; 21 of 1;
Fault 3	10 of {2 2 21 }; 49 of {2 2 1 }; 1 of {212 2 2 }; 2 of {21 1 2 }; 20 of {1 2 2 }; 3 of {12 2 2 }; 6 of {21 2 2 }; 3 of {2 21 21 }; 4 of {2 21 1 }; 2 of {2 2 212 }; 1 of {212 21 2 }; 1 of {2 21 212 }; 1 of {2 2 1212 }; 1 of {2121 2 2 }; 1 of {2 212 12 }; 7 of {2 1 1 }; 5 of {2 2 2121 }; 4 of {2 21212 2 }; 5 of {2 21 2 }; 1 of {21212 21212 1 }; 1 of {21212 1 1 }; 1 of {212 212 1 }; 1 of {12 21 2}; 1 of {212 121 2 }; 5 of {2 1 2 }; 2 of {12 1 2 }; 2 of {212 1 2 }; 1 of {2 1 21 };	75 of 0; 8 of 11; 27 of 110; 6 of 10; 22 of 100; 7 of 1;	143 of 0; 2 of 1;	145 of 0;	145 of 0;	41 of 0; 17 of 2; 53 of 1; 1 of 4; 1 of 10; 6 of 3; 7 of 7; 4 of 6; 7 of 8; 4 of 28; 4 of 29;	134 of 0; 6 of 1; 3 of 2; 1 of 6; 1 of 7;	134 of 0; 11 of 1;
Fault 4	3 of {2 2 2 }; 63 of {212 212 212 }; 12 of {21 21 21 }; 3 of {21212 21212 21212 };	81 of 0;	81 of 0;	69 of 0; 12 of 1;	81 of 1;	3 of 0; 31 of 1; 21 of 3; 16 of 2; 9 of 4; 1 of 5;	81 of 0;	81 of 0;

Table 24: “Yes” set of 11 feature sets in 12 faults.

Yes SET											
	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11
Fault1			{1 4 43 23 23 } {1 4 41 3 3241 } {1 4 434 3 343 } {1 4 41 23 21 } {1 4 41 23 21 } {4 43 4 3 3 } {4 41 4 321 34 } {1 4 4134 3 313 } {2 41 4 31 34 } {2 4 4 3 23 } {2 4314 4 3123 3 } {2 414 4 323 3 } {2 323 4 3 3 } {2 3 4 2 2 } {3 4 414 3 313 } {3 4 414 232 }								
Fault2			{2 3 434 3 3 } {2 323 434 2 2 } {1 434 43 23 23 } {1 434 4 2 2 } {4 4241 434 3141 34 } {4 41 4 341 34 } {4 3 4 34 34 } {2 434 434 3 343 } {2 434 4 32 32 } {3 4 4 3 3 } {1 414 434 232 12 } {1 4 434 323 3 } {1 4 3 2 2 } {1 4234 323 3 3 } {1 434 323 343 3 }								
Fault3			{1 34 23 21 21 } {1 3 312 2 2 } {1 3 2 2 212 } {2 3 23 21 21 } {1 3 4 3 32 } {2 2 3 3 3 } {1 13 4 12 2 } {1 3 12 2 12 } {2 3 12 2 12 } {2 3 212 2 212 } {4 4 313 3 3213 }	{12 12 1 } {1 1 12 } {21 1 12 } {1 212 212}							
Fault4			{1 41 41 31 31 }								
Fault5											
Fault6			{3 1 1 1 1 } {4 1 1 1 1 } {1 1 1 1 1 } {2 1 1 1 1 }								
Fault7											
Fault8			{1 4 4 1 2 } {2 4 4 1 2 } {2 4 4 1 2 } {1 4 4 1 2 } {1 4 4 1 2 }								
Fault9			{4 431 4 321 34 } {3 4 13 43 13 } {1 4 14 32 12 } {2 13 4 12 32 } {2 143 4 12 32 } {1 4 434 23 23 } {1 4 4 321 321 } {1 4 4 23 23 } {2 4 4 32 32 } {2 3 4 3 43 } {1 123 4 12 32 } {1 4 4 321 321 } {1 13 4 12 32 } {1 4 4 23 23 } {1 3 4 3 34 } {1 3 4 3 43 }								
Fault10			{2 31 4 21 2 } {2 143 4 12 2 } {4 3 4 3 3 } {2 41 4 21 2 } {1 4 143 2 12 } {1 4 31 2 21 } {1 4 3 3 3 } {1 4 123 2 12 } {1 4 4 3 2 } {1 4 42 2 32 } {1 4 31 2 21 }								
Fault11											
Fault12			{4 3 3 2 2 } {1 3 4 2 3 } {4 3 3 2 2 } {2 4 3 3 2 } {1 3 4 2 3 } {4 3 3 2 2 } {2 4 3 3 2 } {4 3 3 2 2 } {4 3 3 2 2 } {1 3 4 2 3 } {2 4 3 3 2 } {4 3 3 2 2 }								

Table 25: The equivalence classes of feature sets B1, B2 and B3.

Equivalence classes		
B1	**B2**	**B3**
700 of 0; 227 of 1;	495 of 1; 279 of 2; 153 of 0;	10 of {1 4 43 3 3 }; 7 of {1 4 1 3 1 }; 8 of {2 3 4 3 3 }; 6 of {4 4 4 3 3 }; 16 of {1 4 4 3 3 }; 3 of {1 4 41 3 31 }; 20 of {2 4 4 3 3 }; 3 of {4 41 4 31 34 }; 7 of {1 4 4 2 2 }; 9 of {1 434 4 3 3 }; 2 of {2 3 43 34 34 }; 2 of {1 3 43 3 3143 }; 14 of {1 3 43 3 3 }; 2 of {2 3 434 3 343 }; 60 of {1 3 4 3 3 }; 2 of {2 3 312 23 23 }; 8 of {1 3 4 2 2 }; 2 of {1 3 34 3 3 }; 8 of {1 3 3 2 2 }; 2 of {1 3 413 2 2 }; 2 of {2 3 34 3 3 }; 2 of {3 43 43 3 3 }; 2 of {4 43 43 3 3 }; 3 of {1 43 4 3 3 }; 27 of {1 323 4 3 3 }; 6 of {1 34 4 3 3 }; 3 of {3 43 4 3 3 }; 7 of {3 3 4 3 3 }; 4 of {2 4 434 2 2 }; 3 of {1 43 4 3 23 }; 3 of {4 4 43 3 3 }; 3 of {1 312 43 23 23 }; 3 of {2 3 413 2 2 };
		4 of {2 4 43 2 2 }; 4 of {2 3 34 2 2 }; 4 of {2 4 43 3 34 }; 3 of {2 4 4 1 1 }; 3 of {2 4 2 3 1 }; 3 of {2 4 434 3 1 }; 3 of {2 4 43 1 343 }; 3 of {2 3 43 34 3134 }; 3 of {2 3 43 34 3234 }; 3 of {2 4 43 3 3 }; 3 of {2 4 42 3 3 }; 3 of {2 4 43 34 34 }; 3 of {2 4 42 34 34 }; 6 of {1 3 42 3 3 }; 6 of {1 4 424 3 3 }; 9 of {1 4 434 3 3 }; 3 of {1 3 43 34 3134 }; 3 of {1 3 43 34 3234 }; 3 of {1 4 42 3 3 }; 3 of {1 4 43 34 34 }; 3 of {1 4 42 34 34 }; 4 of {2 431 4 321 34 }; 2 of {2 414 4 313 3 }; 20 of {1 414 4 313 3 }; 8 of {1 414 4 3123 3 }; 2 of {1 414 4 2 2 }; 2 of {1 431 4 31 3 }; 2 of {1 41 4 31 3 }; 2 of {1 431 4 31 34 }; 4 of {1 4134 4 3123 3 }; 4 of {1 4134 4 323 3 }; 2 of {1 414 4 232 2 }; 2 of {1 414 4 212 2 }; 2 of {1 414 4 2132 2 }; 2 of {1 414 4 3213 3 }; 2 of {1 4314 4 3213 3 }; 2 of {2 4 414 2 212 }; 2 of {2 4 4134 3 3143 }; 2 of {2 4 41 34 31 }; 2 of {3 431 4 321 34 }; 36 of {3 414 4 313 3 }; 16 of {3 414 4 3123 3 }; 2 of {3 414 4 3143 3 }; 2 of {3 41 4 31 243 }; 4 of {3 41 4 31 34 }; 2 of {3 41 4 31 3 }; 2 of {3 431 4 31 34 }; 2 of {3 424 4 3134 34 }; 2 of {3 424 4 313 3 }; 2 of {3 414 4 212 2342 };
		2 of {3 414 4 212 232 }; 2 of {3 414 4 212 242 }; 2 of {3 414 4 212 2432 }; 2 of {3 414 3 313 3 }; 4 of {4 4 414 3 313 }; 4 of {4 4 414 232 212 }; 2 of {1 4314 4 313 3 }; 2 of {1 414 4 2312 2 }; 2 of {1 4 2 3 2 }; 9 of {2 4 4 2 2 }; 3 of {2 4 434 3 3 }; 2 of {1 31 4 231 2 }; 2 of {2 3 41 34 31 }; 2 of {2 4 431 3 1 }; 2 of {2 4 431 23 1 }; 4 of {2 4 41 23 21 }; 2 of {1 41 4 21 23 }; 2 of {2 41 1 41 31 }; 2 of {1 31 31 31 31 }; 2 of {2 41 41 431 431 }; 2 of {2 321 31 31 321 }; 25 of {2 4 1 3 1 }; 3 of {4 4 41 343 31 }; 3 of {2 4 1 343 1 }; 4 of {2 4 41 1 1 }; 3 of {2 4 41 1 21 }; 3 of {2 4 41 1 31 }; 3 of {2 4 431 1 341 }; 9 of {2 4 14 32 1 }; 2 of {1 4 43 1 343 }; 3 of {2 434 4 3 3 }; 2 of {1 3 32 3213 3 }; 2 of {3 323 4 34 34 }; 2 of {2 313 313 323 323 }; 18 of {2 3 3 2 2 }; 2 of {1 3 41 3 341 }; 2 of {3 313 414 3413 3413 }; 3 of {1 313 434 2 2 }; 3 of {1 3 434 2 2 }; 3 of {1 3 434 1 1 }; 6 of {1 3 323 2 2 }; 3 of {1 313 3123 323 323 }; 6 of {1 3 434 3 3 }; 3 of {2 4 323 3 3 }; 3 of {1 323 4 32 32 }; 3 of {1 323 434 32 32 }; 3 of {1 3 31 3 31 }; 3 of {1 3 21 23 2341 };
		8 of {2 3 323 2 2 }; 4 of {2 323 32 23 23 }; 5 of {2 3 313 2 2 }; 3 of {2 4 43 23 23 }; 2 of {1 41 1 321 31 }; 2 of {1 434 434 3 3 }; 2 of {2 434 434 343 3 }; 2 of {1 434 1 343 1 }; 2 of {1 4 1 343 1 }; 5 of {1 4 4 32 32 }; 8 of {4 313 414 3413 3413 }; 2 of {2 3 41 34 341 }; 2 of {3 431 431 31 321 }; 2 of {4 431 431 321 321 }; 2 of {3 41 4 431 4 }; 2 of {3 41 4 341 34 }; 4 of {2 4 41 34 321 }; 2 of {2 4 421 3 1 }; 6 of {2 3 1 3 1 }; 3 of {3 41 4 321 34 }; 3 of {2 3 31 4 431 }; 3 of {2 3 1 23 1 }; 3 of {2 4 21 3 321 }; 3 of {2 4 31 1 31 }; 7 of {1 4 31 1 31 }; 4 of {4 4 4 32 32 }; 3 of {2 3 4 3 34 }; 6 of {2 3 3 3 3 }; 3 of {4 4 3 3 3413 }; 3 of {4 4 3 3 313 }; 6 of {1 4 3 3 323 }; 3 of {1 4 3 3 3413 }; 3 of {1 4 3 3 3423 }; 6 of {2 4 1 1 1 }; 4 of {1 14 4 1 32 }; 4 of {1 4 1 1 1 }; 4 of {1 4 14 32 1 };

Table 26: The equivalence classes of feature sets B4 to B11.

Equivalence classes							
B4	**B5**	**B6**	**B7**	**B8**	**B9**	**B10**	**B11**
253 of {2 2 2 }; 29 of {2 2 21 }; 122 of {2 2 1 }; 2 of {212 2 2 }; 6 of {21 1 2 }; 47 of {1 2 2 }; 8 of {12 2 2 }; 17 of {21 2 2 }; 6 of {2 21 21 }; 12 of {2 21 1 }; 6 of {2 2 212 }; 3 of {212 21 2 }; 4 of {2 21 212 }; 4 of {2 2 1212 }; 4 of {2121 2 2 }; 3 of {2 212 12 }; 21 of {2 1 1 }; 15 of {2 2 2121 }; 12 of {2 21212 2 }; 15 of {2 21 2 }; 126 of {212 212 212 }; 70 of {21 21 21 }; 14 of {21212 21212 21212 }; 8 of {21 21 1 }; 2 of {21 21 2121 }; 54 of {1 1 1 }; 5 of {1 21 21 }; 2 of {21 1 21 }; 17 of {12 12 12 }; 2 of {21212 21212 1 }; 2 of {21212 1 1 }; 3 of {212 212 1 }; 4 of {12 21 2 }; 3 of {212 121 2 }; 11 of {2 1 2 }; 3 of {21 1 1 }; 3 of {12 1 2 }; 3 of {212 1 2 }; 2 of {2 1 21 };	614 of 0; 56 of 111; 32 of 11; 88 of 110; 36 of 10; 62 of 100; 32 of 1; 7 of 101;	874 of 0; 53 of 1;	783 of 0; 144 of 1;	536 of 1; 391 of 0;	268 of 0; 356 of 1; 110 of 2; 27 of 4; 6 of 10; 68 of 3; 25 of 7; 15 of 6; 22 of 8; 12 of 28; 12 of 29; 6 of 5;	830 of 0; 41 of 1; 23 of 2; 6 of 6; 6 of 11; 3 of 32; 3 of 22; 3 of 7; 3 of 3; 6 of 5; 3 of 4;	832 of 0; 95 of 1;

For a fault file, if its output of B_j falls in the "YesOrNo" set of RS_{ij}, it indicates that the file could represent an i^{th} fault; and the degree of i^{th} fault will be estimated based on the rough membership function in (66).

$$\mu_X^B(x) = \frac{|[x]_B \cap X|}{|[x]_B|}. \tag{66}$$

As an example, the rough membership calculation for the fault file F0822405.x01 is explained in (67). For fault file F0822405.x01, the feature set B_4 gives an output of {2, 2, 21}. Since set {2, 2, 21} is identified in the "YesorNo" set of RS_{34} 10 times, the fault is possibly a fault 3. In (66), the equivalence class $[x]_B$ is substituted by $[F0822405.x01]_{B_4}$, where x is represented by fault file F0822405.x01 and B is replaced with B_4. Meanwhile, X is replaced by RS_{34}. $[x]_{B_4}$ has 29 elements. Then the degree of overlap between $[x]_{B_4} = [F0822405.x01]_{B_4}$ and RS_{34} can be estimated by (67).

$$\mu_{RS_{34}}^{B_4}(x) = \frac{|[F0822405.x01]_{B_4} \cap RS_{34}|}{|[F0822405.x01]_{B_4}|} = \frac{10}{29} = 34.4\%. \tag{67}$$

It demonstrates that the degree to which class $[F0822405.x01]_{B_4}$ belongs to fault 3 is 34.4%.

7.3 Rough Membership Tables for rmNN Training and Verification

By simply repeating the rough membership computation procedure described in Section 7.2, 11 degrees to which the file F0822405.x01 belongs to fault 3 are obtained based on RS_{3j} ($1 \le j \le 11$) and $[F0822405.x01]_{B_j}$ $(1 \le j \le 11)$. The 11 degrees of membership are represented as a vector, *e.g.*, (0.19, 0.14, 0.33, 0.34, 0.12, 0.16, 0.19, 0.37, 0.28, 0.16, 0.16). In addition, for a fault file *.x01, the membership for each type of fault based on each feature set is derived to transform the training information table (Table 12) and testing information table (Table 13) into 12 rough membership training tables and 12 rough membership testing tables, respectively. One of these rough membership tables is shown in Table 27. The first column indicates the file index. The following 11 columns contain 11 rough memberships for fault 3. The last column is the target that indicates whether it is a fault 3 or not. The value of 1 is for “Yes” and 0.01 is for “No”. Each row in Table 27 is employed as a training set to calibrate the rmNN for fault 3. Table 28 is the rough membership table for fault 3 rmNN verification. It is necessary to point out that both Table 27 and 28 are partial rough membership tables. Twelve complete rough membership training tables as well as 12 testing tables are listed in Appendix C.

Table 27: Partial rough membership table for fault 3 rmNN training.

File Index	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	Target
	11	1	508	1								
1	0.19	0.14	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
4	0.19	0.14	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
7	0.19	0.14	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
31	0.19	0.14	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
55	0.19	0.14	0.5	0.34	0.12	0.16	0.19	0.37	0.15	0.16	0.16	1
69	0.19	0.14	0.5	0.4	0.12	0.16	0.19	0.37	0.15	0.16	0.16	1
70	0.19	0.14	0.33	0.35	0.25	0.16	0.19	0.37	0.15	0.16	0.16	1
91	0.19	0.14	0.25	0.25	0.35	0.16	0.19	0.37	0.15	0.15	0.12	1
92	0.19	0.14	0.25	0.25	0.17	0.16	0.19	0.37	0.15	0.15	0.12	1
93	0.19	0.14	0.25	0.25	0.17	0.16	0.19	0.37	0.17	0.16	0.16	1
99	0.19	0.14	0.33	0.34	0.12	0.16	0.19	0.37	0.28	0.16	0.16	1
103	0.19	0.14	0.33	0.33	0.12	0.16	0.19	0.37	0.27	0.16	0.16	1
104	0.19	0.14	0.36	0.34	0.12	0.16	0.19	0.37	0.32	0.16	0.16	1
128	0.19	0.14	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
131	0.05	0.14	0	0	0.12	0.16	0	0	0.15	0.16	0.16	0.01
141	0.19	0.14	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
150	0.05	0.14	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
162	0.05	0.14	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
202	0.19	0.14	0	0	0	0.16	0.19	0	0.15	0.16	0.16	0.01
209	0.19	0.14	0	0	0.17	0.16	0.19	0	0.15	0.16	0.16	0.01
219	0.19	0.14	0	0	0.12	0.16	0	0	0.15	0.16	0.16	0.01
222	0.19	0.14	0	0	0.12	0.16	0	0	0.15	0.16	0.16	0.01
229	0.19	0.14	0	0	0.12	0.16	0	0	0.15	0.15	0.12	0.01
236	0.19	0.14	0	0	0.12	0.04	0	0	0.15	0.16	0.16	0.01
239	0.05	0.14	0	0	0.12	0.04	0.19	0	0.15	0.16	0.16	0.01
252	0.19	0.17	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
253	0.19	0.17	0	0	0	0.16	0.19	0	0.15	0	0.12	0.01
254	0.19	0.17	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
265	0.19	0.17	0	0	0	0.16	0.19	0	0.15	0.16	0.16	0.01
267	0.19	0.17	0	0	0	0.16	0.19	0	0.15	0.16	0.16	0.01
305	0.19	0.17	0.5	0.43	0.12	0.16	0.19	0.37	0.15	0.16	0.16	1
349	0.19	0.17	0.4	0.33	0.12	0.16	0.19	0.37	0.15	0.17	0.12	1
374	0.19	0.17	0	0	0.12	0.16	0	0	0.15	0	0.12	0.01
391	0.19	0.2	1	0.4	0.12	0.16	0.19	0.37	0.15	0.16	0.16	1
392	0.19	0.2	1	0.67	0.12	0.16	0.19	0.37	0.15	0.16	0.16	1
403	0.19	0.2	1	1	0.12	0.16	0.19	0.37	0.15	0.16	0.16	1
415	0.19	0.2	0.33	0.33	0.35	0.16	0.19	0.37	0.15	0.16	0.16	1
420	0.05	0.2	0	0	0.12	0.16	0	0	0	0.16	0.16	0.01
421	0.19	0.2	0.25	0	0.31	0.16	0.19	0	0.15	0.16	0.16	0.01
422	0.19	0.2	0	0	0.17	0.16	0.19	0	0.15	0.16	0.16	0.01
440	0.19	0.2	0	0	0.12	0.04	0.19	0	0.15	0.16	0.16	0.01
441	0.19	0.2	0	0	0.12	0.04	0.19	0	0.15	0.16	0.16	0.01
447	0.19	0.2	0	0	0.12	0.16	0.19	0	0.09	0.16	0.16	0.01
448	0.19	0.2	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
465	0.19	0.2	0	0	0.12	0.16	0.19	0	0.09	0.16	0.16	0.01
466	0.19	0.2	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
472	0.19	0.2	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
475	0.19	0.2	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
484	0.05	0.2	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
490	0.05	0.2	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
504	0.05	0.2	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
507	0.05	0.2	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
508	0.05	0.2	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01

Table 28: Partial rough membership table for fault 3 rmNN testing.

File Index	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	Target
	11	1	168	1								
1	0.19	0.14	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
19	0.19	0.14	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
30	0.19	0.14	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
46	0.19	0.14	0.4	0.4	0.12	0.16	0.19	0.37	0.15	0.16	0.16	1
53	0.19	0.14	0.25	0.25	0.17	0.16	0.19	0.37	0.15	0.15	0.12	1
54	0.19	0.14	0.4	0.4	0.31	0.16	0.19	0.37	0.15	0.16	0.16	1
56	0.19	0.14	0.33	0.33	0.12	0.04	0.19	0.37	0.15	0.16	0.16	1
57	0.19	0.14	0.33	0.35	0.12	0.16	0.19	0.37	0.28	0.16	0.16	1
63	0.19	0.14	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
82	0.19	0.14	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
86	0.19	0.14	0	0	0.22	0.16	0.19	0	0.15	0.16	0.16	0.01
87	0.19	0.14	0	0	0.22	0.16	0.19	0	0.15	0.16	0.16	0.01
90	0.19	0.14	0	0	0.12	0.16	0	0	0.28	0.16	0.16	0.01
91	0.19	0.14	0	0	0.12	0.16	0	0	0.15	0.15	0.12	0.01
93	0.19	0.14	0	0	0.12	0.04	0	0	0.15	0.16	0.16	0.01
94	0.19	0.14	0	0	0.12	0.04	0.19	0	0.15	0.16	0.16	0.01
111	0.05	0.17	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
112	0.19	0.17	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
116	0.19	0.17	0.4	0.4	0.12	0.16	0.19	0.37	0.15	0.16	0.16	1
120	0.19	0.17	0.39	0.43	0.12	0.16	0.19	0.37	0.15	0.16	0.16	1
121	0.19	0.17	0.4	0.4	0.31	0.16	0.19	0.37	0.15	0.16	0.16	1
127	0.19	0.17	0.4	0.4	0.35	0.16	0.19	0.37	0.15	0.16	0.16	1
128	0.19	0.17	0.39	0.25	0.22	0.16	0.19	0.37	0.09	0.15	0.12	1
129	0.05	0.17	0	0	0.12	0.16	0	0	0.15	0.16	0.16	0.01
130	0.19	0.17	0	0	0.22	0.16	0.19	0	0.15	0.16	0.16	0.01
131	0.19	0.17	0	0	0.17	0.16	0.19	0	0.15	0.16	0.16	0.01
132	0.19	0.17	0	0	0.12	0.16	0	0	0.32	0.16	0.16	0.01
134	0.19	0.17	0	0	0.12	0.16	0	0	0.15	0.16	0.16	0.01
135	0.19	0.17	0	0	0.12	0.16	0	0	0.15	0.13	0.12	0.01
136	0.19	0.17	0	0	0.12	0.16	0	0	0.15	0	0.12	0.01
137	0.19	0.17	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
138	0.19	0.17	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
139	0.19	0.2	1	0.67	0.12	0.16	0.19	0.37	0.15	0.16	0.16	1
141	0.19	0.2	1	1	0.12	0.16	0.19	0.37	0.15	0.16	0.16	1
144	0.19	0.2	0.39	0.67	0.22	0.16	0.19	0.37	0.15	0.16	0.16	1
145	0.19	0.2	0.33	0.33	0.35	0.16	0.19	0.37	0.15	0.16	0.16	1
146	0.05	0.2	0	0	0.12	0.16	0	0	0	0.16	0.16	0.01
147	0.19	0.2	0	0	0.12	0.16	0	0	0.15	0.16	0.16	0.01
148	0.05	0.2	0	0	0.12	0.04	0.19	0	0.15	0.16	0.16	0.01
150	0.19	0.2	0	0	0.12	0.04	0	0	0.15	0.16	0.16	0.01
151	0.19	0.2	0	0	0.12	0.04	0.19	0	0.15	0.16	0.16	0.01
152	0.19	0.2	0	0	0.12	0.04	0.19	0	0.15	0.16	0.16	0.01
153	0.19	0.2	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
156	0.19	0.2	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
158	0.19	0.2	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
161	0.19	0.2	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
167	0.05	0.2	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01
168	0.05	0.2	0	0	0.12	0.16	0.19	0	0.15	0.16	0.16	0.01

7.4 Design of rmNNs for PSFC

Neural networks are collections of massively parallel computation units called neurons. A neuron is a processing element in a neural network. To design a rough membership neural network, the rough neurons should first be defined.

7.4.1 Concepts of Rough Neurons

Typically, a neuron y maps its weighted input from R_n to [0, 1]. A selection of different types of neurons is given in Table 29: common neurons, rough neurons, fuzzy neurons.

Table 29: Different types of neurons.

Common Neuron	Rough membership Neuron	Fuzzy Neurons
$y(x) = g(\sum_i \mathbf{w}_i * \mathbf{x}_i + \nu)$, where input x_i has connection (weight) w_i, which denotes a modifiable neural connection and bias ν, $g(\bullet)$ is called activation function which can be a sigmoidal function such as logistic, tanh and Gaussian function	$y(x) = rmf(x)$ $= \mu_X^B(x) = \frac{\lvert [x]_B \cap X \rvert}{\lvert [x]_B \rvert}$	$y(x1, x2) =$ t-norm $(x1, x2)$ $= x1 * x2$ $y(x1, x2) =$ s-norm $(x1, x2)$ $= x1 + x2 - x1 * x2$ $y(r, x) = r \to x$ $= \begin{cases} \min(1, \frac{x}{r}) = \frac{x}{r} & \text{if } r \geq x \\ 1, & \text{otherwise} \end{cases}$

In the design of the rough membership function neural network (rmNN), the hidden layer consists of fuzzy neurons defined using the t-norm, s-norm and $\to$ (imply operators) from fuzzy set theory. The formal definition for a hidden neuron in an rmNN is given in (68) using the t-norm, s-norm and $\to$ operators. The reason that this form of hidden neuron is applied is because it provides a numerical representation of set intersection (t-norm), set union (s-norm) and implication ($\to$) that works well as a means of aggregating the input from the rough membership functions in the input layer. Let $B, X, [x]_B$ denote a set of features, a set of files with matching fault type based on knowledge, and an equivalence class derived from known objects, respectively. The basic computation steps performed by a rough membership neuron are reflected in the flow chart in Figure 56.

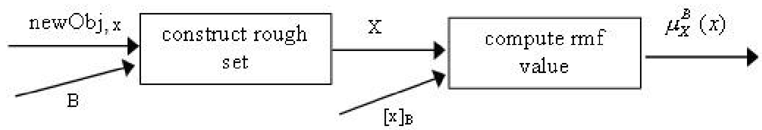


Figure 56: Flow chart for basic Rough Neuron computation.

In Figure 56 x is a newObj, an unclassified fault file. A rough neuron measures the degree of overlap of sets $[x]_B$ and X, and represents certain as well as uncertain classification of the input newObj, x.

7.4.2 Architecture of the Rough Membership Neural Network (rmNN) for PSFC

The architecture of a rmNN for fault classification is dependent on the number of types of available faults. Each fault will have its own rmNN. In the research reported in this thesis, 12 separate rmNNs are employed to classify 12 types of faults. Each rmNN will output an estimation of the degree of one type of fault for a given object. For instance, the output neuron of the k^{th} rmNN will aggregate all contributions from the rough neurons in the first layer, process in the hidden layer and finally output an estimation of the degree of k^{th} fault. The output from 12 rmNNs will be forwarded to a fault type decider neuron. The decider neuron simply picks up all the faults with degree values above 80%. Almost 50% of the power system fault events are multiple faults jointed.

The whole computational framework that contains 12 rmNNs is shown in Figure 57. An example of the k^{th} rmNN is shown in Figure 58, where $rmf^k(B_i(obj_n))$ is the rough membership for fault k based on the output of attribute set B_i for a given object n. The interconnections, ***i.e.***, r_{ij}, w_{ij} and u_j inside the rmNN are shown in Figure 59. There are 11 rough neurons in the first layer and 11 fuzzy neurons in the hidden layer. Each neuron in the first layer is fully connected to the neuron in the hidden layer and each neuron in the hidden layer is fully connected to the output neuron. $rmf^k(B_i(obj_n))$ is simplified as rmf_i^k in Figure 59 as well as in the weights updating formulas.

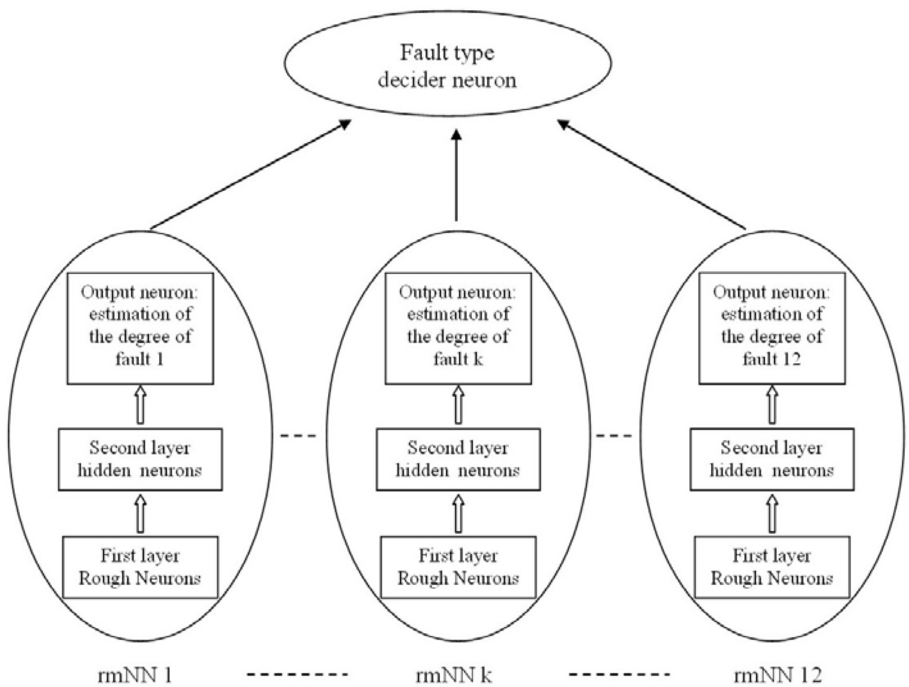


Figure 57: Diagram of connection of 12 rmNNs.

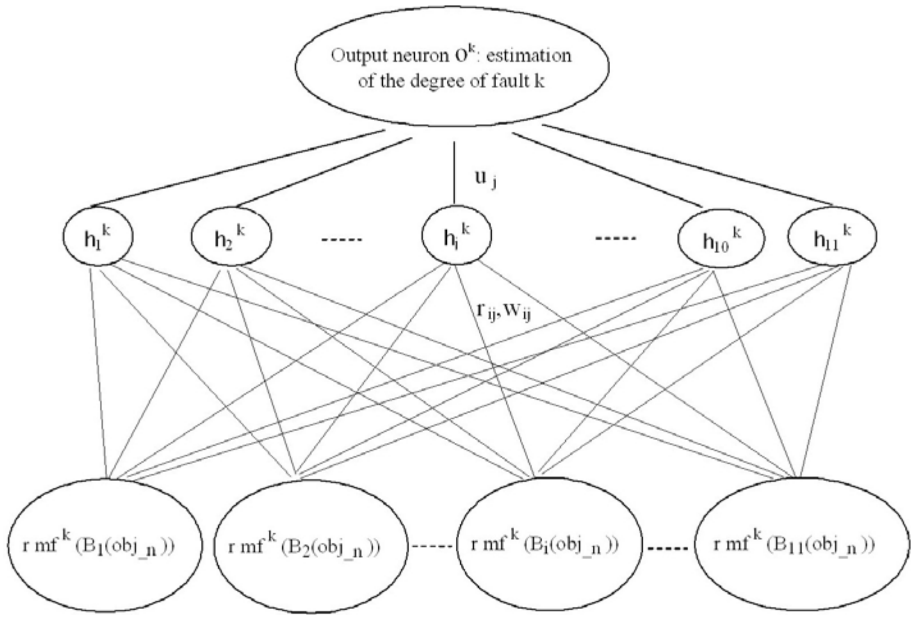


Figure 58: Diagram of rmNN for k^{th} fault classification.

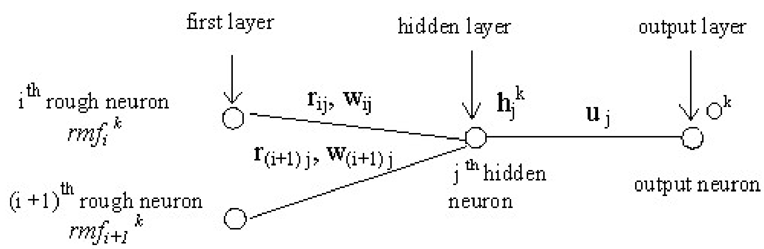


Figure 59: Interconnection of the rmNN for fault k.

The formula for the activation function of the hidden neuron and the output neuron is described in (68) and (71) respectively. The formulas here are with respect to the k^{th} rmNN. The t-$norm$ is defined in (68). The imply operator $\rightarrow$ is defined in (69). The s-$norm$, namely probability sum, is defined in (70).

$$\begin{aligned} h_j^k &= t_{i=1}^{11}[r_{ij} \rightarrow rmf_i^k \; s \; w_{ij}] \\ &= [(r_{1j} \rightarrow rmf_1^k) \; s \; w_{1j}][(r_{2j} \rightarrow rmf_2^k) \; s \; w_{2j}]...[(r_{11j} \rightarrow rmf_{11}^k) \; s \; w_{11j}], \end{aligned} \quad (68)$$

where

$$\begin{aligned}(r_{ij} \rightarrow rmf_i^k) &\equiv min\left(1, \frac{rmf_i^k}{r_{ij}}\right) \\ &= \begin{cases} \frac{rmf_i^k}{r_{ij}}, & if \;\; r_{ij} \geq rmf_i^k, \\ 1, & otherwise. \end{cases} \end{aligned} \quad (69)$$

$$(r_{ij} \rightarrow rmf_i^k) \; s \; w_{ij} = (r_{ij} \rightarrow rmf_i^k) + w_{ij} - (r_{ij} \rightarrow rmf_i^k) w_{ij}. \quad (70)$$

$$O^k = s_{j=1}^{11}[h_j^k u_j]. \quad (71)$$

7.4.3 Weights Adjustment with a Fixed Step Size Gradient Algorithm

In the process of tuning weights in the k^{th} rmNN, r_{ij}, w_{ij} and u_j will be updated by the partial derivative of the error. The error is the square of the difference between the target and the output of the k^{th} rmNN.

$$\begin{aligned} Error &= \left(target^k - O^k\right)^2 \\ &= \left(target^k - s_{j=1}^{11}\left(\left(t_{i=1}^{11}\left(\left(r_{ij} \rightarrow rmf_i^k\right) \; s \; w_{ij}\right)\right) u_j\right)\right)^2 \\ &= F\left(r_{ij}, w_{ij}, u_j\right), \end{aligned} \quad (72)$$

and

$$\begin{aligned} u_j^{(new)} &= u_j^{(old)} - \alpha \frac{\partial Error}{\partial u_j}, \\ w_{ij}^{(new)} &= w_{ij}^{(old)} - \alpha \frac{\partial Error}{\partial w_{ij}}, \\ r_{ij}^{(new)} &= r_{ij}^{(old)} - \alpha \frac{\partial Error}{\partial r_{ij}}, \end{aligned} \quad (73)$$

where α is the learning rate. The effects of the learning rate are discussed in Section 7.6. $u_j^{(new)}$ will be updated by the partial derivative of the error function with respect of $u_j^{(old)}$ in (73).

$$\begin{aligned} \frac{\partial Error}{\partial u_i} &= -\left(target^k - O^k\right) \frac{\partial O^k}{\partial u_i} \\ &= -\left(target^k - O^k\right) \frac{\partial}{\partial u_i} \left[s_{j=1}^{11}[h_j^k u_j]\right], \end{aligned} \quad (74)$$

where i = 1, 2, ... 11. The overall expression can be rewritten by separating the i^{th} component in the overall s-norm composition,

$$\begin{aligned} \frac{\partial O^k}{\partial u_i} &= \frac{\partial}{\partial u_i} \left[A + u_i h_i^k - A u_i h_i^k\right] \\ &= h_i^k (1 - A), \end{aligned} \quad (75)$$

where factor A summarizes the remaining components of the s-norm composition, *i.e.*,

$$A = s_{j=1, j \neq i}^{11} \left[h_j^k u_j\right]. \quad (76)$$

The computation of the connections between the input layer and the hidden layer, *i.e.*, w_{ij} given by the second formula in (73) requires the use of the chaining rule of differentiation. This implies the following,

$$\frac{\partial Error}{\partial w_{ij}} = -\left(target^k - O^k\right) \frac{\partial O^k}{\partial h_j^k} \frac{\partial h_j^k}{\partial w_{ij}}, \quad (77)$$

where w_{ij} refers to the connection from the i^{th} node in the first layer to the j^{th} node in the hidden layer. The $\frac{\partial O^k}{\partial h_j^k}$ factor is expressed as,

$$\frac{\partial O^k}{\partial h_j^k} = u_j(1 - A). \tag{78}$$

For factor $\frac{\partial h_j^k}{\partial w_{ij}}$, the activation function will be applied governing the hidden neuron,

$$\begin{aligned}
\frac{\partial h_j^k}{\partial w_{ij}} &= \frac{\partial}{\partial w_{ij}} \left(t_{l=1}^{11} \left[\left(r_{lj} \rightarrow rmf_l^k \right) \; s \; w_{lj} \right] \right) \\
&= \frac{\partial}{\partial w_{ij}} \left[\left(r_{ij} \rightarrow rmf_i^k \right) \; s \; w_{ij} \right] \left(t_{l=1, l \neq i}^{11} \left[\left(r_{lj} \rightarrow rmf_l^k \right) \; s \; w_{lj} \right] \right). \qquad (79)
\end{aligned}$$

By introducing the notation

$$B = t_{l=1, l \neq i}^{11} \left[\left(r_{lj} \rightarrow rmf_l^k \right) \; s \; w_{lj} \right], \tag{80}$$

(79) can be rewritten as

$$\begin{aligned}
\frac{\partial h_j^k}{\partial w_{ij}} &= \frac{\partial}{\partial w_{ij}} \left(B \left[\left(r_{ij} \rightarrow rmf_i^k \right) \; s \; w_{ij} \right] \right) \\
&= B \frac{\partial}{\partial w_{ij}} \left[\left(r_{ij} \rightarrow rmf_i^k \right) \; s \; w_{ij} \right] \\
&= B \frac{\partial}{\partial w_{ij}} \left(\left(r_{ij} \rightarrow rmf_i^k \right) + w_{ij} - \left(r_{ij} \rightarrow rmf_i^k \right) w_{ij} \right) \\
&= B \left(1 - \left(r_{ij} \rightarrow rmf_i^k \right) \right). \qquad (81)
\end{aligned}$$

The same procedure applies to update the r_{ij} parameter by computing

$$\frac{\partial Error}{\partial r_{ij}} = -\left(target^k - O^k \right) \frac{\partial O^k}{\partial h_j^k} \frac{\partial h_j^k}{\partial r_{ij}}, \tag{82}$$

where r_{ij} refers to the connection from the i^{th} node in the first layer to the j^{th} node in the hidden layer. The $\frac{\partial O^k}{\partial h_j^k}$ factor is expressed in (78), which is the same thing when updating w_{ij}.

$$
\begin{aligned}
\frac{\partial h_j^k}{\partial r_{ij}} &= \frac{\partial}{\partial r_{ij}} \left(B\left[\left(r_{ij} \rightarrow rmf_i^k\right)\ s\ w_{ij}\right]\right) \\
&= B\frac{\partial}{\partial r_{ij}} \left[\left(r_{ij} \rightarrow rmf_i^k\right)\ s\ w_{ij}\right] \\
&= B\frac{\partial}{\partial r_{ij}} \left(\left(r_{ij} \rightarrow rmf_i^k\right) + w_{ij} - \left(r_{ij} \rightarrow rmf_i^k\right) w_{ij}\right) \\
&= B\left(1 - w_{ij}\right) \frac{\partial\left(r_{ij} \rightarrow rmf_i^k\right)}{\partial r_{ij}},
\end{aligned} \tag{83}
$$

and

$$
\begin{aligned}
\frac{\partial\left(r_{ij} \rightarrow rmf_i^k\right)}{\partial r_{ij}} &= \frac{\partial}{\partial r_{ij}} \begin{cases} \frac{rmf_i^k}{r_{ij}}, & if\ \ r_{ij} \geq rmf_i^k, \\ 1, & otherwise. \end{cases} \\
&= \begin{cases} -\frac{rmf_i^k}{r_{ij}^2}, & if\ \ r_{ij} \geq rmf_i^k, \\ 0, & otherwise. \end{cases}
\end{aligned} \tag{84}
$$

The program flowchart for the rmNN calibration is illustrated in Figure 60.

7.4.4 Calibration Results of Selected rmNNs for PSFC

By way of illustration, only the calibration and verification results for fault 3 and 5 rmNNs are illustrated here. The results of all 12 rmNNs are attached in Appendix D.

The error output during the training of the fault 3 and fault 5 rmNNs is shown in Figures 61 and 62 respectively. After 200 learning cycles, the error between the target and the output of fault 3 rmNN is less than 0.8 and of fault 5 rmNN is less than 0.2.

The output for fault 3 and 5 rmNNs compared to their targets after the first learning cycle is shown in Figures 63 and 64 respectively. Figures 65 and 66 illustrate that after 200 learning cycles, the output for fault 3 and 5 rmNNs matches their targets well.

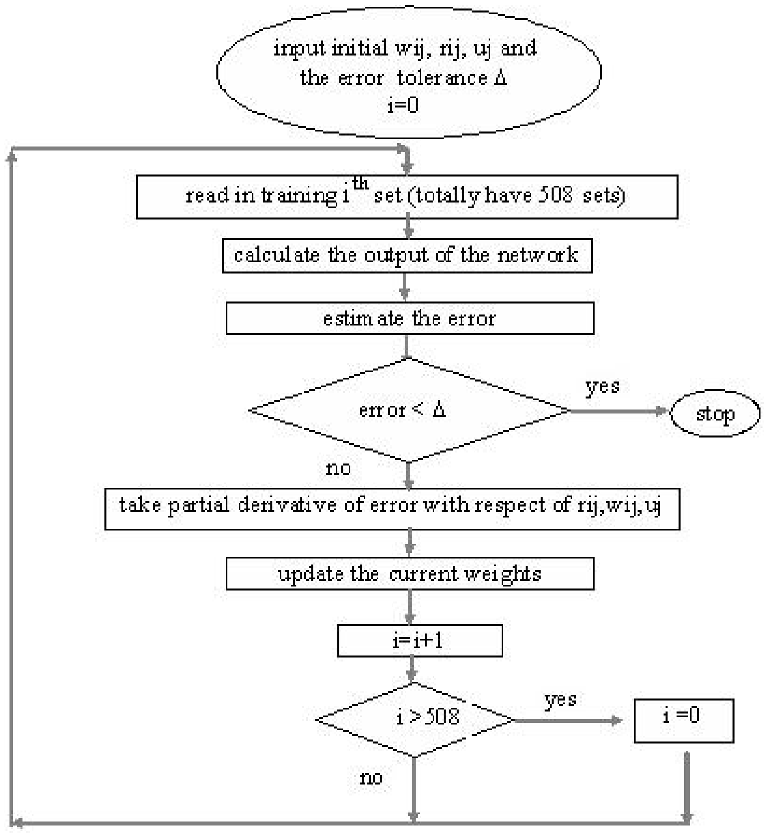


Figure 60: Flowchart of weights updating for rmNN calibration.

It is observed, from Figures 65 and 66, that approximately 70 fault files, which are files from file #73 to file #96, from file #318 to file #348 and from file #405 to file #419, are indicating combinations of fault 3 and 5.

The trained r, w and u parameters for fault 3 and 5 rmNNs are shown in Tables 30 and 31 respectively.

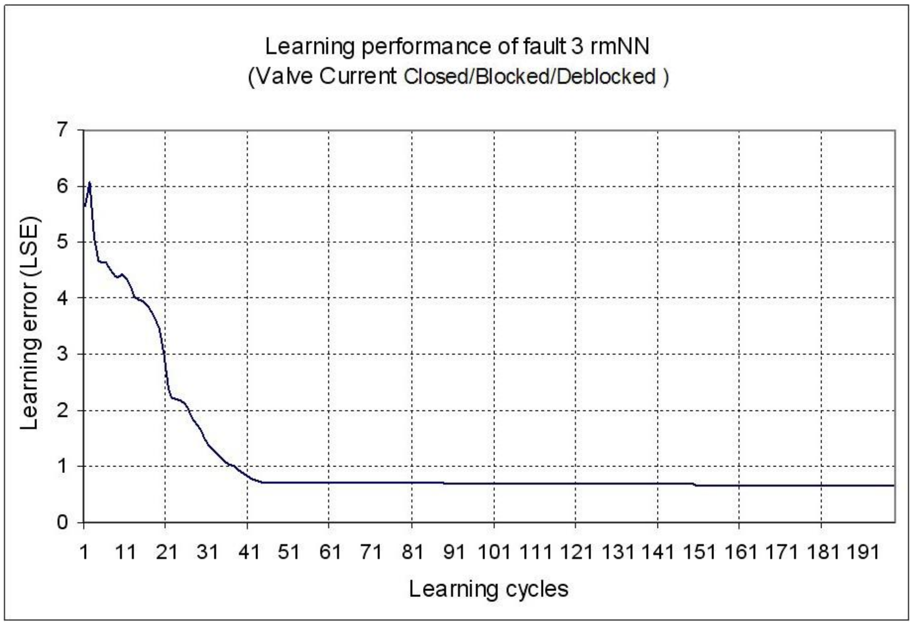


Figure 61: Learning performance of fault 3 rmNN.

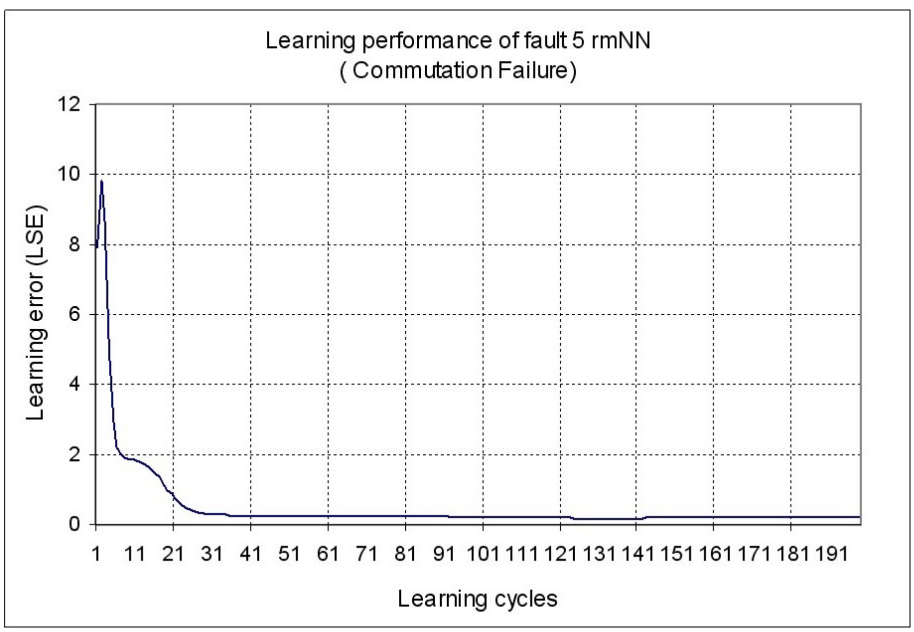


Figure 62: Learning performance of fault 5 rmNN.

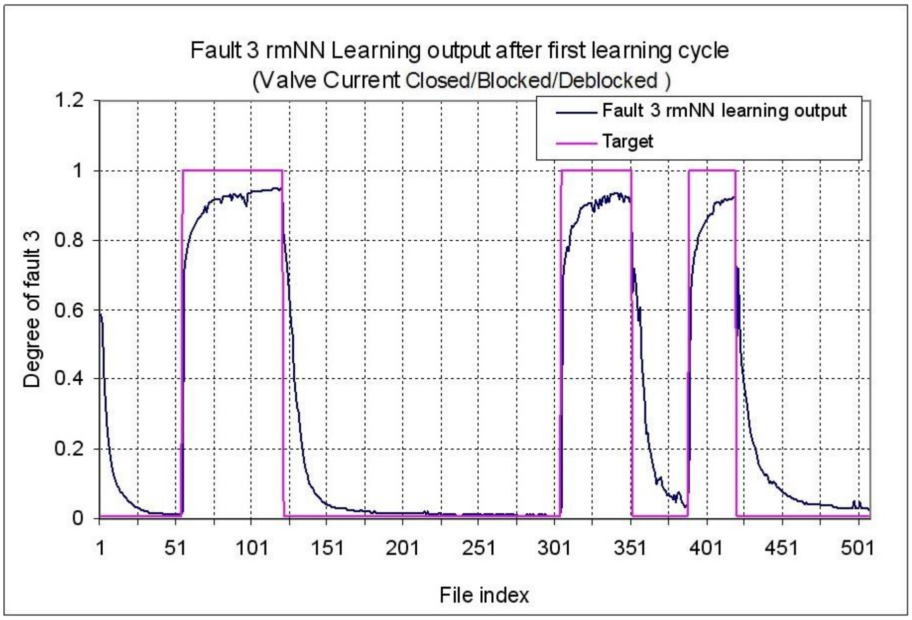


Figure 63: The output for fault 3 rmNN after one learning cycle.

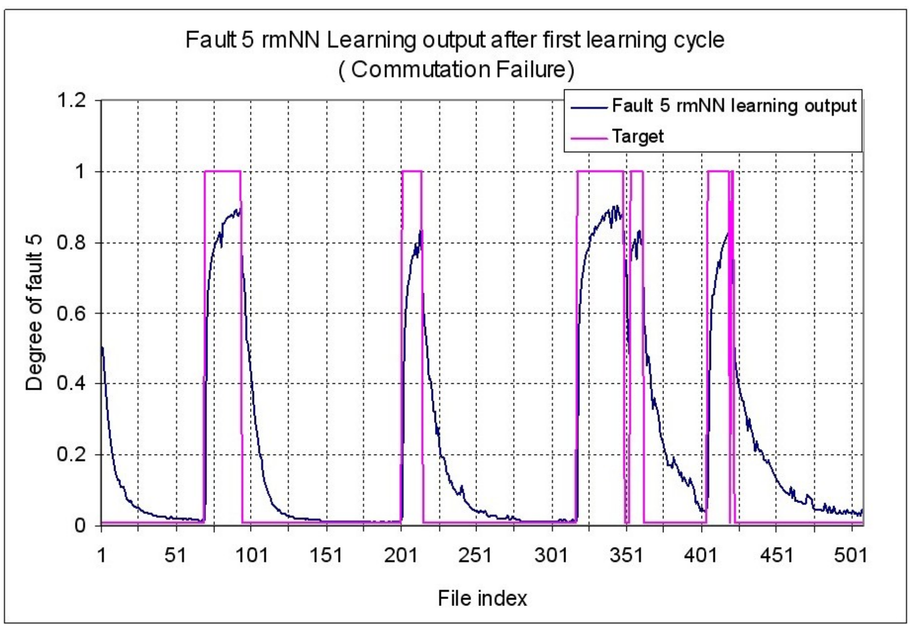


Figure 64: The output for fault 5 rmNN after one learning cycle.

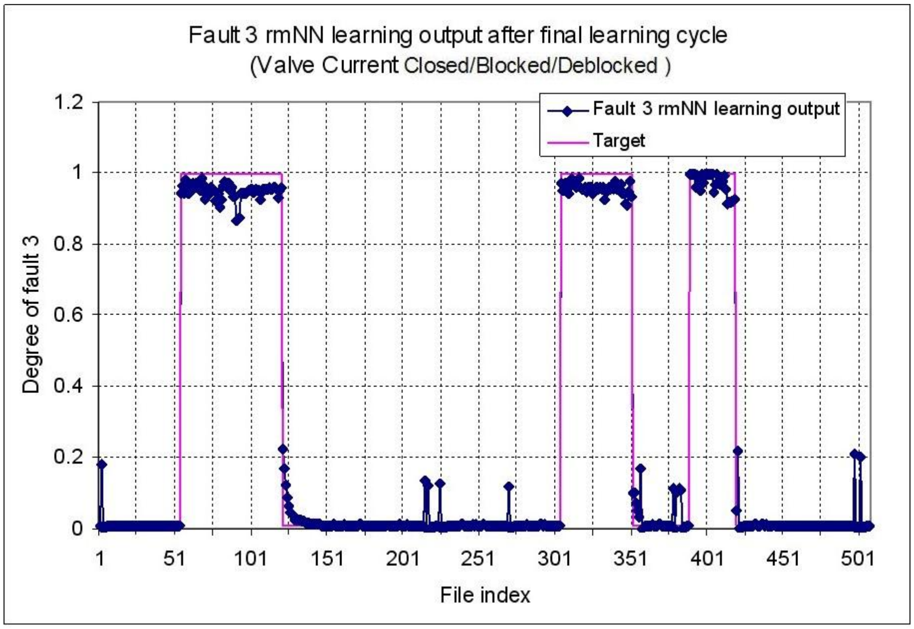


Figure 65: The output for fault 3 rmNN after 200 learning cycles.

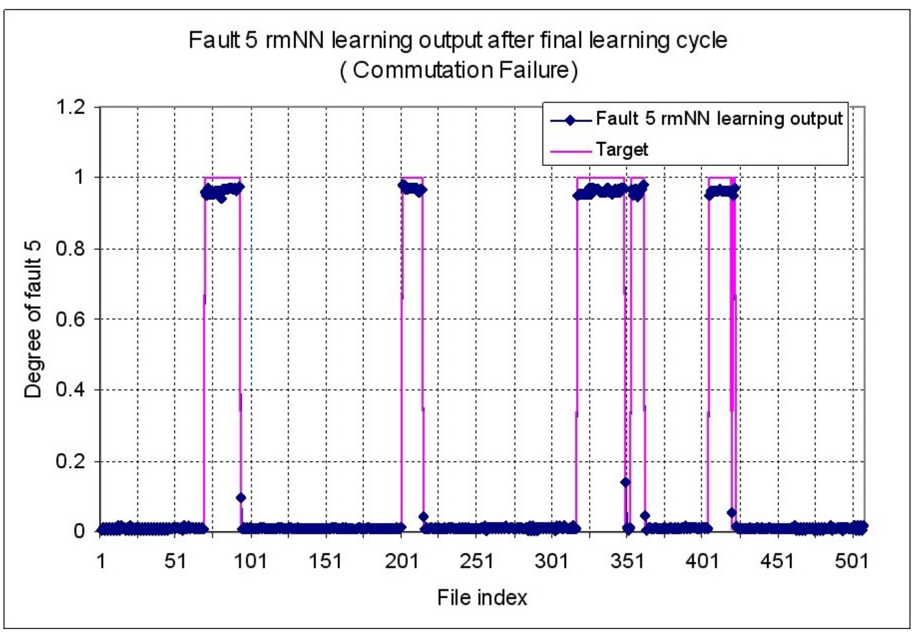


Figure 66: The output for fault 5 rmNN after 200 learning cycles.

Table 30: r, w and u parameters trained for fault 3 rmNN.

r										
0.000	0.060	0.974	0.330	0.120	0.058	0.190	0.999	0.039	0.072	0.120
0.037	0.003	0.255	0.995	0.068	0.040	0.190	0.370	0.043	0.130	0.119
0.198	0.140	0.996	0.517	0.120	0.040	0.190	0.390	0.040	0.130	0.003
0.198	0.140	0.996	0.512	0.120	0.041	0.190	0.390	0.050	0.130	0.047
0.185	0.170	0.827	0.462	0.250	0.317	0.113	0.399	0.340	0.133	0.399
0.048	0.342	0.455	0.708	0.053	0.320	0.169	0.519	0.327	0.044	0.367
0.050	0.133	0.348	0.406	0.214	0.040	0.194	0.996	0.108	0.153	0.120
0.050	0.009	0.253	0.995	0.120	0.040	0.190	0.370	0.040	0.105	0.044
0.363	0.273	0.667	0.607	0.361	0.081	0.292	0.539	0.345	0.150	0.120
0.049	0.137	0.255	0.995	0.120	0.040	0.190	0.370	0.040	0.067	0.030
0.413	0.383	0.361	0.543	0.284	0.146	0.262	0.557	0.289	0.266	0.058
w										
0.176	0.253	0.003	0.000	0.218	0.268	0.116	0.001	0.114	0.050	0.158
0.002	0.118	0.000	0.004	0.206	0.114	0.156	0.000	0.095	0.233	0.307
0.000	0.074	0.003	0.000	0.302	0.106	0.000	0.000	0.023	0.003	0.287
0.000	0.216	0.003	0.000	0.123	0.302	0.000	0.000	0.195	0.111	0.070
0.243	0.308	0.000	0.055	0.011	0.145	0.289	0.162	0.074	0.022	0.192
0.274	0.168	0.000	0.000	0.157	0.113	0.165	0.037	0.131	0.115	0.000
0.207	0.033	0.000	0.000	0.082	0.110	0.000	0.003	0.184	0.148	0.133
0.279	0.311	0.000	0.004	0.200	0.067	0.000	0.000	0.295	0.103	0.229
0.192	0.227	0.000	0.000	0.238	0.036	0.000	0.000	0.191	0.022	0.063
0.088	0.073	0.000	0.004	0.298	0.297	0.000	0.000	0.312	0.193	0.137
0.000	0.071	0.154	0.000	0.177	0.145	0.181	0.041	0.213	0.111	0.231
u										
1.000	1.000	0.990	0.990	0.000	0.000	1.000	1.000	0.000	1.000	0.000

Table 31: r, w and u parameters trained for fault 5 rmNN.

r										
0.000	0.060	0.122	0.100	0.998	0.110	0.120	0.050	0.040	0.103	0.100
0.037	0.003	0.512	0.397	0.476	0.238	0.144	0.342	0.293	0.368	0.384
0.050	0.080	0.187	0.100	0.998	0.110	0.120	0.050	0.040	0.100	0.003
0.048	0.080	0.123	0.100	0.998	0.110	0.120	0.050	0.040	0.101	0.047
0.050	0.080	0.126	0.100	0.998	0.110	0.120	0.044	0.040	0.103	0.101
0.048	0.329	0.419	0.420	0.421	0.309	0.172	0.286	0.324	0.071	0.357
0.209	0.191	0.457	0.353	0.493	0.332	0.212	0.389	0.305	0.087	0.325
0.039	0.009	0.123	0.100	0.998	0.110	0.120	0.034	0.040	0.101	0.044
0.050	0.080	0.217	0.100	0.998	0.110	0.120	0.036	0.040	0.100	0.093
0.050	0.079	0.170	0.100	0.998	0.110	0.120	0.050	0.040	0.101	0.030
0.376	0.358	0.306	0.334	0.482	0.206	0.259	0.335	0.270	0.247	0.058
w										
0.176	0.253	0.000	0.107	0.002	0.256	0.080	0.009	0.092	0.009	0.143
0.002	0.118	0.000	0.078	0.034	0.095	0.176	0.211	0.046	0.196	0.257
0.049	0.074	0.067	0.139	0.002	0.118	0.060	0.021	0.007	0.000	0.287
0.086	0.222	0.000	0.000	0.002	0.282	0.000	0.103	0.152	0.068	0.070
0.255	0.314	0.042	0.120	0.002	0.157	0.264	0.283	0.057	0.000	0.231
0.274	0.166	0.000	0.045	0.040	0.107	0.159	0.210	0.104	0.113	0.000
0.204	0.029	0.157	0.036	0.000	0.080	0.065	0.058	0.140	0.236	0.108
0.284	0.311	0.000	0.000	0.002	0.044	0.000	0.232	0.246	0.050	0.229
0.240	0.239	0.018	0.006	0.002	0.030	0.010	0.142	0.182	0.014	0.059
0.101	0.072	0.063	0.018	0.002	0.287	0.065	0.311	0.270	0.175	0.137
0.000	0.072	0.181	0.003	0.037	0.137	0.173	0.231	0.193	0.105	0.231
u										
1.000	0.050	1.000	1.000	1.000	0.010	0.030	1.000	1.000	1.000	0.000

7.4.5 Verification Results of Selected rmNNs for PSFC

In this section, 168 additional fault files have been used in a test set to verify the performance of the rmNN power fault classification system. The test results for fault 3 and 5 rmNNs are shown in Figures 67 and 68. With the fault detection threshold set to 80%, both rmNNs have 100% classification accuracy.

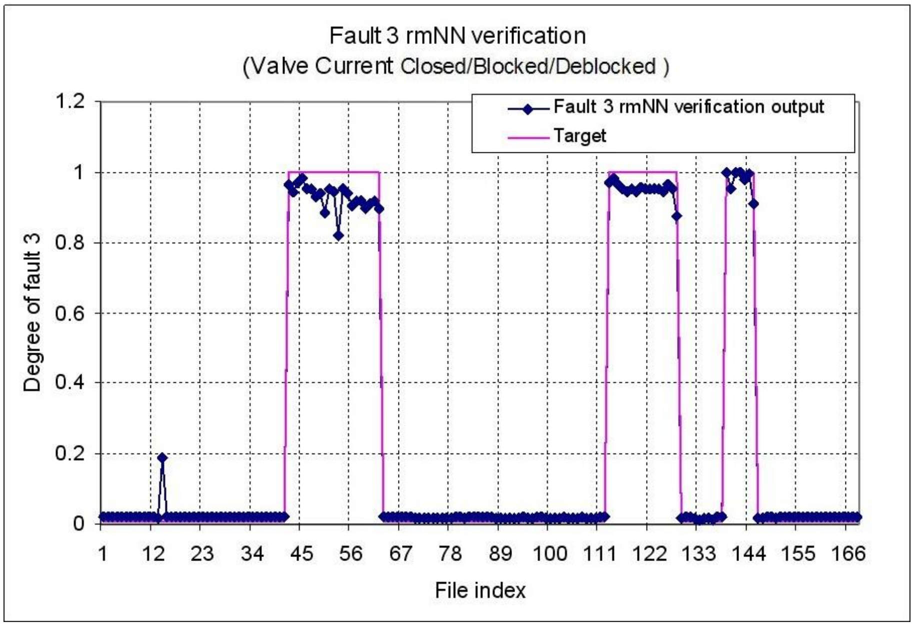


Figure 67: Testing results for fault 3 rmNN.

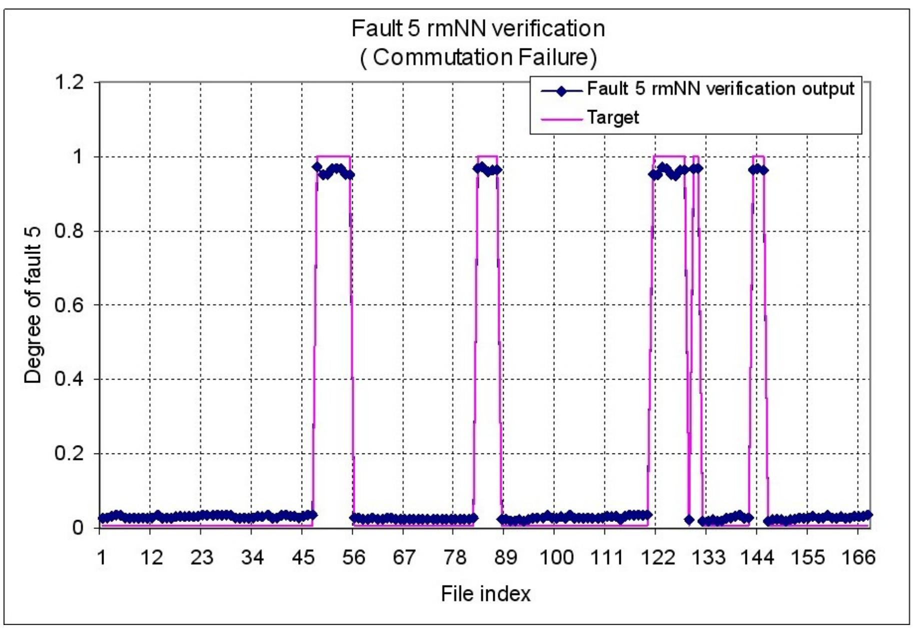


Figure 68: Testing results for fault 5 rmNN.

Table 32 summarizes the accuracy of the rmNN power fault classification system. It has been found that for each type of fault, the more fault files used in training, the more accurate the test results will be. For instance, a greater number of fault files were used to calibrate the rmNNs for faults “Minor AC disturbance”, “AC Disturbance”, “Valve Current Closed/Blocked/Deblocked”, “Line Fault”, and “Commutation Failure”. The test results for these types of faults showed 100% accuracy. By contrast, for faults “Current Arc Back” and “Normal affected by another pole”, 26 and 18 fault files are employed for calibration respectively; and the corresponding accuracy of the test results were comparatively low (Table 32).

Table 32: Accuracy of the rmNN power fault classification system.

Fault type	# of files for verification	# of files incorrect	Accuracy
Minor AC Disturbance	94	0	1.00
AC Disturbance	44	0	1.00
Valve Current Closed/Blocked/Deblocked	44	0	1.00
Line Fault	22	0	1.00
Commutation Failure	25	0	1.00
Pole Voltages/Currents Closed/Blocked/Deblocked	15	0	1.00
Current Arc Back	6	1	0.83
Parallel Operation	9	0	1.00
Pole Current Oscillation	7	0	1.00
Normal Affected by Another Pole	4	1	0.75
Asymmetric Protection	6	0	1.00
Disturbance on DC Voltage	7	0	1.00

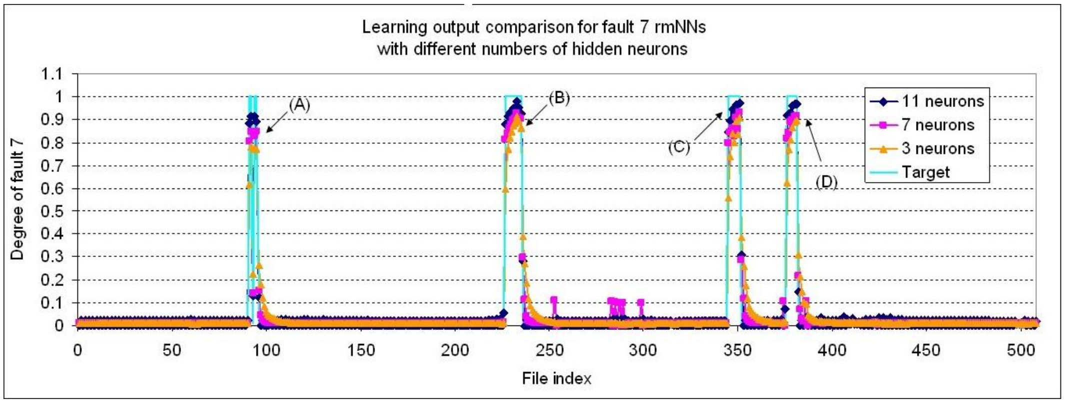
7.5 Effects of the Number of Neurons in the Hidden Layer

For verifying the rmNN performance, the rmNN sensitivity with respect to different numbers of neurons in the hidden layer was analyzed. The rmNNs with 11, 9, 7 or 3 hidden neurons are tested for the learning and verification set. The numerical results provide a very good performance index.

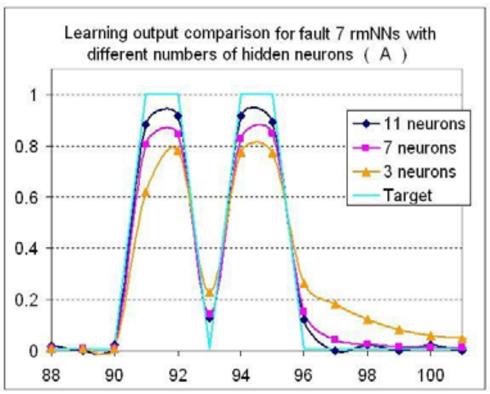
To decide on the number of hidden neurons, the square root of the product of the input and output is a good number with which to begin, in this case, $\sqrt{11 \times 1}$ is 3.3. The test results of the learning and verification performance with 3, 7, 9 and 11 neurons for 12 rmNNs are shown in Appendix E. The test results indicate that the performance with 3 hidden neurons is always the worst case and unacceptable. The performance with 7 and 9 hidden neurons is very close. The case with 11 neurons has the best performance for fault 7 and fault 8 rmNNs.

The learning output comparison for fault 7 rmNN with 3, 7 and 11 hidden neurons is shown in Figure 69(a). The learning output for fault 7 rmNN with 9 hidden neurons is omitted since its performance is close to the rmNN with 7 hidden neurons (Appendix E). Figures 69(b), 69(c), 69(d) and 69(e) show the details of the A, B, C and D parts in Figure 69(a) respectively. They clearly indicate that, for all the true cases, the rmNN with 11 hidden neurons present the highest output, and the rmNN with 3 hidden neurons give the lowest output. The verification output comparison for fault 7 rmNN with 3, 7 and 11 hidden neurons is shown in Figure 70(a). The verification output for fault 7 rmNN with 9 hidden neurons is omitted for the same reason. Figures 70(b), 70(c), 70(d) and 70(e) show the details of the A, B, C and D parts of Figure 70(a) and clearly confirm the results from learning for all the true cases. The rmNN with 11 hidden neurons gives the highest verification score, and the verification output from the one with 3 hidden neurons yields the lowest.

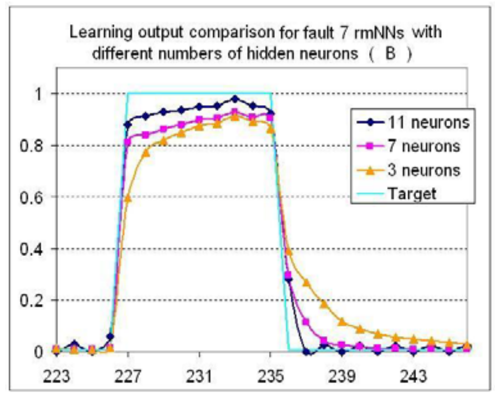
As another example, the learning and verification output comparison for fault 8 rmNN with 3, 7 and 11 hidden neurons is shown in Figures 71 and 72 respectively. It agrees with the learning and verification results for fault 7 rmNN, *i.e.*, different numbers of neurons in the hidden layer considerably affect the performance of the rmNN. The goal is to have not too many but enough hidden neurons to be able to learn correctly. There are no analytically shown facts about the necessary number of hidden neurons, instead more tests are required to find an appropriate number. In addition, some research shows that the redundancy on hidden-layer neurons is useful in the fault tolerance of neural networks, especially for the feedforward networks.



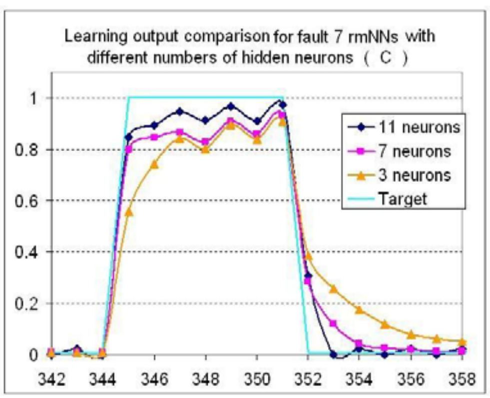
(a) A broad view of the performance comparison for 508 rmNN training files.



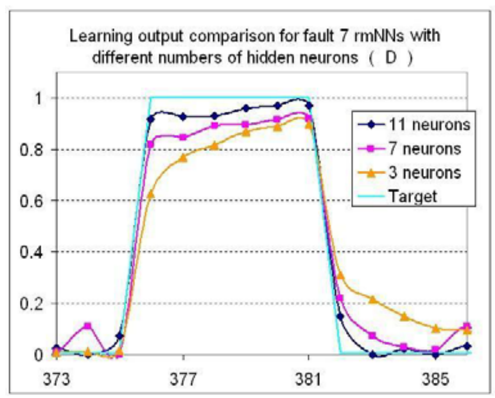
(b) A zoom-in detail of part A in (a).



(c) A zoom-in detail of part B in (a).

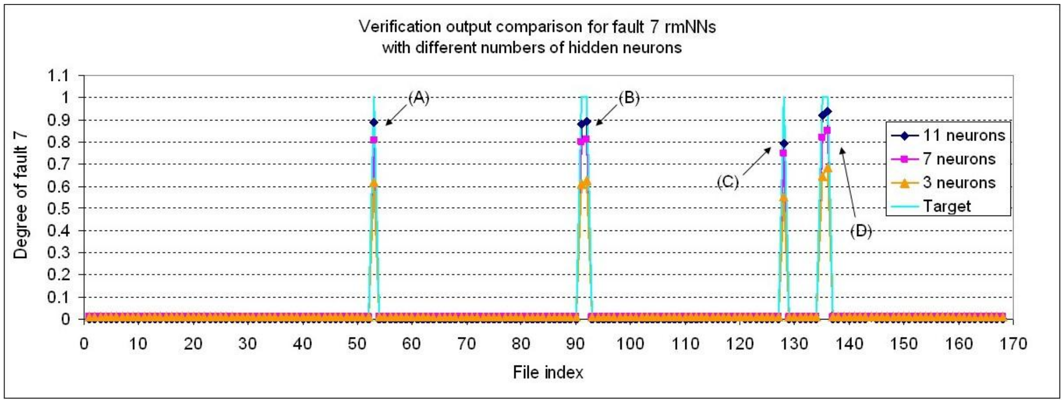


(d) A zoom-in detail of part C in (a).

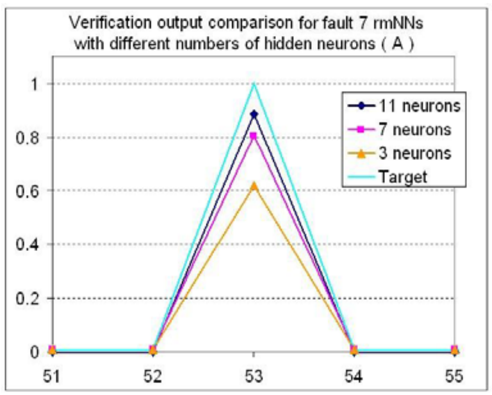


(e) A zoom-in detail of part D in (a).

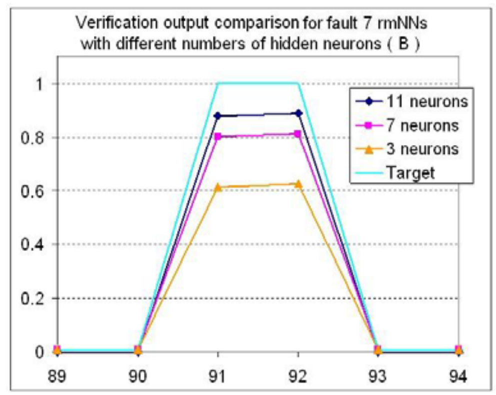
Figure 69: The learning output comparison for fault 7 rmNNs with 3, 7 and 11 hidden neurons.



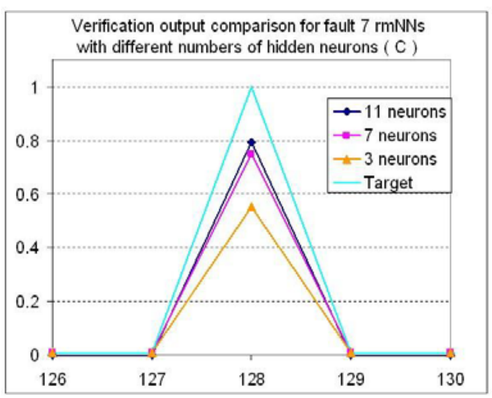
(a) A broad view of the performance comparison for 168 rmNN testing files.



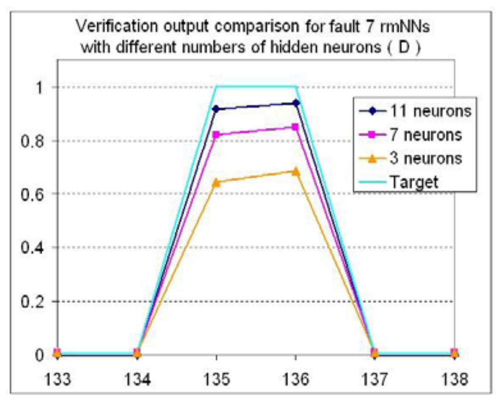
(b) A zoom-in detail of part A in (a).



(c) A zoom-in detail of part B in (a).

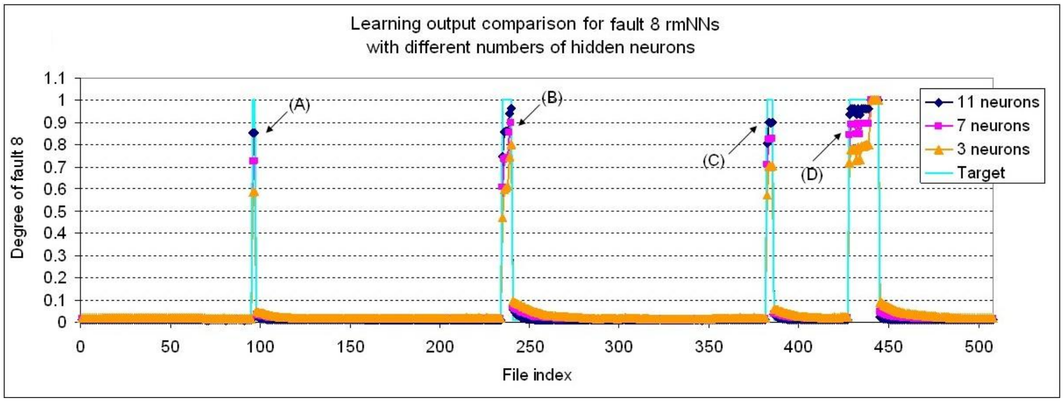


(d) A zoom-in detail of part C in (a).

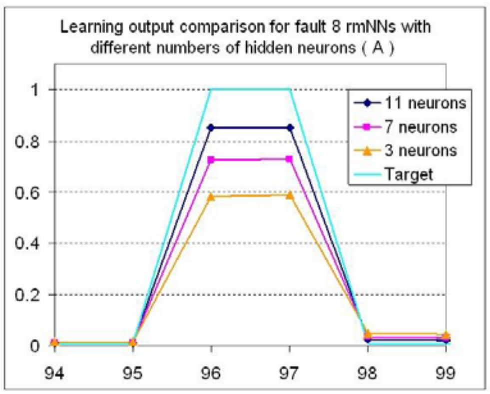


(e) A zoom-in detail of part D in (a).

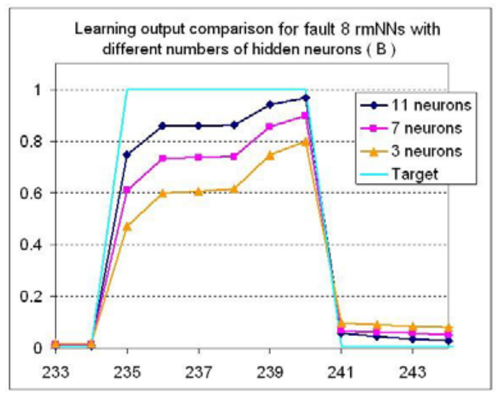
Figure 70: The verification output comparison for fault 7 rmNNs with 3, 7 and 11 hidden neurons.



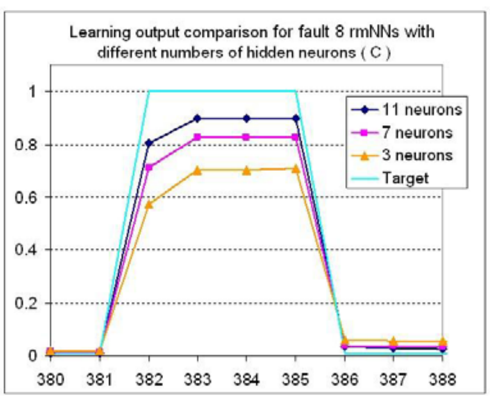
(a) A broad view of the performance comparison for 508 rmNN training files.



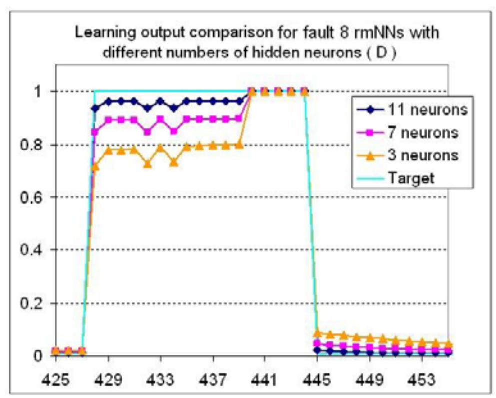
(b) A zoom-in detail of part A in (a).



(c) A zoom-in detail of part B in (a).

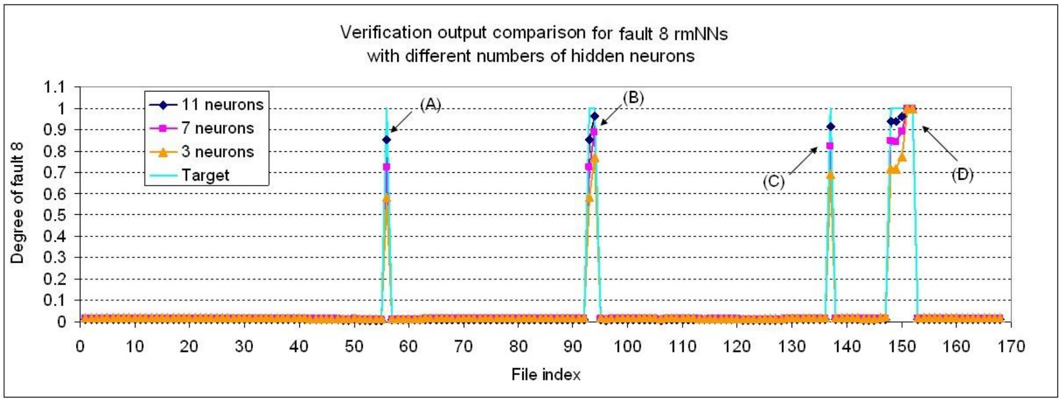


(d) A zoom-in detail of part C in (a).

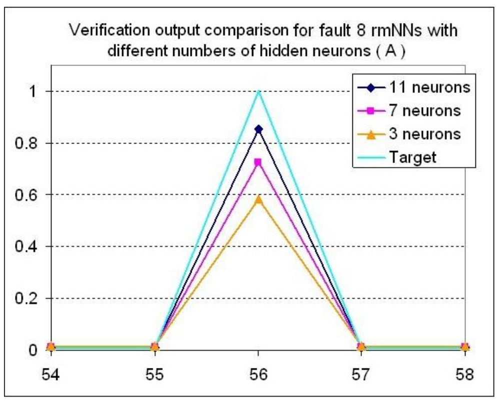


(e) A zoom-in detail of part D in (a).

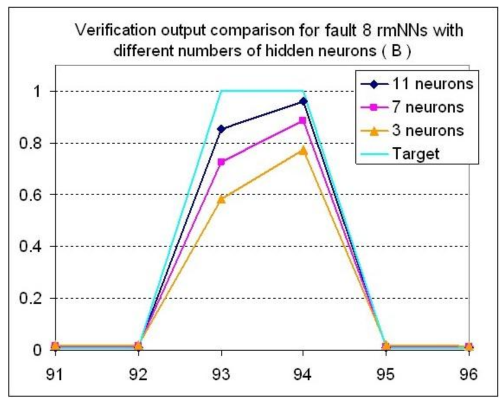
Figure 71: The learning output comparison for fault 8 rmNNs with 3, 7 and 11 hidden neurons.



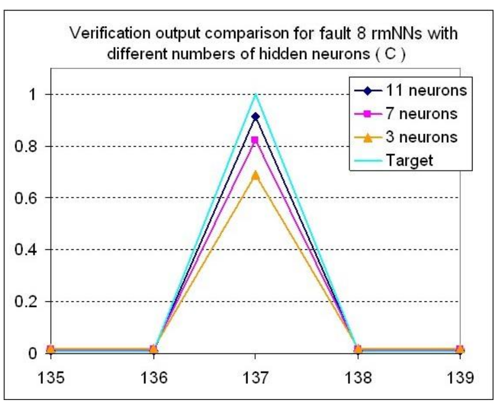
(a) A broad view of the performance comparison for 168 rmNN testing files.



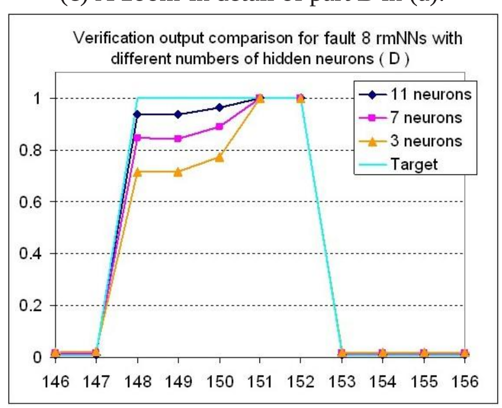
(b) A zoom-in detail of part A in (a).



(c) A zoom-in detail of part B in (a).



(d) A zoom-in detail of part C in (a).



(e) A zoom-in detail of part D in (a).

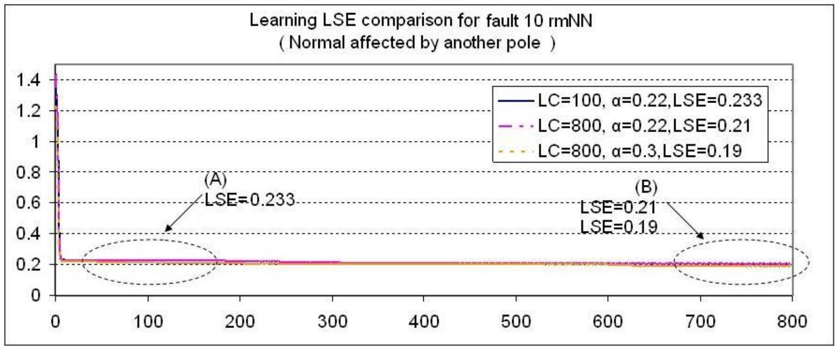
Figure 72: The verification output comparison for fault 8 rmNNs with 3, 7 and 11 hidden neurons.

7.6 Effects of Learning Cycles, Learning Rate and Least Square Error

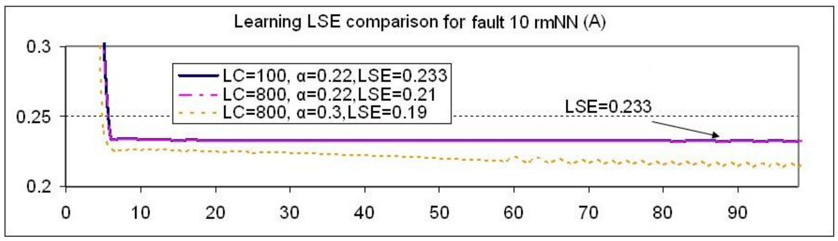
To obtain satisfactory performance for an rmNN, the selection of appropriate learning rates (α) for the rmNN is critical and challenging. Applying an improper learning rate to the rmNN may cause the learning curve of the rmNN to oscillate. For example, suppose a network produces an error of -0.5 and the error was adjusted at an improper learning rate by the network. The new error is +0.5, and the next error is -0.5 again..., so on and so forth. Apparently the learning period this system takes will be endless. On the other hand, if the learning rate is too small, the network parameters will improve toward the best solution, but at a very low speed. It might take hours, even days, to optimize such a network. To gain a good learning rate requires interactive processing to achieve an acceptable overall direction for the search.

It is sometimes seen that the learning error decreases for the learning set of data with more and more learning cycles (LCs), but still does not lead to better classification performance. This suggests that the network is "overfitting" due to some local minimum.

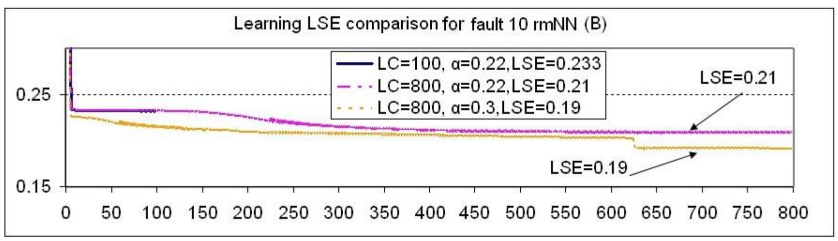
An example of an "overfitting" rmNN is observed when the rmNN is trained for classifying fault 10. Figure 73(a) shows the learning least square error (LSE) comparison for fault 10 rmNN with different LCs and α. The details of A and B parts in Figure 73(a) are illustrated in Figure 73(b) and Figure 73(c), respectively. Learning case 1 has LCs = 100, α = 0.22 and LSE = 0.233; learning case 2 has LCs = 800, α = 0.22 and LSE = 0.21; and learning case 3 has LCs = 800, α = 0.3 and LSE = 0.19. The learning cycles are increasing and the LSEs are decreasing.



(a)



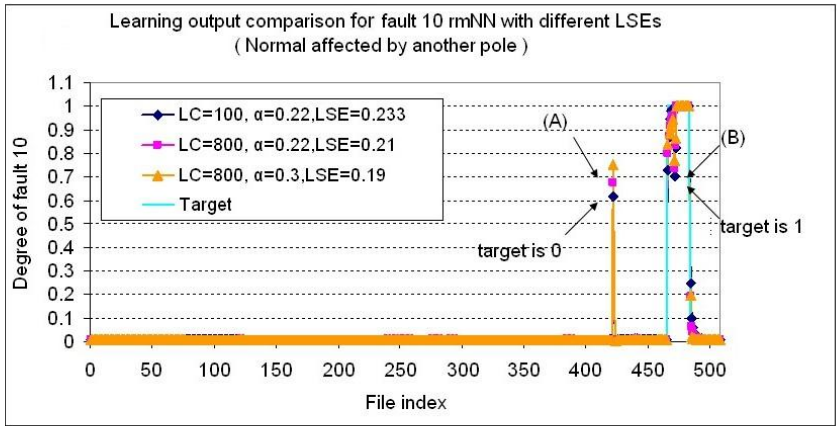
(b)



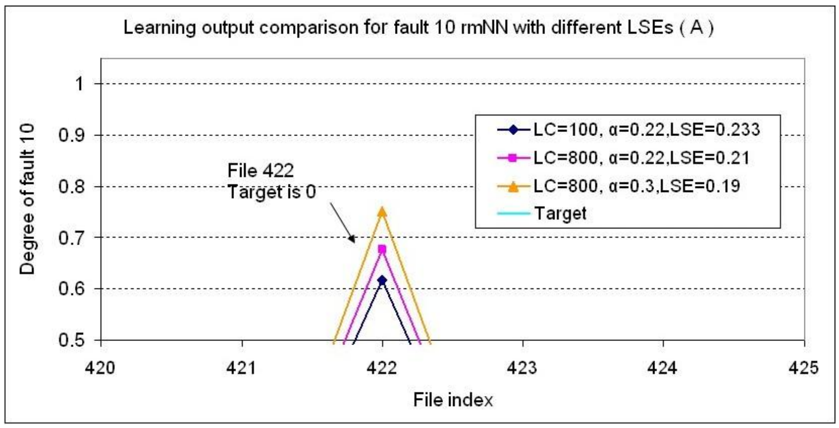
(c)

Figure 73: Learning LSE comparison for fault 10 rmNN with different LCs and α.

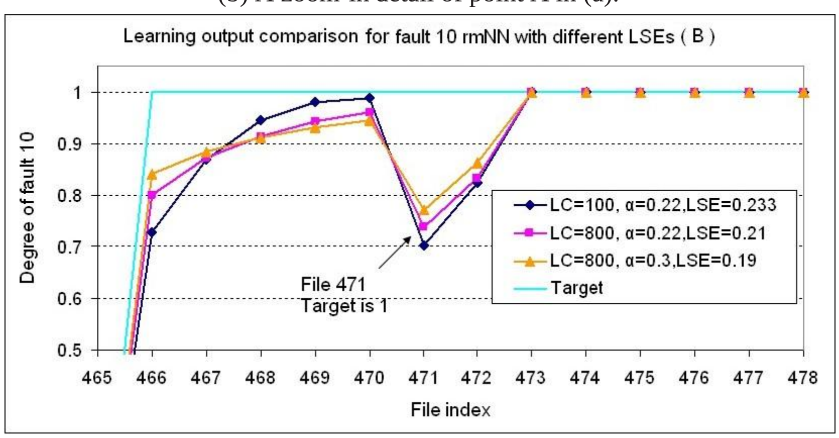
Figure 74(a) illustrates the learning output comparison for fault 10 rmNN in the three cases shown in Figure 73. Figures 74(b) and 74(c) show the learning output at points A and B. Points A and B represent file 422 and file 471, which belong to false case and true case respectively. From Figures 74(b) and 74(c), it is found that for case 1, points A and B are 0.09 apart from each other; for case 2, A and B are closer with a distance of 0.07; and for case 3, A and B locate almost at the same line with a distance of 0.01, and could barely be distinguished. It has been found that the smaller LSE does not lead to better classification performance. The conjecture is that the rmNN is "overfitting", which causes the vagueness (slight difference) between the true and false cases.



(a) A broad view.



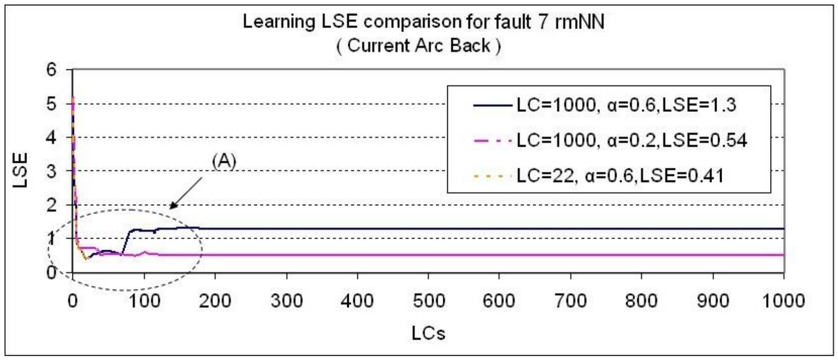
(b) A zoom-in detail of point A in (a).



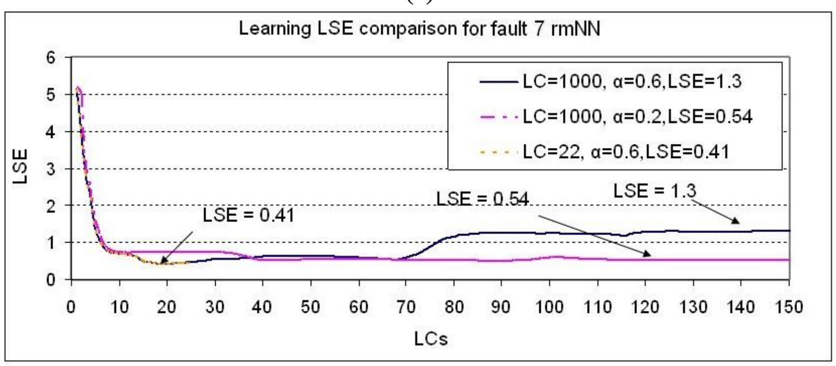
(c) A zoom-in detail of point B in (a).

Figure 74: The learning output comparison for fault 10 rmNN with different LCs and α.

Another example shows the lower output for both learning and verification output for true case files with a smaller LSE. Figure 75 shows the learning least square error (LSE) comparison for fault 7 rmNN with different LCs and α. In learning case 1, LSE = 1.3; learning case 2, LSE = 0.54; and learning case 3, LSE = 0.41. From case 1 to case 3, the LSE decreases.



(a)



(b)

Figure 75: Learning LSE comparison for fault 7 rmNN with different LCs and α.

Figure 76 shows the learning output for the 3 cases. Figures 76(b), 76(c), 76(d) and 76(e) show the details of the A, B, C and D parts in Figure 76(a).

Figure 77 shows the verification output for the 3 cases. Figures 77(b), 77(c), 77(d) and 77(e) show the details of the A, B, C and D parts in Figure 77(a).

It has been shown that, for all the true case files with a target value of 1, case 1 with the higher LSE has the highest output and case 3 with the lowest LSE has the lowest output.

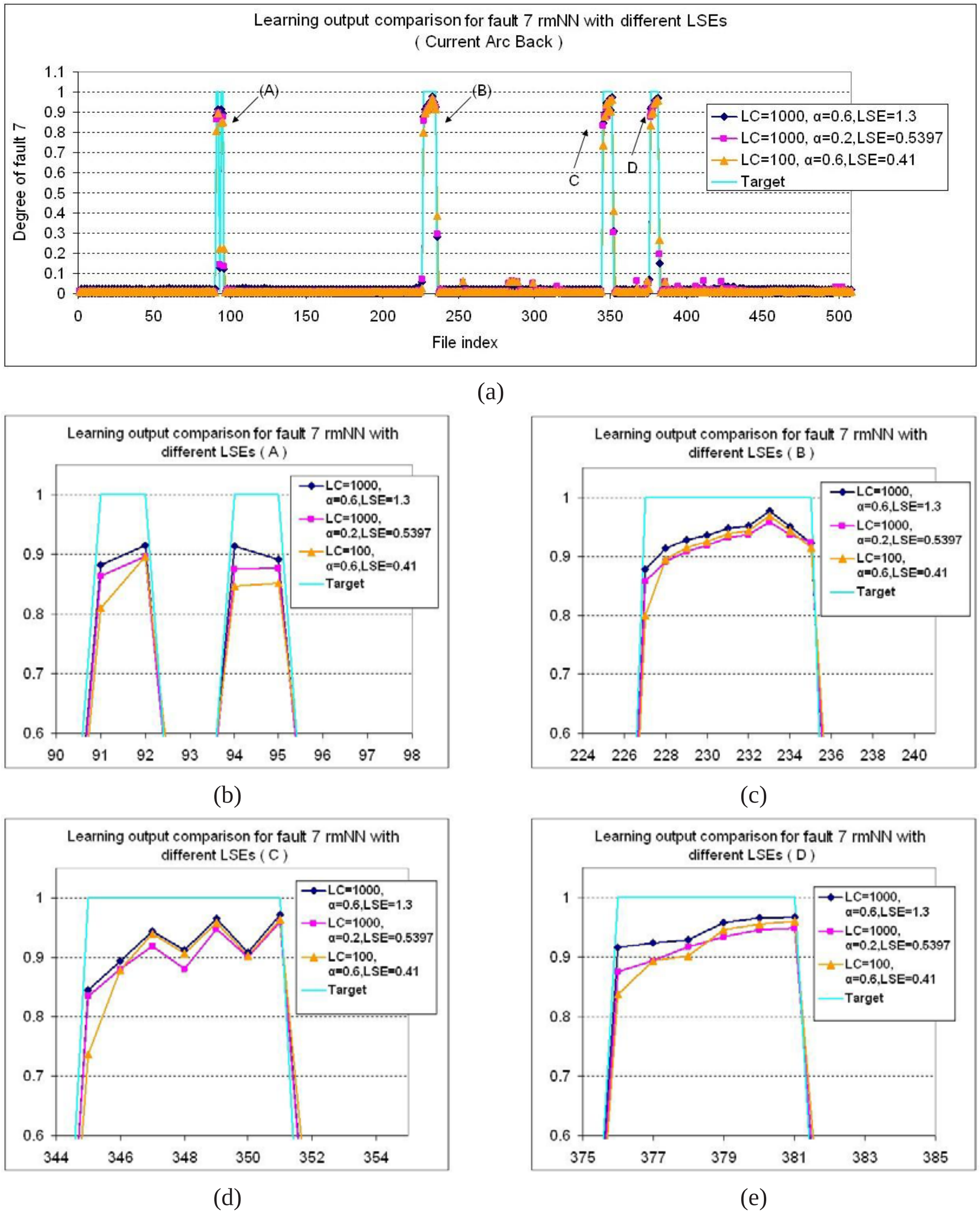


Figure 76: The learning output comparison for fault 7 rmNN with different LCs and α.

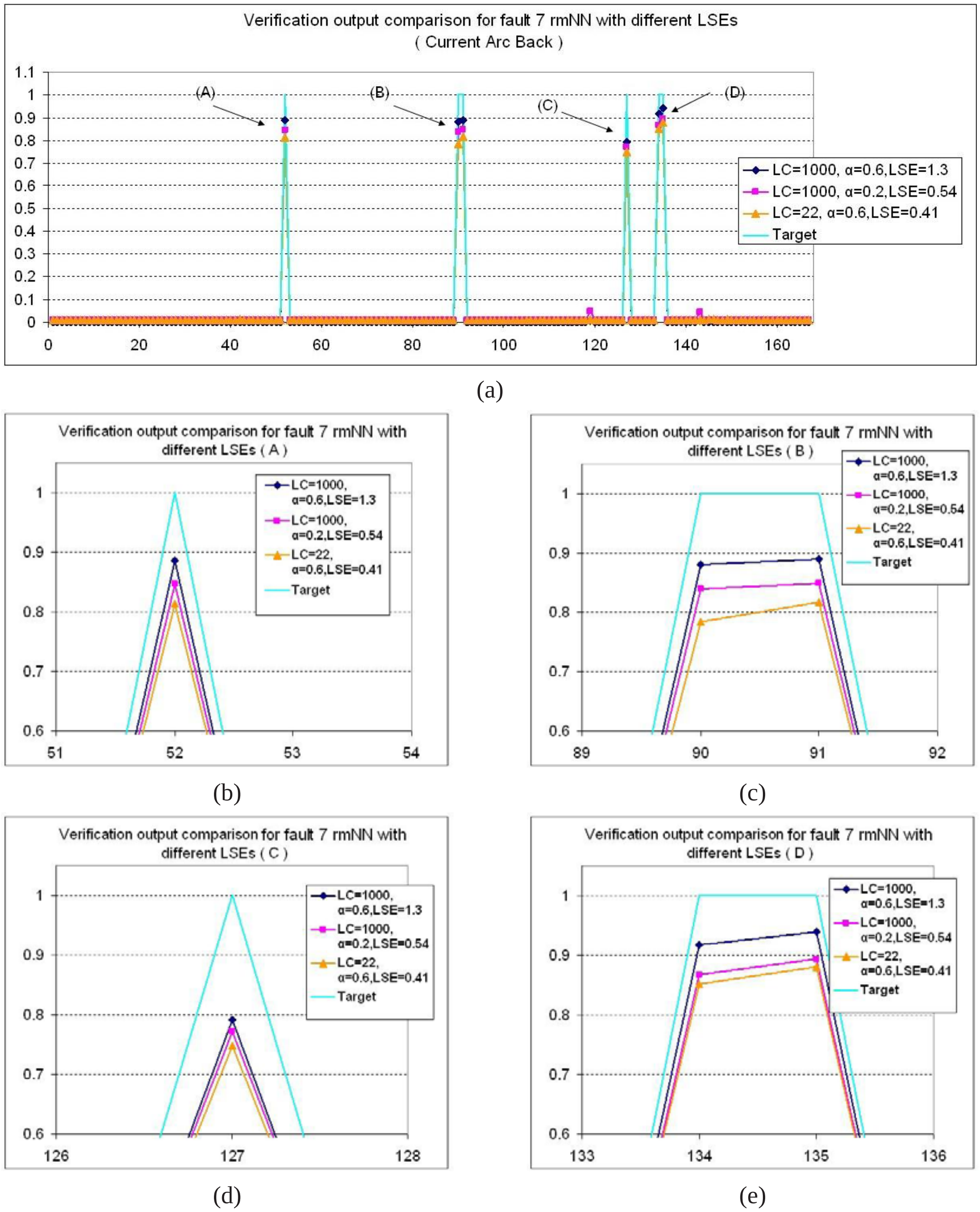


Figure 77: The verification output comparison for fault 7 rmNN with different LCs and α.

The learning cycles, learning rates applied and the learning errors received for 12 rmNNs are listed in Table 33.

Table 33: The learning cycles, learning rate and learning error for 12 rmNNs.

	Training parameters		
	Learning cycles (LC)	Learning rate (α)	Least square error (LSE)
Fault 1 rmNN	200	0.01	0.2658
Fault 2 rmNN	200	0.01	0.8579
Fault 3 rmNN	200	0.1	0.6608
Fault 4 rmNN	800	0.1	0.1653
Fault 5 rmNN	200	0.1	0.2295
Fault 6 rmNN	1000	0.2	0.4061
Fault 7 rmNN	1000	0.6	1.3
Fault 8 rmNN	800	0.02	0.1439
Fault 9 rmNN	800	0.1	0.0892
Fault 10 rmNN	100	0.22	0.233
Fault 11 rmNN	800	0.1	0.86
Fault 12 rmNN	100	0.01	0.0634

7.7 Implementation of rmNN for PSFC

After theoretical development and computer simulation, the next sought-after step is to build the software package for the implementation of rmNN power system fault classification with a user friendly interface. The software package provides the following functions:

- Feature extraction
- Rough set construction
- Rough membership computation

- Rough membership neural network calibration and verification

- Power system fault type detection

C++ programs as well as the executable codes have been developed for each function. The executables are called and embedded in a LabVIEW program, which creates a flexible and scalable user interface. With LabVIEW, users can interface with real-world signals, analyze data for meaningful information, and share results through intuitive displays and reports. The screen snapshot of the user interface for rmNN power system fault classification is shown in Figure 78.

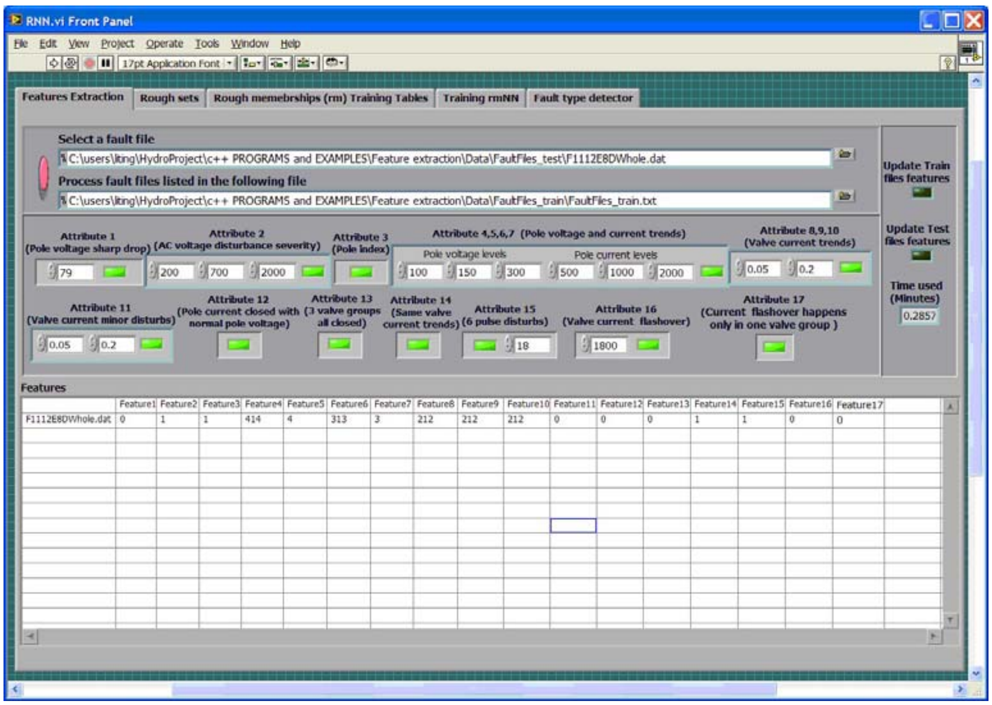


Figure 78: The user interface for rmNN power system fault classification.

The main GUI (graphic user interface) window for the rmNN PSFC contains five tabs that are created for the five functions. The first tab is designed for the features extraction function seen in Figure 78. Users are allowed to select either a fault file list or a specific fault file to be processed. An example of a fault file list is FaultFiles_train.txt, which contains the file names of all fault files for training. The 23 signals are analyzed and 17 features are derived for each fault file. The thresholds applied in the features extraction can be adjusted to achieve optimized feature values for the best fault classification performance. When partial features need to be updated, the switch allocated for each feature can be individually turned off to disable the feature extraction operation.

The second tab allows accessing the rough set construction function shown in Figure 79. The equivalence classes, the B-low approximations (Yes Sets) and approximation boundary sets (YesOrNo Sets) are created based on features values of all training fault files. The rough sets need to be restructured whenever the features values are updated.

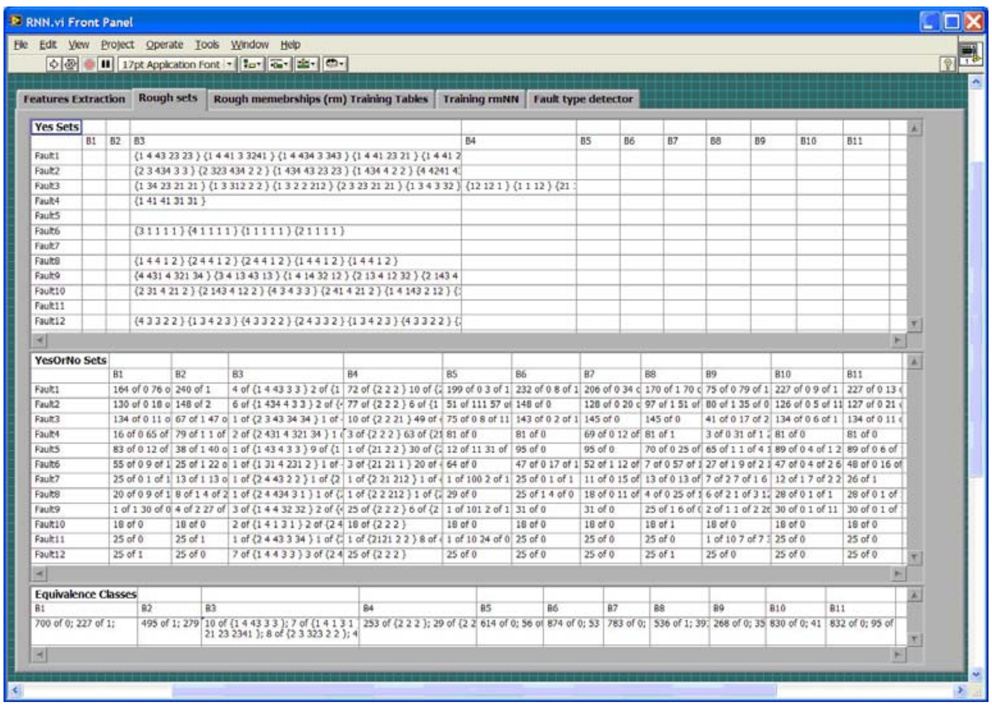
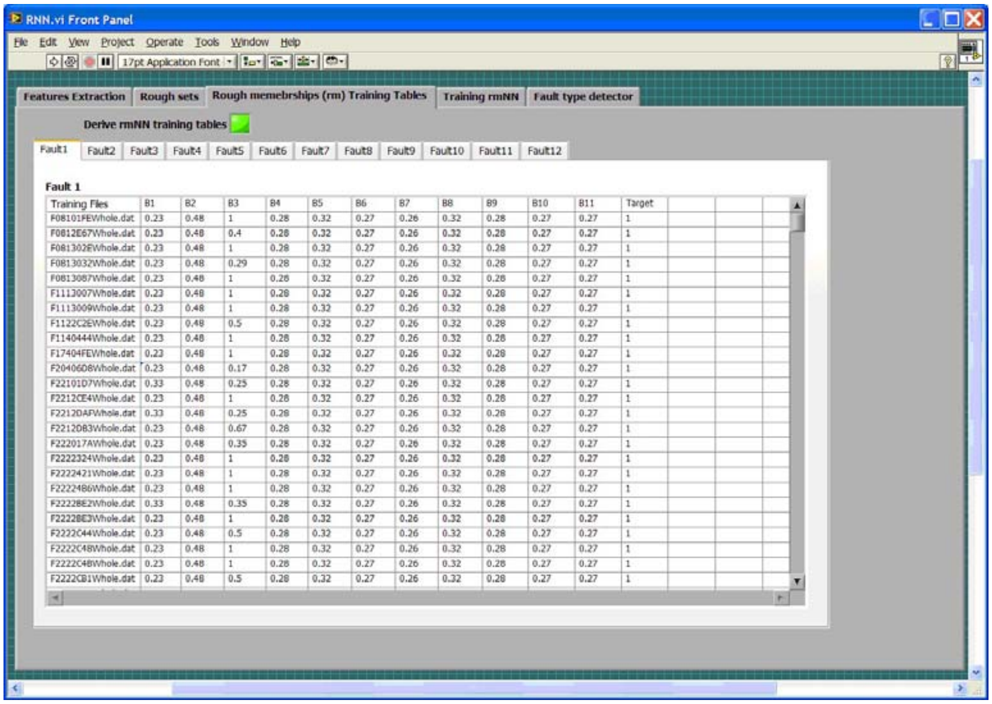


Figure 79: Rough set construction for rmNN power system fault classification.

The third tab is for the rough membership computation function illustrated in Figure 80. Twelve rough membership training tables, which contain the training sets for the calibration of twelve rmNNs, have been derived. These training tables need to be re-generated as well whenever the features values are adjusted.



Training Files	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	Target
F08101FEWhole.dat	0.23	0.48	1	0.28	0.32	0.27	0.26	0.32	0.28	0.27	0.27	1
F0812E67Whole.dat	0.23	0.48	0.4	0.28	0.32	0.27	0.26	0.32	0.28	0.27	0.27	1
F081302EWhole.dat	0.23	0.48	1	0.28	0.32	0.27	0.26	0.32	0.28	0.27	0.27	1
F0813032Whole.dat	0.23	0.48	0.29	0.28	0.32	0.27	0.26	0.32	0.28	0.27	0.27	1
F0813087Whole.dat	0.23	0.48	1	0.28	0.32	0.27	0.26	0.32	0.28	0.27	0.27	1
F1113007Whole.dat	0.23	0.48	1	0.28	0.32	0.27	0.26	0.32	0.28	0.27	0.27	1
F1113009Whole.dat	0.23	0.48	1	0.28	0.32	0.27	0.26	0.32	0.28	0.27	0.27	1
F1122C2EWhole.dat	0.23	0.48	0.5	0.28	0.32	0.27	0.26	0.32	0.28	0.27	0.27	1
F1140444Whole.dat	0.23	0.48	1	0.28	0.32	0.27	0.26	0.32	0.28	0.27	0.27	1
F17404FEWhole.dat	0.23	0.48	1	0.28	0.32	0.27	0.26	0.32	0.28	0.27	0.27	1
F20406D8Whole.dat	0.23	0.48	0.17	0.28	0.32	0.27	0.26	0.32	0.28	0.27	0.27	1
F22101D7Whole.dat	0.33	0.48	0.25	0.28	0.32	0.27	0.26	0.32	0.28	0.27	0.27	1
F2212CE4Whole.dat	0.23	0.48	1	0.28	0.32	0.27	0.26	0.32	0.28	0.27	0.27	1
F2212DAFWhole.dat	0.33	0.48	0.25	0.28	0.32	0.27	0.26	0.32	0.28	0.27	0.27	1
F2212DB3Whole.dat	0.23	0.48	0.67	0.28	0.32	0.27	0.26	0.32	0.28	0.27	0.27	1
F222017AWhole.dat	0.23	0.48	0.35	0.28	0.32	0.27	0.26	0.32	0.28	0.27	0.27	1
F2222324Whole.dat	0.23	0.48	1	0.28	0.32	0.27	0.26	0.32	0.28	0.27	0.27	1
F2222421Whole.dat	0.23	0.48	1	0.28	0.32	0.27	0.26	0.32	0.28	0.27	0.27	1
F22224B6Whole.dat	0.23	0.48	1	0.28	0.32	0.27	0.26	0.32	0.28	0.27	0.27	1
F2222BE2Whole.dat	0.33	0.48	0.35	0.28	0.32	0.27	0.26	0.32	0.28	0.27	0.27	1
F2222BE3Whole.dat	0.23	0.48	1	0.28	0.32	0.27	0.26	0.32	0.28	0.27	0.27	1
F2222C44Whole.dat	0.23	0.48	0.5	0.28	0.32	0.27	0.26	0.32	0.28	0.27	0.27	1
F2222C48Whole.dat	0.23	0.48	1	0.28	0.32	0.27	0.26	0.32	0.28	0.27	0.27	1
F2222C4BWhole.dat	0.23	0.48	1	0.28	0.32	0.27	0.26	0.32	0.28	0.27	0.27	1
F2222CB1Whole.dat	0.23	0.48	0.5	0.28	0.32	0.27	0.26	0.32	0.28	0.27	0.27	1

Figure 80: Rough membership computation for rmNN power system fault classification.

The forth tab allows accessing the rough membership neural network calibration and verification function shown in Figure 81. Twelve pages have been developed inside this function for calibration of twelve rmNNs respectively. The learning rate and cycles are the parameters to be adjusted until the best classification performance is achieved. The learning error, learning output and test result for each rmNN are displayed in three graphs respectively and the final calibrated weights are reported in the table at the bottom-right quarter of the tab window.

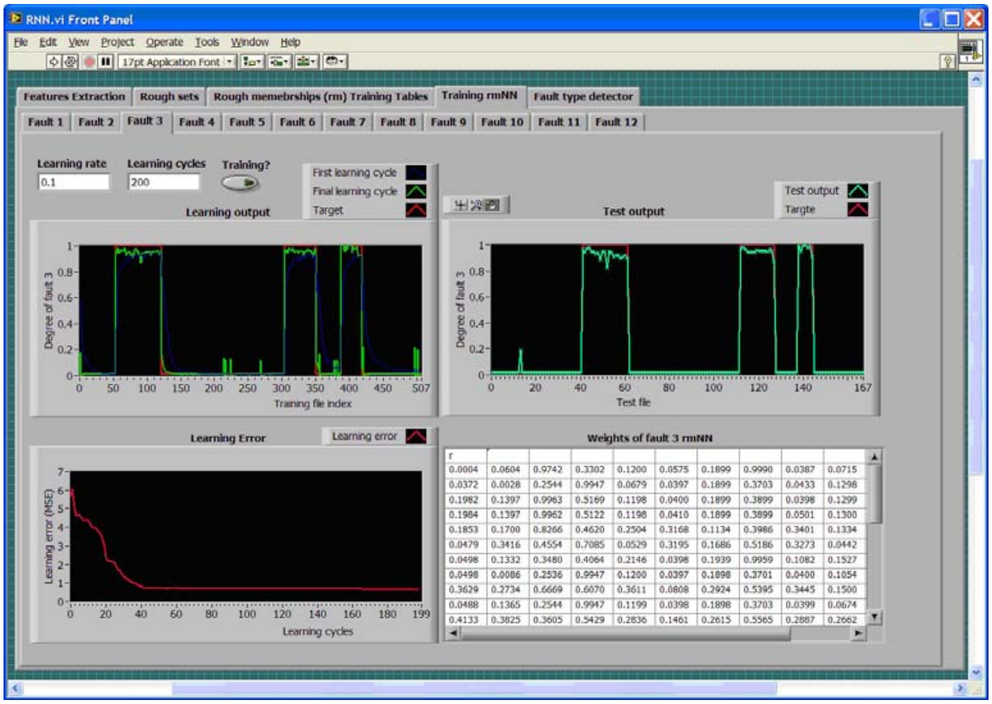


Figure 81: rmNN calibration and verification function for rmNN power system fault classification.

The final function is for the power system fault type detection, which is illustrated in Figure 82. For a fault file to be classified, 17 features are derived by function 1 and loaded by this fault type detector. The 11 rough memberships, a rough membership set, based on 11 feature sets associated with a type of fault are estimated. Consequently a total of 12 rough membership sets are derived and input to 12 trained rmNNs respectively. Twelve degrees, one for each fault type, are estimated by the 12 rmNNs and forwarded to a decider neuron, which picks up the faults with degree output greater than the preset threshold. For example, in Figure 82, the threshold is set as 0.8; the faults detected for fault file F1112E80Whole.dat are a combination of fault 1 and fault 4 with a degree of 0.911 and 0.985 respectively. The fault types are indicated in a text box. The threshold can be adjusted lower to allow more fault types to be considered.

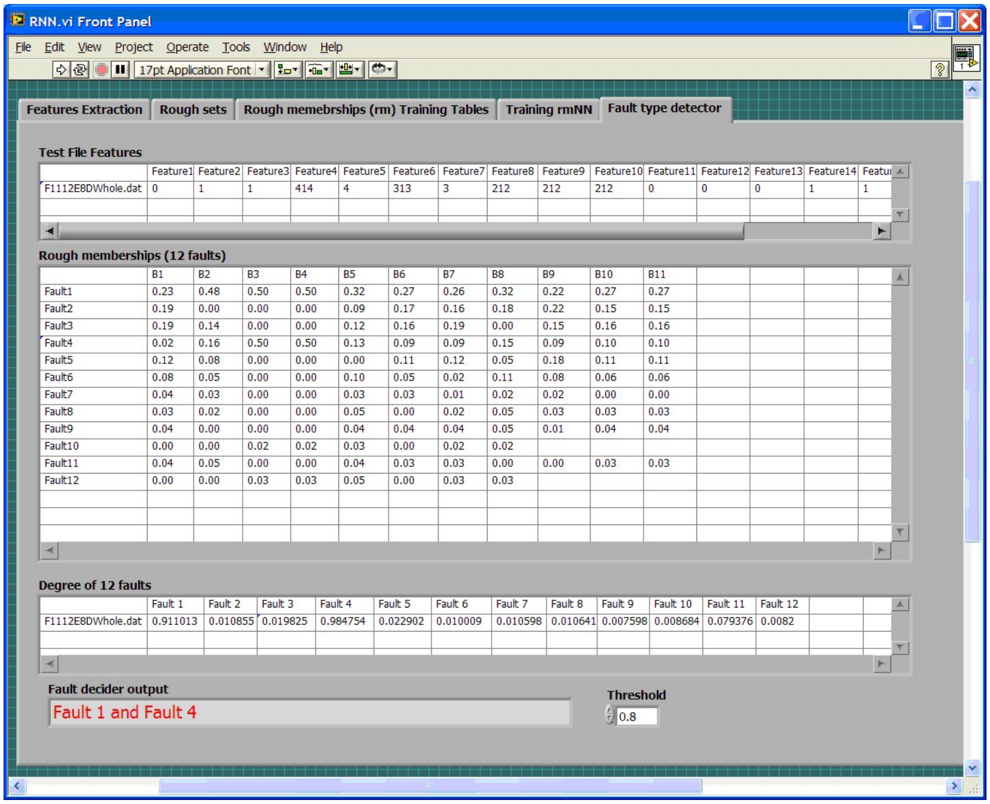


Figure 82: Power system fault type detection.

The developed graphic user interface makes the power system fault classification easy to operate for users. The software integrates features extraction, rough sets construction, rough membership computation, rmNNs training, verification and fault type detection into one system, which provides the possibility for further implementation of an adaptive learning real-time fault classification system. Currently, the most time consuming component of the software package is the features extraction, which takes two to three seconds and needs to be computationally optimized. The rough membership computation and fault type detection take less than one second, which meets the need of real-time.

8 Normal Artificial Neural Network (ANN) for PSFC

Nomenclature

Symbol	Brief Explanation
$B_i(Obj\_n)$	Output of attribute set B_i for a given fault data file n
r_{ij}	Weights on the connection from i^{th} neuron in the input layer to the j^{th} neuron in the hidden layer in the first and second type of artificial neuron networks
w_{jk}	Weights on the connection from j^{th} neuron in the hidden layer to the k^{th} neuron in the output layer in the first type of artificial neuron network
h_j	Output of the j^{th} neuron in the hidden layer in the first type of artificial neuron network
O_k	The degree estimation of the k^{th} type of fault, the output of the k^{th} neuron in the output layer in the first type of artificial neuron network
$g(\cdot)$	The logistic sigmoid activation function in normal artificial neuron network
w_j	Weights on the connection from j^{th} neuron in the hidden layer to the output neuron in the second type of artificial neuron network
h_j^k	Output of the j^{th} neuron in the hidden layer in k^{th} second type of artificial neuron network
O^k	The degree estimation of the k^{th} type of fault, the output of the k^{th} second type of artificial neuron network.

Before using the rmNN to classify the power system faults, two normal artificial neural networks were investigated. Each row in the information table (Table 12) is an input to the neurons in the first layer of both ANNs. No rough membership is computed. The performance of these two types of ANNs is addressed in detail in the following two sections.

8.1 A Single ANN for PSFC

First, a single ANN was considered to classify 12 types of faults. The single ANN is designed with 3 layers, containing 17 neurons in the input layer and hidden layer and 4 neurons in the output layer. Each neuron in the output layer indicates one possible type of fault. As seen in Table 12, some *.x01 files are involved with 4 types of faults, *i.e.*, F082016A.x01 is intervened with faults 1, 3, 5 and 7. The four output neurons are expected to output 1, 3, 5 and 7 respectively when processing fault F082016A.x01.

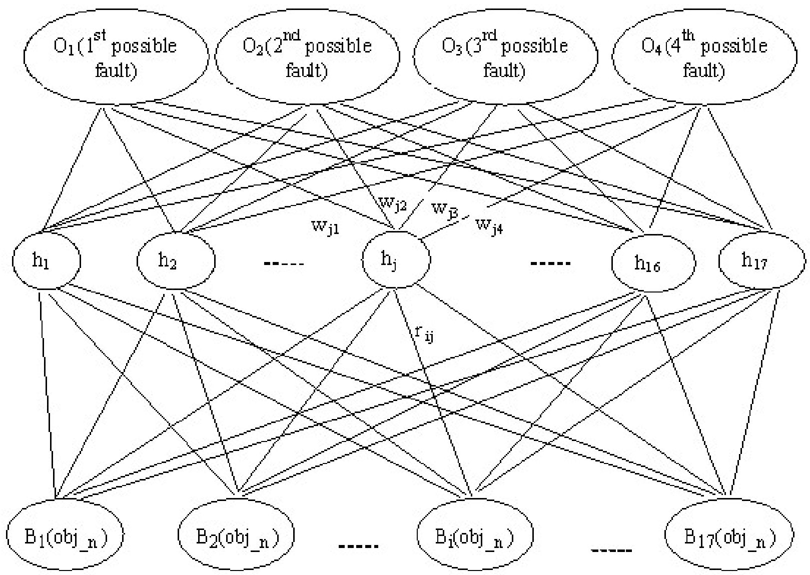


Figure 83: The architecture of the single ANN power fault classification system.

The architecture of the single ANN is illustrated in Figure 83, where

$$h_j = g\left(\sum_{i=1}^{17} r_{ij} B_i\left(obj\_n\right)\right), \tag{85}$$

$$O_k = \sum_{j=1}^{17} w_{jk} h_j, \tag{86}$$

and $g(\cdot)$ is the logistic sigmoid activation function given by

$$g(a) = \frac{1}{1 + e^{-a}}. \tag{87}$$

The 17 neurons in the first layer receive values from functions representing 17 features. Unlike an rmNN, the features are not grouped. Back-propagation is adopted as the network training function. A gradient descent learning function is used as the weights updating function, and the least squared error function is used as the learning performance evaluation function. The learning performance is shown in Figure 84. After 1600 learning cycles, the LSE converged to 1.23.

Once the ANN is calibrated, 168 test files are applied to evaluate the accuracy of this ANN fault classification system. Four neuron output compared with four targets are displayed in Figures 85, 86, 87 and 88, respectively. It is noticeable that over 60% of testing files failed the verification.

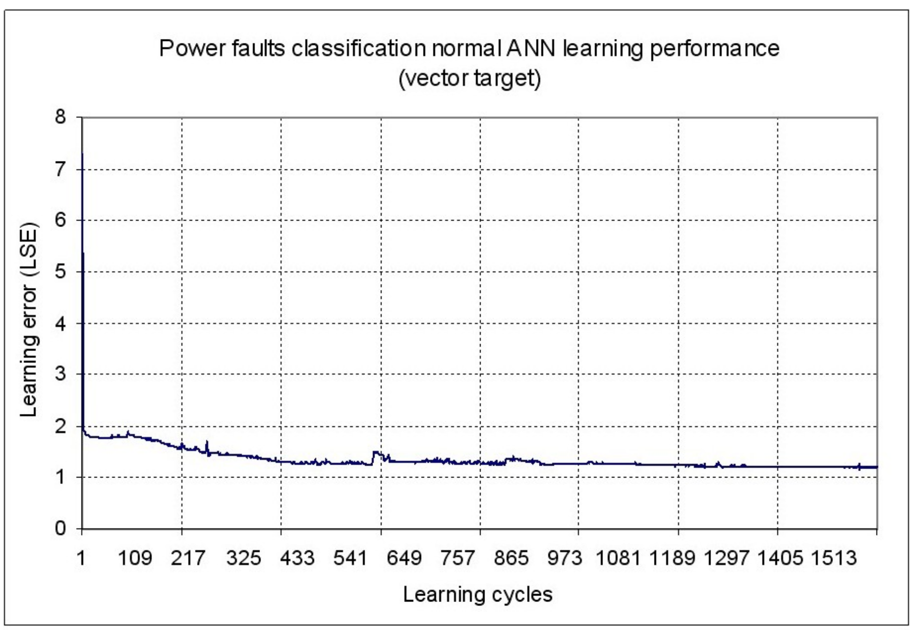


Figure 84: Learning performance of the single ANN for power system fault classification.

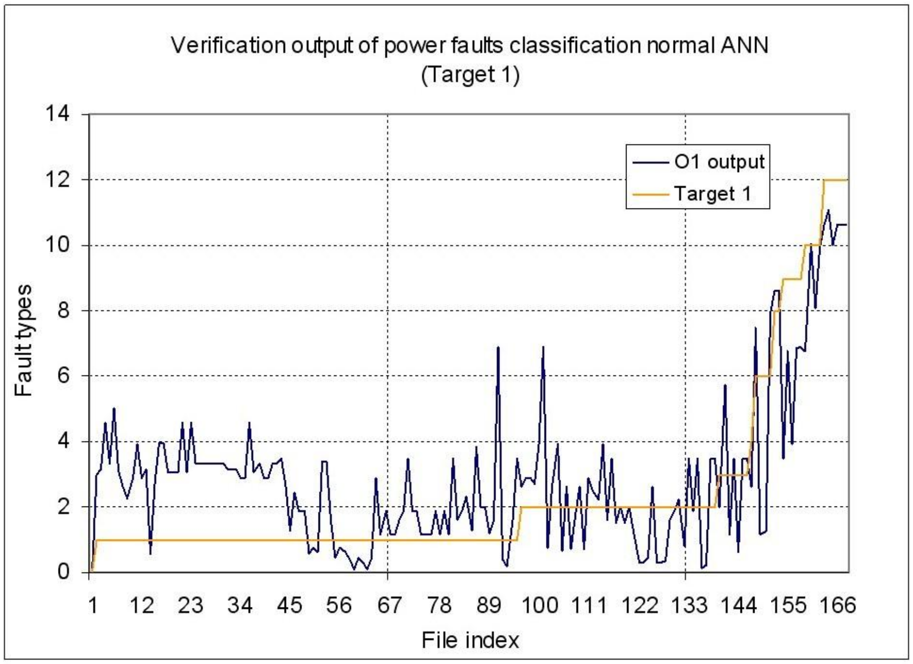


Figure 85: Target 1 verification.

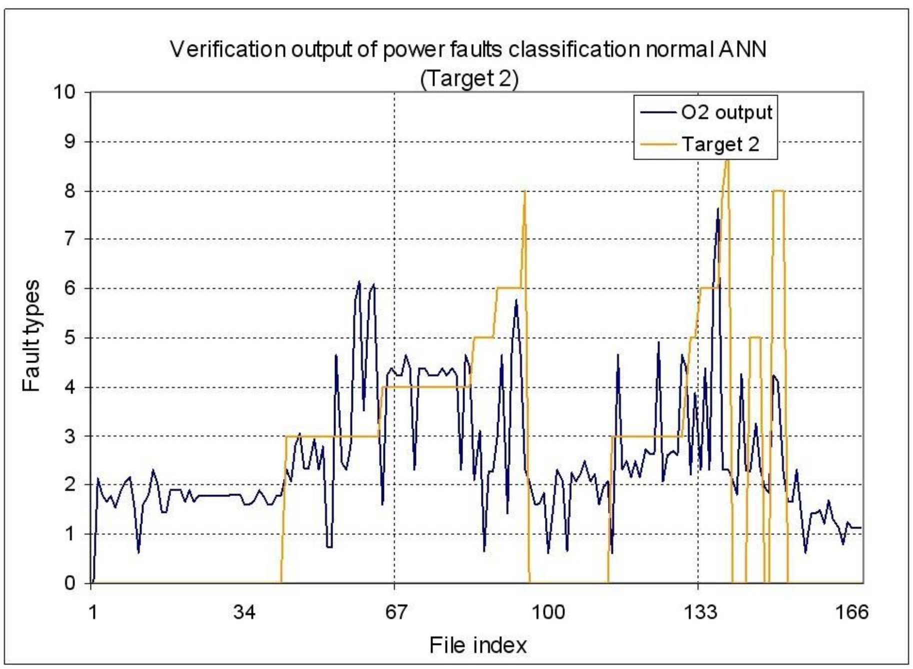


Figure 86: Target 2 verification.

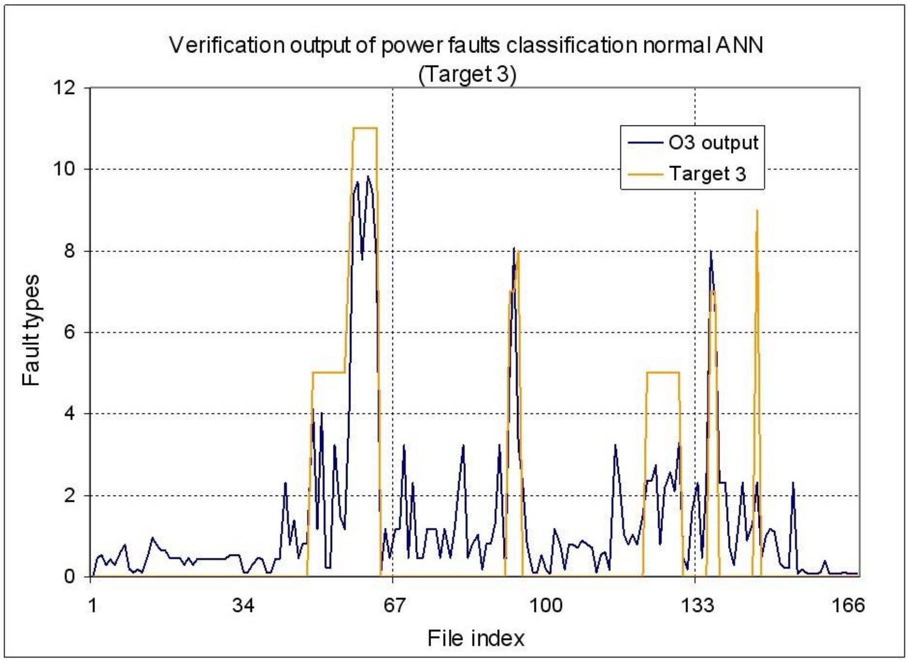


Figure 87: Target 3 verification.

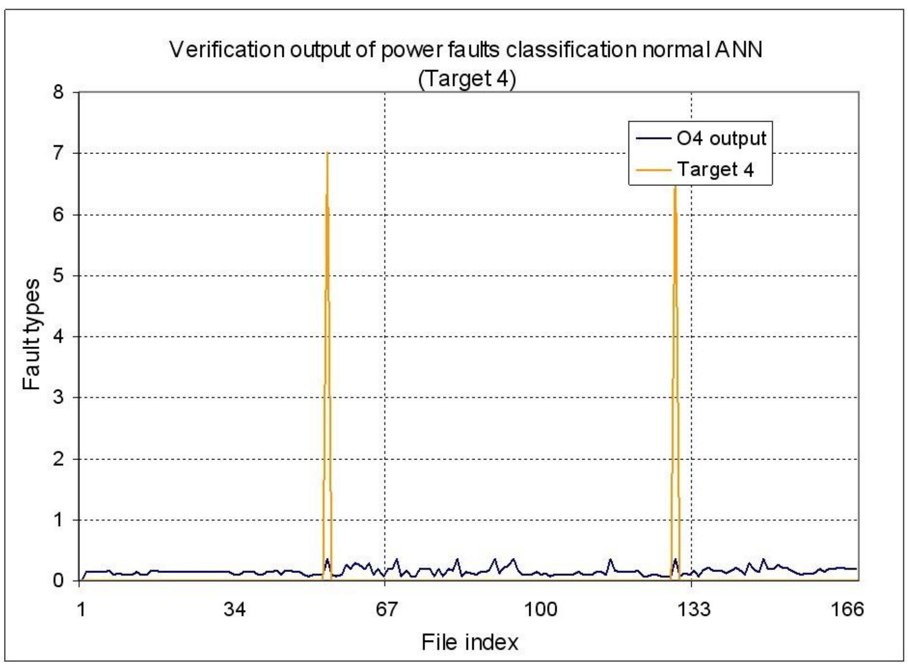


Figure 88: Target 4 verification.

8.2 Twelve Sub-ANNs for PSFC

The basic architecture of the second ANN power fault classification system is similar to the rmNNs described earlier. Twelve ANNs are created and each of them classifies one type of fault. An example of ANN for k^{th} fault classification is illustrated in Figure 89. The output from the 12 ANNs are the estimations of the degrees of the 12 faults respectively. The output from the 12 ANNs are forwarded to a decider neuron, which simply picks up the faults with degrees above a preset threshold.

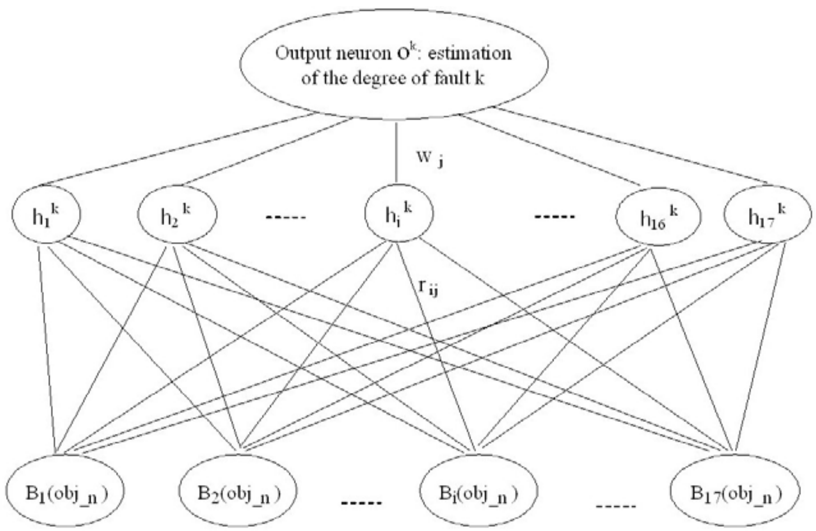


Figure 89: The architecture of a sample ANN for k^{th} fault classification.

In Figure 89,

$$h_j^k = g\left(\sum_{i=1}^{17} r_{ij} B_i\left(obj\_n\right)\right), \tag{88}$$

$$O^k = \sum_{j=1}^{17} w_j h_j^k, \tag{89}$$

and $g(\cdot)$ is the logistic sigmoid activation function same as the one applied in the first type of ANN.

The 17 neurons in the first layer receive values from functions representing 17 features. Again, the features are not grouped. Backpropagation is still used as the network training function, the gradient descent learning function as the weights learning function, and the least squared error function as the learning performance evaluation function. The learning performance for fault 3 and 5 ANN is shown in Figures 90 and 91 respectively. After 800 learning cycles, both LSEs are approximately 10.

After 800 learning cycles, 168 test files are applied to evaluate the performance of the ANNs. The testing output for fault 3 and 5 ANN are displayed in Figures 92 and 93 respectively. It is obvious that the threshold to pick up the fault has to be reduced to 60% to generate better accuracy. The results for the calibrations and verifications of the 12 ANNs are attached in Appendix F. The classification accuracy is listed in Table 34. Compared with the rmNN system, the accuracy of the ANN fault classification system is fairly poor. It either produces low detection accuracy for the desired faults or generates a great number of unexpected false alarms.

The failure of both ANN fault classification systems is possibly due to the input, which are the 17 feature values. Consider the feature 5 (Pole Current Trend), two possible codes are “313” and “343”. They are very close in terms of the values of these two numbers when treated by the ANN system. But “313” usually happens in fault 4 (Line Faults), and “343” happens in fault 1 (Minor AC Disturbance). The rough membership computation distinguishes these two numbers by assigning each of them with the degree of each type of fault, which greatly improves the quality of the feature information and consequently the classification performance.

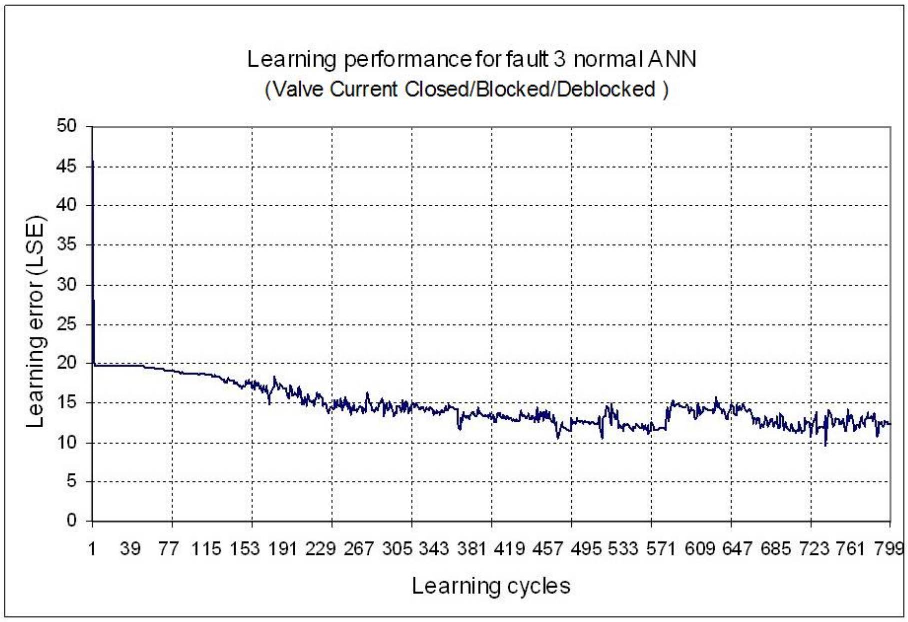


Figure 90: The learning performance for fault 3 ANN.

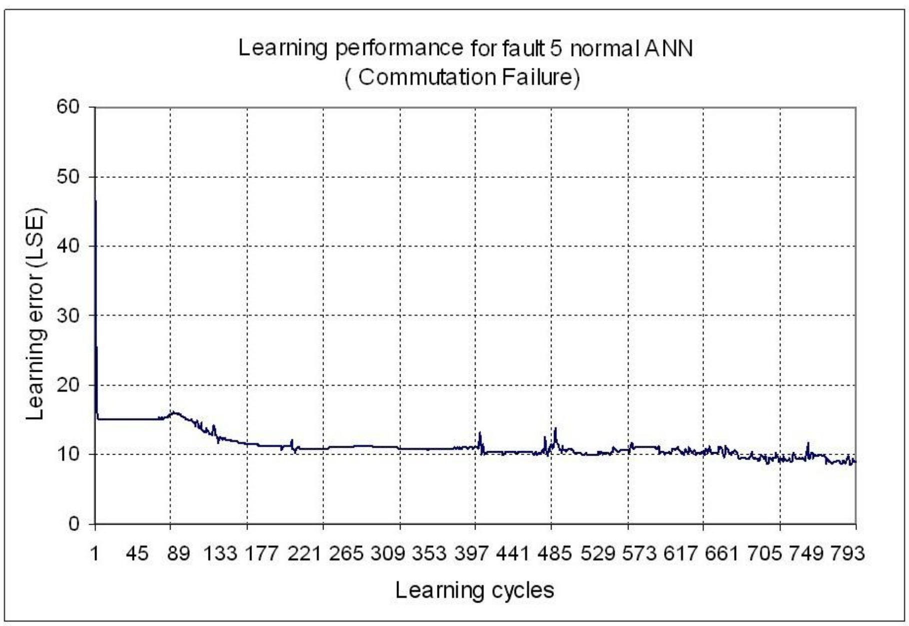


Figure 91: The learning performance for fault 5 ANN.

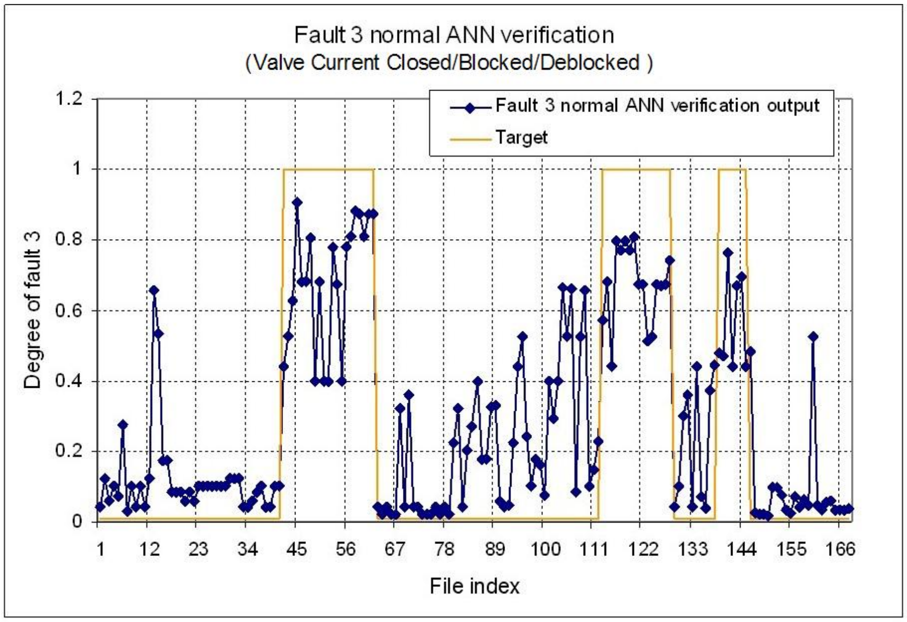


Figure 92: Fault 3 ANN verification.

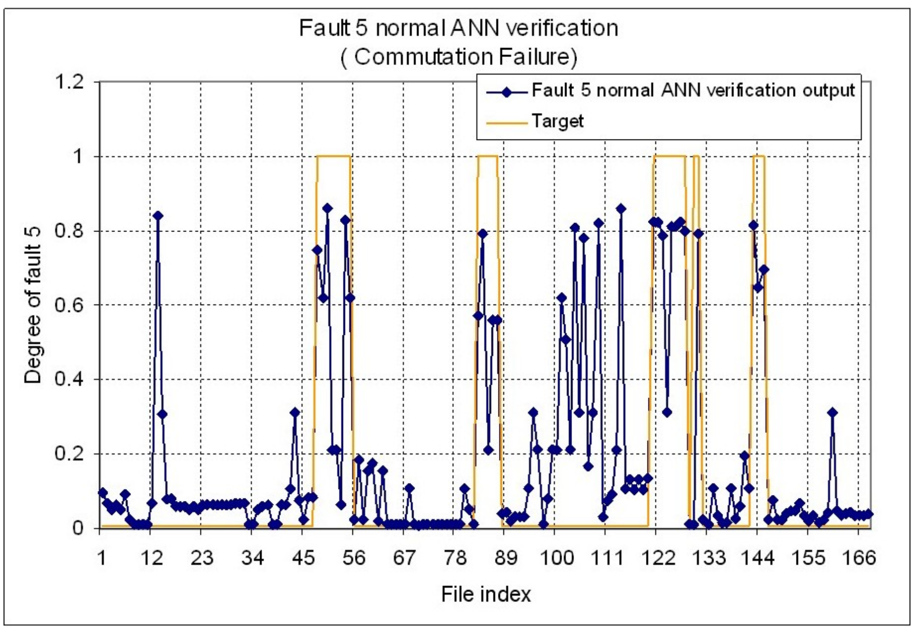


Figure 93: Fault 5 ANN verification.

Table 34: Accuracy of 12 ANNs for PSFC.

Fault type	# of files for verification	# of files incorrect	Accuracy
Minor AC Disturbance	94	93	0.011
AC Disturbance	44	25	0.432
Valve Current Closed/Blocked/Deblocked	44	14	0.68
Line Fault	22	13	0.41
Commutation Failure	25	5	0.8
Pole Voltages/Currents Closed/Blocked/Deblocked	15	7	0.53
Current Arc Back	6	1	0.83
Parallel Operation	9	5	0.44
Pole Current Oscillation	7	7	0
Normal Affected by Another Pole	4	4	0
Asymmetric Protection	6	5	0.17
Disturbance on DC Voltage	7	7	1

9 Classifier Fusion

Nomenclature

Symbol	Brief Explanation
LMD	Linear Mean and Deviation
SORM	Sum of 11 Rough Memberships

9.1 Motivation in Using a Second Complementary Classifier

A number of classifier fusion methods have been recently developed and lead to potential improvement in classification performance. In this section, a second successful classifier based on mean and standard deviation evaluation of the sum of 11 rough memberships is proposed. The goal is to take advantage of the diversity of two classifiers to improve the performance of PSFC.

To achieve high overall performance of the classification function, the performance of each individual classifier has to be optimized prior to using it within any fusion schemes. That is, the fusion scheme will be able to improve the overall classification result relative to the performance of the individual classifiers. If several classifiers with only marginal performance are being used, the results cannot necessarily be expected to reach high performance. On the other hand, if several classifiers are used that work exceptionally well, any further gains will be exceedingly hard to accomplish because the opportunity for diversity is diminished.

Recall the performance of the 12 rmNNs. Table 35 lists the minimum rmNN output for true cases and the maximum rmNN output for false cases in both learning and verification.

The classification performance of the rmNNs for fault 1, fault 2, fault 4, fault 5 and fault 12 are excellent and both the learning and verification output for the true cases have high scores over 0.9, while for the false cases have low scores less than 0.16.

Table 35: Maximum and minimum rmNN output for false and true cases, respectively.

	Learning		Verification	
	Lowest output for true cases	Highest output for false cases	Lowest output for true cases	Highest output for false cases
fault 1	**0.94**	**0.08**	**0.93**	**0.01**
fault 2	**0.90**	**0.16**	**0.90**	**0.04**
fault 3	0.87	0.22	0.82	0.19
fault 4	**0.98**	**0.06**	**0.98**	**0.01**
fault 5	**0.94**	**0.14**	**0.95**	**0.04**
fault 6	0.74	0.50	0.84	0.27
fault 7	0.85	0.31	0.79	0.01
fault 8	0.75	0.06	0.85	0.01
fault 9	0.87	0.15	0.87	0.01
fault 10	0.68	0.56	0.66	0.01
fault 11	0.61	0.34	0.81	0.39
fault 12	**0.99**	**0.01**	**0.99**	**0.01**

Faults 1, 2, 4, 5 and 12 do not need to be reinforced by a second complementary classifier. However, for faults 3, 6, 7, 8, 9, 10 and 11 classification, a second LMD classifier is introduced to fusion the output from rmNNs in order to increase the overall PSFC accuracy.

9.2 Linear Mean-Deviation (LMD) Based Classifier

The input for the linear mean and deviation based (LMD) classifier is the sum of 11 rough memberships (SORM) in the training and testing tables for rmNNs. Figure 94 shows the SORMs of 508 training files for fault 7.

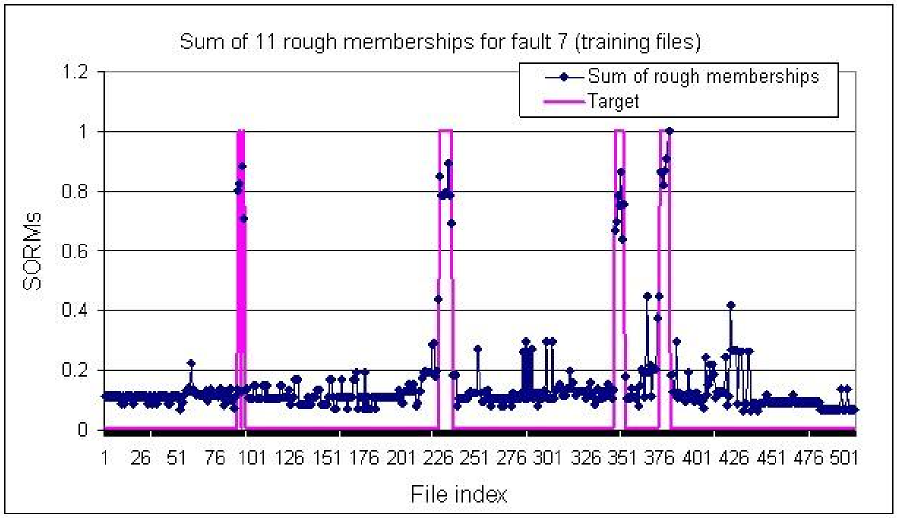


Figure 94: The SORMs of 508 training files for fault 7.

From all the points of SORM output, three sets are constructed. Set 1 consists of all the points of true case with SORM values over 0.85. Set 2 contains all the points of true case with SORM values less than 0.85. Set 3 collects all the points of false case. These 3 sets are illustrated in Figure 95.

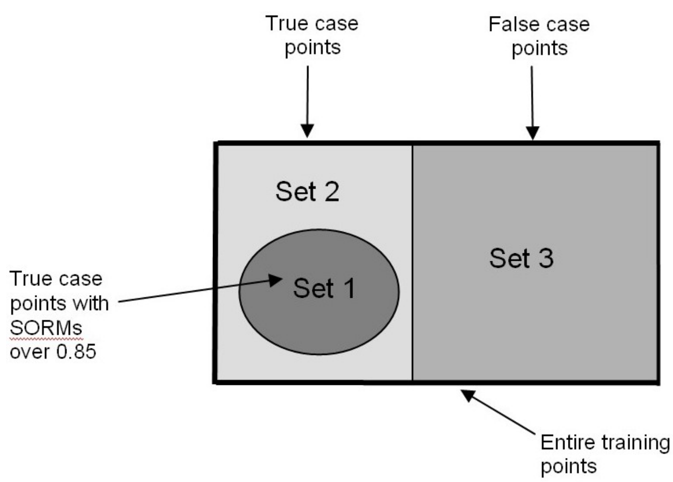


Figure 95: Three sets of SORMs.

The points in sets 2 and 3 will be employed to estimate the mean and deviation values to establish the distribution functions of set 2 and set 3. Assuming that x_{ij} is the j^{th} point in set i and there are N_i points in set i. The mean and absolute deviation for set i, *i.e.*, μ_i and dev_i are defined as follows.

$$\mu_i = \frac{1}{N_i}\sum_{j=1}^{N_i} x_{ij}. \tag{90}$$

$$dev_i = \frac{1}{N_i}\sum_{j=1}^{N_i} |x_{ij} - \mu_i|. \tag{91}$$

The Gaussian distribution function of set i, $f_i(x)$, is defined as,

$$f_i(x) = \frac{1}{\sqrt{2\pi(dev_i)^2}} e^{\frac{-(x-\mu_i)^2}{2(dev_i)^2}}. \tag{92}$$

Take fault 7 as an example, the mean and deviation of sets 2 and 3 are calculated, and listed in Table 36.

Table 36: The mean and deviation of sets 2 and 3 for fault 7 training files.

Fault 7 training files			
set 2		set 3	
μ_2	dev_2	μ_2	dev_3
0.758	0.12	0.126	0.324

The degree of fault 7 will then be calculated as described in (93)

$$deg(x) = \begin{cases} x, & if \;\; x \geq 0.85, \\ \frac{f_2(x)+1-f_3(x)}{2}, & if \;\; x < 0.85. \end{cases} \tag{93}$$

Keep the SORM as the degree of fault 7, if it is bigger or equal to 0.85. $\frac{f_2(x)+1-f_3(x)}{2}$ is only applied to the points with SORM values that are less than 0.85. In this way, the degree of fault 7 of the points in set 2 is raised. This method is applied to faults 3, 6, 7, 8, 9, 10 and 11. For example, the degree of fault 7 of 508 training files is shown in Figure 96.

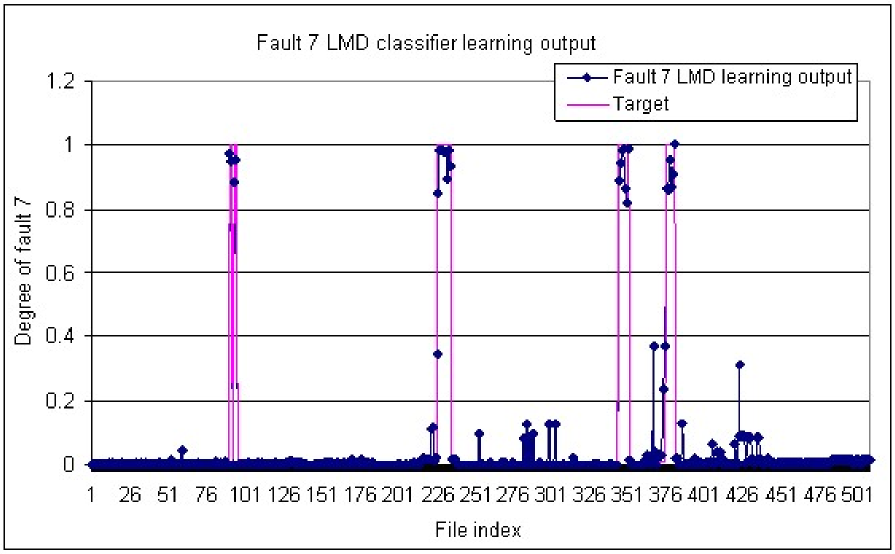


Figure 96: Fault 7 LMD classifier output for 508 training files.

Use the training files, the mean and deviation of sets 2 and 3 can be estimated to set up the distribution function for the points of true case with SORM values less than 0.85 and the distribution function for the points of false case. The trained distribution functions will be applied to the test points to estimate the degree of a type of fault. In this example, the degree of fault 7 is estimated.

The SORMs of fault 7 for 168 testing files are shown in Figure 97, and the fault 7 LMD classifier output for 168 testing files are shown in Figure 98. The degrees of fault 7 for all the true case points are above 0.87. One point, file 128, exists in the verification output for fault 7 rmNN. It has a low estimation of the degree of fault 7, which is only 0.79 (Table 35, Section 9.1).

The fault 10 LMD classifier results are illustrated in Figures 99, 100, 101 and 102. The SORMs of 508 training files are shown in Figure 99, and the LMD classifier training output is shown in Figure 100. The SORMs of 168 testing files are shown in Figure 101, and the

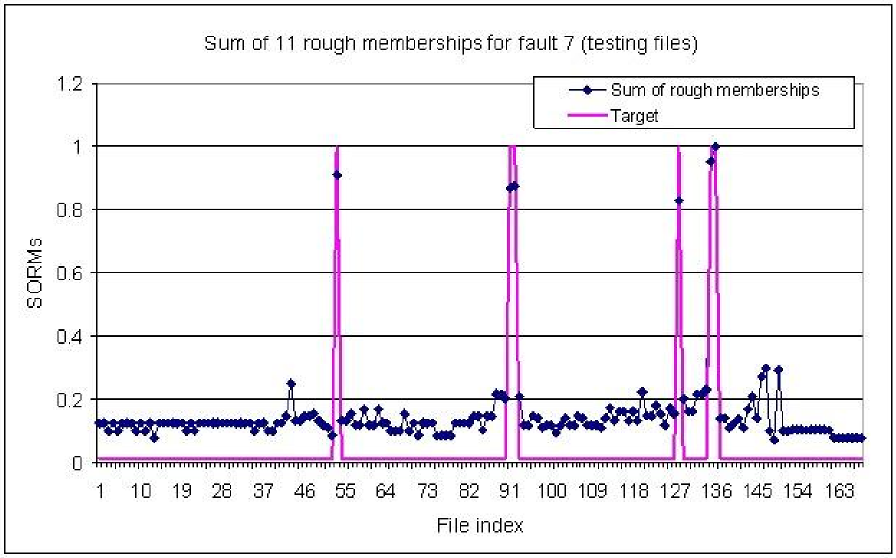


Figure 97: The SORMs of 168 testing files for fault 7.

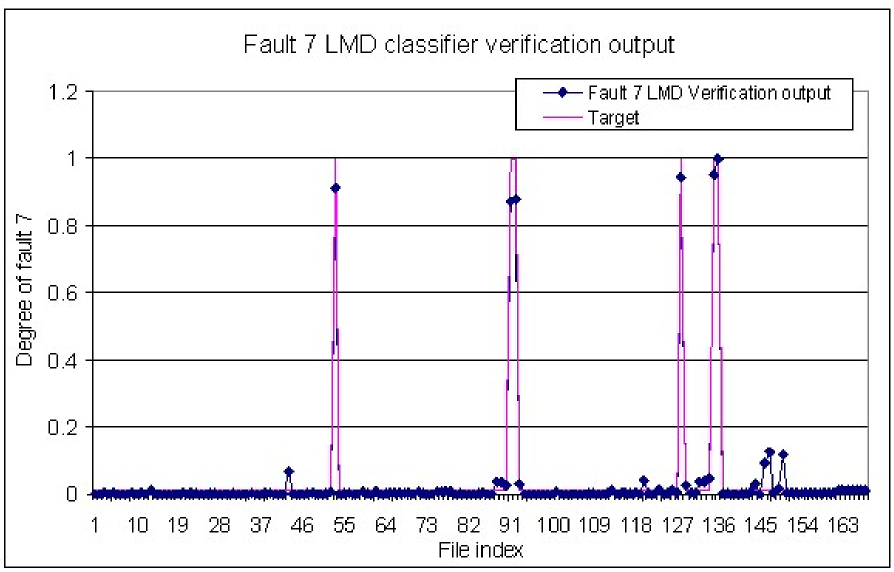


Figure 98: Fault 7 LMD classifier output for 168 testing files.

LMD classifier testing output is shown in Figure 102. In Figure 99, it is very clear that only two points (file 471 and file 472) have low SORM output (*i.e.*, 0.745 and 0.746) and they are almost at the same level. The distribution function of set 2 is designed based on these two points and the degree of fault 10 for these two points from the LMD classifier is high and raised to 0.985. In the test, the degree of fault 10 for file 159 is boosted to 0.984 as well. The rmNN classifier testing output for this point, however, is as low as 0.66 (Table 35, Section 9.1).

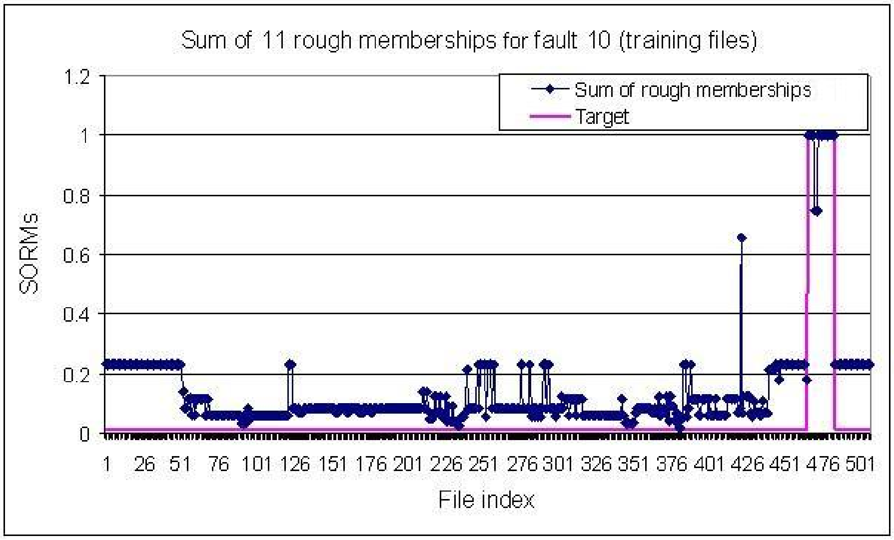


Figure 99: The SORMs of 508 training files for fault 10.

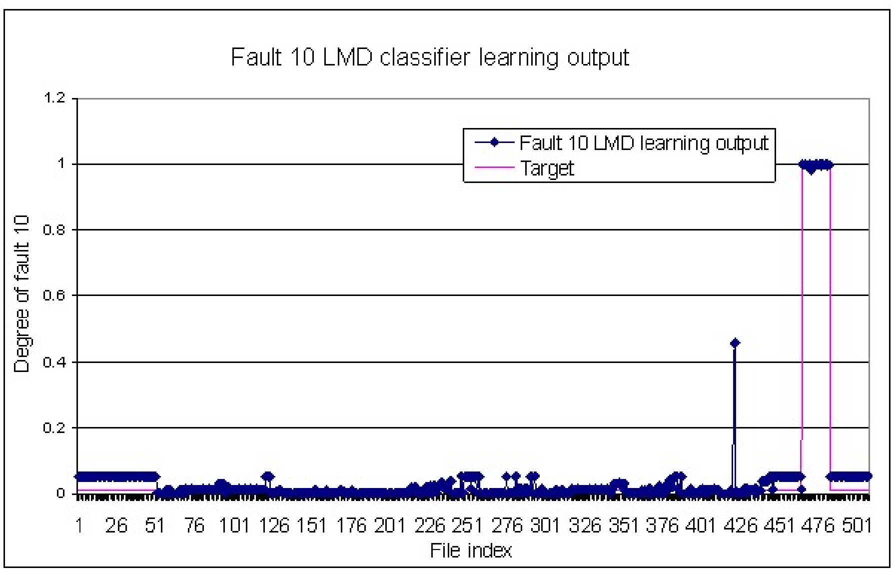


Figure 100: Fault 10 LMD classifier output for 508 training files.

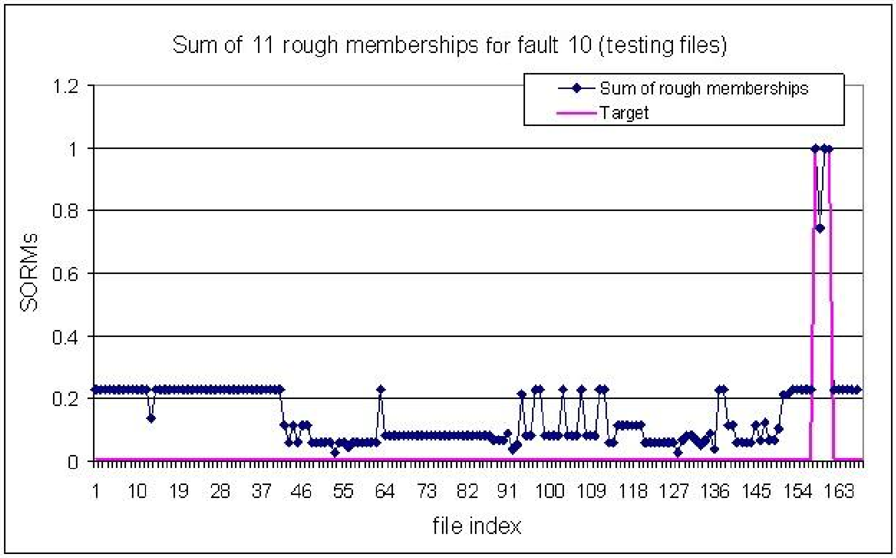


Figure 101: The SORMs of 168 testing files for fault 10.

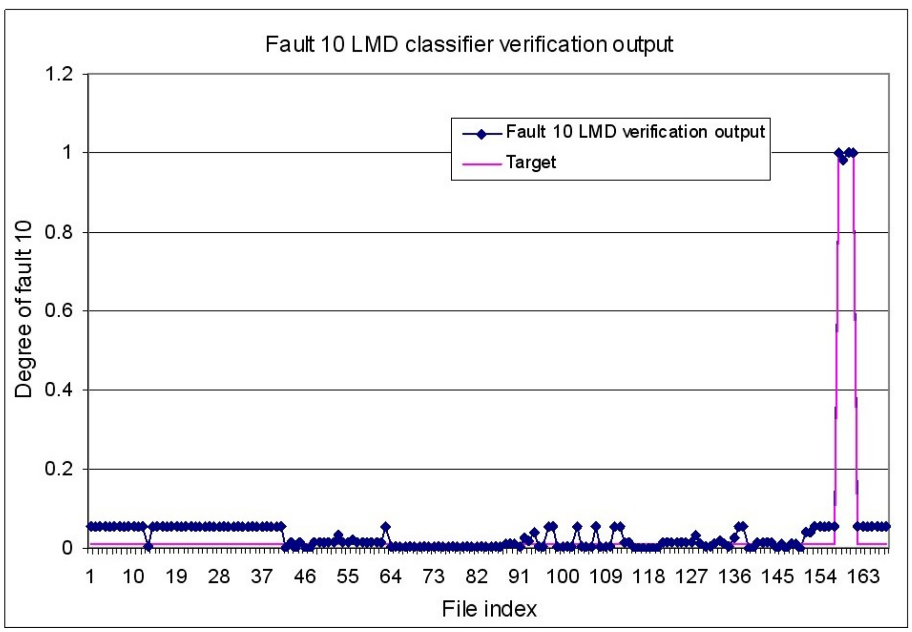


Figure 102: Fault 10 LMD classifier output for 168 testing files.

The LMD classifier training and testing results for fault 6 are illustrated in Figures 103 to Figure 106. There is one point, file 90, in the fault 6 LMD classifier testing output, which gives a low estimation of the degree of fault 6. The degree of fault 6 is only 0.786.

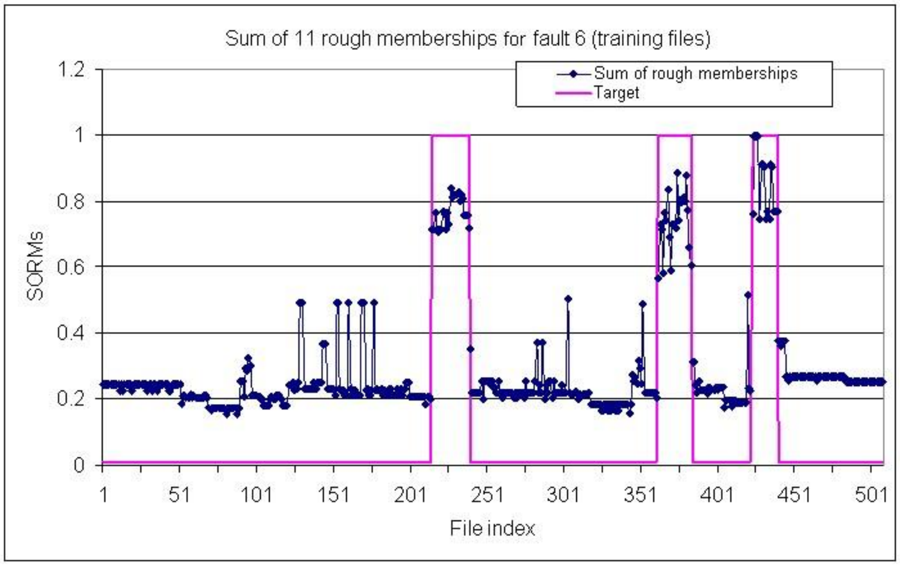


Figure 103: The SORMs of 508 training files for fault 6.

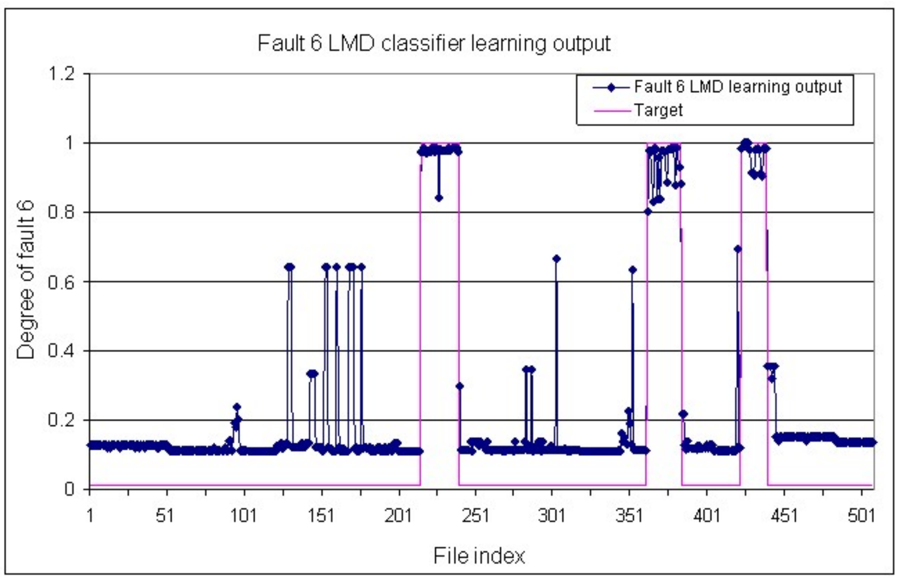


Figure 104: Fault 6 LMD classifier output for 508 training files.

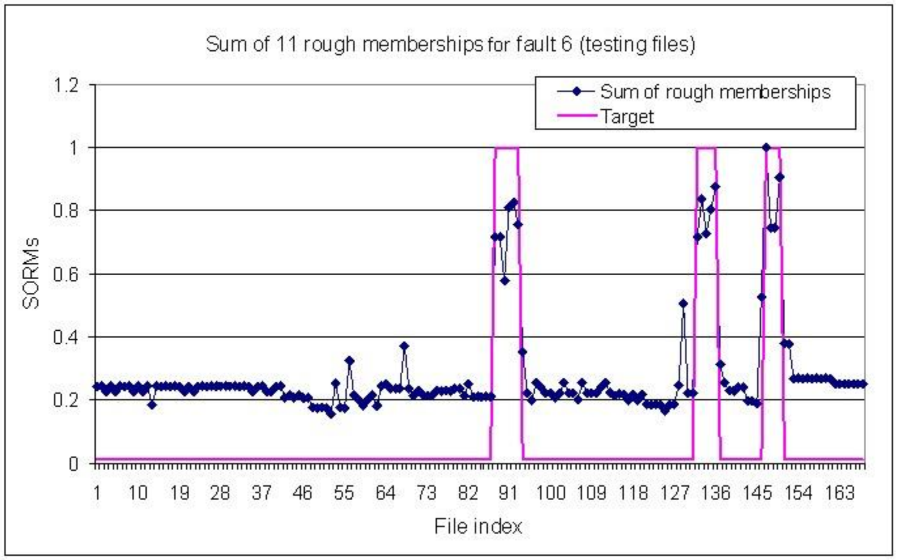


Figure 105: The SORMs of 168 testing files for fault 6.

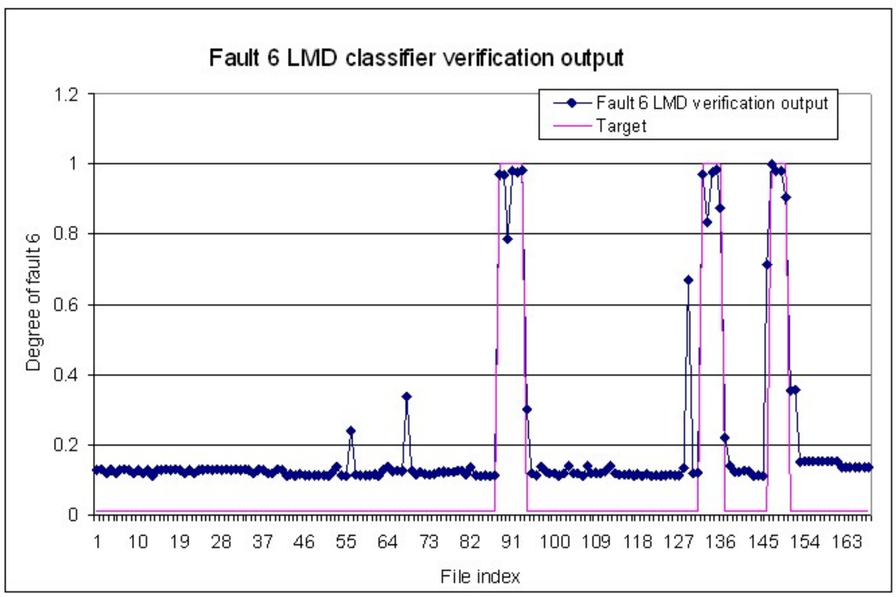


Figure 106: Fault 6 LMD classifier output for 168 testing files.

The SORMs for faults 3, 6, 7, 8, 9, 10 and 11 of both training and testing files are listed in Appendix G. The fault 3, 6, 7, 8, 9, 10 and 11 LMD classifier training and verification output is included in Appendix H.

Table 37 summarizes the accuracy of the LMD classifiers. Except that the accuracy for fault 6 classification is 0.93, the accuracy for all of the other 6 faults is 100%. LMD classifier considers the isolated points with medium and low SORM values. For fault 7, “Current Arc Back”, and fault 10, “Normal affected by another pole”, only 26 and 18 fault files are employed for calibration respectively; but the LMD classifier test result is 100% accurate. On the other hand, the rmNN classifier gives poorer results when it deals with a fault with less files participate in learning. For fault 7 and 10, the rmNN classifier verification accuracy is only 0.83 and 0.75 respectively (Table 32, Section 7.4.5).

Table 37: Accuracy of the LMD power fault classification system.

Fault type	# of files for verification	# of files incorrect	Accuracy
Fault 3: Valve Current Closed/Blocked/Deblocked	44	0	1
Fault 6: Pole Voltages/Currents Closed/Blocked/Deblocked	15	1	0.93
Fault 7: Current Arc Back	6	0	1
Fault 8: Parallel Operation	9	0	1
Fault 9: Pole Current Oscillation	7	0	1
Fault 10: Normal Affected by Another Pole	4	0	1
Fault 11: Asymmetric Protection	6	0	1

One point that needs mentioning is that the LMD classifier is not suitable for the classification for all 12 faults. Look at the SORMs for the 508 training files of fault 1 and fault 2 (Figures 107 and 108), where the SORMs of many false and true cases are comparable, which causes the failure of the LMD classifier. The good thing is that the accuracy of the rmNN classifier for these two faults is excellent and compensates the weakness of the LMD classifier.

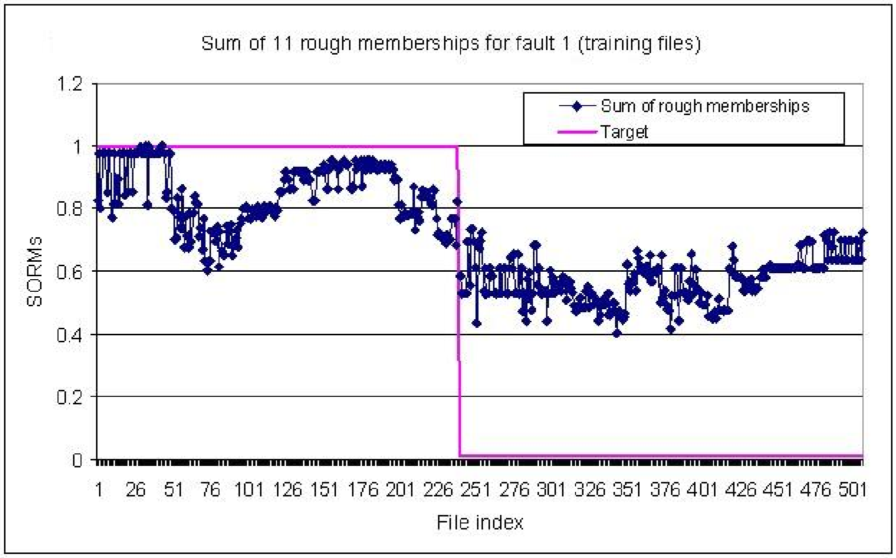


Figure 107: The SORMs of 508 training files for fault 1.

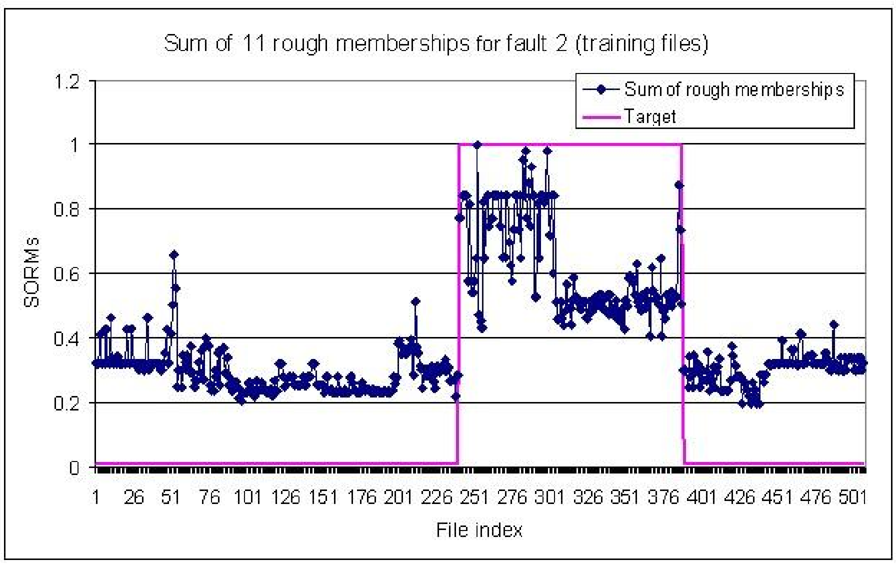


Figure 108: The SORMs of 508 training files for fault 2.

9.3 Correlation of the rmNN and LMD Classifier

In classifier fusion, it is desirable to use classifiers that not only offer reasonable performance but also have a mutually low correlation. If two classifiers are completely redundant, many fusion schemes not only will not gain anything, but will actually exhibit poorer performance. Obviously, some degree of confirmatory information is desirable, but it is the complementary information that gives the multi-classifier fusion a chance for success.

In this section, the correlation of the rmNN and LMD classifier will be estimated to prove that they are good candidates for classifier fusion.

According to the 2-Classifier correlation analysis mentioned in Section 3.7, the formula for the correlation is:

$$\rho_2 = \frac{2 \times N^{FF}}{N^{TF} + N^{FT} + 2 \times N^{FF}}, \tag{94}$$

where,

TT represents that the output of the rmm_NN is T and the output of the LMD is T;

TF represents that the output of the rmm_NN is T and the output of the LMD is F;

FT represents that the output of the rmm_NN is F and the output of the LMD is T; and

FF represents that the output of the rmm_NN is F and the output of the LMD is F; and the following two methods are also applied for the correlation evaluation.

1. Try 3 thresholds for the 'true' decision making;

2. Consider both the training and testing files.

The correlation estimations of rmNN and LMD classifier for faults 3, 6, 7, 8, 9, 10 and 11 are listed in the following 3 tables for 3 thresholds respectively. Table 38 shows that the correlations are all 0 for 7 faults when the threshold = 0.8 for 'true' decision making. Table 39 shows that the correlations are still 0 when the threshold is pushed to 0.85. Table 40 shows that the correlation for fault 6 and 11 are increased to 0.2 and 0.267 respectively when the threshold is pushed to 0.86. But 0.2 and 0.267 still have a reasonably low correlation level to ensure the success of the classifier fusion.

Table 38: 2-Classifier correlation estimation (Threshold for true case is 0.8).

Threshold 0.8						
	Total true cases	TT	TF	FT	FF	correlation
fault 3	189	189	0	0	0	0
fault 6	79	77	1	1	0	0
fault 7	32	31	0	1	0	0
fault 8	38	37	0	1	0	0
fault 9	38	38	0	0	0	0
fault 10	22	19	0	3	0	0
fault 11	31	30	0	1	0	0

Table 39: 2-Classifier correlation estimation (Threshold for true case is 0.85).

Threshold 0.85						
	Total true cases	TT	TF	FT	FF	correlation
fault 3	189	188	0	1	0	0
fault 6	79	70	6	3	0	0
fault 7	32	29	1	2	0	0
fault 8	38	33	3	2	0	0
fault 9	38	37	1	0	0	0
fault 10	22	18	0	4	0	0
fault 11	31	20	0	11	0	0

Table 40: 2-Classifier correlation estimation (Threshold for true case is 0.86).

Threshold 0.86						
	Total true cases	TT	TF	FT	FF	correlation
fault 3	189	188	0	1	0	0
fault 6	79	70	5	3	1	0.2
fault 7	32	27	3	2	0	0
fault 8	38	27	3	8	0	0
fault 9	38	37	1	0	0	0
fault 10	22	18	0	4	0	0
fault 11	31	18	0	11	2	0.266667

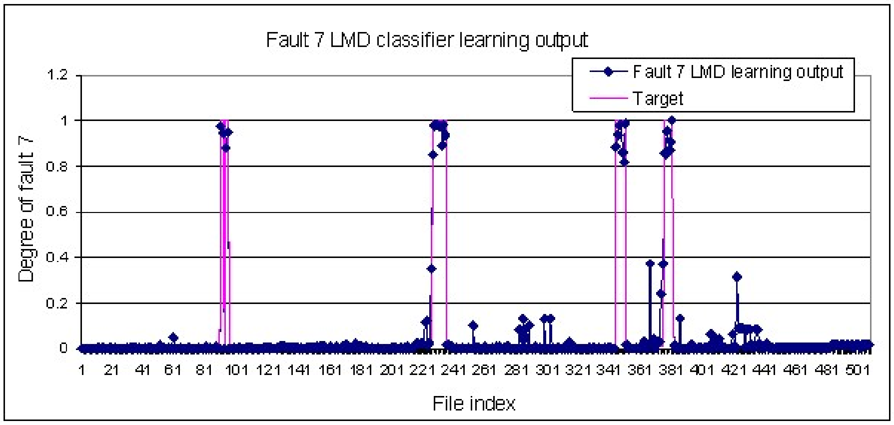
9.4 Results of the rmNN and LMD Classifier Fusion

The fusion methods are less important than the diversity of the classifier team, but still need to consider which method is more suitable for specific problem solving.

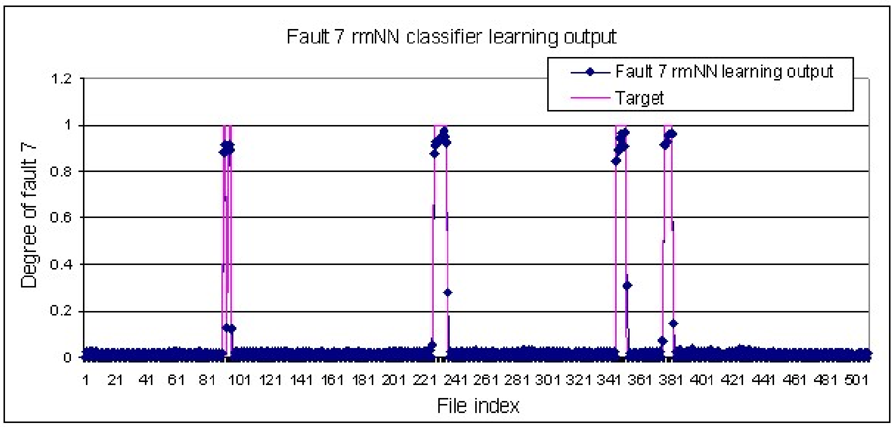
The classifier fusion function for two classifiers can be minimum, maximum, average, median and oracle. The majority vote usually applies when having more than two classifiers. The minimum will not help in this PSFC system. The maximum and oracle emphasize the possible true points and it is easy to generate a false alarm. The average and median methods are relatively soft and safe and their performances are approximately the same. The average method is tried in this PSFC system and tested out to gain excellent classification performance.

Once again, take fault 7 as an example. The training output for fault 7 LMD and the rmNN classifier are shown in Figures 109(a) and 109(b) respectively. The output is the degree of fault 7. The average of the two training output is shown in Figure 109(c). The lowest point in the true cases from the LMD classifier is at file 350 with a value of 0.817. On the other hand, the lowest point in the true cases from the rmNN classifier is at file 345 with a value of 0.845. After averaging, the degree of fault 7 for file 350 is increased to 0.862, which is the lowest point after classifier fusion. In the training, the rmNN classifier helps to lift the lowest point and improve the PSFC performance.

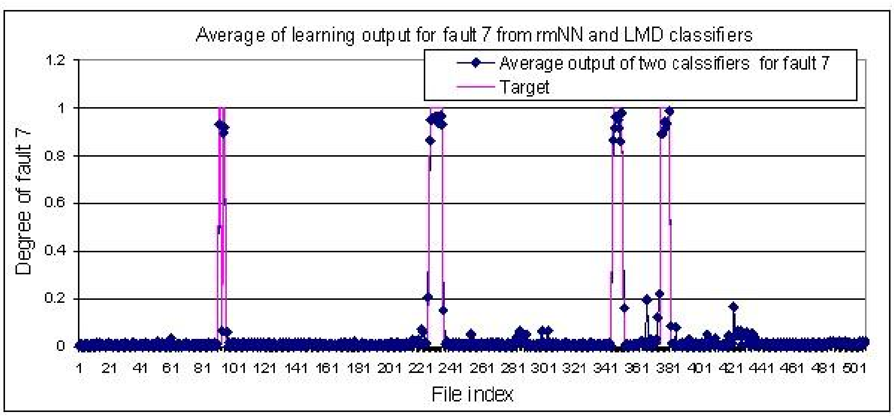
Now consider the verification results, which are illustrated in Figures 110(a), 110(b) and 110(c). The lowest point in the true cases from the LMD classifier is at file 91 with a value of 0.869. On the other hand, the lowest point in the true cases from the rmNN classifier is at file 128 with a value of 0.792. After averaging, the degree of fault 7 for file 128 is increased to 0.867, which is the lowest point after classifier fusion; and the overall performance of the PSFC is improved. In the testing process, the LMD classifier helps to lift the lowest point and improve the PSFC performance.



(a) The learning output for fault 7 from the LMD classifier.

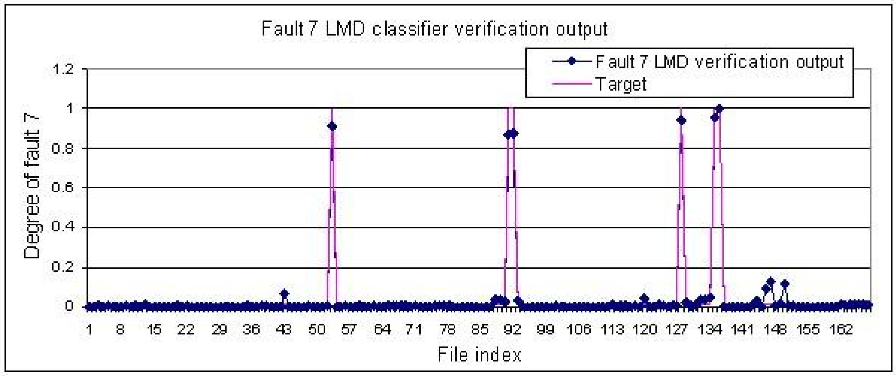


(b) The learning output for fault 7 from the rmNN classifier.

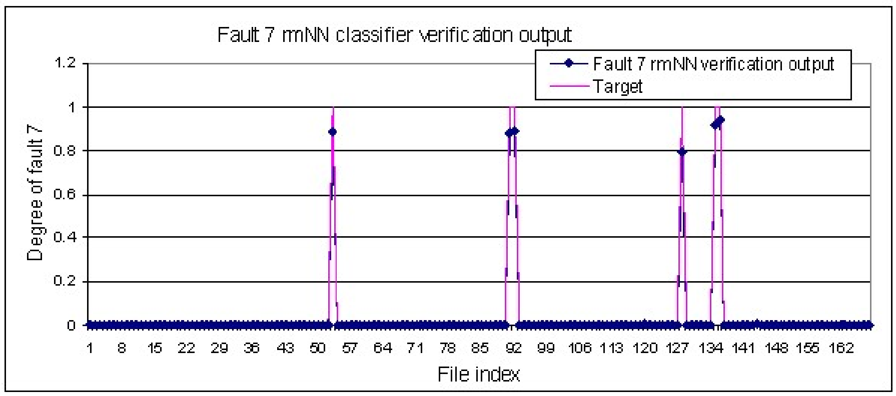


(c) The average of two learning output for fault7 from the LMD and rmNN classifiers.

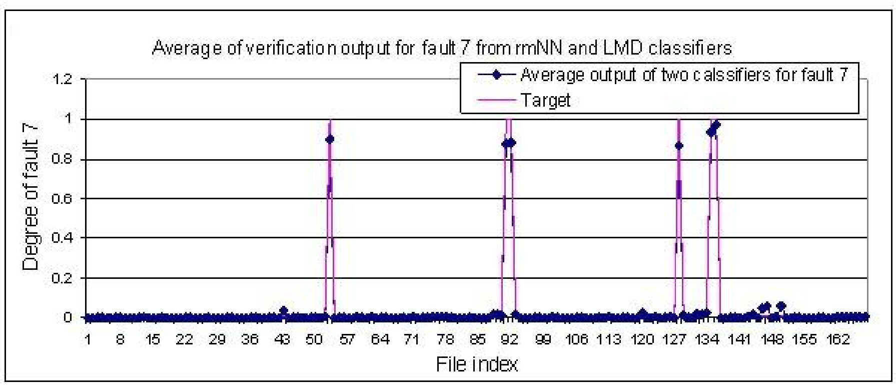
Figure 109: The learning output for fault 7 after the fusion of the LMD and rmNN classifiers.



(a) The verification output for fault 7 from the LMD classifier.



(b) The verification output for fault 7 from the rmNN classifier.



(c) The average of two verification output for fault7 from the LMD and rmNN classifiers.

Figure 110: The verification output for fault 7 after the fusion of the LMD and rmNN classifiers.

The Learning and testing results for faults 3, 6, 7, 8, 9, 10 and 11 after the classifier fusion have been illustrated in Appendix H. The overall improvement of the PSFC performance will be discussed via Tables 41 and 42.

Table 41 lists the minimum learning output from the rmNN, LMD and fusioned classifier for all the true cases in the training set. It is apparent that, for faults 6, 8, 10 and 11, the minimums from rmNNs are lower than 0.8. After classifier fusion the minimums are all above 0.8. On the other hand, for fault 6, the minimum from LMD is 0.793, lower than 0.8. After classifier fusion, the minimum output is raised to 0.811.

Table 41: Minimum learning output from the rmNN, LMD and fusioned classifier.

Evaluation of classifier fusion performance (Training)						
	Minimum learning output of rmNN		Minimum learning output of LMD		Minimum learning output of fusioned classifier	
	value	file index	value	file index	value	file index
Fault 3	0.871	92	0.899	401	0.901	92
Fault 6	0.742	383	0.793	362	0.811	383
Fault 7	0.845	345	0.817	350	0.862	350
Fault 8	0.748	235	0.835	384	0.861	382
Fault 9	0.87	414	0.840	387	0.896	387
Fault 10	0.68	471	0.984	471	0.832	471
Fault 11	0.61	93	0.854	103	0.803	93

Table 42 lists the minimum verification output from the rmNN, LMD and fusioned classifier for all the true cases in the testing set. It is apparent that, for faults 7 and 10, the minimums from rmNNs are 0.792 and 0.656, both lower than 0.80. After classifier fusion, the minimums are all raised above 0.82. On the other hand, for fault 6, the minimum from LMD is 0.786, lower than 0.8. After classifier fusion, the minimum output is raised to 0.833.

Table 42: Minimum verification output from the rmNN, LMD and fusioned classifier.

Evaluation of classifier fusion performance (Verification)						
	Minimum verification output of rmNN		Minimum verification output of LMD		Minimum verification output of fusioned classifier	
	value	file index	value	file index	value	file index
Fault 3	0.819	53	0.894	139	0.863	53
Fault 6	0.839	93	0.786	90	0.833	90
Fault 7	0.792	128	0.869	91	0.867	128
Fault 8	0.852	56	0.838	137	0.875	137
Fault 9	0.868	145	0.920	138	0.932	145
Fault 10	0.656	159	0.984	159	0.820	159
Fault 11	0.808	60	0.926	60	0.867	60

The accuracy of the PSFC, which benefits from the fusion of the rmNN and LMD classifiers is listed in Table 43. The threshold for ‘true’ decision making is still 0.8.

Table 43: The accuracy of the PSFC system.

Fault type	# of files for verification	# of files incorrect	Accuracy
Minor AC disturbance	94	0	1
AC Disturbance	44	0	1
Valve Current Closed/Blocked/Deblocked	44	0	1
Line Fault	22	0	1
Commutation Failure	25	0	1
Pole Voltages/Currents Closed/Blocked/Deblocked	15	0	1
Current Arc Back	6	0	1
Parallel Operation	9	0	1
Pole Current Oscillation	7	0	1
Normal Affected by Another Pole	4	0	1
Asymmetric Protection	6	0	1
Disturbance on DC Voltage	7	0	1

It is obvious that the overall performance of the PSFC was improved via the fusion of the two classifiers, the rmNN and LMD. The two classifiers provide complementary information that gives the 2-classifier fusion method a chance to succeed. The accuracy of the PSFC is 100%, which provides confident information for fault decision making and enhances the quality of the power system protection functionality.

10 Conclusion

This thesis introduces a rough set approach to power system fault classification. A form of rough neural computing based on the use of rough membership functions is introduced in the design of what is known as a rough membership function neural network (rmNN). A rough membership function makes it possible to measure the degree that any specified object with given feature values belongs to a given set X. The set X in this application is a set of fault files, which represent the same type of fault. Each rmNN has 3 layers: input, hidden, and output. The input layer contains what are known as rmf neurons, *i.e.*, neurons that compute the degree overlap between a specific class containing objects representing a fault type and a set of sample objects representing fault signals to be classified. The neurons in the hidden layer aggregate the output from the rmf neurons. The hidden layer neurons are designed using fuzzy set theory, which is ideally suited for numerical representation of aggregated rmf neuron output. The output neuron of an rmNN estimates the degree of a specific type of fault.

The most significant contribution of this research is a demonstration that the rough membership function successfully distinguishes objects with similar feature values. This makes rmNN a reasonable choice as a power system fault classifier.

A C++ and Labview based graphic user interface is implemented for the rmNN classifier, which makes the power system fault classification easy to operate.

To further improve the performance of the proposed approach to power system fault classification, a 2-Classifier fusion method has been introduced. This fusion method takes into account both the results of the rmNN classifier as well as a linear mean and standard deviation (LMD) based classifier. The correlation of the rmNN and LMD classifiers was estimated and has proved to be low enough to ensure that these two classifiers provide complementary information and are good candidates for classifier fusion. The ‘average’ method was selected as a fusion function.

Future work will include an extension of the Transcan^{TM} system used by Manitoba Hydro. In addition, it is also possible to consider various forms of unsupervised, adaptive learning as a means of classifying power system faults.

References

[1] Alkoot, F. and Kittler, J. Experimental Evaluation of Expert Fusion Strategies. Pattern Recognition Letters, Vol. 20 (11) (1999): 1361-1369.

[2] Van Breukelen, M., Duin, R., Tax, D. and Den Hartog, J. Combining Classifiers for the Recognition of Handwritten Digits. Proceedings of First IAPR TC1 Workshop Statistical Techniques in Pattern Recognition (1997): 13-18.

[3] Brigham, E.O. The Fast Fourier Transform and Its Applications. Prentice-Hall, NJ, USA (1988).

[4] Burrus, C.S., Gopinath, R.A., and Guo, H. Introduction to Wavelets and Wavelet Transforms. Prentice Hall, New York, USA (1998).

[5] Cajori, F. A History of Mathematical Notations. The Open Court Publishing Co., Illinois, USA, Vol. 1-2 (1929).

[6] CEIDS (Consortium for Electric Infrastructure to Support a Digital Society) The Cost of Power Disturbances to Industrial and Digital Economy Companies - Executive Summary. June (2001).

[7] Chaari, O., Meunier, M. and Brouaye, F. Wavelets: A New tool for the Resonant Grounded Power Distribution system Relaying. IEEE Transactions on Power Delivery, Vol. 11 (3) (1996): 1301-1308.

[8] Chanda, D., Kishore, N.K. and Sinha, A.K. Application of Wavelet Multiresolution Analysis for Classification of Faults on Transmission lines. Poster Session, IEEE TENCON 2003: Proceedings of the IEEE Region 10 Conference on Convergent Technologies for Asia-Pacific Region, Bangalore, India, Vol. 4 (2003): 1464-1469.

[9] Chen, S., Cowan, C. and Grant, P. Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks. IEEE Transactions on Neural Networks, Vol. 2 (2) (1991): 302-309.

[10] Cheng, H. and Elangovan, S. A B-Spline Wavelet Based Fault Classification Scheme for High Speed Protection Relaying. Electric Machines and Power Systems, Vol. 28 (4) (2000): 313-324.

[11] Cohen, L. Time-Frequency Analysis. Prentice-Hall, NJ, USA (1994).

[12] Drisen, J., Van Craenenbroeck, T., Reekmans, R. and Van Dommelen, D. Analyzing Time-Varying Power System Harmonics Using Wavelet Transform. In proceedings of IEEE Instrumentation and Measurement Technology Conference, Brussels, Belgium, Vol.1 (1996): 474-479.

[13] Dugan, R.C., McGranaghan, M.F. and Beaty, H.W. Electrical Power Systems Quality. McGraw-Hill, NY, USA (1996).

[14] Eistenstein, G., Note. Crelle's Journal XXVII (1844): 91.

[15] Elman, J. Finding Structure in Time. Cognitive Science, Vol. 14 (1990): 179-211.

[16] Equivalence class notation:
`http://en.wikipedia.org/wiki/Equivalence_class`

[17] Fernández, R. and Rojas, H.N. An Overview of Wavelet Transforms Application in Power Systems. 14th Power Systems Computation Conference, Sevilla, Spain, Session 1, Paper 6, (2002): 1-8.

[18] Freeman, J.A. and Skapura, D.M. Neural Networks: Algorithms, Applications, and Programming Techniques. Addison-Wesely Publishing Company, MA, U.S.A. (1991).

[19] Gaouda, A.M., Salama, M.A., Sultan, M.R. and Chikhani, A.Y. Power Quality Detection and Classification Using Wavelet-Multiresolution Signal Decomposition. IEEE Transactions on Power Delivery, Vol. 14 (4) (1999): 1469-1476.

[20] Goebel, K., Yan, W. and Cheetham, W. A method to calculate classifier correlation for decision fusion. Proceedings IDC 2002, Adelaide, (2002): 135-140.

[21] Gole, A.M. Course 24.799 - HVDC Transmission Course Notes, Department of Electrical and Computer Engineering, University of Manitoba, Canada. (1995).

[22] Hagan, M., Demuth, H.B. and Beale, M. Neural Network Design. PWS Publishing Company, Boston, U.S.A. (1996).

[23] Han, L., Peters, J.F., Ramanna, S. and Suraj, Z. Towards a power system fault classification system: A rough neurocomputing approach. In: Z. Suraj (Ed.). Sixth International Conference on Soft Computing and Distributed Processing. Rzeszow, Poland. (2002): 89-92.

[24] Han, L. and Peters, J.F. Heuristic method in searching for global minima in high-voltage AC error. In: Kinsner, W., Sebak, A., Ferens, K. (Eds.). Canadian Conference on Electrical and Computer Engineering (CCECE'02). Winnipeg, Manitoba, Canada. (2002): 776-781.

[25] Haselsteiner, E. What Elman Networks Can not Do. IEEE World Congress on Computational Intelligence, the 1998 IEEE International joint Conference, Vol. 2 (1998): 1245-1249.

[26] Imriš, P. and Lehtonen, M. Transient Based Earth Fault Location In 110KV Subtransmission Networks. 15th Power Systems Computation Conference. Liége, Belgium, Vol. 23 (2005): 1-4.

[27] Kaiser, G. A Friendly Guide to Wavelets. Birkhäuser, Boston (1994).

[28] Kashyap, K.H. and Shenoy, U.J. Classification of Power System Faults using Wavelet Transforms and Probabilistic Neural Networks. Proceedings of the 2003 International Symposium on Circuits and Systems. Vol. 3 (2003): 423-426.

[29] Kawady, T.A.S. Fault Location Estimation in Power Systems with Universal Intelligent Tuning. Dissertation for Doctor of Philosophy. Elektrotechnik und Informationstechnik. Der Technischen Universität Darmstadt (2005).

[30] Kittler, J., Hatef, M., Duin, R. and Matas, J. On Combining Classifiers. IEEE Transaction on Pattern Analysis and Machine Intelligence. Vol. 20 (3) (1998): 226-239.

[31] Komorowski, J., Pawlak, Z., Polkowskis, L. and Skowron, A. Rough Fuzzy Hybridization - A New Trend in Decision Making. Pal, S.K. and Skowron, A. (Eds.). Springer Verlag Publisher, Singapore (1999).

[32] Komorowski, J., Pawlak, Z., Polkowski, L. and Skowron, A. Rough sets: A tutorial. In: [31] (1999): 3-98.

[33] Konrad, E., Orłowska, E. and Pawlak, Z. Knowledge Representation Systems. Institute for Computer Science, Polish Academy of Sciences. Report 433 (1981).

[34] Kuncheva, L.I. A theoretical Study on six Classifier Fusion Strategies. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 24 (2), (2002): 281-286

[35] Kuncheva, L., Bezdek, J. and Duin, R. Decision Templates for Multiple Classifier Fusion: An Experimental Comparison. Pattern Recognition. Vol. 34 (2) (2001): 299-314.

[36] Lee, C.H., Wang, Y.J. and Huang, W.L. A Literature Survey of Wavelets in Power Engineering Applications. Proceedings National Science Council. ROC(A) Vol. 24 (4) (2000): 249-258.

[37] Liang, J., Elangovan, S. and Devotta, J.B.X. A wavelet multiresolution analysis approach to fault detection and classification in transmission lines. International Journal of Electrical Power and Energy Systems. Vol. 20 (5) (1998): 327-332.

[38] Lu, Y., Sundararjan, N. and Saratchandran, P. Performance Evaluation of a Sequentional Minimal Radial Basis Function (RBF) Neural Network Learning Algorithm. IEEE Transactions on Neural Networks Vol. 9 (2) (1998): 308-318.

[39] Martin, F. and Aguado, J.A. Wavelet-based ANN approach for transmission line protection. IEEE Transactions on Power Delivery. Vol. 18 (4) (2003): 1572-1574.

[40] Masters, T. Practical neural network recipes in C++. Academic press, Inc., FL (1995).

[41] Orłowska, E. Semantics of Vague Concepts, Applications of Rough Sets, Institute for Computer Science, Polish Academy of Sciences. Report 469, (1982).

[42] Pal, S.K., Peters, J.F., Polkowski, L. and Skowron, A. Rough-neural computing: An introduction. In: [43] (2004): 15–42.

[43] Pal, S.K., Polkowski, L. and Skowron, A. (Eds.). Rough-Neural Computing. Techniques for Computing with Words. Springer-Verlag, Berlin, Germany (2004).

[44] Pal, S.K. and Skowron, A. (Eds.). Rough-Fuzzy Hybridization: A New Trend in Decision-Making. Springer-Verlag, Singapore Pte. Ltd. (1999).

[45] Pal, S.K., Polkowski, L., Peters, J. and Skowron, A. Rough neurocomputing: An Introduction. Pal, S.K., Polkowski, L. and Skowron, A. (Eds.). Rough-Neuro Computing. Berlin, Springer (2003): 16-43.

[46] Papoulis, A. The Fourier Integral and Its Applications. McGraw-Hill, NY (1962): 241-242.

[47] Pawlak, Z. and Skowron, A. Rudiments of rough sets. Information Sciences, vol. 177 (2007): 3-27.

[48] Pawlak, Z. and Skowron, A. Rough sets: Some extensions. Information Sciences, vol. 177 (2007): 28-40.

[49] Pawlak, Z. and Skowron, A. Rough sets and Boolean reasoning. Information Sciences, vol. 177 (2007): 41-73.

[50] Pawlak Z. Classification of Objects by Means of Attributes. Institute for Computer Science, Polish Academy of Sciences. Report 429 (1981).

[51] Pawlak Z. Rough Sets, Institute for Computer Science, Polish Academy of Sciences. Report 431 (1981).

[52] Pawlak, Z. Rough sets, International J. Comp. Inform. Science. Vol. (11) (1982): 341–356.

[53] Pawlak, Z. and Skowron, A. Rough membership functions. Yager, R. et al. (Eds.). Advances in Dempster Shafer Theory of Evidence. Wiley, N.Y., U.S.A. (1994): 251–271.

[54] Perunicic, B., Mallini, M., Wang, Z., Liu, Y. and Heydt, G.T. Power Quality Disturbance Detection and Classification Using Wavelets and Artificial Neural Networks. The 8th International Conference on Harmonics and Quality of Power (1998): 77-82.

[55] Peters, J.F., Han, L. and Ramanna, S. Rough neural computing in signal analysis. Computational Intelligence 1 (3) (2001): 493-513.

[56] Peters, J.F. and Skowron A. Rough Sets: An Introduction. Springer, Berlin, in press.

[57] Peters J.F., Henry C. and Gunderson D.S. Biologically-inspired approximate adaptive learning control strategies: A rough set approach. International Journal of Hybrid Intelligent Systems (2007), in press.

[58] Peters, J.F., Skowron, A., Han, L. and Ramanna, S. Towards rough neural computing based on rough membership functions: Theory and Application. Ziarko, W., Yao, Y. (Eds.). Rough Sets and Current Trends in Computing. Lecture Notes in Artificial Intelligence 2005. Berlin, Springer-Verlag (2005): 611-618

[59] Peters, J.F., Ramanna, S., Suraj and Z., Borkowski, M. Rough neurons: Petri net models and applications. Pal, S.K., Polkowski, L., Skowron, A. (Eds.). Rough-Neuro Computing. Berlin, Springer (2003): 471-489.

[60] Peters, J.F., Skowron, A., Suraj, Z. and Han, S. Design of rough neurons: Rough set foundation and Petri net model. Ras, Z.W., Ohsuga, S. (Eds.). Lecture Notes in Artificial Intelligence. Berlin, Springer-Verlag (2000): 283-291

[61] Petrakos, M., Kannelopoulos, I., Benediktsson, J. and Pesaresi, M. The effect of correlation on the accuracy of the combined classifier in decision level fusion. Proceedings IEEE 2000 International. Geo-science and Remote Sensing Symposium. Vol. 6 (2000): 2623-2625.

[62] Polkowski, L. Advances in Soft Computing: Rough Sets Mathematical Foundation. Phyica-Verlag Publisher, A Springer Verlag Company, Berlin (2002).

[63] Ribeiro, P.F. Wavelet transform: An advanced tool for analysing non-stationary harmonic distortion in power system. Proceedings of IEEE ICHPS, VI. Bologna, Italy (1994): 925-293.

[64] Robertson, D.C., Camps, O. and Mayer, J. Wavelets and power system transients. SPIE international symposium on optical engineering in aerospace sensing. (1994): 474-487.

[65] Robertson, D.C., Camps, O.I., Mayer, J.S. and Gish, W. Wavelets and electromagnetic power system transients. IEEE Transactions on Power Delivery. Vol. 11 (2) (1996): 1050-1058.

[66] Ramaswamy, S., Kiran, B.V., Kashyap, K.H. and Shenoy, U.J. Classification of Power System Transients using Wavelet Transforms and Probabilistic Neural Networks. IEEE Tencon 2003: Proceedings of the IEEE Region 10 Conference on Convergent Technologies for the Asia-Pacific. Taj Residency, Bangalore, India. Vol. 4, (2003): 1272-1276.

[67] Rough Set Database System (RSDS):
`http://rsds.wsiz.rzeszow.pl/rsds.php`

[68] Tax, D., Duin, R. and Van Breukelen, M. Comparison between Product and Mean Classifier Combination Rules. Proceedings Workshop Statistical Pattern Recognition (1997): 165-170.

[69] Tumer, K. and Ghosh, J. Linear and Order Statistics Combiners for Pattern Classification. Combining Artificial Neural Nets. Sharkey, A. (Ed.). (1999): 127-161.

[70] Santoso, S., Powers, E. and Hofmann, P. Power quality assessment via wavelet transform analysis. IEEE Transactions on Power Delivery. Vol. 11 (2) (1996): 924-930.

[71] Santoso, S., Powers, E.J., Grady, W.M. and Parsons, A.C. Power Quality Disturbance Waveform Recognition Using Wavelet-Based Neural Classifier. I. Theoretical Foundation. IEEE Transactions on Power Delivery. Vol. 15, (1) (2000): 222-228.

[72] Transcan^{TM} Recording System - Operation and Software. Mehta Tech. Inc. (1994).

[73] Valens, C. A Really Friendly Guide to Wavelets. ©1999-2004
`http://perso.orange.fr/polyvalens/clemens/wavelets/wavelets.html`

[74] Wang, M. and Mamishev, A.V. Classification of Power Quality Events Using Optimal Time-Frequency Representations – Part 1: Theory. IEEE Transactions on Power Delivery. Vol. (19) (3) (2004): 1488-1495.

[75] Wang, W., Rowe, G.I. and Mamishev, A.V. Classification of Power Quality Events Using Optimal Time-Frequency Representations – Part 2: Application. IEEE Transactions on Power Delivery. Vol (19) (3) (2004): 1496-1503.

[76] Woodford, D.A. HVDC Transmission. Technical Report, Manitoba HVDC Research Centre. (1998).

[77] Zhao, W., Song, Y.H. and Min, Y. Wavelet analysis based scheme for fault detection and classification in underground power cable systems. Electric Power Systems Research. Vol. (53) (2000): 23-30.

[78] Zwillinger, D. (Ed.) CRC Standard Mathematical Tables and Formula. Boca Raton, FL: CRC Press (1995): 223.