ANALOG CIRCUIT FAULT CLASSIFICATION BASED ON MACHINE LEARNING ALGORITHM

PROJECT REPORT

PHASE II

Submitted by

MANJUPARKAVI R

Register No: 15MAE006

in partial fulfilment for the requirement of award of the degree

of

MASTER OF ENGINEERING

in

APPLIED ELECTRONICS

Department of Electronics and Communication Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Chennai)

COIMBATORE – 641 049

ANNA UNIVERSITY, CHENNAI 600 025

APRIL 2017

BONAFIDE CERTIFICATE

Certified that this report titled **"ANALOG CIRCUIT FAULT CLASSIFICATION BASED ON MACHINE LEARNING ALGORITHM"** is the bonafide work of **MANJUPARKAVI R [Reg. No. 15MAE006]** who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form part of any another project or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other student.

SIGNATURE

Ms. M.SHANTHI

PROJECT SUPERVISOR

DEPARTMENT OF ECE

Kumaraguru College of Technology

Coimbatore – 641 049

SIGNATURE

Prof. K.RAMPRAKASH

HEAD OF PG PROGRAMMES

DEPARTMENT OF ECE

Kumaraguru College of Technology

Coimbatore – 641 049

The Candidate with **Register No. 15MAE006** is examined by us in the Project viva-voice examination held on ………………

…………………………..
INTERNAL EXAMINER

…………………………
EXTERNAL EXAMINER

ACKNOWLEDGEMENT

First, I would like to express my praise and gratitude to the Lord, who has showered his grace and blessings enabling me to complete this project in an excellent manner.

I express my sincere thanks to the management of Kumaraguru College of Technology and Joint Correspondent **Shri Shankar Vanavarayar** for his kind support and for providing necessary facilities to carry out the work.

I would like to express my sincere thanks to our beloved Principal **Dr.R.S.Kumar Ph.D.,** Kumaraguru College of Technology, who encouraged us with his valuable thoughts.

I would like to thank **Prof. K.Ramprakash M.E.,** Head of PG Programmes, Electronics and Communication Engineering, for his continuous encouragement and motivation throughout the course of this project work.

In particular, I wish to thank with everlasting gratitude to the project coordinator **Ms.S.Nagarathinam M.E., (Ph.D).,** Assistant Professor-III **,**Department of Electronics and Communication Engineering, for her expert counselling and guidance to make this project to a great deal of success.

I extend my heartfelt thanks to my project guide **Ms.M.Shanthi M.S., (Ph.D).,** Associate Professor, for her immense contribution, guidance, support and constructive criticism not only during this project but also during this two years of master program.

Lastly, I would like to thank my friends and family for standing by me and encouraging me throughout the project.

ABSTRACT

The role of System-on-Chip (SOC) technology has dramatically boosted the importance of analog circuitry, moving it more into mainstream integrated circuit(IC) design. The improvements in IC technology and co-existence of analog and digital signals make testing a challenging task. There are numerous testing tools available for fault diagnosis in digital circuits but only a certain number of fault diagnosis techniques are available for analog circuits. The fault diagnosis of analog circuits are generally classified into Simulation After Test (SAT) and Simulation Before Test (SBT). SBT is suitable for recent day research work and is the most preferred one. Fault in analog circuits are classified into hard faults and soft faults. Soft faults cause performance degradation so diagnosis of such faults based on simulation method is focused in this project.

A soft fault diagnostic system based on Multilayer Extreme Learning Machine (MLELM) and Kernel Extreme Learning Machine (KELM) with Particle Swarm Optimization (PSO) is proposed in this project. State Variable Filter (SVF) and Sallenkey Band Pass Filter (SKBPF) are chosen as the benchmark circuits. Inability to classify faults for multilayer by Extreme Learning Machine (ELM), we use the MLELM approach to classify the faults. MLELM is a Multilayer feed forward neural network in which each hidden layer is constructed by Extreme Learning Machine-Autoencoder (ELM-AE). ELM-AE is an unsupervised neural network which reproduces the input signal as much as possible. The fault dictionary constructed from the features of the Circuit Under Test (CUT) is used for fault detection and classification. The parameters obtained from the fault dictionary are normalized in the range of -1 to 1. Then MLELM algorithm is applied on the normalized values for the fault classification. Results shows that MLELM algorithm has faster responsiveness, better scalability and much faster learning speed.

KELM is an infinite single-hidden layer feedforward neural network (SLFNs). KELM improves the stability and performance by using kernel matrix instead of computing the hidden layer matrix. Kernel matrix is a low-rank decomposition matrix and it improves the generalization performance. In order to improve the classification accuracy, kernel parameter of kernel function need to be optimized, it is done by using PSO approach. PSO is a population based powerful optimization technique developed by Eberhart and Kennedy and it is inspired by the social behaviour of bird flocking. KELM with PSO provides higher classification accuracy and generalization performance than ELM and MLELM algorithm by minimizing the training error and output weight. The results of all the three algorithms are compared and the results prove that KELM with PSO algorithm outperforms other two algorithms in terms of generalization performance and classification accuracy.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVATIONS	NOMENCLATURE
SOC	System-on-chip
CMOS	Complementary Metal Oxide Semiconductor
IC	Integrated circuits
SAT	Simulation After Test
MLELM	Multilayer Extreme Learning Machine
CUT	Circuit Under Test
ATE	Automatic Test Equipments
ICT	In- Circuit Tester
FT	Functional Tester
ELM	Extreme Learning Machine
SVM	Support Vector Machine
LS-SVM	Least Square Support Vector Machine
PSVM	Proximal Support Vector Machine
PSO	Particle Swarm Optimization
SVF	State Variable Filter
SKBPF	Sallenkey Bandpass Filter
ELM-AE	Extreme Learning Machine -Auto Encoder
KELM	Kernel Extreme Learning Machine

DELM	Deep Extreme Learning Machine
SLFN	Single Layer Feedforward Neural Network
AI	Artificial Intelligence
BPNN	Backward Propagation Neural Network
EEG	Electroencephalography
BCI	Brain Computer Interface
SAE	Stacked Auto Encoder
LGT	Langatate
ISOP	Irredundant Sum of Products
LDA	Latent Dirichilet Allocation
DBN	Deep Belief Networks
PCA	Principal Component Analysis
DMOZ	Deutsche Medizinische Online Testing
TF-IDF	Term Frequency Inverse Document Frequency
SBT	Simulation Before Testing
NB	Naïve Bayes
GA	Genetic Algorith
LSWSVM	Least Squares Wavelet Support Vector Machine
HELM	Hierarchical Extreme Learning Machine
PLS	Partial Least Square
WTGS	Wind Turbine Generator System

PCA	Principal Component Analysis
RP	Random Projection
NMP	Non-negative Matrix Factorization
SAE	Stacked Auto Encoder
DBN	Deep Belief Network
MH	Multi Hypothesis
RKELM	Reduced Kernel Extreme Learning Machine
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
FPR	False Positive Rate
FNR	False Negative Rate
PPV	Positive Predictive Value
NPV	Negative Predictive Value

CHAPTER 1
INTRODUCTION

System–on-chip (SOC) has improved the importance of analog circuits. Analog and digital circuits are now being integrated into a SOC. Because of the smaller size of IC'S, it increases the functional complexity. The improvements in IC technology and co-existence of analog and digital signals make testing a challenging task. Therefore electronic tests are system dependent and there are different fault diagnosis methods based on the signal nature. Testing of analog circuit is not fully automated as compared to digital testing and the cost of analog testing is very high. There are very limited numbers of testing tools available for analog and mixed signal circuits. So analog circuits demands substantial research and needs improved development in the area of fault diagnosis. There are two methods available for performing testing in analog circuits, they are Specifications based testing and functional testing. The specification based testing is performed mainly to check whether the circuit or design has met the specifications, the functional testing is performed to check the functionality of the circuit within the standard input.

1.1 FAULT MODELS IN ANALOG AND MIXED SIGNAL SYSTEMS

Test can be performed at several levels of IC fabrication like wafer level, package level, module level and system level. Testing of circuits means the identification of faults in the circuit. Faults in the analog integrated circuits may occur due to defects in the manufacturing process which leads to failures. Faults may also occur due to defective components, short circuit in signal lines, breaks in signal lines, lines shortened to power supply or ground, excessive delays, etc. There are three types of faults. They are temporary faults, delay faults and permanent faults

1.1.1 Temporary Faults

The temporary faults are those faults which are transient and exist only for a short duration of time.

1.1.2 Delay Faults

The faults which have impact on the operating speed of the circuit are called delay faults.

1.1.3 Permanent Faults

Permanent faults are those type of faults which are present in the circuit long enough to be observed during the test time. There are two types of permanent faults, they are hard faults and soft faults.

1.1.3.1 Hard Faults

Hard or catastrophic faults are the changes in the circuit that cause the circuit to fail catastrophically. These faults include shorts, opens or large variations in design parameters. These faults are caused by major structural deformations or extreme out-of-range parameters and lead to malfunctioning of the circuit. Catastrophic faults are further classified in to stuck-open and stuck-short faults.

1.1.3.1a Stuck-Open Faults

The stuck open fault is the fault in which the component terminals are out of contact with the rest of the circuit which creates a high resistance at the incident of the fault in the circuit. Open faults can be simulated by adding a high resistance in series (Rs =100 MΩ) with the component to be faulted.

1.1.3.1b Stuck-Short Faults

The stuck short fault is the short between the terminals of the component. It is essentially shorting out the component from the circuit. Short faults can be simulated by adding a small resistance in parallel (Rp =1Ω) with the component. The stuck-open and stuck-short faults can be simulated in a resistor, capacitor, MOSFET. The figure 1.1 shows the stuck-open and stuck short faults for resistor and capacitor.

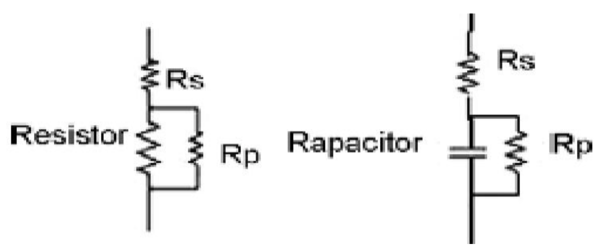

Figure 1.1 Stuck-open and Stuck-short fault models in resistor and Capacitor

The figure 1.2 shows stuck open and short fault models for MOSFET device. The stuck open fault in MOSFET can be modeled by connecting high resistance in series either to the drain or source of the component.

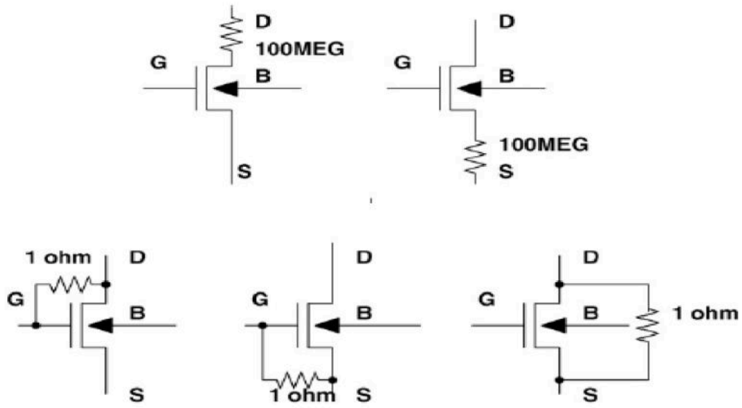

Figure 1.2 Stuck-open and Stuck-short fault models for MOSFET

1.1.3.2 Soft Faults

Soft or Parametric faults are the statistical variations in the manufacturing process conditions that cause performance degradation of the circuit. These faults occur mainly because of ageing, manufacturing tolerances or parasitic effects. These faults involve parameters deviations from their nominal value which exceeds from their tolerance band. These faults result from local and global defects.

1.2 FAULT DETECTION

The current approach to detect manufacturing faults in electronic circuit uses several forms of Automatic Test Equipments (ATE), In- Circuit Tester (ICT) and Functional Tester (FT). ICT require physical access to notes or points on the circuit in order to perform the necessary testing. Analog fault diagnosis approaches are generally classified into two types. They are

1. Simulation After Test (SAT)
2. Simulation Before Test (SBT)

1.2.1 Simulation After Test

SAT approach is very efficient for soft faults diagnosis because they are based on linear network models. In SAT approach simulation is performed to identify the network parameters and it is carried out at the time of testing. The component values are used for fault detection and these values are measured from the voltage and current measurements. The components are identified as fault components if the range exceeds the tolerance limit. SAT method is also called as topological method because it uses circuit topology for fault identification.

1.2.1.1 Simulation After Test Methods

Parameter identification technique works on the basis that it identifies all the network parameters from the available independent variables. Parameter identification technique is classified into two types based on the nature of diagnosis equations. They are linear and nonlinear techniques. The major problem in parameter identification is the ability to access test points. There are not enough test points to test all components are each added test points is too expensive to accept.

All the parameters cannot be identified if the measurements are limited. Fault verification techniques assume that only limited number of parameters is faulty and rest of the parameters are fault free. In this technique the whole circuit is partitioned into two groups called group 1 and group 2. Among the two groups group 1 consists of fault free components (nominal components) and group 2 consists of faulty components. The measurements and characteristics of group 1 are used to calculate the input and output from group 2. If the parameters of both the group are similar then the parameters from the group 2 are shifted to group 1 and this process is repeated until satisfactory verification is achieved.

Optimization technique is used to find most likely fault elements. L2 approximation technique, Quadratic approximation technique and L1 are most widely used optimization techniques for fault classification. The elements are said to be faulty if the changes from nominal values are large.

1.2.2 Simulation Before Test

SBT methods are based on building a fault dictionary in which the nominal circuit behaviours in DC, frequency or time domain are stored. The fault dictionary also consists of the responses of the circuit for various anticipated faults. There are two important SBT methods used for fault diagnosis.

1.2.2.1 Simulation Before Test Methods

Fault dictionary technique consists of fault free and anticipated faulty cases of a circuit under test. The anticipated faulty cases are based on the field experience gained by the engineer. Fault simulation plays an important role in the construction of fault dictionary. The efficiency and effectiveness of the technique depends on many factors. The main factors are proper choice of stimulus, selection of test measurement optimization and fault isolation. The statistical approach is based on constructing the statistical database or fault dictionary by performing large number of simulations to characterize the network statistically. The statistical database helps in obtaining the probability error in each and every component of the circuit. The component with highest probability is considered as faulty component. Diagnosis of soft faults using the SBT fault diagnosis approach for two filter circuits is carried out in this project using proposed machine learning algorithm.

1.3 MACHINE LEARNING

Machine learning is the subfield of computer science that allows computers, the capacity to learn without being programmed explicitly. Machine learning is a type of artificial intelligence (AI) used in the field computer science, probability theory, and optimization theory which allows complex tasks to be solved for which a logical/procedural approach would not be possible or feasible. Machine learning algorithms are categorized as supervised or unsupervised.

1.3.1 Supervised Learning

Supervised learning is a type of machine learning task which infers a function with labeled training data. This algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. The figure 1.3 shows the typical supervised learning model. The main aim of supervised learning algorithm is to build a model that makes predictions based on the learning. From the figure, the known set of inputs (Text, image or any other data) and their responses are given to the algorithm. The algorithm trains the model to generate reasonable predictions for the response of new data.

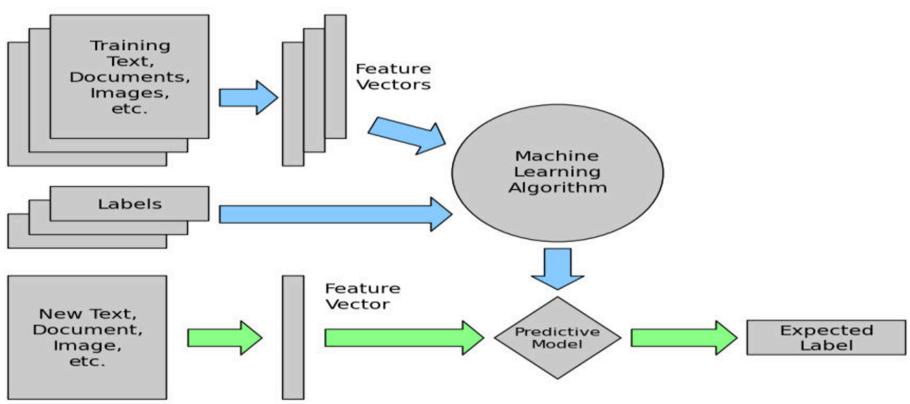

Figure 1.3 Supervised Learning Model

1.3.2 Unsupervised Learning

Unsupervised learning draws inferences from the datasets consisting of input data without labeled responses. Since the examples given to the learner are unlabeled, there is no error or reward signal to evaluate the potential solution. The figure 1.4 shows the unsupervised learning model. The inputs (Text, image, etc) are given as input to the model without any label. The inputs are grouped in to several groups based on some criteria or some learning model and the algorithm adapts to the data and trains if any new input is given to the algorithm based on the statistical properties.

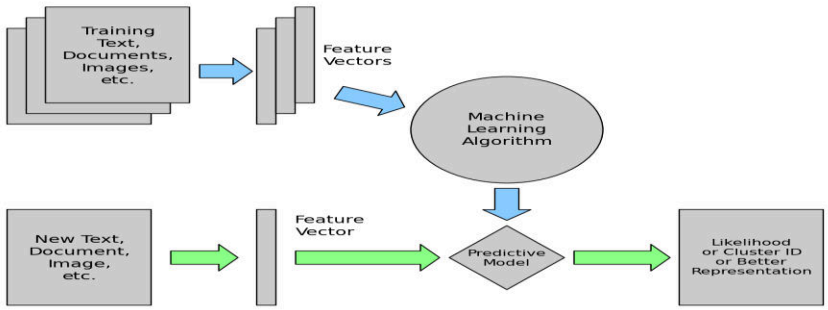

Figure 1.4 Unsupervised Learning Model

There are many machine learning algorithms they are Naïve Bayes Classification, Decision tree learning, Association rule learning, Artificial neural network or neural network , Support vector machines, Clustering , Sparse dictionary learning. Artificial neural networks are computational models inspired by biological neural networks are used to approximate functions that are generally unknown.

- Contribution of this project:

  A special type of multilayer layer feed forward neural network called Multilayer Extreme learning machines (MLELM) and Kernel ELM with PSO have been used in wide variety of fields, evidently for the first time, this algorithm is proposed for analog circuit fault classification in this project work.

1.4 OUTLINE OF THE REPORT

The third chapter deals with the fault diagnostic framework, the benchmark circuits, their transfer function and the data sets generated from the benchmark circuits, followed by performance metrics calculation from the confusion matrix. Chapter 4 presents the proposed methodology MLELM and ELM-AE with the simulation results. Chapter 5 deals with the KELM algorithm and introduction of optimization technique called PSO, this chapter also contains the result of proposed method KELM with PSO and it also includes the comparison results of all the proposed methodologies. Finally the chapter 6 concludes the project work with the performance analysis of all the proposed algorithms.

CHAPTER 2

LITERATURE REVIEW

2.1 NEURAL NETWORKS BASED PARAMETRIC TESTING OF ANALOG IC

V.Stopjakova proposed a new parametric approach in analog integrated circuits for detecting the faults [1]. For detecting the faults, the feedforward neural network which is trained by back-propagation method is proposed. The classification is done by sensing differences observed in dynamic supply current of faulty and faulty free circuits. Two stage CMOS operational amplifier is taken as the CUT for this method and it is simulated in HSPICE circuit simulator. The identification is performed in both frequency and time domain. It was shown that neural networks are very efficient and flexible approach for fault identification of analog circuits. The proposed method provides an eminent classification of about 97% even for the faulty input waveforms.

2.2 FAULT DIAGNOSIS OF ANALOG CIRCUITS WITH TOLERANCES USING ARTIFICIAL NEURAL NETWORKS

A simple method for diagnosing analog faults using artificial neural networks is proposed by Ying Deng. The k-fault diagnosis method and artificial backward propagation neural network (BPNN) are the two proposed approach in this paper [2]. The k-fault diagnosis method belongs to the category of SAT. This method locates the fault without tolerance in circuits. But for the circuit with tolerance, this method is very slow. So for improving the on-line characteristic and to achieve robustness, k-fault diagnosis method with BPNN is proposed in this paper. Simulation of the proposed method is carried in the resistive circuit with 8 resistors and each resistor has a tolerance of ±5 %. Simulation result shows that proposed method is fast and robust for analog circuit fault diagnosis with tolerances.

2.3 ANALOG TESTING WITH TIME RESPONSE PARAMETERS

Ashok Balivada proposed a simple test generation algorithm which derives sinusoidal test waveform for detecting fault classes [3]. The amplitude and phase errors are obtained from the steady state time response waveform which helps in the classification of large number of faults. For example, proposed technique is applied to Biquad filter. Parameters like delay, rise-time and overshoot are the criteria for faulty behaviour and this faulty behaviour is detected using time saturated ramp waveforms as tests and the use of associated ramp response. All these parameters are computed using simple algorithms from closed form expressions of the sinusoidal and ramp response.

2.4 FAULT DETECTION AND CLASSIFICATION IN LINEAR INTEGRATED CIRCUITS: AN APPLICATION OF DISCRIMINATION ANALYSIS AND HYPOTHESIS TESTING

To detect and classify faults in linear IC, standard multivariate technique of hypothesis testing and discrimination analysis is proposed in this paper [4]. These methods are very useful for tracking IC failures during method of preprocessing or assessing the failure in IC. Fault simulation is performed by using Monte Carlo Simulation. One-stage Amplifier, Differential Pair Circuit, comparator, Elliptical Filter, Analog Multiplexer are used as the circuit under test in this paper. Results indicate that statistical methods are investigated potentially produce low detection and classification error rates. There are three limitations are described for this technique. First, parametric drift will result more fault classes. Second, more complex IC circuits will cause high probability of multiple simultaneous faults. Third one is reliance on realistic fault, misclassification costs and fault models

2.5 TEST GENERATION ALGORITHM FOR ANALOG SYSTEMS BASED ON SUPPORT VECTOR MACHINE

Ting Long, Houjun Wang and Bing Long (2010) proposed a test generation algorithm for CUT based on Support Vector Machine (SVM) [5].Two-pole, three-pole and five-pole active filters are used as the CUT. In digital domain, test pattern and response are determined. Then the classification of the response space is done by using SVM. It also proposes an algorithm for calculating test sequence for input stimuli using the SVM results. Precision of test generation is improved by using numerical experiments. SVM method can be used for classification for the problems like mixed response spaces and non-linear classification problems. The advantage of SVM test generation method is that the output responses of the CUT can be used directly for classification and fault detection.

2.6 TEST GENERATION FOR LINEAR TIME INVARIANT ANALOG CIRCUITS

Chen-Yang Pan and Kwang-Ting (1999) Cheng proposed a novel and cost effective testing technique for parametric faults which generates small number of test patterns in multidimensional space using hyperplanes [6]. Three-pole, five-pole and BICMOS operational amplifier are used as the circuit under test in this approach. The major goal of this approach is to find test sets to achieve desired level of correct classification with minimal test application time and this objective is achieved by successive application of each test set. The residual response of previous test might affect the output response of the current test and may cause measurement errors. The next test measurement cannot start unless residual response becomes negligible. This observation implies that the overall test application is reduced which limits the speed of the approach. Therefore, it generalizes the arbitrarily linear independent vectors can be used as the test sequence. This approach results less than 10% misclassification using several test sets from hyperplanes or sampled points on a sinusoidal signal and each consists of a small number of test patterns.

2.7 NEURAL-NETWORK BASED ANALOG-CIRCUIT FAULT DIAGNOSIS USING WAVELET TRANSFORM AS PREPROCESSOR

For analog circuits, fault diagnosis based on backpropagation neural networks is proposed by mehran Aminian [7]. For preprocessing, wavelet decomposition, principal component analysis and data normalization methods are used. A 25 kHz sallenkey bandpass filter and two stage four-op-amp Biquad filter are used as the circuit under test in this study.

At first, wavelet decomposition is used to preprocess the impulse response and it remarkably reduces the number of inputs and minimizing its processing and training time. The second preprocessing is carried out by using principal component analysis and it further reduces the dimensionality of the input space and minimizes diagnostic errors. Finally, data normalization is used for enhancing the dataset features and it avoids the dynamic variance in one or more dimensions in input space. Compared with neuromorphic analyzers, backpropagation neural networks, expose that our system needs a much smaller network and it function remarkably better in analog circuits fault diagnosis.

2.8 APPLYING WAVELET SUPPORT VECTOR MACHINE TO ANALOG CIRCUIT FAULT DIAGNOSIS

For analog circuit diagnosis, genetic algorithm (GA) and least squares wavelet support vector machine (LSWSVM) using vector wavelet kernel function is proposed in this paper by Zuo lei [8]. Wavelet package is used as a tool for extracting the features, the GA-LSWSVM is then applied to the filter circuit after trained by GA. The filter circuit used here is biquadratic filter. PSPICE software is used to simulate failure state of the circuit. The simulation results show that the method can increase the accuracy and generalization ability. GA-LSWSVM has the accuracy of diagnosis over 98%, LSWSVM method has the accuracy about 96%, and using RBF kernel function of the LSSVM it achieves accuracy of about 93%. By using genetic algorithm for LSWSVM, it not only improves the classification ability but also reduce the complexity of the method.

2.9 EXTREME LEARNING MACHINE FOR MULTILAYER PERCEPTRON

Extreme Learning Machine for Multilayer Perceptron proposed by Jiexiong Tang provides new ELM-based hierarchical learning framework to overcome shallow architecture and feature learning while processing natural signals [9]. Original ELM is outperformed by the proposed framework in various simulations and it achieves high level representation using layer wise encoding methods. Original single-layer ELM is outperformed by hierarchical ELM (HELM) in classification and learning accuracy. Comparing with other MLP training methods HELM leads with faster training speed and better performance. Moreover, compared with other MLP training methods, the training of HELM is much faster and achieves higher learning accuracy. We also verified the generality and capability of HELM for practical computer vision applications. In theses applications, HELM functions as a feature extractor and classifier, and it achieves more robust and better performance than relevant state-of-the-art methods.

2.10 EXTREME LEARNING MACHINE: A NEW LEARNING SCHEME OF FEEDFORWARD NEURAL NETWORKS

A simple and efficient learning algorithm called ELM is proposed in this paper [10]. To overcome the issues like local minima, overfitting and improper learning rate, some methods such as weight decay and early stopping methods are used in these classical learning algorithms. Unlike the traditional learning algorithms which only work for differentiable activation functions but ELM algorithm is used to train SLFNs with many non-differentiable activation functions.

The performance comparison of the ELM algorithm and with many rpopular algorithms is conducted for a real medical diagnosis problem: Diabetes, using the “Pima Indians Diabetes Database” produced in the Johns Hopkins University, 1988. The experimental results show that the ELM may learn faster than SVM by a factor up to thousands. In our simulations especially for forest type prediction application, the response speed of trained SVM is very slow. So it is not easy for SVMs to make prediction in real-time application while the ELM appears to be suitable in an application which has fast prediction and response capability.

2.11 A FAULT DIAGNOSIS METHOD BY USING EXTREME LEARNING MACHINE

An improved algorithm of T-PLS by using extreme learning machine is proposed in this paper [11]. In industry, product quantities of sensors are difficult to measure. The relationship between the process variables and quality variables are used to predict the product quality information and it is used for fault diagnosis like partial least squares (PLS), latent structures algorithm and so on. Experiment results proves that ELM based quality prediction performance is superior than PLS, it can better reflect the internal relations between X and Y. So it is used for the decomposition process of projection space. There is no need for parameters of ELM to be adjusted iteratively.

2.12 EXTREME LEARNING MACHINE:THEORY AND APPLICATIONS

This paper proposes a new learning algorithm called ELM which overcomes the drawbacks of feed forward neural network [12]. The main drawback of the feed forward neural is slow gradient-based learning algorithms are used to train the network and the parameters are tuned using iteratively. This algorithm provides good generalization performance at extremely better learning speed. The experimental outcome based on a several artificial and real benchmark function approximation and classification problems, including very large complex applications show that the new algorithm can provide good generalization performance in several cases and learning can be thousands of times faster than conventional popular learning algorithms for feed forward neural networks. The traditional classic gradient-based learning algorithms may face several issues like local minima, improper learning rate and over fitting, etc. In order to avoid these problems, several methods include early stopping methods and weight decay method can be used often in these classical learning algorithms. The ELM tends to reach the solutions straightforward ignoring such trivial issues. A simple comparison between the ELM and SVM has also been conducted in our simulations, showing that the ELM may learn faster than SVM by a factor up to thousands.

2.13 EXTREME LEARNING MACHINE FOR REGRESSION AND MULTICLASS CLASSIFICATION

In this paper a new regression algorithm called Extreme Learning Machine is presented. ELM is a SLFNs, has hidden layer called feature mapping need not be tuned. This paper describes that ELM provides a unified learning platform it can be applied for regression and multiclass classification applications and it has milder optimization constraints compared toleast square SVM (LS-SVM) and proximal SVM (PSVM) [13]. Compared to ELM, LS-SVM and PSVM achieve suboptimal solutions and require higher computational complexity and ELM can approximate any target continuous function and classify any disjoint regions. The simulation results verifies that ELM has better scalability and achieve similar or better generalization performance at much faster learning speed than traditional SVM and LS-SVM.

2.14 WHAT ARE EXTREME LEARNING MACHINES? FILLING THE GAP BETWEEN FRANK ROSENBLATT'S DREAM AND JOHN VON NEUMANN'S PUZZLE

ELM is an emergent machine learning technique which has acquired attention to wide researchers around the world. This paper clarifies that ELM algorithm manage to address the open problem which has puzzled the neural networks, neuroscience and machine learning communities over the past 60 years: whether hidden neurons needed to be tuned for learning and proved that in contrast to the common knowledge and conventional neural network learning tenets, hidden neurons do not need to be iteratively tuned [14].

Unlike ELM, none of those earlier works gives theoretical foundations on feedforward neural networks with randomly assigned hidden nodes. ELM is suggested for both single-hidden-layer feedforward network and multi-hidden-layer feedforward networks. Compared with ELM, SVM and LS-SVM tend to provide only suboptimal solutions and SVM and LS-SVM do not consider feature representations in multi-hidden-layer feedforward networks.

2.15 OPTIMIZATION METHOD BASED EXTREME LEARNING MACHINE FOR CLASSIFICATION

G.B Huang, X.Ding and H.Zhou (2010) proposed a least square based approach called ELM for training feed forward networks [15]. ELM shows good performance in regression and classification applications. This paper shows further studies in ELM and extends it to specific type of generalized SLFNs called support vector network. This paper shows that SVM's maximal margin property and minimal norm of weights theory of feed forward neural networks are consistent under ELM learning framework and ELM has special separability feature and it has less optimization constraints compared to SVM. The simulation results prove that ELM used for classification tends to achieve better generalization performance than traditional SVM. It is proven that ELM for classification is less sensitive to user specified parameters and it can be implemented easily. In SVM some of the training data may not be linearly separable so it permits training error. In ELM, all the training data are linearly separable and it also permits training error to eliminate possible over fitting and to minimize test errors to improve generalization performance.

2.16 TEST GENERATION ALGORITHM FOR FAULT DETECTION OF ANALOG CIRCUITS BASED ON EXTREME LEARNING MACHINE

Test Generation Algorithm for Fault Detection of Analog Circuits Based on ELM proposed by Jingyu Zhou, Shulin Tian, Chenglin Yang, and Xuelong Ren which is proposed to apply traditional functional testing to ICs and SoCs [16]. This paper provides an advanced test generation algorithm for analog circuits. ELM-based classification algorithm is less time-consuming than SVM and has better computational complexity of learning algorithm. This algorithm avoids precision reduction under compressing and saves time efficiently. ELM-based algorithm proposed in this paper has much simpler processes due to signal generator and test as well as tradeoff parameters not sensitive to in terms of classification accuracy in the Computational Intelligence and Neuroscience.

2.17 DEEP EXTREME LEARNING MACHINE AND ITS APPLICATION IN EEG CLASSIFICATION

The concept of deep learning was first proposed by Hinton and Salakhutdinov in 2006, who presented deep structure of multilayer autoencoder. This paper highlights the need of MLELM and deep extreme learning machine (DELM) for the classification of the visual feedback experiment and the second brain-computer interface (BCI) competition datasets, using MATLAB [17]. Activation functions used for ELM, MLELM, and DELM is sigmoid function and for KELM is kernel function. ELM, MLELM, and DELM were performed 100 times, and their average values are reported. It is observed from the experiment that DELM testing accuracy is greater than MLELM, average values of DELM testing accuracy are higher than ELM and KELM. Although DELM have advantages, there are some difficulties observed which should be improved, such as the number of hidden layer nodes, hidden layer activation function, and layer parameter that are difficult to determine.

2.18 ENCRYPTED IMAGE CLASSIFICATION BASED ON MULTILAYER EXTREME LEARNING MACHINE

The author of this paper Weiru Wang, states that effective search algorithm is required to crawl out the images with queried objects from databases [18]. Privacy protection is a serious issue, decrypting and then classifying millions of images becomes a heavy burden for computation. In this paper, the author proposed an encrypted image classification algorithm based on MLELM that is able to directly classify encrypted images without decryption. Experiments were conducted and the results convey that the proposed method is secure, efficient and accurate for classifying encrypted image. The framework is capable to extract higher order features from the encrypted images for classification directly. Through experiments, the framework that is proposed based on MLELM achieves accuracy of 90.44 and 79.83% on MNIST, respectively encrypted under data encryption standard and advanced encryption standard.

2.19 REPRESENTATIONAL LEARNING FOR FAULT DIAGNOSIS OF WIND TURBINE EQUIPMENT: A MULTI-LAYERED EXTREME LEARNING MACHINES APPROACH

To avoid sudden interruption and reduction in maintenance cost, reliable fault diagnosis is crucial for the wind turbine generator system (WTGS) is discussed by Zhi-Xin Yang in this paper [19]. The conditional data generated from WTGS operating in a tough environment is always high-dimensional and dynamical. To notify these challenges, we propose a new fault diagnosis scheme which is designed by multiple extreme learning machines in a hierarchical structure. The framework enables representational feature learning as well as fault classification. To evaluate its performance, tests in comparison are carried out on a wind turbine generator simulator. The results show that the evolved diagnostic framework accomplishes the best performance among the compared approaches in terms of accuracy and effectiveness. In order to verify the effectiveness of the proposed scheme, this paper results in various coupling of methods to realize the contrast experiments. The framework is successfully applied on recognizing the wrong patterns coming from the WTGS system. Unlike the well adopted data preprocessing methods using a combination of wavelet packet transform, time-domain statistical features and kernel principal component analysis, the proposed MLELM could leverage the down-streamed classification accuracy in around 5%–10% for different corresponding classifiers.

2.20 CLASSIFICATION BASED ON MULTILAYER EXTREME LEARNING MACHINE FOR MOTOR IMAGERY TASK FROM EEG SIGNALS

To improve the Classification accuracy of motor imagery electroencephalogram (EEG), Jun Miao introduces a classification system based on MLELM [20]. In the current system, the combination of LDA and PCA is chosed for feature extraction and MLELM for classification. The data used here is gathered from BCI competition 2003 data set Ia, which is a high quality of the data set issued by University of Tübingen, Institute of Medical Psychology and Behavioral Neurobiology, Niels Birbaumer. In the experiment, our method is compared with other methods of ELM, such as Constrained-ELM, kernel-ELM and V-ELM on the same dataset. The experimental results show that MLELM is much more capable for motor imagery EEG data and it has better performance than the others. Here the method is just applied to the binary-class EEG data. However, there is an improvement needed in the method for multi-classification of EEG.

2.21 DIMENSION REDUCTION WITH EXTREME LEARNING MACHINE

The main aim of the dimension reduction algorithms like Principal Component Analysis (PCA), random projection (RP), auto-encoder (AE) and Non-negative Matrix Factorization (NMF), is to minimize the irrelevant information or noise from the data. The features of PCA and linear AE are not able to indicate data as parts. Non-linear AE and NMF have slow learning speed. In this paper, dimension reduction framework which represents data as parts, has rapid learning speed and learns the between-class scatter subspace is proposed by yan yang [21]. This paper examines the linear and nonlinear dimension reduction framework like ELM-AE and Sparse Extreme Learning Machine Auto-Encoder (SELM-AE).SELM-AE and the hidden neurons in ELMAE need not be tuned in contrast to the weight auto-encoder (TAE) and their parameters are initialized using orthogonal and sparse random weights. Experimental results on USPS handwritten digit recognition dataset, NORB object recognition data set and CIFAR-10 object recognition dataset shows that the efficacy of ELM-AE and SELM-AE in terms of sparsity, discriminative capability, Normalized Mean Square Error and training time. The experimental results show that the SELM-AE (linear and non-linear) and ELM-AE (linear and non-linear) learning the features more localized than PCA.

2.22 MULTILAYER EXTREME LEARNING MACHINE WITH SUBNETWORK NODES FOR REPRESENTATION LEARNING

This paper represents the general architecture of MLELM with sub network nodes by Yimin Yang [22], showing that the proposed method results a representation learning platform with unsupervised/supervised and compressed/sparse representation learning. The experimental results on ten image datasets and 16 classification datasets shows that, compared to other conventional feature oriented learning methods, the proposed MLELM with sub network nodes performs competitively or much better than other feature learning methods. The learning speed of the proposed method can be several times faster than deep networks such as deep belief networks (DBNs) and stacked auto-encoders (SAE). Furthermore, our platform can provide much better generalization performance than other feature extraction methods such as linear graph embedding, isometric projection, and linear discrimination analysis, etc. The proposed method provides much better reconstructions than DBN and PCA. The experimental results shows that the proposed method can provide a similar or much better generalization performance compared to other representation learning methods.

2.23 STUDY ON SUITABLY AND IMPORTANCE OF MULTILAYER EXTREME LEARNING MACHINE FOR CLASSIFICATION OF TEXT DATA

The author of this paper Rajendra Kumar Roul, proposes a new method for text classification [23]. This study proposes an efficient method that uses the concept of connected component of a graph and Wordnet along with four established feature selection techniques. The behavior of different classifiers are studied and compared using these four feature selection techniques, and it is observed that MLELM achieves 72.28% on DMOZ dataset using TF-IDF and 81.53% on 20 Newsgroup dataset using BNS as feature selection technique.

The experimental work which focused on text classification process is carried out on two standard datasets: DMOZ (Open Directory Project) and 20-Newsgroups. The performance of different classifiers is compared in the experimental section, and it has been observed that MLELM leads the other established classifiers including ELM and SVM. MLELM yields a very good result which demonstrates the efficiency of our approach compared to other existing approaches.

2.24 ON THE KERNEL EXTREME MACHINE CLASSIFIER

This paper discusses about the kernel version of the ELM classifier with SLFN of infinite hidden layer [24]. The kernel matrix is computed using the kernel formulation and the activation function and the obtained kernel matrix is a low –rank matrix. The activation function used here is RBF and sigmoid functions. The ELM space obtained from this process is used for training the network by using original ELM formulation. The algorithm is executed on the different data sets like Libras, Madelon, Opt.Digits, segmentation and the results indicate that the low-rank decomposition based ELM space leads to best performance when compared to the standard random input weights generation.

2.25 AN IMPROVED KERNEL BASED EXTREME LEARNING MACHINE FOR ROBOT EXECUTION FAILURES

This paper introduces novel learning algorithm KELM along with particle swarm optimization approach for the classification or prediction of robot execution failures [25]. This algorithm produces higher accuracy when the learning samples are very limited and even with the erroneous data. The higher accuracy of the algorithm is mainly due to the parameters of the kernel function, these parameters of the neural network are adjusted for searching the optimal values by particle swarm optimization technique. The simulation results indicate that the algorithm shows better accuracy and better generalization performance compared to the other traditional neural network and ELM algorithms

2.26 KERNEL-BASED MULTILAYER EXTREME LEARNING MACHINES FOR REPRESENTATION LEARNING

MLELM was adapted to SAE for representation learning; here training time is reduced with higher accuracy [26]. Although, MLELM suffers from the drawbacks like manual tuning of number of hidden nodes in each and every layer, suboptimal model generalization and large reconstruction error. To overcome these drawbacks, a kernel version of MLELM is developed. Benchmark data sets of different sizes are employed for the estimation of MLKELM. Experimental results show that the accuracy is improved over benchmark data set is up to 7%.

2.27 FAST DETECTION OF IMPACT LOCATION USING KERNEL EXTREME LEARNING MACHINE

Traditional learning algorithms are time consuming because of their computational complexity so that the real-time requirement applications cannot be fulfilled in practical applications. So kernel ELM algorithm is used to predict the impact location on a plate, which is an illustrative example for the aircraft health monitoring and maintenance is proposed [27]. KELM is used to predict impact locations on the x- and y-coordinates of impact locations. The speed of proposed KELM is 1.43 times and 35.2 times faster than SVM and BPNN. The comparison result reveals the effectiveness of kernel ELM for impact detection, shows that kernel ELM has better accuracy than SVM.

2.28 SPECTRAL-SPATIAL CLASSIFICATION OF HYPERSPECTRAL IMAGE BASED ON KERNEL EXTREME LEARNING MACHINE

To integrate spectral and spatial information for hyperspectral image classification and to exploit the benefits of spatial features, KELM classifier is proposed in this paper [28]. Particularly, two-dimensional Gabor filter was employed to extract the spatial features in the PCA domain. Multihypothesis (MH) prediction preprocessing is used to integrate the spectral and spatial information. The proposed MH-prediction-based KELM classifier and Gabor-filtering-based KELM are verified based on two real hyperspectral datasets. RBF kernel is used for both KELM and SVM, SVM is implemented by using libSVM package and KELM uses the implementation available in ELM website.

Experimental results revealed that the KELM algorithm can outperform the Gabor-filtering-based SVM and MH-prediction-based SVM. Particularly, proposed method achieved over 16% and 9% classification accuracy improvement for both Indian Pines dataset and the University of Pavia dataset. MH-KELM outperformed MH-SVM by 5% for the Indian Pines dataset and Gabor-KELM outperformed Gabor-SVM by 1.3% for the University of Pavia dataset.

2.29 ILLUMINATION CORRECTION OF DYEING PRODUCTS BASED ON GREY-EDGE AND KERNEL EXTREME LEARNING MACHINE

Kernel extreme learning machine based color constancy is applied it to textile illumination correction is proposed in this paper [29]. In addition, a new effective and low dimensional color feature extraction method which is based on Grey-Edge framework is used to replace the old traditional high dimensional binary chromaticity histogram.

Totally 1150 indoor textile images were collected under the light source of D65, D55, D50 and the standard illuminant A. The Sigmoid function is chosen as the activation function. The proposed method is compared with the support vector regression and extreme learning machine on the same image set. The experimental results show that KELM performs better than ELM and SVR and it method reduces the median and root mean square errors with approximately 6%, 11%, 43% and 48% respectively.

2.30 MODELING AND OPTIMIZATION OF BIODIESEL ENGINE PERFORMANCE USING KERNEL-BASED EXTREME LEARNING MACHINE AND CUCKOO SEARCH

Due to the absence of randomness and for better generalization performance, KELM has been preferred in this paper to construct a biodiesel engine model based on experimental data [30]. By using KELM method, a biodiesel engine model is first created based on experimental data and cuckoo search is then employed to calculate the optimal biodiesel ratio. Even if the sample data used in this study suffered from the data scarcity and exponentiality problems, logarithmic transformation of dependent variables is applied to fix these problems at the same time. For demonstrating the effectiveness of the KELM algorithm, a comparison is between the K-ELM model and the LS-SVM model is made, under the same sample data sets. The experimental result shows that K-ELM can achieve better performance than LS-SVM, resulting a reliable prediction result for Optimization.

CHAPTER 3

FAULT DIAGNOSTIC FRAMEWORK

3.1 PROPOSED DESIGN

The proposed design consists of order of steps as shown in figure 3.1.The basic purpose is to determine the transfer function of the benchmark circuit and the faults are injected by varying the component value with step size of 10% within the tolerance limit of ±50% and it is simulated to obtain the features gain, pole selectivity and frequency. Single fault is introduced to one component at a time with other fault free components taking different random values within their tolerance limit. Fault dictionary is constructed and it is separated in to training and testing samples, then it is normalized and these samples are given as input to the MLELM and KELM with PSO algorithm for fault classification.

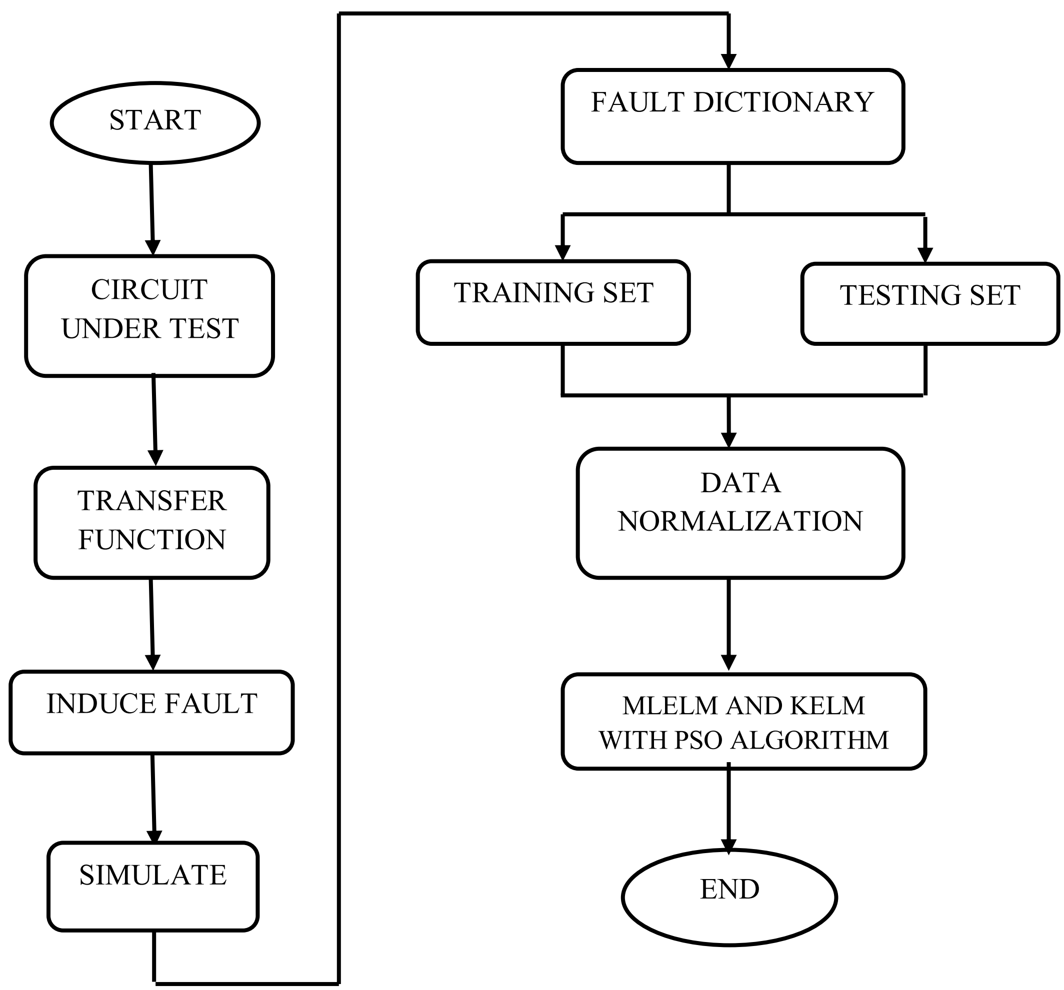

Figure 3.1 Fault Detection Framework

3.2 CIRCUIT UNDER TEST

3.2.1 State Variable Filter Circuit

The state variable filter which is a type of active filter having simultaneous low-pass, high-pass and band-pass output responses. State variable filter uses three or more operational amplifier circuits cascaded to produce individual filter outputs. State variable filter as shown in figure 3.2 is a second-order RC active filters consisting of two identical op-amp integrators with each one behave as a first-order, single pole low pass filter, a summing amplifier around which we can set the filters gain and damping feedback.

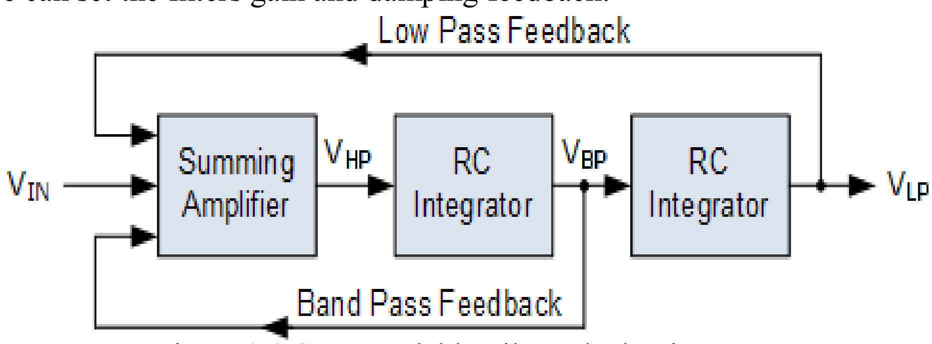

Figure 3.2 State Variable Filter Block Diagram

The output signals from the three op-amp stages are fed back to the input allowing us to define the state of the circuit. The major advantages of a state variable filter design is that all three of the filters main parameters, Gain (K), corner frequency (f_C) and the filters selectivity (Q) can be adjusted or set independently without affecting the filter performance. An added advantage over bi-quad filter is that only one coefficient is needed, rather than their five coefficients.

3.2.1.1 SVF Transfer Function

The transfer function is the ratio of output voltage to the input voltage. Any Linear time invariant system can be represented as a state-space model, with n state variables for an nth-order system. The low pass and high pass output's are phase inverted while the band pass output maintains them in phase. The gain of each output is an independent variable. Due to temperature variation, component value may vary but must be in tolerance limit.

The nominal values of the circuit components are:

R1 = R2 = R3 = R4 = R5 = 10kΩ;

R6 = 3kΩ;

R7 = 7kΩ;

C1 = C2 = 20nF.

All the parameters were assigned with $\pm$10% torelence.

The voltage transfer function of the second-order SVF (Figure 3.3), considering its low-pass output (LPO) is given by

$$\frac{V_{LPO}}{V_{input}} = \frac{-R_5}{R_1}\left[\frac{\frac{R_2/R_5}{R_3C_1R_4C_2}}{S^2+\frac{\left(1+\frac{R_2}{R_5}+\frac{R_2}{R_1}\right)s}{\left(1+\frac{R_7}{R_6}\right)R_3C_1}+\frac{R_2/R_5}{R_3C_1R_4C_2}}\right] \tag{3.1}$$

Comparing the equation 3.1 with second order low-pass filter transfer function, we get the following relations for k, $\dot{\omega}_0$ and Q.

$$\text{Gain, } K = \frac{R5}{R1} \tag{3.2}$$

$$\text{Pole frequency, } \dot{\omega}_0 = \sqrt{\frac{\text{R2/R5}}{\text{R3C1R4C2}}} \tag{3.3}$$

$$\text{Pole selectivity, } Q = \sqrt{\left(\frac{R3C1}{R4C2}\right)\left(\frac{R2}{R5}\right)}\;\frac{1+\frac{R7}{R6}}{1+\frac{R2}{R5}+\frac{R2}{R1}} \tag{3.4}$$

Therefore for the LPO of filter with nominal values of the components yields k= 1.0, Q = 1.11 and f_o = 796HZ.

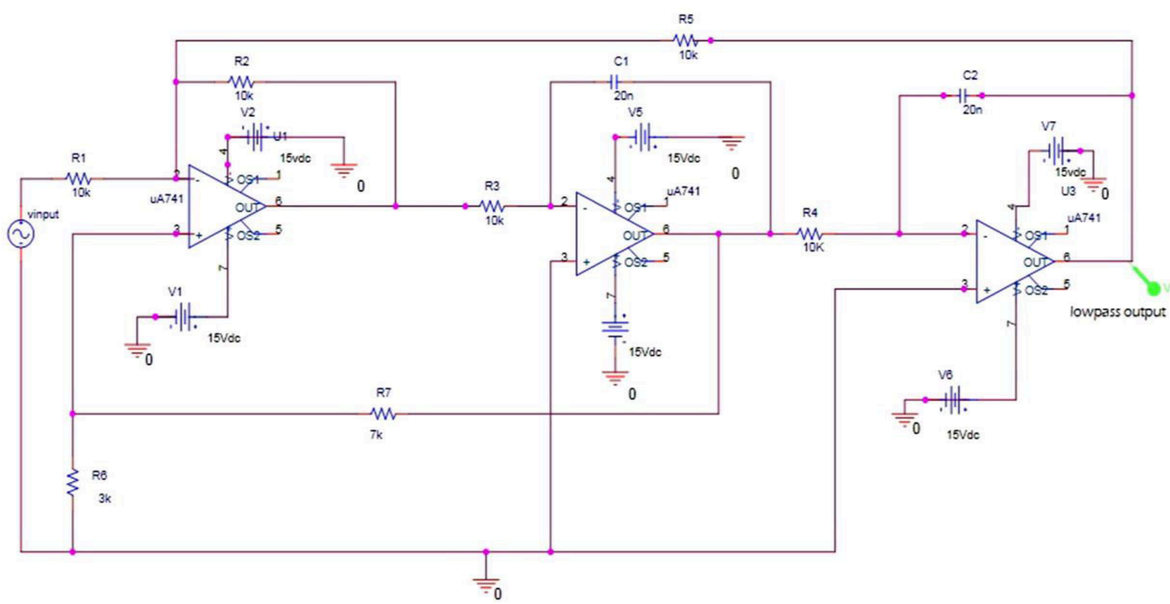

Figure 3.3 State Variable Filter

3.2.1.2 Sample Fault Database For SVF Circuit

Fault dictionary is generated by injecting fault to all the components and then evaluating the parameters K, Q, f. The fault dictionary contains samples of both fault-free and faulty values. There are two types of fault dictionaries, fault dictionary with single fault and fault dictionary with double fault. Single fault dictionary is constructed by injecting fault to a single component and the other component values are varied within their tolerance limit. Double fault is constructed by injecting faults to two components at a time and other components are varied within their tolerance limit. In this project, fault dictionary with single fault is considered. There are totally 9 components in the circuit so the total fault injected is 9. From the fault dictionary, samples for training and testing are obtained. A sample fault dictionary is given below in the table 3.1.

Table 3.1 Fault Dictionary Samples

Target	Resistor	Gain	Pole Selectivity	Frequency
1	11000	0.910237	1.151268524	795.2619181
1	11000	0.909171	1.147036158	798.0061491
1	11000	0.907227	1.143966426	796.6476759
1	11000	0.908585	1.148800112	798.3129886
1	12000	0.833162	1.178369422	796.769863
1	12000	0.832309	1.175471185	796.2174176
2	14000	0.998964	1.037240938	943.1519069
2	14000	1.001707	1.036592011	940.5867861
2	14000	0.99729	1.041261579	943.0619943
2	14000	0.996838	1.038652605	943.5703492
2	14000	0.998326	1.038511084	940.3097349
2	14000	1.001366	1.042958186	941.0638149
3	13000	1.002881	1.26095022	698.8666136
3	14000	0.997876	1.313955515	673.5067958
3	14000	0.998392	1.320197132	674.4033499
4	13000	1.000863	0.969920385	698.1284696

3.2.1.3 Data Normalization

All the input parameters except the expected targets in the fault dictionary is normalized in the range [-1 to 1], because the MLELM and KELM with PSO algorithm works only on the normalized data set. To normalize the vector in this range, we need to use minimum and maximum value. The normalization is performed separately on training and testing data sets for SVF. The normalization vector helps in reducing the computational complexity. Table 3.2 shows the sample normalized fault dictionary of SVF circuit.

Table 3.2 Normalized Fault Dictionary Samples

Target	Resistor	Gain	Pole Selectivity	Frequency
1	-0.2	0.552968	-0.680494607	-0.17937494
1	1	-0.77877	-0.101292239	-0.17285171
1	0.866667	-0.71602	-0.136993204	-0.17553874
1	0.6	-0.55543	-0.213752791	-0.17706586
1	0.466667	-0.45493	-0.27710344	-0.17553013
2	0.733333	-0.64404	-0.177739734	-0.17812384
2	-0.06667	-0.3369	-0.248901217	-0.63050879
2	0.466667	-0.33597	-0.364467981	-0.0398042
2	0.866667	-0.33853	-0.456573248	0.345462317
2	0.733333	-0.33911	-0.426738061	0.218437645
3	-0.06667	-0.33345	-0.638641006	0.378275636
3	0.6	-0.33474	-0.156227158	-0.4269329
3	0.6	-0.32994	-0.146510751	-0.42303652

3.2.2 SKBPF Circuit

The Sallen–Key Bandpass filter is also known as voltage control –voltage source topology. It is one of the most extensively used filter topologies and it is used to implement second order active filter. This filter uses unity gain voltage amplifier with infinite input impedance and zero output impedance. It can be used to implement low-pass, band-pass and high-pass structure. The super-unity-gain amplifier allows for very high Q factor and passes band gain without the use of inductors. The Sallen-Key band pass filter structure shown in figure 3.4 is mainly used because the section gain is fixed by the other parameters and there is a vast stretch in component values, especially capacitors.

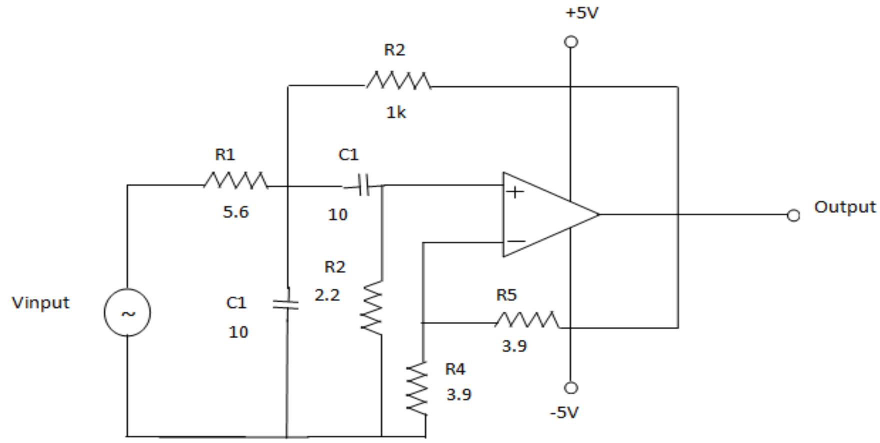

Figure 3.4 Sallenkey Bandpass Filter

3.2.2.1 SKBPF Transfer Function

The nominal values of the circuit components are given below:

R1 = 5.6kΩ;

R2 = 1kΩ;

R3 = 2.2kΩ;

R4 = R5 = 3.9kΩ;

C1 = C2 = 10 nF.

All the components were assigned ±5%.
The voltage transfer function of the Sallen- key band pass filter circuit is given by

$$H(s) = \frac{V_0(s)}{V_{in}(s)} \tag{3.5}$$

$$= \frac{\frac{ks}{R_1C_1}}{S^2+\left(\frac{1}{R_2C_1}+\frac{1}{R_3C_2}+\frac{1}{R_3C_1}+\frac{1-K}{R_2C_1}\right)S+\frac{R_1+R_2}{R_1R_2R_3C_1C_2}} \tag{3.6}$$

we get the following relations for K, ώ0, and Q.

$$Gain, K = \frac{k}{R_1C_1} \tag{3.7}$$

$$Pole\ Frequency, \omega_0 = \sqrt{\frac{R_1+R_2}{R_1R_2R_3C_1C_2}} \tag{3.8}$$

$$Poleselectivity, Q_p = \frac{\sqrt{\frac{R_1+R_2}{R_1R_2R_3C_1C_2}}}{\frac{1}{R_1C_1}+\frac{1}{R_3C_1}+\frac{1}{R_3C_2}+\frac{1-k}{R_2C_1}} \tag{3.9}$$

Therefore for the sallenkey bandpass filter with nominal values of the components yields k = 75,987, Q = 8.34 and fo = 25KHZ.

3.2.2.2 Fault Dictionary Creation

The procedure for creation of fault dictionary is as same as SVF fault creation. The transfer function is simulated with faults injected to the components. The fault injection is done to the extent of ±50% deviation from nominal value with a step size of 5%. There are totally 7 components in the circuit so the total fault injected is 7 which correspond to the number of classes. Fault dictionary is generated injecting fault to all component and evaluating the parameter K, Q, f. The input sample of size 1443×4 is obtained. Fault dictionary are separated into train samples and test samples randomly. A sample fault dictionary is given below in the table 3.3.

Table 3.3 Fault dictionary samples for SKBPF circuit

Target	Resistor	Gain	Pole Selectivity	Frequency
1	5880	72386.7	7.71696	24649.35
1	5880	72265.8	10.7935	24585.49
1	5880	72539.8	9.61000	24486.47
1	5880	72265.3	9.25718	24636.08
1	6160	69007.6	8.721377	24576.05
1	6160	69197.11	8.825022	24429.72
1	6160	69143.26	13.54071	24793.58
1	6160	69060.86	9.881653	24920.91
1	6160	69136.2	10.43722	24629.85
2	1050	69141.01	7.135413	24434.82
2	1050	69255.5	12.06381	24806.14
2	1050	69065.08	14.17858	24690.2
2	1050	69089.22	10.76941	24732.67

3.2.2.3 Data Normalization For SKBPF Circuit

The procedure for the normalization of data is as same as SVF data normalization. The data are normalized in the range of [-1 to +1]. To normalize the vector in this range, we need to use minimum and maximum value. The normalization is performed separately on training and testing data sets for SKBPF. The normalization vector helps in reducing the computational complexity. Table 3.4 shows the sample normalized fault dictionary of SKBPF circuit.

Table 3.4 Normalized fault dictionary

Target	Resistor	Gain	Pole Selectivity	Frequency
1	0.4	-0.5646	-0.1477	-0.3744
1	0.4	-0.5651	-0.1473	-0.3893
1	0.4	-0.5644	-0.1491	-0.4208
1	0.4	-0.5661	-0.148	-0.4064
1	0.4	-0.5649	-0.1488	-0.3858
2	-0.75	-0.4942	-0.1511	-0.4632
2	-0.75	-0.5011	-0.1514	-0.4496
2	-0.75	-0.4967	-0.1516	-0.4428
2	-0.75	-0.4959	-0.1509	-0.4659
2	-0.75	-0.4922	-0.1511	-0.4654
3	-0.45	-0.498	-0.1432	-0.4632
3	-0.45	-0.4974	-0.1434	-0.4681
3	-0.45	-0.4882	-0.138	-0.4445

3.3 PERFORMANCE METRICS CALCULATION

The performances of both the algorithms are evaluated by using confusion matrix. A confusion matrix is an approach for describing the performance of a classification algorithm on the dataset for which the true values are known. Confusion matrix is also called as error matrix and it is a specific table layout which allows performance visualization of the algorithm and it is mainly used for supervised learning. Each column of the matrix represents the predicted class instances and each row represents the instances of actual class. The figure 3.5 shows the general confusion matrix for 3 class classification.

		Predicted class				
		Class 1	Class 2	Class 3	Class 4	Class 5
Actual Class	Class 1	True positives				
	Class 2		True positives			
	Class 3			True positives		
	Class 4				True positives	
	Class 5					True positives

Figure 3.5 Confusion matrix

The element (m, n) in the confusion matrix is the number of samples whose predicted class is m and whose actual class is n. The diagonal elements represent the correctly classified samples.

3.3.1 Possible Outcomes of Confusion Matrix

The confusion matrix consists of information about actual class and predicted class. The matrix describes the four different possible outcomes of the result. They are True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN).

i. TRUE POSITIVE

The correct positive prediction denotes true positive. In the confusion matrix it corresponds to the diagonal element of the corresponding class.

For class 1 TP = Confusion matrix (1, 1).

ii. TRUE NEGATIVE

The correct negative prediction denotes the true negative. In the confusion matrix it corresponds to the sum of the columns and rows by excluding that particular class.

For class 1 TN= Confusion matrix (2, 2), (2,3), (2,4), (2,5), (3,2) , (3,3), (3,4), (3,5), (4,2) , (4,3), (4,4), (4,5), (5,2) , (5,3), (5,4) and (5,5).

iii. FALSE POSITIVE

The incorrect positive prediction denotes the false positive. In the confusion matrix it corresponds to the sum of the values in the corresponding column.

For class 1 FP= Confusion matrix (2, 1), (3, 1), (4, 1) and (5, 1)

iv. FALSE NEGATIVE

The incorrect negative prediction denotes the false negative. In the confusion matrix it corresponds to the sum of the values in the corresponding row.

For class 1 FN= Confusion matrix (1, 2), (1, 3), (1, 4) and (1, 5)

These measures are used for analyzing the performance metrics. The performance metrics are accuracy, sensitivity, specificity, error, precision, f-measure, FPR, FNR, PPV and NPV

v. ACCURACY

The accuracy is the proportion of the total number of predicted samples that were correct. Generally, it is the ratio of correctly predicted samples to total predictions made. It is also calculated by 1 – ERROR.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.10)$$

vi. SPECIFICITY

Specificity is the measure of true negative rate which is the number of correct negative predictions by total number of negatives.

$$Specificity = \frac{TN}{TN+FP} \quad (3.11)$$

vii. SENSITIVITY

Sensitivity which is also known as recall is a true positive rate. It is calculated as the number of correct positive predictions by the total number of positives.

$$Sensitivity = \frac{TP}{TP+FN} \quad (3.12)$$

viii. ERROR

Error is the measure of misclassification rate which is the proportion of incorrect classification to the total number of samples.

$$Error = \frac{FP+FN}{TP+TN+FP+FN} \quad (3.13)$$

ix. Precision

Precision is the reproducibility of measurement.

$$Precision = \frac{TP}{TP+FP} \quad (3.14)$$

x. F-measure

F-measure is the measure of test accuracy. It is the harmonic mean of precision and recall.

$$f - measure = \frac{2 \times Precision \times Recall}{Precision+Recall} \quad (3.15)$$

xi. FPR

FPR is the False Positive Rate which is the proportion of the absent events that yield positive test outcomes. It is the probability of a positive test result given an absent event.

$$FPR = \frac{FP}{FP+TN} \quad (3.16)$$

xii. FNR

FNR is the False Negative Rate which is the proportion of events that are being tested that yield negative test incomes (i.e.) the conditional probability of a negative test given that the event being looked for has taken place.

$$FNR = \frac{FN}{TP+FN} \tag{3.17}$$

xiii. PPV

PPV is the Positive Predictive Values which is the proportion of subjects with positive results and was correctly diagnosed.

$$PPV = \frac{TP}{TP+FN} \tag{3.18}$$

xiv. NPV

NPV is the Negative Predictive Value which is the proportion of subjects with negative results and was correctly diagnosed.

$$NPV = \frac{TN}{FN+TN} \tag{3.19}$$

CHAPTER 4
MULTILAYER EXTREME LEARNING MACHINE IMPLEMENTATION FOR DETECTING ANALOG CIRCUIT FAULTS

4.1 INTRODUCTION OF MLELM

Multilayer ELM is an effective machine learning approach based on the architecture of artificial neural network and is motivated by deep learning and extreme learning machine [17]. Deep learning has been proposed by Hinton and Salakhutdinov (2006) who uses deep structure of multilayer auto encoder and created a multilayer neural network on the unsupervised data. Working in this direction, Kasun et al. (2013) proposed multilayer ELM which performs unsupervised learning from layer by layer and it does not need to iterate during the training and hence, it does not spend a long time during training phase.

4.1.1 Architectural Description of MLELM

In the architecture of MLELM shown in figure 4.1, output weights β^1 of (a) ELM-autoencoder (AE) is denoted according to input data x are the layer 1 weights of MLELM. (b) Output weights are denoted as β^{i+1} in ELM-AE, according to ith hidden layer and output hi of MLELM are the i + 1th layer weights of MLELM; (c) Regularized least squares are used in calculating the output layer weights of MLELM. In each layer, MLELM uses ELM-AE to train the parameters and hidden layer activation functions of MLELM can be either linear or nonlinear piecewise. If the ith hidden layer activation function of MLELM is g(x), then the parameters between the ith hidden layer and the (i−1) hidden layer of MLELM are trained by ELM-AE, and then, the activation function also be g(x).

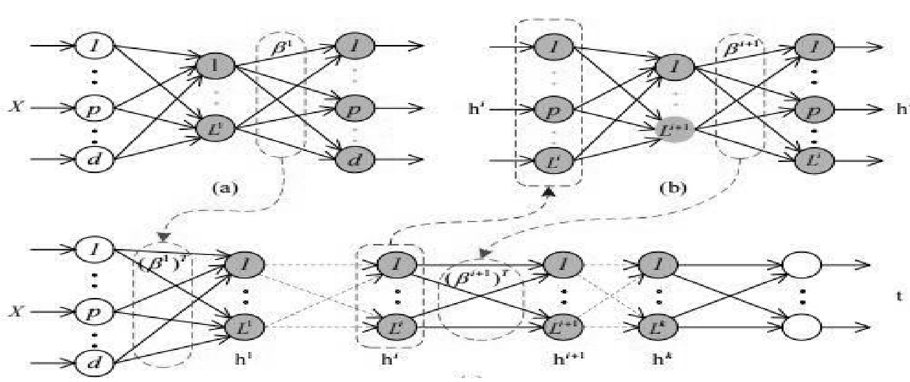

Figure 4.1ELM-AE and Multilayer ELM

Some of the broad application of MLELM includes EEG classification, fault diagnosis of wind turbine, text data classification, brain tumor detection and classification, 3d feature learning and image classification. In this work, MLELM algorithm is proposed for the fault classification of analog circuits.

For MLELM the following steps are involved,

- **Step 1:** Assign weights between hidden nodes and input nodes and the bias of the hidden nodes to be orthogonal
- **Step 2:** Calculate the hidden layer output matrix
- **Step 3:** The output weight $\boldsymbol{\beta}$ is decided on the basis of the number of nodes in input and hidden layer
- **Step 4:** Calculate the training and testing accuracy from the miss-classification training and testing rate. Calculate the training and testing time is from the CPU time
- **Step 5:** Evaluate the performance metrics based on the confusion matrix.

4.2 ELM- AUTOENCODER

ML-ELM is an ELM algorithm with multiple hidden layers between input and output layer while each hidden layer is constructed by ELM-AE [18]. Autoencoder is an artificial neural network which is frequently used in deep learning algorithm. Autoencoder is an unsupervised neural network, here the outputs and inputs of the autoencoder are same. Like ELM, ELM-AE shown in figure 4.2 also has n input layer nodes, hidden layer of L nodes and n output layer nodes. There are two major differences exist between them which are:

i. ELM is a supervised neural network but ELM-AE is an unsupervised one and the output of ELM is a class label but ELM-AE has output is same as the input.

ii. In ELM, input weights and biases are randomly assigned, but they are orthogonal in ELM-AE. Orthogonalization of randomly assigned weights and biases will increase the generalization performance of ELM-AE. This is the most important difference when compared with ELM.

Depends upon the number of hidden layer nodes, the ELM-AE can be classified into the following three categories, they are

(i) Compressed representation ($n > L$):

In compressed representation, features are represented from a higher – dimensional (or sparse) input signal space to a feature space of lower-dimensional (or compressed)

(ii) Equal dimension representation (n = L):

In this type representation, the dimension of input signal space and feature space needed to be equal.

(iii) Sparse representation (n < L):

It is just the reverse of compressed representation where the features needed to be represented from a lower-dimensional input signal space to a higher-dimensional (or sparse) feature space.

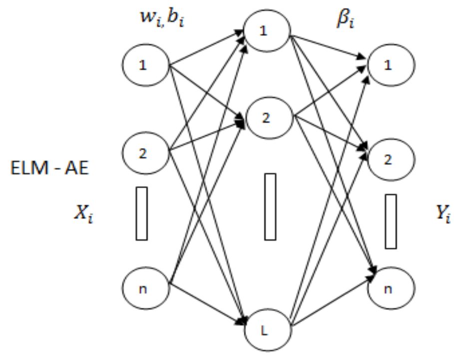

Figure 4.2 ELM-Autoencoder

4.3 FORMULA DESCRIPTION

The multilayer ELM is remarkably faster than deep networks because iterative tuning is not required here and obtained similar or better performance compared to deep networks. In order to perform unsupervised learning, few modifications done in ELM-AE whose working principle is similar to ELM, which are described as follows,

(1) The output of the auto encoder and the input data remain same for every hidden layer. Therefore, for every input data ***X***: ***Y*** = ***X***

(2) In order to improve the performance, we need to consider the weights and the biases to be orthogonal and can be represented as follows:

$$h = g(w \cdot x + b), w^T \cdot w = I \text{ and } b^T \cdot b = 1 \quad (4.1)$$

where $w = [\ w_1, \ldots, w_L\ ]^T$ and $b = [\ b, \ldots, b_L\ ]^T$ are the orthogonally generated random weights and bias between the input layer anf hidden layer respectively.here g(.) is the activation function of hidden layer.

(3)The output weight β is decided based on the following conditions:

i. if $n > L$ then

$$\beta = \left(\frac{I}{C} + H^T H\right)^{-1} H^T X \quad (4.2)$$

ii. if $n = L$ then

$$\beta = H^{-1}X \tag{4.3}$$

iii. if $n < L$ then

$$\beta = H^{T}(\frac{I}{C} + H^{T}H)^{-1}X \tag{4.4}$$

where C is a scalable parameter which adjusts structural and experiential risk. ELM-AE is used for training the parameters in each layer of MLELM.

where $H = [h_1, \ldots, h_N]$ is the output of the hidden layer and each h_i is L dimensional vector, $X = [x, \ldots, x_N]$ are N input and output samples of ELM-AE.Note that in ELM-AE input samples are equal to output data.

According to Huang et al. (2012), the output weight β of ELM-AE can be used to represent the features of the input data via singular values. Multilayer extreme learning machine is an ELM learning algorithm with multiple hidden layers between the input and output layers. Between the input layer and final hidden layer, the outputs of each and every layer are wired to the inputs of the successive layer (equation 4.5). Each hidden layer weights are initialized using ELM-AE which executes layer-wise unsupervised training. It has fast training speed and doesn't require fine tuning.

$$X^{k} = f(\beta^{k}X^{k-1}) \tag{4.5}$$

where X^{k} is the kth layer input data, X^{0} will be the input data layer and β^{k} is the kth hidden layer output weight. f(.) is chosen as linear if number of hidden nodes in kth hidden layer L^{k} is equal to the number of hidden nodes in (k-1)th hidden layer L^{k-1}.otherwise choose f(.) as nonlinear piecewise, such as sigmoid function g.

4.4 PROPOSED FAULT CLASSIFICATION USING MLELM

The flow diagram of the proposed MLELM algorithm is shown in figure 4.3. The training and testing samples are obtained from the fault dictionary. 75% data are chosen for training and 25% data is chosen for testing. The testing and training data are chosen randomly. The training and testing data are normalized in the range -1 to 1. The normalized training and testing data is given as the input to the algorithm. Different hidden neurons with different layers are used for the MLELM. MLELM has input parameters such as training data, testing data, number of layers, number of hidden nodes, and signal parameters. The output from the algorithm implementation is the correct detection of fault index as per the target defined.

The testing and training samples are given as input to the MLELM algorithm. Number of layers and neurons are initialized and inputted to the algorithm. Input weights and bias of hidden neurons are generated random manner which obeys orthogonal property. Compute the hidden layer matrix using activation function. Based on n and L values, output weights are computed. Compute the expected target from output weight and hidden layer matrix. Finally compute the classification accuracy based on the actual and expected target.

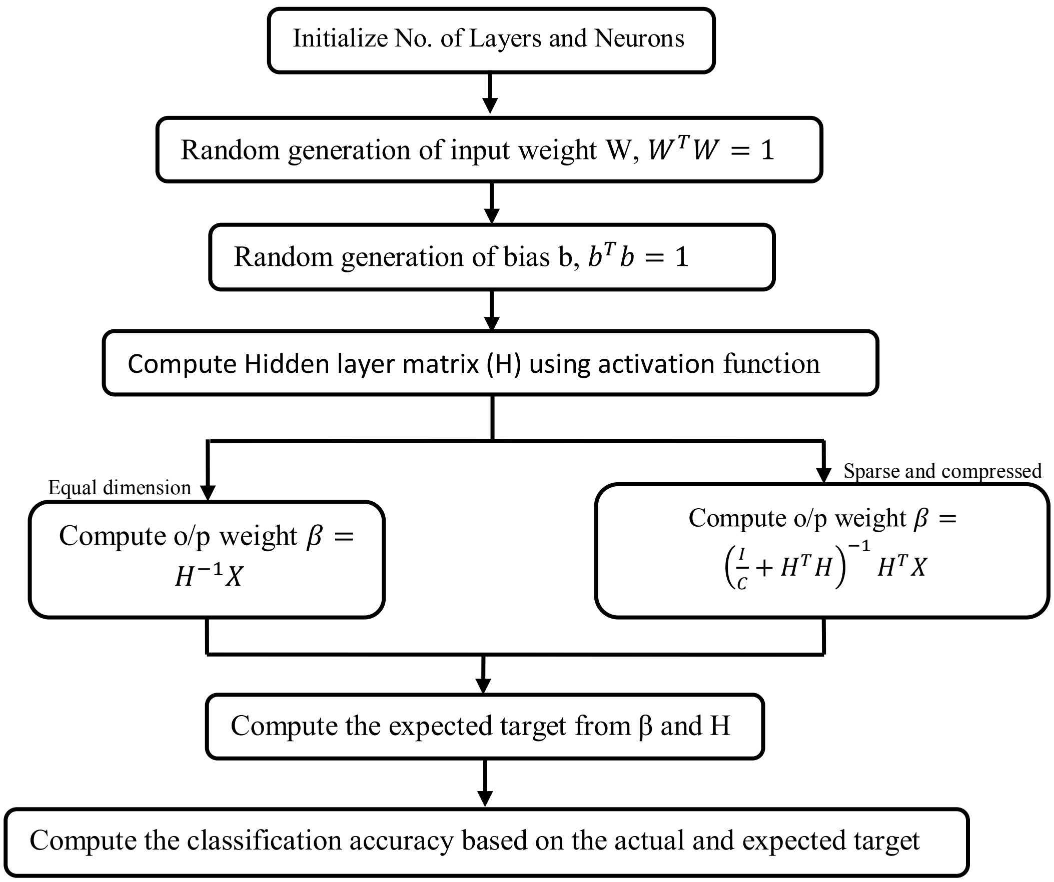

Figure 4.3 Flowchart for MLELM

4.5 ACTIVATION FUNCTIONS

In computational networks, activation function corresponding to a node is the output of that node with respect to input or set of inputs. In neural network, it is the function used to transform the activation level of a neuron into the output signal. Five types of activation functions are generally used, they are sigmoid, sine, hard-limit, triangular basis and radial basis.

For MLELM algorithm, sigmoid activation function gives higher accuracy for SVF circuit and triangular basis activation function gives higher accuracy for SKBPF circuit.

4.5.1 Sigmoid Activation Function

"S" shape (sigmoid curve) function called sigmoid function is the most commonly used activation function. It is real-valued and differentiable, having either a non-positive or non-negative first derivative which is bell shaped. There are two types of sigmoid functions, they are logistic and hyperbolic tangential function. The value of logistic function ranges from 0 to 1 where the output value is either binary or varies from 0 to 1. Sometimes, the logistic sigmoid function is suitable for particular problems only. So tangential function is used, its value ranges from -1 to +1. The expression for sigmoid activation is given by equation 4.5 and it shown in figure 4.4.

$$f(x) = \frac{1}{1+e^{-\beta x}} \tag{4.5}$$

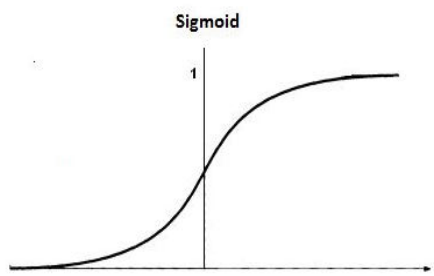

Figure 4.4 Sigmoid Activation Function

4.5.2 Triangular Basis Activation Function

Triangular basis (tribas) activation function is a neural network transfer function. It calculates the layers output from the net input. The expression for tribas function in equation 4.6 and it is showed in figure 4.5.

$$f(x) = tribas(x) \tag{4.6}$$

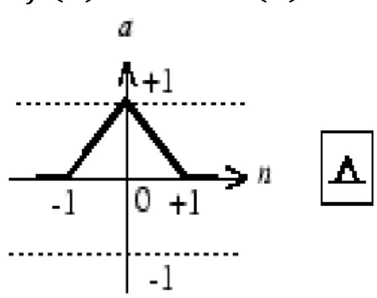

Figure 4.5 Triangular Basis Activation Function

4.6 SIMULATION RESULTS

4.6.1 Performance of SVF Circuit

Detection and classification of faults is done by using MLELM algorithm in MATLAB tool version 2013a.The single fault dataset of SVF benchmark circuit is taken as the first dataset, here out of 1853 samples, 1403 samples are used for training and 450 samples are used for testing the network. These testing and training samples are given as input to MLELM for fault classification.

By varying the number of layers in MLELM, testing time, training time, testing accuracy and training accuracy are calculated and listed in table 4.1. Here, hidden nodes are assigned as 20 and the layers of MLELM algorithm are varied within the range of 2 to 10 for determining both training and testing accuracy.

Table 4.1 Accuracy calculation for different hidden layers

Layers	**Training Accuracy (in %)**	**Training Time (in sec)**	**Testing Accuracy (in%)**	**Testing Time (in sec)**
2	84.96	0.0107	67.78	0.0015
3	82.32	0.0129	76.67	0.0034
4	83.61	0.0413	80.67	0.0073
5	**83.46**	0.0245	**83.11**	0.0023
6	82.04	0.0131	76.00	0.0018
8	83.82	0.0152	75.56	0.0018
10	84.03	0.0173	73.67	0.0021

The values of training and testing accuracy increases gradually until layer 5 and fluctuate beyond layer 5. Therefore, after layer 5, training accuracy starts increasing and testing accuracy starts decreasing. So, when the hidden layers are assigned as 5, we get better accuracy for both training and testing.Training and testing accuracy for MLELM are calculated for different number of hidden nodes and it is shown in figure 4.6. Here the number of layers is assigned with 5 and the hidden nodes are varied to measure accuracy.

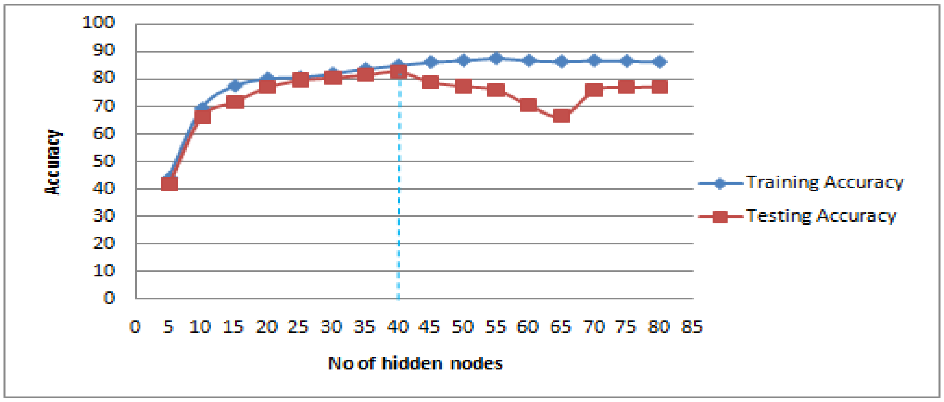

Figure 4.6 Training and Testing accuracy vs number of hidden nodes

The training accuracy increases as the number of node increases and remains stable as the number of nodes increased beyond 40. The testing accuracy increases as the number of node increases and starts decreasing as the number of nodes increased beyond 40. Therefore 5 hidden layers and 40 Hidden nodes are chosen as input for MLELM for high accuracy in both training and testing.

The MLELM algorithm is executed by varying the activation functions. The results for single fault for the varied activation function are shown in figure 4.7

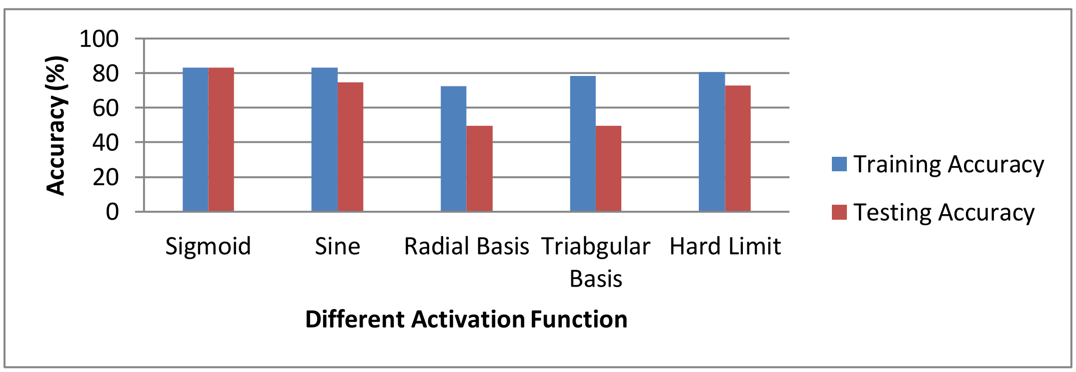

Figure 4.7 Accuracy calculation for different activation function

The results of the varied activation function shows that the sigmoid activation function gives the higher training and testing accuracy compared to the other activation functions. The hidden layer and hidden node numbers for sigmoid activation is varied and the 5 layers and 40 hidden nodes gives the higher accuracy compared to the others.

Hence, 5 layers, 40 hidden nodes and sigmoid activation function are chosen as the standard input data for attaining higher classification accuracy. By assigning this, training and testing performance for SVF single fault using MLELM algorithm is analyzed using confusion matrices are tabulated in table 4.2

Table 4.2 Performance calculation for testing dataset

Fault Index	TP	TN	FP	FN	Accuracy	Error	Precision	Sensitivity	Specificity	Fmeasure	FPR	FNR	PPV	NPV
R_1±50%	46	393	7	4	0.975556	0.024444	0.867925	0.92	0.9825	0.983204	0.0175	0.08	0.867925	0.989924
R_2±50%	39	382	18	11	0.935556	0.064444	0.684211	0.78	0.955	0.728972	0.045	0.22	0.684211	0.97201
R_3±50%	39	394	6	11	0.962222	0.037778	0.866667	0.78	0.985	0.821053	0.015	0.22	0.866667	0.97284
R_4±50%	42	378	22	8	0.933333	0.066667	0.65625	0.84	0.945	0.736842	0.055	0.16	0.65625	0.979275
R_5±50%	28	394	6	22	0.937778	0.062222	0.823529	0.56	0.985	0.666667	0.015	0.44	0.823529	0.947115
R_6±50%	38	400	0	12	0.973333	0.026667	1	0.76	1	0.863636	0	0.24	1	0.970874
R_7±50%	41	393	7	9	0.964444	0.035556	0.854167	0.82	0.9825	0.836735	0.0175	0.18	0.854167	0.877612
C_1±50%	48	393	7	2	0.98	0.022222	0.872727	0.96	0.9825	0.914286	0.0175	0.04	0.872727	0.994937
C_2±50%	48	392	8	2	0.977778	0.022222	0.857143	0.96	0.98	0.90566	0.02	0.04	0.857143	0.994924
Avg	41	391	9	9	0.96	0.04	0.831402	0.82	0.9775	0.818562	0.0225	0.18	0.831402	0.977723

Testing performance of MLELM algorithm is listed in table 4.2. As an average out of 450 samples, True positive detected during training is 41, leading to false negative of 9 and True negative identified is 391, leading to a false positive of 9.

The results show that the classifier has higher accuracy for fault index 8. The same fault index has higher sensitivity, Fmeasure, PPV and NPV. The fault index 6 has high precision which shows that the performance or classification reproducibility is obtained. Therefore the overall testing accuracy is 96%. The figure 4.8 shows the testing and training accuracy for individual fault indexes.

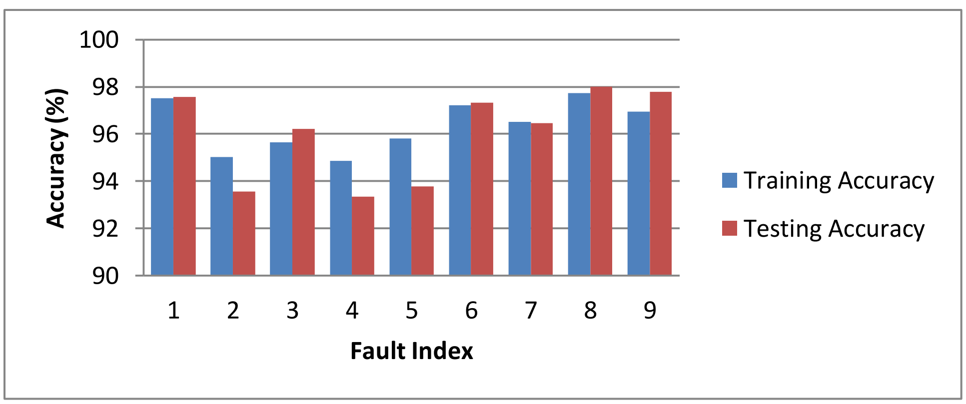

Figure 4.8 Training and Testing accuracy for individual fault indexes

4.6.2 Performance of SKBPF Circuit

The single fault dataset of SKBPF benchmark circuit is taken as the second dataset, here out of 1443 samples, 1093 samples are used for training and 350 samples are used for testing the network. These testing and training samples are given as input to MLELM for fault classification.

Testing time, training time, testing accuracy and training accuracy are calculated by varying the number of layers in MLELM and tabulated in table 4.3. Here, the performance of the SKBPF circuit with single fault is analysed through MLELM algorithm by assigning the hidden nodes as 40 and changing the number of layers for determining both training and testing accuracy.

Table 4.3 Accuracy calculation for different hidden layers

Layers	Training Accuracy (in %)	Training Time (in sec)	Testing Accuracy (in%)	Testing Time (in sec)
2	84.53	0.0114	74.22	0.0015
3	83.25	0.0162	74.67	0.0036
4	82.32	0.0129	76.58	0.0034
5	**81.25**	0.0167	**81.11**	0.0062
6	83.25	0.0184	77.11	0.0032
8	86.24	0.0375	71.33	0.0035
10	87.10	0.0425	63.78	0.0087

The values of training and testing accuracy increases gradually until layer 5 and fluctuate beyond layer 5. Therefore, when the hidden layers are assigned as 5, we get training accuracy as 81.24% at 0.0167 sec and testing accuracy as 81.11% at 0.0062 sec.

By assigning the number of layers as 5 and the hidden nodes are varied to measure both training and testing accuracy of SKBPF circuit by MLELM algorithm and it is shown in figure 4.9.

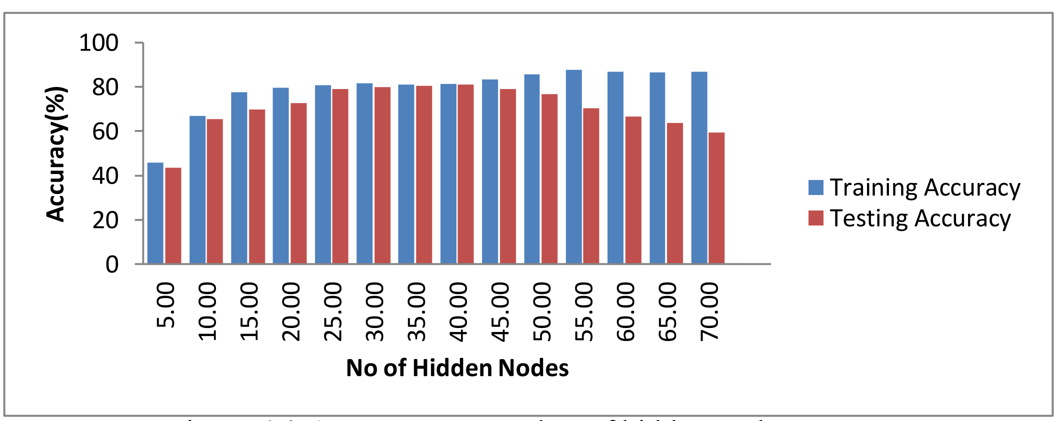

Figure 4.9 Accuracy vs Number of hidden nodes

The training accuracy increases as the number of node increases and become stable as the number of nodes increased beyond 40. The testing accuracy increases as the number of node increases and starts decreasing as the number of nodes increased beyond 40. Therefore 5 hidden layers and 40 Hidden nodes are chosen as input for MLELM for high accuracy in both training and testing. Then the behavior of SKBPF benchmark circuit is examined by changing the different activation functions available in the algorithm and by changing the hidden node numbers. The result of single fault for the varied activation function is summarized in table 4.4.

Table 4.4 Accuracy calculation for varied activation function

Activation Function	Training Accuracy (in %)	Training Time (in sec)	Testing Accuracy (in%)	Testing Time (in sec)
Sigmoid	78.77	0.0294	66.86	0.0078
Sine	75.39	0.088	68.87	0.0016
Radial Basis	70.36	0.0978	55.14	0.0064
Triangular Basis	**81.25**	0.0167	**81.11**	0.0062
Hard Limit	75.67	0.0443	68.57	0.0058

The tabulated results indicate that triangular basis function shows higher accuracy for training and testing compared to other activation function. The performance of all the activation function is analyzed with 40 hidden nodes because it gives reasonable training and testing accuracy in minimum time.

Therefore, 5 layers, 40 hidden nodes and triangular basis activation function are chosen as the input data for attaining better classification accuracy. By assigning this, training and testing performance for SKBPF circuit is analyzed using confusion matrices and are tabulated in table 4.5

Table 4.5 Testing Performance of SKBPF circuit

Fault Index	TP	TN	FP	FN	Accuracy	Error	Precision	Sensitivity	Specificity	Fmeasure	FPR	FNR	PPV	NPV
R_1±50%	50	293	7	0	0.98	0.02	0.877193	1	0.976667	0.934579	0.023333	0	0.877193	1
R_2±50%	49	293	1	1	0.994186	0.005814	0.98	0.98	0.996599	0.98	0.003401	0.02	0.98	0.996599
R_3±50%	44	299	1	6	0.98	0.02	0.977778	0.88	0.996667	0.926316	0.003333	0.12	0.977778	0.980328
R_4±50%	47	282	18	3	0.94	0.06	0.723077	0.94	0.94	0.817391	0.06	0.06	0.723077	0.989474
R_5±50%	36	299	1	14	0.957143	0.042857	0.972973	0.72	0.996667	0.827586	0.003333	0.28	0.972973	0.955272
C_1±50%	20	277	23	30	0.848571	0.151429	0.465116	0.4	0.923333	0.430108	0.076667	0.6	0.465116	0.90228
C_2±50%	24	271	29	26	0.842857	0.157143	0.45283	0.48	0.903333	0.466019	0.096667	0.52	0.45283	0.912458
Avg	38	288	12	12	0.93468	0.06532	0.778424	0.771429	0.961895	0.768857	0.038105	0.228571	0.778424	0.962344

Testing performance of MLELM algorithm is listed in table 4.5. As an average out of 350 samples, True positive detected during training is 38, leading to false negative of 12 and True negative identified is 288, leading to a false positive of 12.

The results show that the classifier has higher accuracy for fault index 2 and has least accuracy for fault index 6. The fault index 2 has higher precision, sensitivity, Fmeasure, PPV and NPV. Therefore the overall testing accuracy is 93.468%. The figure 4.10 shows the testing and training accuracy for individual fault indexes.

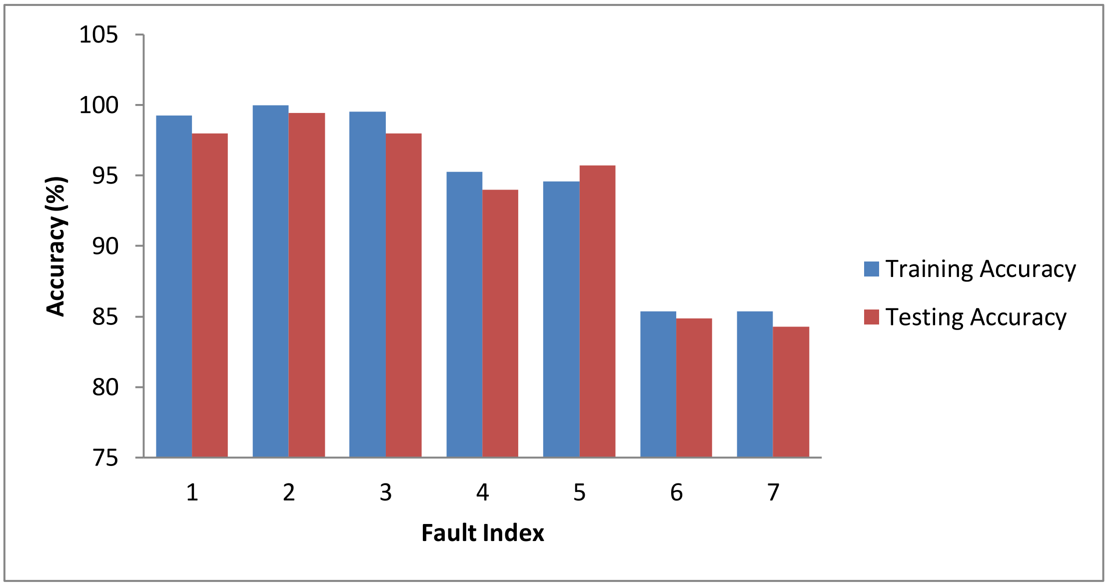

Figure 4.10 Training and Testing accuracy for individual fault indexes

CHAPTER 5

KERNEL EXTREME LEARNING MACHINE WITH PSO

5.1 INTRODUCTION TO KELM

The Kernel methods are new class of algorithms which reduces the cost function. It is extensively used for regression and classification problems due to high generalization performance and mathematical rigor of the field. Generally, KELM is a single hidden-layer feed forward neural network. Compared with ELM algorithm, the feature mapping of hidden layer need not be known and the number of hidden neurons is not chosen in the KELM. The stability and generalization performance of the ELM algorithm is determined by these input parameters. KELM improves the stability and performance by eliminating feature mapping of hidden neurons and with the group of activation functions. Furthermore, the KELM learning algorithm has similar or better generalization performance and is more stable compared with ELM and it is faster than SVM [13]. KELM has kernel parameters which are optimized and it improves the generalization performance compared to ELM.

KELM is widely used for Spectral-Spatial Classification of Hyperspectral Image, Fast detection of impact location, Illumination correction of dyeing products, Representation Learning, Modeling and optimization of biodiesel engine performance and prediction of robot execution failures. In this work, to avoid the application of time consuming algorithm for the determination of the ELM space dimensionality and performance, KELM is used for the applications of fault classification in analog circuits.

5.1.1 Mathematical Model

The kernel version of ELM is similar to ELM in generating input weights randomly, the only difference between ELM and KELM is that the hidden layer output is not calculated they are inherently encoded called as ELM kernel matrix and they are defined as $K = \varphi^T\varphi$ where φ represents the training data representations in ELM space .

In KELM the kernel matrix defined on the input data determines the ELM space. The kernel version of ELM is obtained from the output function of ELM by replacing the hidden layer output matrix by kernel matrix [25]. The N arbitrary distinct samples $\{(x_i, t_i) \mid x_i \in R^n$ is the input vector, $t_i \in R^m$ is the corresponding output vector, i=1,2,......,N$\}$ then the output function in ELM is stated as

$$f_L(x) = \sum_{i=1}^{L} \beta_i h_i(x) = h(x)\beta \tag{5.1}$$

where L is the number of hidden nodes and $\beta = [\beta_1, \beta_2, \dots, \beta_l]$ is the vector of output weights and $h(x) = [h_1(x), h_2(x), \dots, h_L(x)]$ is the output vector of the hidden layer with respect to the input x and it maps the data from input space to the ELM feature space.

For reducing the training error and to improve the performance, the output weights and training error should be minimized at the same time, that is

$$\text{Minimize: } \|\beta\|, \|H\beta - T\| \tag{5.2}$$

where $\|\beta\|$ is the output weight and the $\|H\beta - T\|$ training error.

According to Karush-Kuhn-Tuker theorem (KKT), the output weight β can be written as

$$\beta = H^T \left(\frac{1}{C} + HH^T\right)^{-1} T \tag{5.3}$$

where H is the hidden layer output matrix, C is the regularization coefficient and T is the expected output matrix of the input samples.

The corresponding output function of the ELM is:

$$f(x) = h(x)\beta = h(x)H^T \left(\frac{1}{C} + HH^T\right)^{-1} T \tag{5.4}$$

In ELM, a feature mapping h(x) is usually known to users. If the feature mapping of h(x) is unknown then the kernel matrix is used to determine the ELM feature space which is defined based on the Mercer's conditions is defined as

$$M = HH^T: m_{ij} = h(x_i)h(x_j) = k(x_i, x_j) \tag{5.5}$$

where $k(x_i, x_j)$, is a kernel function. Now the output function f(x) of KELM can be written as:

$$f(x) = [k(x, x_1), \dots, k(x, x_N)] \left(\frac{1}{C} + M)\right)^{-1} T \tag{5.6}$$

where $M = HH^T$ is the kernel function of hidden neurons of single hidden layer feed-forward neural networks.

From the above mentioned equations, it can be obtained that by introducing the kernel function into ELM algorithm gets the least-squares optimal solution and more stable than basic ELM. In addition, compared with the traditional methods, KELM has the advantage of multi-output, which can reduce training time greatly.

There are 4 different kernel functions available in KELM for the computation of kernel matrix. The four kernel functions are RBF kernel, linear kernel, polynomial kernel and wavelet kernel which satisfy the Mercer condition available from the existing literature. It is observed that different types of kernel activation functions have great influence on the performance of KELM. In this kernel-based ELM, the hidden layer feature mapping h(x) need not to be known to the user and the number of hidden nodes L need not be specified. Among the four kernels, RBF kernel is chosen as standard kernel function for the applications because of the higher performance in terms of accuracies in lesser time compared to the other kernels. RBF kernels can be randomly generated instead of being tuned. This allows the centers and impact widths of RBF kernels to randomly generate and analytically calculate the output weights instead of iterative tuning. The kernel function of ELM can be any nonlinear bounded integral function which is almost continuous anywhere.

5.2 PARTICLE SWARM OPTIMIZATION

5.2.1 Introduction of PSO

In KELM learning algorithm, kernel parameter should be chosen properly for improving the generalization performance of neural networks. In [13], the parameters are tried in a wide range which is time consuming. Although, how to choose the optimized value of the kernel parameter has not been resolved yet. Therefore in this work, an optimization technique is proposed to the KELM algorithm for choosing the optimal parameter of kernel function.

There are two groups of optimistic algorithm, they are deterministic and stochastic algorithms. Mostly, deterministic algorithms are effective for unimodal functions which have one global optimum and need gradient information. But, stochastic algorithms do not need any properties of the objective function. Therefore, more attention has been paid to stochastic algorithms recently which are Genetic Algorithm, Differential Evolution, Particle Swarm Optimization, Ant Colony Optimization and Artificial Bee Colony.

Among these stochastic algorithms, PSO is a relatively new, modern, and powerful optimization method developed by Eberhart and Kennedy. PSO is inspired by the behavior of a flock of birds. Assume a scenario: a group of birds are searching for the food in an area randomly. There is only one piece of food in searched area. They do not know where the food is. But they know about their position and how far the food is. Now the best strategy is to follow the bird which is nearer to the food. PSO learns about the scenario and using it for solving the optimization problems. Here, each solution in the search space is like a "bird" which is known as particle. Every particle has its own fitness values evaluated by the fitness function which is to be optimized. The particles constitute a swarm, fly through the search space by following the particles for finding the best solution.

5.2.2 Basic Algorithm of PSO

Initially, PSO begins with a set of random particles and then explore the optimal solution by updating each generation. In all iteration, each and every particle is updated by two best solutions. The first one is called pbest for an individual particle and the second one is gbest for best in the whole population. The main concept of PSO lies in moving every particle towards its pbest and gbest values in their path. If the dimension of PSO search space is D and the population size is Ń. Then, x_i^d and v_i^k are the current position and velocity of the ith particle at iteration t in the search space.

The core of PSO algorithm is updating the formula of each particle, which is represented as follows. Equation 5.7 calculates the new velocity for each particle based on their previous velocity. Equation 5.8 represents the calculation of new position of the particles in the next iteration.

$$v_i^k(t+1) = \omega . v_i^k(t) + c_1 . rand(\,)\left(p_i^k(t) - x_i^k(t)\right) + c_2 . rand()\left(g_i^k(t) - x_i^t(t)\right) \qquad (5.7)$$

$$x_i^k(t+1) = x_i^k(t) + v_i^k(t+1)$$

$$1 \le i \le Ń, 1 \le k \le D \qquad (5.8)$$

where p_i^k is the best position of ith particle has achieved so far in the search space, then g_i^k denotes the global best position in the entire swarm until now.

This algorithm uses a few parameters that can affect its performance. The basic PSO parameters are number of iterations, swarm size or number of particles, inertia weight and acceleration constants. Population size or swarm size is the number of particles in the swarm and it is problem dependent. From the number of factual studies, it is shown that most of the PSO implementations use the population size n, belongs to [20, 60]. Here, in this project population size assigned as 60.

The numbers of iterations are assigned to obtain a good result and it is problem independent. The small number of iteration can stop the search process early, while large number of iteration has the effect of unwanted computational complexity and more time. Here, in this work number of iteration is assigned as 50. ω is the inertia weight ,it is first introduced by Shi and Eberhart. Its function is to balance global and local exploration. The researchers have found that the best performance is obtained by inertia weight in the range $\omega < 1$. Here, inertia weight is assigned as 0.9.

Two parameters, c_1 and c_2 are responsible for the behavior of the swarm. They are called acceleration constants, because they are the reason for the control of magnitude towards the particles personal best and the global best. c_1 controls the cognitive aspect which is the adjustment towards the personal best and c_2 controls the social aspect which is the adjustment towards the global best. For the canonical parameter settings, assign c_1 and c_2 as 2.0 as the standard value. The prominent reason is that it will adjust the search cover all surrounding regions which is centered at pbest and gbest. Then, rand() is the random number between 0 and 1. The process for implementing the PSO algorithm is shown in the figure 5.1

The following steps are considered for the PSO algorithm.

Step 1: Initialize the particles which is population size based on kernel function with random position and velocity for each particle

Step2: Evaluate the fitness function of each particle

Step3: If the fitness value (p) is better than pbest, assign pbest = p

Step4: Then set best value of pbest as the gbest

Step5: Based on the equation 5.7 and 5.8, velocity and position of the particle is updated and repeated until the maximum iteration time is satisfied

Step 6: The optimal parameter (gbest) of kernel function can be determined

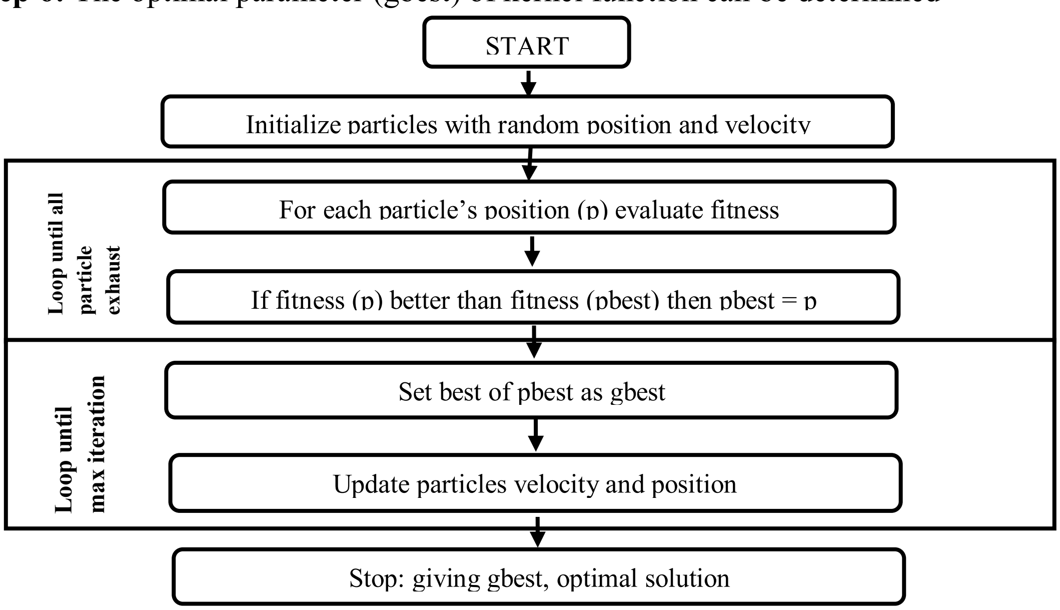

Figure 5.1 Flowchart for PSO Algorithm

5.3 FAULT CLASSIFICATION USING KELM WITH PSO

The flow diagram of the proposed KELM with PSO algorithm is shown in figure 5.2. The data sets of SVF and SKBPF benchmark circuits are taken as the input to the algorithm. 75% data are chosen for training and 25% data is chosen for testing. The testing and training data are chosen randomly. KELM has input parameters such as training data, testing data, regularization coefficient and kernel function. The output from the algorithm implementation is the correct detection of fault index as per the target defined.

The testing and training samples are given as input to the KELM algorithm. Initialize the regularization coefficient and Kernel function. Determine the optimal kernel parameter for better generalization performance using optimization technique like PSO. Then compute the kernel matrix using kernel parameter and kernel function. Based on the kernel matrix and target values, output weights are computed. Compute the expected target from output weight and kernel matrix. Finally compute the classification accuracy based on the actual and expected target. The performance analysis of the algorithm is measured by performance metrics determined from the confusion matrix.

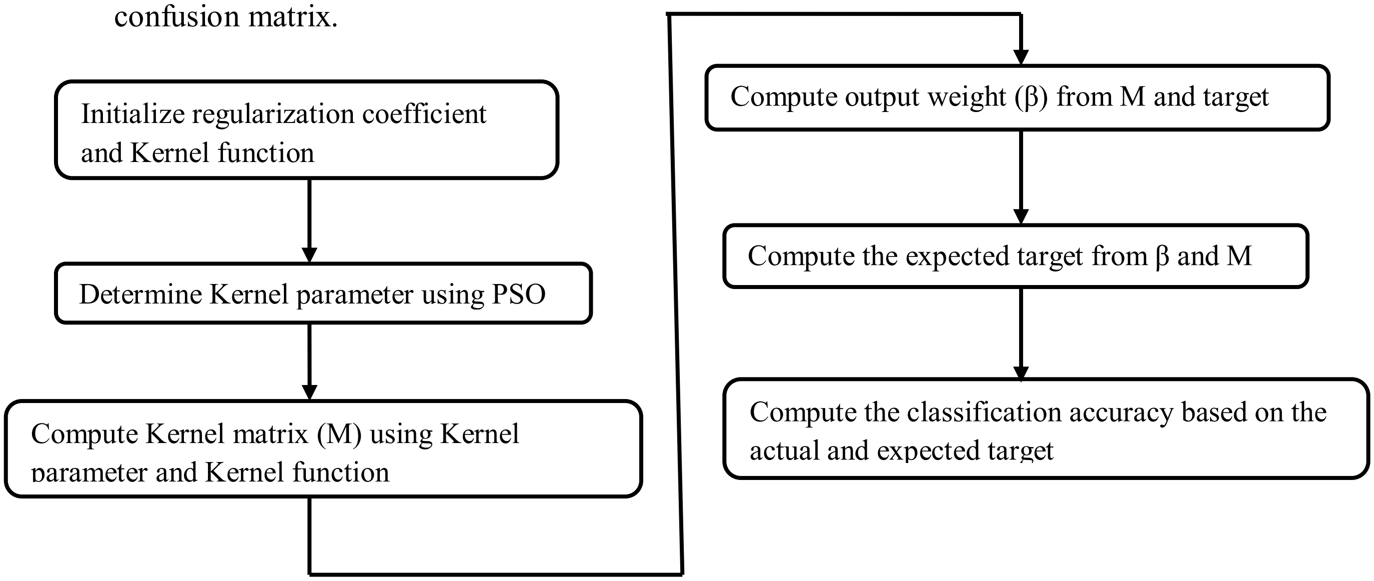

Figure 5.2 Flowchart for KELM with PSO

5.4 SIMULATION RESULTS

5.4.1 Performance of SVF circuit

The KELM with PSO algorithm for SVF single fault is executed. The single fault dataset of SVF benchmark circuit is taken as the first dataset, here out of 1853 samples, 1403 samples are used for training and 450 samples are used for testing the network.

By varying the kernel type in KELM, testing time, training time, testing accuracy and training accuracy are calculated and listed in Table 5.1. The KELM algorithm has 4 different types of kernel, they are RBF kernel, Linear Kernel, Poly kernel and wave kernel. The kernel parameter is given in scalar form for RBF and linear kernel. The kernel parameter is given as vector for polynomial and wavelet kernel.

Table 5.1 Performance of various kernels

Kernel Type	Kernel Parameter	Training Accuracy (in %)	Training Time (in sec)	Testing Accuracy (in %)	Testing Time (in sec)
RBF Kernel	**0.0357**	**100**	0.2661	**94.89**	0.0601
Linear Kernel	0.0182	26.09	0.3471	22.00	0.0274
Poly Kernel	[0.0395 30]	84.25	0.5845	68.00	0.1089
Wave kernel	[0.002 0.1300 0.5284]	95.26	1.7139	86.52	0.0869

The tabulated results show that for RBF Kernel, the optimized kernel parameter is 0.0357. For Linear Kernel, the optimized kernel parameter is 0.0182. For Polynomial Kernel, the optimized kernel parameter is [0.0395 30]. For Wavelet Kernel, the optimized kernel parameter is [0.002 0. 1300 0.5284].

Among the entire kernels, RBF kernel with 0.0357 as kernel parameter gives the best result in terms of time and accuracy. Figure 5.3 shows the training and testing accuracy performance for different Kernel function.

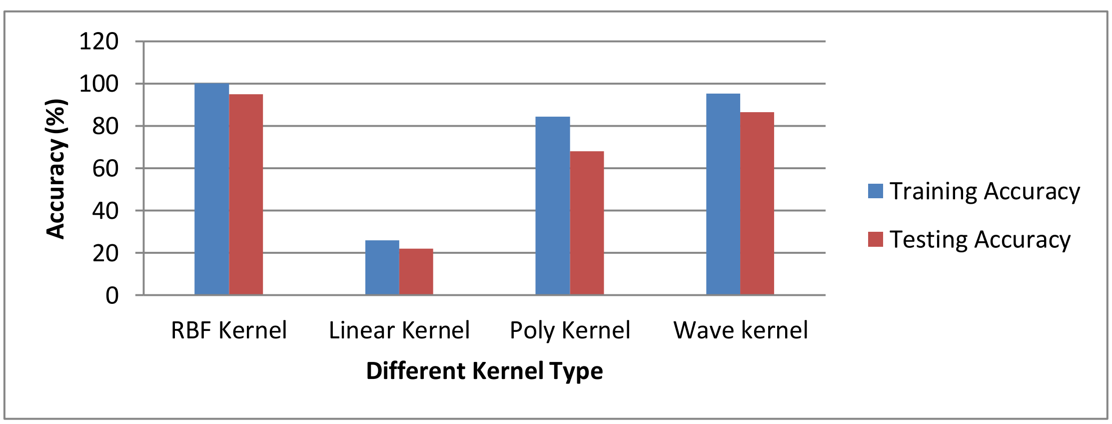

Figure 5.3 Accuracy performance for different Kernel function

Hence, RBF kernel with 0.0357 as kernel parameter is chosen as the optimum value for attaining higher classification accuracy. By assigning this, training and testing performance for

SVF single fault using KELM with PSO algorithm is analyzed using confusion matrices and it is tabulated in table 5.2

Table 5.2 Testing data result for SVF circuit

Fault Index	TP	TN	FP	FN	Accuracy	Error	Precision	Sensitivity	Specificity	Fmeasure	FPR	FNR	PPV	NPV
$R_1 \pm 50\%$	45	400	0	5	0.988889	0.011111	1	0.9	1	0.947368	0	0.1	1	0.987654
$R_2 \pm 50\%$	46	398	2	4	0.986667	0.013333	0.958333	0.92	0.995	0.938776	0.005	0.08	0.958333	0.99005
$R_3 \pm 50\%$	48	387	13	2	0.966667	0.033333	0.786885	0.96	0.9675	0.864865	0.0325	0.04	0.786885	0.994859
$R_4 \pm 50\%$	48	398	2	2	0.991111	0.008889	0.96	0.96	0.995	0.96	0.005	0.04	0.96	0.995
$R_5 \pm 50\%$	46	400	0	4	0.991111	0.008889	1	0.92	1	0.958333	0	0.08	1	0.990099
$R_6 \pm 50\%$	50	398	2	0	0.995556	0.004444	0.961538	1	0.995	0.980392	0.005	0	0.961538	1
$R_7 \pm 50\%$	48	400	0	2	0.995556	0.004444	1	0.96	1	0.979592	0	0.04	1	0.995025
$C_1 \pm 50\%$	49	397	3	1	0.991111	0.008889	0.942308	0.98	0.9925	0.960784	0.0075	0.02	0.942308	0.997487
$C_2 \pm 50\%$	47	399	1	3	0.991111	0.008889	0.979167	0.94	0.9975	0.959184	0.0025	0.06	0.979167	0.992537
Avg	47	397	3	3	0.988642	0.011358	0.954248	0.948889	0.993611	0.949922	0.006389	0.051111	0.954248	0.993635

Testing performance of KELM with PSO algorithm is listed in table 5.2. As an average out of 450 samples, True positive detected during training is 47, leading to false negative of 3 and True negative identified is 397, leading to a false positive of 3.

The results show that the classifier has higher accuracy when the fault index 6 and 7. Similarly the table results for testing shows that all the fault indexes are classified with minimum error. The classifier has least accuracy for fault index 3 with minimum precision and PPV. Therefore the overall testing accuracy is 98.86%. The figure 5.4 shows the testing and training accuracy for individual fault indexes.

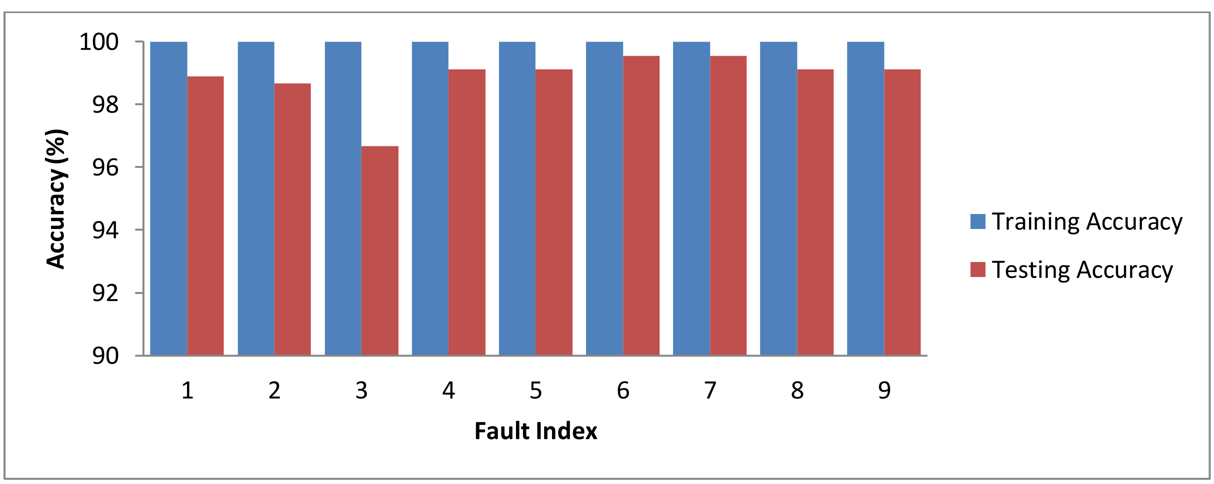

Figure 5.4 Accuracy calculation for individual fault indexes

5.4.2 Performance of SKBPF Circuit

Detection and classification of faults is done by using KELM with PSO algorithm in MATLAB tool version 2013a.The single fault dataset of SKBPF benchmark circuit is taken as the second dataset, here out of 1443 samples, 1093 samples are used for training and 350 samples are used for testing the network.

By varying the kernel type in KELM, testing time, training time, testing accuracy and training accuracy are calculated and listed in table 5.3. The KELM algorithm has 4 different types of kernel, they are RBF kernel and Linear Kernel has kernel parameter in scalar form and Poly kernel and wave kernel has kernel parameter in vector form.

Table 5.3 Performance of various kernels

Kernel Type	Kernel Parameter	Training Accuracy (in %)	Training Time (in sec)	Testing Accuracy (in %)	Testing Time (in sec)
RBF Kernel	**3.5312×10^{-5}**	**98.99**	0.1582	**85.14**	0.0552
Linear Kernel	1.6715×10^{-4}	37.24	0.2972	37.43	0.0298
Poly Kernel	[0.0006 0.9379]	68.00	1.3029	52.29	0.0620
Wave kernel	[0.0351 0.1845 0.5464]	92.37	1.7139	64.52	0.0869

The tabulated results show that for RBF Kernel, the optimized kernel parameter is 3.5312×10^{-5}. For Linear Kernel, the optimized kernel parameter is 1.6715×10^{-4}. For Polynomial Kernel, the optimized kernel parameter is [0.0006 0.9379]. For Wavelet Kernel, the optimized kernel parameter is [0.0351 0.1845 0.5464].

Among the entire kernels RBF kernel with 3.5312×10^{-5} as kernel parameter gives the best result in terms of training and testing accuracy and figure 5.5 shows the accuracy performance.

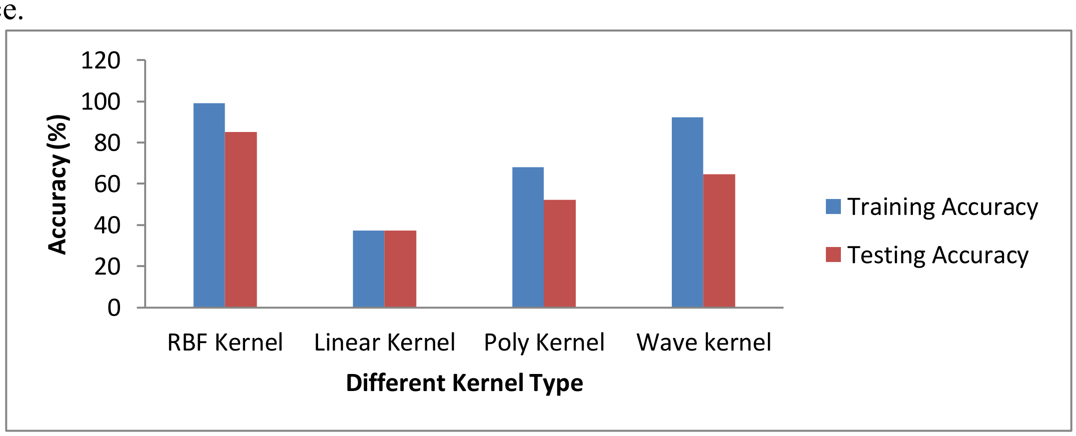

Figure 5.5 Accuracy comparison for different Kernel type

Hence, RBF kernel with 3.5312 $\times 10^{-5}$ as kernel parameter is chosen as the standard input data for attaining better classification accuracy. By assigning this, training and testing performance for SKBPF single fault using KELM with PSO algorithm is analyzed using confusion matrices are tabulated in table 5.4

Table 5.4 Testing data result for SKBPF circuit

Fault Index	TP	TN	FP	FN	Accuracy	Error	Precision	Sensitivity	Specificity	Fmeasure	FPR	FNR	PPV	NPV
R_1±50%	50	288	12	0	0.965714	0.034286	0.806452	1	0.96	0.892857	0.04	0	0.806452	1
R_2±50%	50	288	0	0	1	0	1	1	1	1	0	0	1	1
R_3±50%	47	300	0	3	0.991429	0.008571	1	0.94	1	0.969072	0	0.06	1	0.990099
R_4±50%	47	300	0	3	0.991429	0.008571	1	0.94	1	0.969072	0	0.06	1	0.990099
R_5±50%	50	300	0	0	1	0	1	1	1	1	0	0	1	1
R_6±50%	33	276	24	17	0.882857	0.117143	0.578947	0.66	0.92	0.616822	0.08	0.34	0.578947	0.94198
R_7±50%	21	284	16	29	0.871429	0.128571	0.567568	0.42	0.946667	0.482759	0.053333	0.58	0.567568	0.907348
Avg	42	292	8	8	0.957551	0.042449	0.850424	0.851429	0.975238	0.847226	0.024762	0.148571	0.850424	0.975647

Testing performance for SKBPF circuit is listed in table 5.4. As an average out of 350 samples, True positive detected during training is 42, leading to false negative of 8 and True negative identified is 292, leading to a false positive of 8. The results show that the classifier has higher accuracy when the fault index is 2 and 5. Similarly the table results for testing shows that all the fault indexes are classified with minimum error. The fault index 6 and 7 has minimum accuracy compared with other fault indexes which indicates the performance reproducibility of these two fault indexes are very less. Therefore the overall testing accuracy is 95.75%. The figure 5.6 shows the testing and training accuracy for individual fault indexes

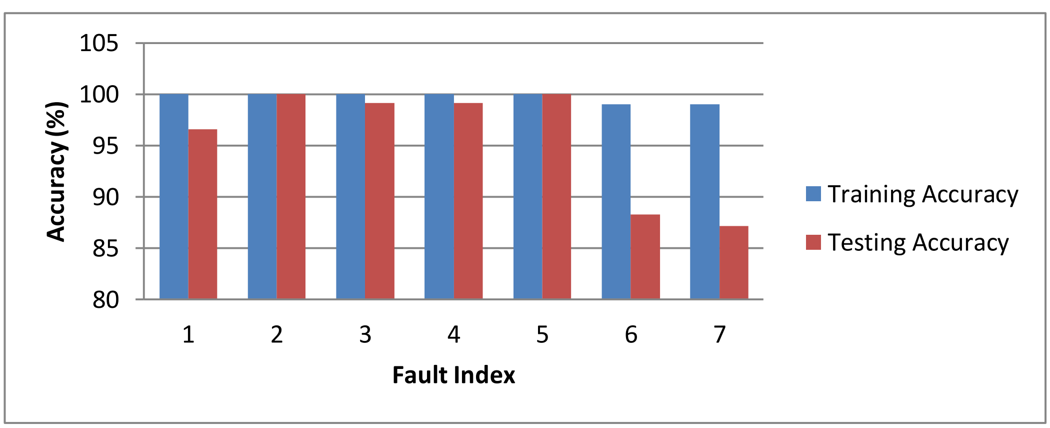

Figure 5.6 Accuracy calculation for individual fault indexes

5.5 PERFORMANCE COMPARISON OF METHODOLOGIES

The algorithms MLELM and KELM with PSO are proposed in this project are compared with ELM. ELM is the basic algorithm used to train the network with the random generated input weights, this algorithm gives better performance compared to traditional algorithms but due to the inability to classify faults for multilayer by Extreme Learning Machine, we use the MLELM approach to classify the faults. The next proposed algorithm is the KELM with PSO, here kernel parameter of kernel function is optimized to improve generalization performance. This algorithm shows higher performance compared to the other two proposed algorithms because it reduces the cost function and it uses only the kernel matrix and the training sample for the computation of output weight and classification unlike ELM and MLELM uses, input weight, bias, hidden neurons for the output weight computation and classification. The data sets namely SVF with single faults and SKBPF with single faults are given as input for evaluating the performance of all the proposed algorithms. The training and testing results of all the algorithms are compared separately for each datasets.

5.5.1 Performance Comparison For Testing Dataset

The table 5.5 shows the testing results comparison of all the algorithms for datasets of SVF and SKBPF circuits. The table 5.5 results show that KELM with PSO algorithm has higher testing classification accuracy of 98.86 % with minimum error compared to the other two other two algorithms for SVF circuit. KELM with PSO has 95.75% training classification accuracy with minimum error compared to the other two algorithms for SKBPF circuit.

Table 5.5 Testing Results Comparison with Accuracy and error

Algorithm	**Accuracy (%)**		**Error (%)**	
	SVF	SKBPF	SVF	SKBPF
ELM	93.9753	90.535	6.0247	9.465
MLELM	96	93.468	4	6.532
KELM with PSO	98.8642	95.7551	1.1358	4.2449

Figure 5.7 shows the comparison of precision, sensitivity and specificity for all the three algorithm of both the circuits. KELM has high precision, sensitivity and specificity.

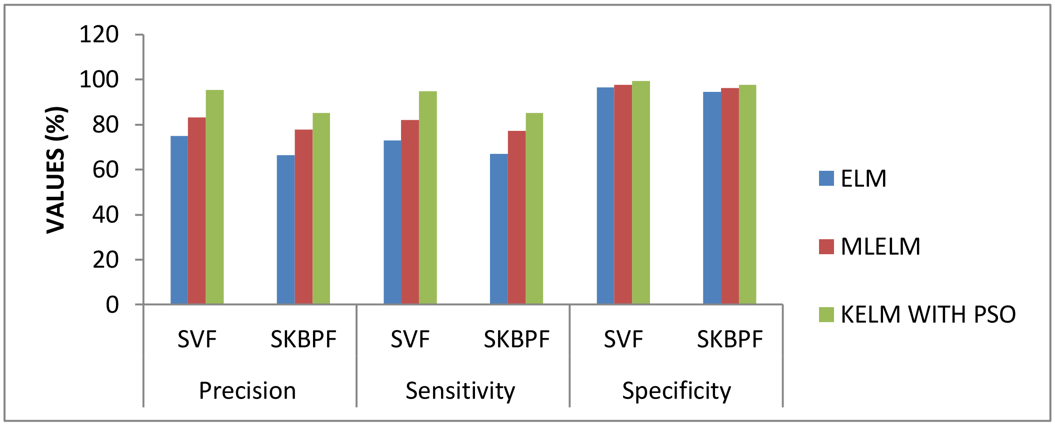

Figure 5.7 Testing Results Comparison with Precision, Sensitivity and Specificity

For testing, comparison of Fmeasure, FPR and FNR for all the three algorithm of both SVF and SKBPF circuit is shown in table 5.6. The result indicates that KELM with PSO has the high Fmeasure, FPR and FNR compared to other algorithms.

Table 5.6 Training Results Comparison with Fmeasure, FPR and FNR

Algorithm	**Fmeasure (%)**		**FPR(%)**		**FNR(%)**	
	SVF	SKBPF	SVF	SKBPF	SVF	SKBPF
ELM	72.8889	65.9467	3.3889	5.5203	27.1111	33.1429
MLELM	81.8562	76.8857	02.25	03.8105	18	22.8571
KELM with PSO	94.9922	84.7226	0.6389	02.4762	05.1111	14.8571

For testing dataset, Positive predictive value and negative predictive value for all the algorithms are compared and shown in figure 5.8. For testing dataset, KELM with PSO has the high PPV and NPV compared to other algorithms.

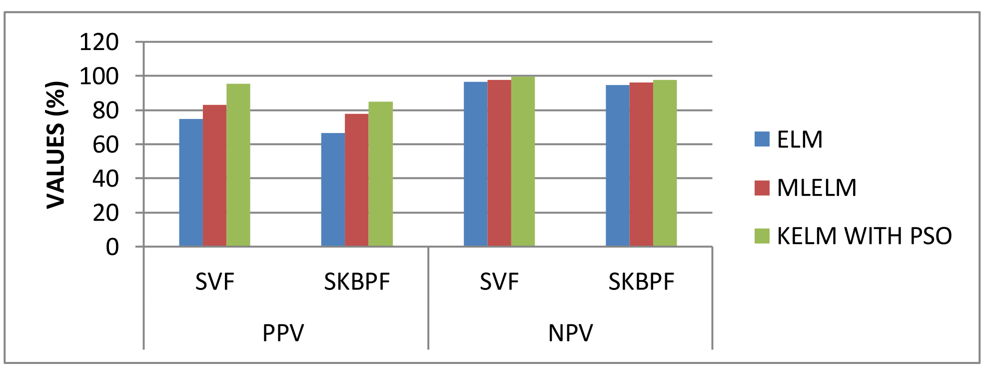

Figure 5.8 Testing Results Comparison with PPV and NPV

From the figure 5.7and 5.8 and table5.5 and 5.6, KELM with PSO has higher performance in both circuits compared to other two algorithms.

CHAPTER 6
CONCLUSION

The soft fault detection experimented using MLELM and KELM with PSO are compared with ELM algorithm. MLELM is a multiple hidden layer feed forward neural network and it avoids fine-tuning the network. So it does not spend more time during training phase. Here the number of layers and number of hidden nodes are varied and the training and testing accuracy for the state variable filter circuit and sallenkey Bandpass filter are calculated. The result shows that our proposed framework based on MLELM which consists of 5 layers and 40 neurons achieves training accuracy of 83.46% at 0.0245 sec and testing accuracy of 83.11% at 0.0023sec for SVF and training accuracy of 81.25% at 0.0167sec and testing accuracy of 81.11% at 0.0062for SKBPF.

KELM is an infinite SLFN which uses low rank decomposition matrix defined on the input data which improves the classification accuracy by optimizing the kernel parameter by PSO, further algorithm chooses hidden nodes based on the application which further improves the performance. SVF single fault detection using KELM with PSO for RBF kernel results in 100 % training accuracy and 94.89 % testing accuracy. SKBPF single fault detection using KELM with PSO for RBF kernel results in 98.99 % training accuracy and 85.14% testing accuracy.

The results obtained for the benchmark circuits are also analyzed for all the three algorithms based on the measures like accuracy, error, precision, sensitivity, specificity, Fmeasure, FNR, FPR, PPV and NPV values. The comparison shows that KELM with PSO has higher training and testing accuracy measures compared to other two algorithms, and has higher performance measures compared to other two algorithms. For SVF circuit KELM with PSO gives 100% training accuracy where as other two algorithms gives accuracy less than 90%. Hence, KELM with PSO has higher classification accuracy and better generalization performance with less computational time when compared to all the other two algorithms.

REFERENCES

[1] Stopjakova, V., Micusik, D., Benusková, L., & Margala, M. (2002). Neural networks-based parametric testing of analog IC. In *Defect and Fault Tolerance in VLSI Systems, 2002. DFT 2002. Proceedings. 17th IEEE International Symposium on* (pp. 408-416). IEEE.

[2] Deng, Y., He, Y., & Sun, Y. (2000). Fault diagnosis of analog circuits with tolerances using artificial neural networks. In *Circuits and Systems, 2000. IEEE APCCAS 2000. The 2000 IEEE Asia-Pacific Conference on* (pp. 292-295). IEEE.

[3] Balivada, A., Chen, J., & Abraham, J. (1996). Analog testing with time response parameters. *IEEE Design & Test of Computers*, *13*(2), 18-25.

[4] Epstein, B. R., Czigler, M., & Miller, S. R. (1993). Fault detection and classification in linear integrated circuits: An application of discrimination analysis and hypothesis testing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *12*(1), 102-113.

[5] Long, T., Wang, H., & Long, B. (2011). Test generation algorithm for analog systems based on support vector machine. *Signal, Image and Video Processing*, *5*(4), 527-533.

[6] Pan, C. Y., & Cheng, K. T. (1999). Test generation for linear time-invariant analog circuits. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, *46*(5), 554-564.

[7] Aminian, M., & Aminian, F. (2000). Neural-network based analog-circuit fault diagnosis using wavelet transform as preprocessor. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, *47*(2), 151-156.

[8] Lei, Z., Ligang, H., Wang, Z., & Wuchen, W. (2010, March). Applying wavelet support vector machine to analog circuit fault diagnosis. In *Education Technology and Computer Science (ETCS), 2010 Second International Workshop on* (Vol. 1, pp. 75-78). IEEE.

[9] Tang, J., Deng, C., & Huang, G. B. (2016). Extreme Learning Machine for Multilayer Perceptron. *IEEE transactions on neural networks and learning systems*, *27*(4), 809.

[10] Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2004, July). Extreme learning machine: a new learning scheme of feedforward neural networks. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on* (Vol. 2, pp. 985-990). IEEE.

[11] Wang, C., Wen, C., & Lu, Y. (2015, January). A fault diagnosis method by using extreme learning machine. In *Estimation, Detection and Information Fusion (ICEDIF), 2015 International Conference on* (pp. 318-322). IEEE.

[12] Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2006). Extreme learning machine: theory and applications. *Neurocomputing*, *70*(1), 489-501.

[13] Huang, G. B., Zhou, H., Ding, X., & Zhang, R. (2012). Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *42*(2), 513-529.

[14] Huang, G. B. (2015). What are extreme learning machines? Filling the gap between Frank Rosenblatt's dream and John von Neumann's puzzle. *Cognitive Computation*, *7*(3), 263-278.

[15] Huang, G. B., Ding, X., & Zhou, H. (2010). Optimization method based extreme learning machine for classification. *Neurocomputing*, *74*(1), 155-163.

[16] Zhou, J., Tian, S., Yang, C., & Ren, X. (2014). Test generation algorithm for fault detection of analog circuits based on extreme learning machine. *Computational intelligence and neuroscience*, *2014*, 55.

[17] Ding, S., Zhang, N., Xu, X., Guo, L., & Zhang, J. (2015). Deep extreme learning machine and its application in EEG classification. *Mathematical Problems in Engineering*, *2015*.

[18] Wang, W., Vong, C. M., Yang, Y., & Wong, P. K. (2016). Encrypted image classification based on multilayer extreme learning machine. *Multidimensional Systems and Signal Processing*, 1-15.

[19] Yang, Z. X., Wang, X. B., & Zhong, J. H. (2016). Representational learning for fault diagnosis of wind turbine equipment: a multi-layered extreme learning machines approach. *Energies*, *9*(6), 379.

[20] Duan, L., Bao, M., Miao, J., Xu, Y., & Chen, J. (2016). Classification Based on Multilayer Extreme Learning Machine for Motor Imagery Task from EEG Signals. *Procedia Computer Science*, *88*, 176-184.

[21] Kasun, L. L. C., Yang, Y., Huang, G. B., & Zhang, Z. (2016). Dimension Reduction with Extreme Learning Machine. *IEEE Transactions on Image Processing*, *25*(8), 3906-3918.

[22] Yang, Y., & Wu, Q. J. (2016). Multilayer extreme learning machine with subnetwork nodes for representation learning. *IEEE transactions on cybernetics*, *46*(11), 2570-2583.

[23] Roul, R. K., Asthana, S. R., & Kumar, G. (2016). Study on suitability and importance of multilayer extreme learning machine for classification of text data. *Soft Computing*, 1-18.

[24] Iosifidis, A., Tefas, A., & Pitas, I. (2015). On the kernel extreme learning machine classifier. *Pattern Recognition Letters*, *54*, 11-17.

[25] Li, B., Rong, X., & Li, Y. (2014). An improved kernel based extreme learning machine for robot execution failures. *The Scientific World Journal*, *2014*.

[26] Wong, C. M., Vong, C. M., Wong, P. K., & Cao, J. (2016). Kernel-Based Multilayer Extreme Learning Machines for Representation Learning. *IEEE Transactions on Neural Networks and Learning Systems*.

[27] Deng, W. Y., Ong, Y. S., & Zheng, Q. H. (2016). A fast reduced kernel extreme learning machine. *Neural Networks*, *76*, 29-38.

[28] Chen, C., Li, W., Su, H., & Liu, K. (2014). Spectral-spatial classification of hyperspectral image based on kernel extreme learning machine. *Remote Sensing*, *6*(6), 5795-5814.

[29] Zhou, Z., Xu, R., Wu, D., Zhu, Z., & Wang, H. (2016). Illumination correction of dyeing products based on Grey-Edge and kernel extreme learning machine. *Optik-International Journal for Light and Electron Optics*, *127*(19), 7978-7985.

[30] Wong, P. K., Wong, K. I., Vong, C. M., & Cheung, C. S. (2015). Modeling and optimization of biodiesel engine performance using kernel-based extreme learning machine and cuckoo search. *Renewable Energy*, *74*, 640-647.

LIST OF PUBLICATIONS

LIST OF CONFERENCES

1 Ms.R.ManjuParkavi, Ms.M.Shanthi, and Ms.M.C.Bhuvaneswari, "Multilayer Extreme learning machine based fault classification in sallenkey bandpass filter", in International conference on IoT, Data Science and Security (ICIDS'17) on 7^{th} January at PSG college of Technology, Coimbatore.

2 Ms.R.ManjuParkavi, Ms.M.Shanthi, and Ms.M.C.Bhuvaneswari, "Analog circuit fault classification using Multilayer Extreme Learning Machine", in International conference on Data Science and Engineering (ICDSE'17) 20^{th} January at PSG college of Technology, Coimbatore.

3 Ms.R.ManjuParkavi, Ms.M.Shanthi, and Ms.M.C.Bhuvaneswari, "Extreme Learning Machine Algorithm and Its Application", in International conference on Latest Trends in Engineering ,Science, Humanities and Management 26^{th} February at Indian Federation of United Nations Associations, New Delhi.

LIST OF JOURNALS

1 Ms.R.ManjuParkavi, Ms.M.Shanthi, and Ms.M.C.Bhuvaneswari, "Recent Trends in ELM and MLELM: A review " in Advances in Science, Technology and Engineering Systems Journal Vol. 2, No. 1, 69-75 (2017) .

2 Ms.R.ManjuParkavi, Ms.M.Shanthi, and Ms.M.C.Bhuvaneswari, "Extreme Learning Machine Algorithm and Its Application", in International Journal of Electrical and Electronics Engineers Vol.No.9, Issue No.01,January-June 2017.