ANALOG FAULT CLASSIFICATION IN FILTER CIRCUIT USING BINARY BAT ALGORITHM AND SVM PRINCIPLE

A PROJECT REPORT

Submitted by

SHANTHINI D
Register No: 15MAE010

in partial fulfillment 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-641049

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2017

BONAFIDE CERTIFICATE

Certified that this project report titled **"ANALOG FAULT CLASSIFICATION IN FILTER CIRCUIT USING BINARY BAT ALGORITHM AND SVM PRINCIPLE"** is the bonafide work of **SHANTHINI.D [Reg. No. 15MAE010]** who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

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 University **Register No.15MAE010** was 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 me with his valuable thoughts.

I would like to express my sincere thanks and deep sense of gratitude to my HOD, **Dr.K.Malarvizhi M.E.,Ph.D.,** and **Prof. K.Ramprakash M.E.,** for their valuable suggestions and encouragement which paved way for the successful completion of the project.

In particular, I wish to thank with everlasting gratitude to the Project Coordinator **Ms.S.Nagarathinam M.E., (Ph.D).,** Asst. Professor-III, Department of Electronics and Communication Engineering, for her suggestions throughout the course of this project work.

I am greatly privileged to express my heartfelt thanks to my project guide **Ms.M.Shanthi M.S., (Ph.D).,** Associate Professor, Department of Electronics and Communication Engineering, for her expert counselling and guidance to make this project to a great deal of success and I wish to convey my deep sense of gratitude to all teaching and non-teaching staff of ECE Department for their help and cooperation.

Finally, I thank my parents and my family members for giving me the moral support and abundant blessings in all of my activities and my dear friends who helped me to endure my difficult times with their unfailing support and warm wishes.

Abstract

Analog circuit fault diagnosis problem can be modeled as a pattern recognition problem and is solved by machine learning algorithm. Support Vector Machine (SVM) is often chosen as the learning machine because of its good generalization ability in small sample decision problem. This paper provides a fault detection model for analog filter circuit based on Polynomial Coefficients and V-Transform Coefficients and using Support Vector Machines Classifier. V-Transform is a non-linear transform that increases the sensitivity of Polynomial Coefficients with respect to circuit component's variation by three to five times. It makes the original Polynomial Coefficients monotonic. SVM is used for fault classification in the two feature sets. The fault classifier is a multi-class classifier based on the traditional "one against rest" SVC (Support Vector Machine Classifier) which is used to train the feature samples. To increase the classification accuracy and to reduce the execution time, feature subset selection is performed for the two feature sets using Binary Bat Algorithm (BBA). The method using BBA provides better performance in dealing with fault diagnosis problems. The classification accuracy for both the Polynomial Coefficients and the V-Transform Coefficients can also be improved by varying the kernel parameters c and epsilon combined with the SVM algorithm for the three kernel functions such as Polynomial kernel (POLYkernel), Radial Basis Function (RBF) kernel, and the Pearson VII kernel (PUK) function. PUK kernel function provides better classification accuracy compared to the other two kernel functions. Furthermore to increase the classification accuracy, HSVM (Hierarchical SVM) is used which is an extension of the Support Vector Machines to handle multi-class problems. HSVM for multi-class classification is a decision tree with an SVM at each node. The performance of the two circuit schemes based on the variations of the SVM classification are analyzed and their results are compared on the different metrics applied on the two coefficients.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	**iv**
	LIST OF TABLES	**viii**
	LIST OF FIGURES	**ix**
	LIST OF ABBREVIATIONS	**x**
1	**INTRODUCTION**	**1**
	1.1 Overview	1
	1.2 Fault diagnosis methods	1
	1.3 Analog Circuit Faults	2
	1.4 Benchmark Circuit	2
	1.5 SVM classification	2
	1.5.1 Feature selection	3
	1.5.2 Kernel functions	3
	1.5.3 Hierarchical support vector machines	3
	1.6 Properties of support vector machines	4
	1.7 Benefits of SVM	4
	1.8 Applications of SVM	4
2	**LITERATURE SURVEY**	**5**
	2.1 Multi-tone testing of linear and non-linear analog circuits using Polynomial coefficients	5
	2.2 Parametric fault diagnosis of nonlinear analog circuits using polynomial coefficients	5
	2.3 Neural network based fault diagnosis in analog electronic circuit using polynomial curve fitting	6
	2.4 Polynomial coefficient based DC testing of nonlinear analog circuits	6
	2.5 Non-linear analog circuit test and diagnosis under process variation using V-transform coefficients	6
	2.6 Parametric fault testing of non-linear analog circuits based on Polynomial and V-transform coefficients	7
	2.7 Parametric fault detection of analogue circuits	7
	2.8 One-class classifier based on SBT for analog circuit fault diagnosis	8
	2.9 Naïve-Bayes Guided bat algorithm for feature selection	8
	2.10 A new metaheuristic bat inspired algorithm	9
	2.11 A binary bat algorithm for feature selection	9
	2.12 A novel approach of analog circuit fault diagnosis using support vector machine classifier	10
	2.13 Fault diagnosis in analog electronic circuits-The SVM approach	10
	2.14 Feature selection and parameter optimization for support vector machines: a new approach based on genetic	10

	algorithm with feature chromosomes	
	2.15 Feature selection for support vector machines by means of genetic algorithms	10
	2.16 Optimization of SVM multi-class by particle-swarm (PSO-SVM)	11
	2.17 Feature selection for multi-class problems using support vector machines	11
	2.18 Performance of SVM based on PUK kernel in comparison to SVM based on RBF kernel in prediction of yarn tenacity	11
	2.19 A support vector hierarchical method for multi-class classification and rejection	12
	2.20 A comparison of methods for multiclass support vector machines	12
	2.21 A hierarchy of support vector machines for pattern detection	13
3	**FAULT FEATURE EXTRACTION**	**14**
	3.1 Introduction	14
	3.2 Circuit under test (CUT)	15
	3.3 Transfer function	16
	3.4 Fault dictionary	17
	3.5 Polynomial coefficients	18
	3.5.1 Fault dictionary using Polynomial coefficients	19
	3.6 V-Transform coefficients	20
	3.6.1 Fault dictionary using V-transform coefficients	21
4	**SUPPORT VECTOR MACHINES (SVM)**	**23**
	4.1 SVM classification	23
	4.1.1 Multi-class classification	25
	4.2 Feature subset extraction	26
	4.2.1 Binary bat algorithm	26
	4.2.2 Feature subset construction	30
	4.2.3 Fitness function	30
	4.3 Kernel functions of SVM	30
	4.3.1 Polynomial kernel	31
	4.3.2 Radial basis kernel	32
	4.3.3 Pearson VII kernel	32
	4.4 Simulation results	32
	4.4.1 Simulation results for the feature set	32
	4.4.1.1 Accuracy	33
	4.4.1.2 Error	33
	4.4.1.3 Sensitivity	33
	4.4.1.4 Specificity	33
	4.4.1.5 Precision	33
	4.4.1.6 F-measure	34
	4.4.2 Simulation results for the feature subset	35
	4.4.3 Simulation results for the different kernel functions	35

5	**HIERARCHICAL SUPPORT VECTOR MACHINES**	**46**
	5.1 Overview	46
	5.2 Hierarchical classifier design	47
	5.3 Clustering rules	48
	5.4 Simulation results	49
6	**SUMMARY**	**50**
7	**CONCLUSION**	**52**
	REFERENCES	**53**
	PUBLICATIONS	**56**

LIST OF TABLES

TABLE NO	CAPTION	PAGE NO
3.1	Sample fault dictionary of polynomial coefficients for R1 component	20
3.2	Sample fault dictionary of v-transform coefficients for R1 component	22
4.1	Comparison of results for the feature set	34
4.2	Comparison of results for the feature subsets using BBA	35
4.3	Simulation results for the polynomial kernel function with $\varepsilon = 0.1$	36
4.4	Simulation results for the polynomial kernel function with $\varepsilon = 0.001$	36
4.5	Simulation results for the polynomial kernel function with $\varepsilon = 10^{-12}$	37
4.6	Simulation results for the RBF kernel function with $\varepsilon = 0.1$	37
4.7	Simulation results for the RBF kernel function with $\varepsilon = 0.001$	38
4.8	Simulation results for the RBF kernel function with $\varepsilon = 10^{-12}$	38
4.9	Simulation results for the PUK kernel function with $\varepsilon = 0.1$	39
4.10	Simulation results for the PUK kernel function with $\varepsilon = 0.001$	39
4.11	Simulation results for the PUK kernel function with $\varepsilon = 10^{-12}$	40
5.1	HSVM classification	49
6.1	Accuracy comparison between the feature sets and the subsets using BBA	50
6.2	Accuracy comparison between the kernel functions	51
6.3	Training and testing accuracy of HSVM	51

LIST OF FIGURES

FIGURE NO	CAPTION	PAGE NO
3.1	Flowchart for the feature extraction process	14
3.2	Biquad filter circuit	16
4.1	Support vectors	24
4.2	SVM-Multi-class classification	25
4.3	Echolocation behavior of bats	27
4.4	Pseudo code of BBA	29
4.5	Accuracy plot for Polynomial Vs V-transform coefficients with $c = 0.001$ and $\varepsilon = 0.1$	40
4.6	Accuracy plot for Polynomial Vs V-transform coefficients with $c= 1$ and $\varepsilon = 0.1$	41
4.7	Accuracy plot for Polynomial Vs V-transform coefficients with $c = 10^3$ and $\varepsilon = 0.1$	41
4.8	Accuracy plot for Polynomial Vs V-transform coefficients with $c = 0.001$ and $\varepsilon = 0.001$	42
4.9	Accuracy plot for Polynomial Vs V-transform coefficients with $c = 1$ and $\varepsilon = 0.001$	42
4.10	Accuracy plot for Polynomial Vs V-transform coefficients with $c = 10^3$ and $\varepsilon = 0.001$	43
4.11	Accuracy plot for Polynomial Vs V-transform coefficients with $c = 0.001$ and $\varepsilon = 10^{-12}$	43
4.12	Accuracy plot for Polynomial Vs V-transform coefficients with $c = 1$ and $\varepsilon = 10^{-12}$	44
4.13	Accuracy plot for Polynomial Vs V-transform coefficients with $c = 10^3$ and $\varepsilon =10^{-12}$	44
5.1	An example of Hierarchical classification structure	47
5.2	Illustration of the three clustering rules	48

LIST OF ABBREVIATIONS

SVM	Support Vector Machine
BBA	Binary Bat Algorithm
SVC	Support Vector Machines Classifier
POLY kernel	Polynomial kernel
RBF	Radial Basis Function
PUK	Pearson VII Kernel
HSVM	Hierarchical support Vector Machine
SAT	Simulation After Test
SBT	Simulation Before Test
ANN	Artificial Neural Network
CUT	Circuit Under Test
VTC	V-Transform Coefficients
LNA	Low Noise Amplifier
ATPG	Analog Test Pattern Generator
ATRC	Analog Test Response Comparator
ACUT	Analog Circuit Under Test
ADC	Analog to Digital Converter
BW	Bandwidth
PSO	Particle Swarm Optimization
FDT	Fault Driven Test
SDT	Specification Driven Test
GA	Genetic Algorithm
DAGSVM	Directed Acyclic Graph Support Vector Machine
SAB	Single Amplifier Biquad
OVA	One Against All
OVO	One Against One
ACO	Ant Colony Optimization
BA	Bat Algorithm
PCA	Principle Component Analysis
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
PPV	Positive Predictive Value

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Fault diagnosis is an important problem of analog circuit testing. With the rapid development of modern electronic circuits and systems, fault diagnosis of analog circuits has become an increasingly important task in the world today. Fault diagnosis of analog filter circuits has been an active research area for decades, but it is still complicated due to poor fault models, component tolerances and non-linearity effects of analog circuits. Given the circuit topology and nominal circuit parameter values, fault diagnosis is to obtain the exact information about the faulty circuit based on the analysis of the limited measured circuit responses. Fault diagnosis of analog circuits is essential for analog and mixed-signal systems testing and maintenance both during the design process and the manufacturing process of VLSI ASICs. In the past several years, analog circuit fault diagnosis based on back propagation technique was used which resulted in excess raining time and the performance was poor. A new method of fault diagnosis of analog filter circuits using Support Vector Machines Classifier were employed which resulted in increase in accuracy and reduction in execution time.

1.2 FAULT DIAGNOSIS METHODS

Many analog diagnosis algorithms have been proposed and they fall into two categories: (i) Simulation after test (SAT) and (ii) Simulation before test (SBT). The SAT diagnosis implements costly circuit simulation and computation in test phase, which should be carried out in real time. SBT is more acceptable because it eliminates the on-line simulation and need only once off-line computation effort before test activities. Also there is no limitation for the application of SBT in test domain (parameters, frequency and time) and circuit type (linear or non-linear). Among all the SBT methods, data-driven methods, such as the artificial neural networks (ANNs) and Support Vector Machines (SVMs) are more suitable for analog fault diagnosis, since they do not need any explicit model.

1.3 ANALOG CIRCUIT FAULTS

Faults in analog circuits are broadly classified into two types: (i) Catastrophic faults (hard fault) and (ii) Parametric faults (soft fault). A catastrophic fault is one in which discrete component of a circuit is destroyed (e.g short circuit, open circuit as well as topological change). With parametric fault, the component is still functioning but out of nominal tolerance band (i.e out of specification). Faults can also be categorized into single faults and multiple faults. In the proposed method, single fault in analog filter circuits are diagnosed.

Many research works suggested the detection of faults in analog circuits using bandwidth, upper and lower cut-off frequencies, polynomial coefficients, supply current waveforms, peak amplitude, peak frequency and higher order sensitivity coefficients. In this proposed work, fault detection of analog filter circuits using Polynomial coefficients and V-Transform coefficients are used to detect the parametric faults.

1.4 BENCHMARK CIRCUIT

The benchmark circuits are a set of analog and mixed-signal circuits provided for the evaluation and performance of different testing approaches. However, the fault models for these benchmark circuits, along with a list of standard faults and range of acceptable component variations are not specified but they are a major concern in analog device testing. A circuit is designed to meet the tolerances associated with the specific requirement. Due to the very nature of the manufacturing process and working environment of the designed circuit, the values of the parameters often change. These variations are acceptable as long as circuit response is within specified limits. A known range of acceptable values for a circuit component parameter is necessary to establish the fault-free behavior for a given circuit, which can then be used to detect a fault.

1.5 SVM CLASSIFICATION

SVM classifier is used for fault classification in analog filter circuits. SVM is one of the most efficient machine learning algorithms, which is mostly used for pattern recognition and classification. SVM usually deals with pattern classification that means this algorithm is used

mostly for classifying the different types of patterns. SVM classifiers are robust, accurate and very effective even in cases where the number of training samples is very small. SVMs are essentially binary classifiers, but they can even handle multi-class problems. The multi-class classification based on the conventional "one against rest" SVC is used for the classification of Polynomial and V-transform coefficients. The "one against rest" SVC is faster and memory efficient and is widely used for multi-class classification problems. The classification is done in order to increase the accuracy and reduce the execution time of the process.

1.5.1 FEATURE SELECTION

Feature selection also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features for use in model construction. In machine learning, the problem of supervised classification is concerned with using labeled examples to induce a model that classifies objects into a finite set of known classes. Some of the features may be irrelevant or redundant. Avoiding irrelevant or redundant features is important because they may have a negative effect on the accuracy of the classifier. In addition, by using fewer features we may reduce the cost of acquiring the data and improve the comprehensibility of the classification model.

1.5.2 KERNEL FUNCTIONS

Support Vector Machines along with kernel based algorithms provide good classification results than Artificial Neural Networks (ANNs) for most of the benchmark problems. Kernel methods used in SVM were applied to a variety of problems such as classification and regression. The different kernel functions available in SVMs are the Polynomial kernel, Radial Basis kernel, Pearson VII kernel functions. On selection of the kernel function, the different parameters have to be varied in order to obtain higher classification accuracy.

1.5.3 HIERARCHICAL SUPPORT VECTOR MACHINES

HSVM is the extension of the SVM for multi-class problems. It is a decision tree with an SVM at each node. At the root node of the decision tree, all classes are available for prediction. The number of classes available for prediction keeps decreasing as the tree gets descended. The speed and accuracy of HSVM depends on its tree structure. HSVM overcomes the disadvantages

of one-against-one and one-against-all classifiers by using a binary hierarchical classification structure.

1.6 PROPERTIES OF SUPPORT VECTOR MACHINES

- Flexibility in choosing a similarity function.
- Ability to handle large feature spaces.
- Provides better feature selection properties.
- Achieve high generalization by maximizing the margin.
- Support an efficient learning of nonlinear functions by kernel trick.

1.7 BENEFITS OF SVM

- Effective in high dimensional spaces.
- Effective when number of dimensions is greater than number of samples.
- Uses a subset of training points called Support Vectors, so it is memory efficient.
- Different kernel function can be specified for the decision function.

1.8 APPLICATIONS OF SVM

- Digital Image Analysis
- Text Categorization
- Character Recognition
- Bioinformatics

CHAPTER 2

LITERATURE SURVEY

2.1 Multi-Tone Testing of Linear and Nonlinear Analog Circuits using Polynomial Coefficients

A method of testing for parametric faults of analog circuits based on a polynomial representation of fault-free function of the circuit is presented in this paper. The response of the circuit under test (CUT) is estimated as a polynomial in the applied input voltage at relevant frequencies in addition to DC. Classification of CUT is based on a comparison of the estimated polynomial coefficients with those of the fault free circuit. By expanding polynomial coefficients at critical frequencies the fault coverage is significantly improved, yielding a minimum size of detectable faults in some parameters as low as 5%. The proposed method is illustrated for a benchmark elliptic filter. It is shown to uncover several parametric faults causing deviations as small as 5% from the nominal values.

2.2 Parametric Fault Diagnosis of Nonlinear Analog Circuits using Polynomial Coefficients

In this paper, a method for diagnosis of parametric faults in analog circuits using polynomial coefficients of the circuit model is presented. In addition to the work proposed in the above paper, where the circuit response is modeled as polynomial for uncovering parametric faults in nonlinear circuits, the author has proposed diagnosis of such faults using sensitivity of coefficients of the estimated polynomial to circuit parameters. The proposed method requires no design for test hardware as might be added to the circuit by some other methods. The method has been extended to sensitivity based fault diagnosis with probabilistic confidence levels in parameter drifts. The author has also demonstrated diagnosis of several parametric faults with confidence levels up to 98.9% in the benchmark elliptic filter.

2.3 Neural Network based Fault Diagnosis in Analog Electronic Circuit using Polynomial Curve Fitting

The use of the neural network for parametric fault diagnosis in an analog circuit, based upon the polynomial curve fitting coefficients of the output response of an analog circuit is presented in this study. Building upon the theory of polynomial coefficients a parametric fault diagnosis methodology is proposed. A polynomial of suitable degree is fitted to the output frequency response of an analog circuit. The coefficients of the polynomial attain different values under faulty and non faulty conditions. These polynomial coefficients are used as signature for the training, validation and test sets for artificial neural network. Using these features of polynomial coefficients, a BPNN is used to detect the parametric faults. Simulation results are presented for a benchmark biquad filter circuit. Single resistance and capacitance faults of ±1% to ±50% deviation from nominal values were correctly diagnosed.

2.4 Polynomial Coefficient Based DC Testing of Non-Linear Analog Circuits

DC testing of parametric faults in non-linear analog circuits based on polynomial approximation of the functionality of fault free circuit is presented. Classification of circuit under test (CUT) is based on comparison of estimates of polynomial coefficients with those of the fault free circuit. The method needs very little augmentation of circuit to make it testable as only output parameters are used for classification. Possible fault diagnosis in conjunction with sensitivity of polynomial coefficients is also presented in this paper. The minimum size detectable faults of some of the parameters in circuits are as low as 10% which implies impressive fault coverage. The method has been extended to sensitivity based fault diagnosis with probabilistic confidence levels in parameter drifts.

2.5 Non-Linear Analog Circuit Test and Diagnosis under Process Variation using V-Transform Coefficients

A new approach for test and diagnosis of non-linear circuits based on a transformation of polynomial expansion of the circuit is demonstrated. The V- Transform acts on the polynomial expansion of the circuit's function. The V-Transform renders the polynomial coefficients monotonicity and enhances their sensitivity. The minimum sizes of detectable faults in some of

the circuit parameters are as low as 5% which implies that impressive fault coverage can be achieved with V-Transform Coefficients (VTC). The use of VTC shows a reduction in masking of parametric faults due to process variation. The method is then extended to sensitivity based fault diagnosis by evaluating VTC at different frequencies.

2.6 Parametric Fault Testing of Non-Linear Analog Circuits Based on Polynomial and V-Transform Coefficients

In this paper, the polynomial coefficient and V-transform coefficient based testing of parametric faults in linear and non-linear analog circuits is proposed. This paper describes two analog circuit test schemes for high resolution fault detection. The first scheme uses polynomial coefficients of the circuit's input- output response for fault detection. The second scheme uses a transformation on the polynomial coefficients for fault detection. Further, these methods are extended to sensitivity based fault diagnosis of parameter drifts with probabilistic confidence levels. V-transform is a non- linear transform that increases the sensitivity of polynomial coefficients with respect to circuit component variations by three to five times. In addition, it makes the original polynomial coefficients monotonic. Using simulation, the proposed test method is shown to un-cover most parametric faults in the range of 5–15% on a low noise amplifier (LNA) and an elliptic filter bench-mark. The experimental results demonstrate the benefits of V-transform.

2.7 Parametric Fault Detection of Analogue Circuits

This paper presents a new testing approach for analog circuits based on the digital signature analysis. In this paper, the efficient parametric fault detection approach for analog circuits using the simulation environment is presented. This approach has three main parts, an analog test pattern generator (ATPG), an analog test response comparator (ATRC), and an analog circuit under test (ACUT) model, build in the PSpice circuit simulator. The proper ATPG is designed to sweep the applying sinusoidal frequencies to match the frequency domain of the ACUT. The output test response of the ACUT is acquired via the analog-to-digital converter (ADC). The ATRC accumulates digital samples of the output response from the ADC to generate a digital signature that can characterize the situation of the ACUT. The signature comparison is achieved

based on signature boundaries based on the worst-case analysis. It combines effective parameters of the transfer function of the ACUT with respect to the component variations. These parameters are the band-with and the passband transmission. Using the signature curve, a parametric fault of each component of the ACUT can be detected under the sweep sinusoidal frequency of the ATPG. Based on this curve, the relation between digital signatures and component variations of the ACUT combines the effects of the bandwidth (BW) and the passband transmission (Amax) on the output response of the ACUT during component variations. In some cases, the signature curve is affected with the bandwidth only during the constant variation of Amax. In some other cases, the signature curve is affected with the Amax only during the constant variation of bandwidth. In other cases, the signature curve is affected with both the bandwidth and the Amax. The presented testing approach is applied to the analog benchmark circuit to validate the presented testing approach.

2.8 One-class classifier based on SBT for analog circuit fault diagnosis

In OCB-SBT, many predefined fault classes are constructed from their corresponding fault simulation samples. One-class classifier based SBT diagnosis framework is proposed in order to overcome the shortcomings of MCB-SBT methods. The important feature of the proposed framework is robust and reliable to the effects due to not only component tolerances and nonlinearity but also poorly defined fault classes and low testability that are familiar in analog circuit fault diagnosis.

2.9 Naive Bayes – Guided Bat Algorithm for Feature Selection

A new hybrid feature selection algorithm has been proposed in this paper. Bio-inspired method called Bat Algorithm hired Naïve-Bayes algorithm to intelligently select the most convenient feature that could maximize the classification accuracy while ignoring redundant and noisy features. The performance of the proposed algorithm was compared with three other feature selection algorithms using twelve benchmark datasets obtained from different domains. Four types of performance measures were evaluated such as the number of features, classification accuracy, stability and generalization. Compared to the other algorithms, Naïve Bayes – guided Bat Algorithm (BANB) produced less number of features and the classification

accuracy obtained was better. In terms of stability and generalization of results, the presented algorithm is more stable than the other algorithms.

2.10 A New Metaheuristic Bat-Inspired Algorithm

A new metaheuristic algorithm named Bat algorithm for solving continuous constrained optimization problems is presented in this paper. The algorithm is based on the echolocation behavior of bats. The advantages of the existing algorithms are combined in the bat algorithm. The new solutions are generated by adjusting frequencies, loudness and pulse emission rates and the acceptance of the proposed solution depends on the quality of the solutions controlled or characterized by loudness and pulse rate which are in turn related to the closeness or fitness of the locations or solutions to the global optimal solution. The proposed algorithm is compared with the other existing algorithms such as genetic algorithm and particle swarm optimization algorithms. Simulation results show that the proposed algorithm is superior to the other existing algorithms.

2.11 A Binary Bat Algorithm for Feature Selection

In this paper, an optimization problem known as the feature selection problem is focused in order to obtain the most important information from a given set of features. A new nature inspired feature selection technique known as the Binary Bat Algorithm (BBA) is proposed which is based on the behavior of the bats. A binary version of the well-known continuous-valued Bat Algorithm was derived in order to position the bats in binary coordinates along the corners of the search space, which represents a string of bits that encodes whether a feature will be selected or not. The wrapper approach is used along with Optimum-Path Forest classifier in order to find the set of features that maximizes the accuracy in a validating set. The experiment for the proposed method was conducted with five public datasets and it was compared with several meta-heuristic algorithms such as Particle Swarm Optimization (PSO), FFA and GSA to show the robustness and the good generalization capability of the bat inspired technique. The experimental results demonstrated that the proposed technique performed well compared to the other techniques.

2.12 A novel approach of analog circuit fault diagnosis using support vector machine classifier

The proposed method presents a novel approach of diagnosing actual analog circuits using improved support vector machine classifier (SVC) .The fault classifier is based on the conventional "one against rest" svc, which is then used to train the feature samples. Two experiments based on DAC and DSP are demonstrated to validate the proposed method and the results given by the experiment yields that the proposed svc is suited to be applied in the domain of analog testing, if proper parameters are chosen.

2.13 Fault diagnosis in analog electronic circuits – The SVM approach

In this proposed method, the application of the SVM algorithm has been used for diagnosis and tests of analog electronic circuits. The diagnosis procedure belongs to simulation-before-test techniques. The SVM has been applied for the fault-driven test (FDT) and the specification-driven test(SDT). The SVM classifies features which are calculated from the time domain responses. Results obtained from this approach prove a high detection and localization level of circuit states with the use of the SVM classifier.

2.14 Feature selection and parameter optimization for support vector machines: A new approach based on genetic algorithm with feature chromosomes

In this paper, a new approach based on genetic algorithm with feature chromosomes, is proposed to simultaneously optimize the feature subset and the parameters for SVM. Compared with GA without feature chromosomes, the proposed approach not only has higher classification accuracy and smaller feature subsets, but also has fewer processing time.

2.15 Feature selection for support vector machines by means of genetic algorithms

The proposed method consists of a special Genetic Algorithm, which especially takes into account the existing bounds on the generalization error for support vector machines instead of performing cross-validation. This is computationally much faster as each feature subset needs to

be trained only once. Additionally to the selection of the feature subset, kernel parameters can also be optimized such as the regularization parameter C of the SVM by means of GA.

2.16 Optimization of SVM Multiclass by Particle Swarm (PSO-SVM)

This paper proposes a PSO-SVM technique to optimize the performance of SVM classifier. In many classification problems, the performances of a classifier are often evaluated by a factor (the rate of error). The factor is not well adapted to the multi-class problems. Therefore an evolutionary method for optimizing this factor is obtained. The optimization method used in this paper is the Particle Swarm Optimization (PSO) technique which makes it possible to optimize the performance of SVM classifier. The experimental results show that the approach PSO-SVM gives a better classification in terms of accuracy even though the execution time is increased.

2.17 Feature Selection for Multi-class Problems Using Support Vector Machines

In this paper, feature selection for multi-class problems using support vector machines (SVM) is presented. Since feature selection can remove the irrelevant features and improve the performance of learning systems, it is a crucial step in machine learning. The feature selection methods using support vector machines have obtained satisfactory results. In this paper, a prediction risk based feature selection method using multiple classification support vector machines is proposed. The performance of the proposed method is compared with the previous methods of optimal brain damage based feature selection methods using binary support vector machines. The results of experiments on UCI data sets show that prediction risk based feature selection method obtains better results for multiple classification problems.

2.18 Performance of SVM based on PUK kernel in Comparison to SVM Based on RBF Kernel in Prediction of Yarn Tenacity

In the proposed method, a new kernel function of SVM based on the Pearson VII function has been applied and compared with the commonly applied kernel functions such as Polynomial and Radial Basis function (RBF) to predict yarn tenacity. The SVM models based on RBF and PUK kernel shows the same applicability, suitability and performance to map the nonlinear

relation between input and output data for predicting the yarn tenacity. A comparison of SVM models based on RBF and PUK kernels with ANN (Artificial Neural Network) model shows that the two SVM models have similar prediction performances as ANN model.

2.19 A Support Vector Hierarchical Method for Multi-class Classification and Rejection

This paper presents a new soft-decision hierarchical classifier to address multi-class classification and rejection problems, with focus on distortion invariant object recognition. The hierarchical structure is designed by the weighted support vector k-means clustering method. SVRDMs (support vector representation and discrimination machine) are used at each node as the classifiers to provide good generalization and rejection ability, which cannot be achieved by standard SVMs. The new aspect of this paper is that it provides remarks on the hierarchical design method, including the hierarchical clustering rule and discussed the meaning and the use of the probabilities in the soft-decision hierarchical SVRDM classifiers. The results were compared with the hierarchical SVM classifiers, the standard One-VS-All SVM and SVRDM classifiers and it was found that the new classifier gave better Pc and P_{FA}. Excellent classification and rejection results were obtained in initial tests on the COIL-100 database with aspect variations, while no prior work considered rejection of false classes for this database.

2.20 A Comparison of Methods for Multiclass Support Vector Machines

This paper presents a decomposition implementation for all-together methods such as binary classification and multi-class classification. Their performances are then compared with three methods based on binary classification such as “one-against-one”, “one-against-all”, and directed acyclic graph SVM (DAGSVM). The experiments indicate that for problems with larger dataset the “one-against-one” and DAG methods are more suitable for practical use than the other methods.

2.21 A Hierarchy of Support Vector Machines for Pattern Detection

In this proposed work, a computational design for pattern detection based on a tree-structured network of support vector machines (SVMs) is introduced. The objective is to design and build a network which balances overall error and computation. An SVM is associated with each cell in a recursive partitioning of the space of patterns called hypotheses into increasingly finer subsets. The hierarchy is traversed coarse-to-fine and each chain of positive responses from the root to a leaf constitutes detection. Initially SVMs were constructed for each cell with no constraints. This free network is then perturbed, cell by cell, into another network, which is graded in two ways: First, the number of support vectors of each SVM is reduced by clustering in order to adjust to a pre-determined, increasing function of cell depth and the Second, the decision boundaries are shifted to preserve all positive responses from the original set of training data. The limits on the number of clusters result from minimizing the mean computational cost of collecting all detections subject to a bound on the expected number of false positives. When applied to detecting faces in cluttered scenes, the patterns correspond to poses and the free network is already faster and more accurate than applying a single pose-specific SVM many times. The graded network promotes very rapid processing of background regions while maintaining the discriminatory power of the free network.

CHAPTER 3

FAULT FEATURE EXTRACTION

3.1 INTRODUCTION

The proposed methodology mainly consists of three stages such as fault signature extraction, preprocessing techniques and the fault classification. The detailed flow of the methodology consisting of Polynomial coefficient and V-Transform coefficient based fault detection of analog filter circuit using SVM classification is illustrated in the fig 3.1.

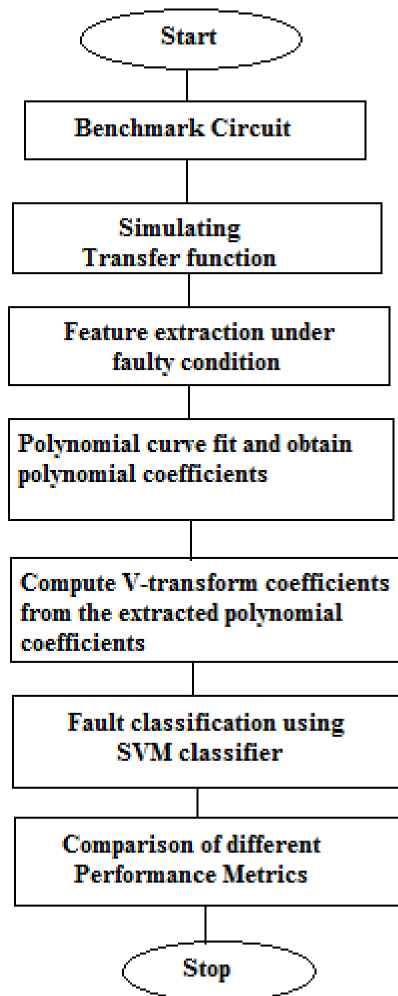


FIG 3.1: FLOWCHART FOR THE FEATURE EXTRACTION PROCESS

3.2 CIRCUIT UNDER TEST (CUT)

The proposed methodology is validated by considering the benchmark circuit namely Two Thomas Biquad filter. The three-op-amp Biquad filter is used as the circuit under test for the proposed method. In signal processing, a digital biquad filter is a second order recursive linear filter, containing two poles and two zeros. Biquad is an abbreviation of biquadratic which refers to the fact that in the z-domain, its transfer function is the ratio of two quadratic functions. It is also sometimes called as the "ring of 3" circuit. Because of coefficient sensitivities in higher order filters, the biquad is often used as the basic building block for more complex filters.

Biquad filters are typically active and implemented with the single-amplifier biquad (SAB) or two-integrator-loop topology. The SAB topology uses feedback to generate complex poles and possibly complex zeros. In particular, the feedback moves the real poles of an RC circuit in order to generate the proper filter characteristics. The two-integrator-loop topology is derived from rearranging a biquadratic transfer function. The rearrangement will equate one signal with the sum of another signal, its integral and the integral's integral. By using different states as output, any kind of second-order filter can be implemented. The SAB topology is sensitive to component choice and can be more difficult to adjust. Hence, usually the term biquad refers to the two-integrator-loop state variable filter topology.

The Biquad filter is an active RC-topology used to realize both band-pass and low-pass responses. Biquad filter consists of three operational amplifiers cascaded to produce the two output responses. It consists of two integrators and an inverter. The basic configuration of the biquad filter is shown in the figure and it can be used as either the low-pass or the band-pass filter depending on where the output signal is taken from.

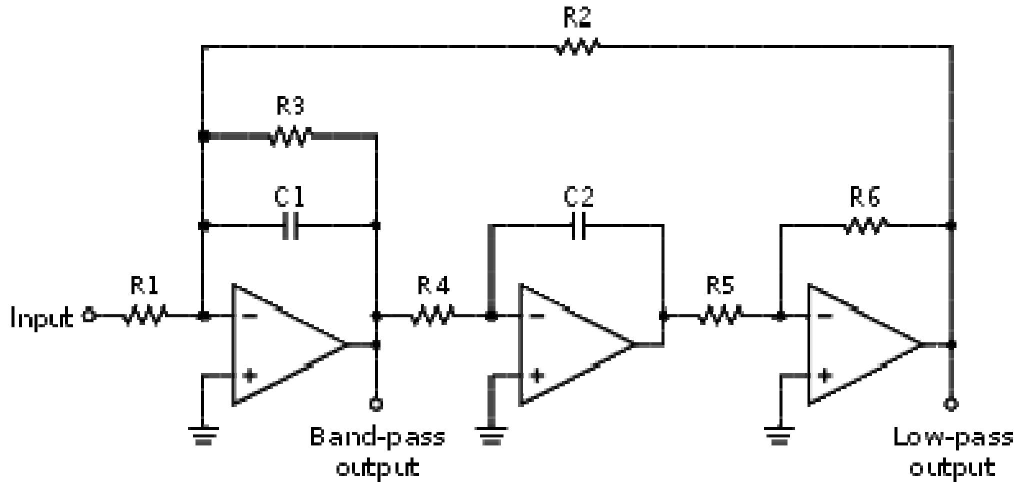


FIG 3.2: BIQUAD FILTER CIRCUIT

In a biquad filter, as the fc changes, the bandwidth stays constant, but the Q value changes. Thus if we change fc in the frequency domain, as fc increases the Q value increases and as fc decreases, the Q value decreases. It allows very high Q values. It can have Q factor values of 400 or greater.

The circuit can be configured as the 3 or 4 amplifier configuration and it is less sensitive to external component variations. The 3 or 4 amplifier circuit draws more power and it requires more design time, but it gives better performance when compared to single amplifier configuration. The advantage of using this filter is that it has fewer components, versatile, parasitic insensitive and simple.

3.3 TRANSFER FUNCTION

The biquad filter is a type of linear filter that implements a transfer function that is the ratio of the two quadratic functions. The transfer function of the circuit is the ratio of the output voltage to the input voltage (i.e) Vo/Vi and it varies for various types of output. The transfer function is computed to obtain the magnitude and phase of the frequency response of the circuit. The transfer function of the biquad filter is represented as

$$H(s) = \frac{G_p \omega^2}{S^2 + \frac{\omega}{Q} S + \omega^2} \qquad (1)$$

Where G_p is the gain of the filter, Q is the quality factor and w is the frequency. The bandwidth is given by

B=w/Q

The transfer function for obtaining the band-pass output is given by

$$\frac{V_o BP(s)}{V_{in}(s)} = -\frac{S\frac{1}{C_1 R_1}}{S^2 + S\frac{1}{C_1 R_2} + \frac{R_4}{C_1 C_2 R_3 R_5 R_6}} \tag{2}$$

The transfer function for obtaining the low-pass output is given by

$$\frac{V_o LP(s)}{V_{in}(s)} = -\frac{\frac{R_4}{C_1 R_2 R_5 R_6 R_1}}{S^2 + S\frac{1}{C_1 R_2} + \frac{R_4}{C_1 C_2 R_3 R_5 R_6}} \tag{3}$$

where Vo is the output voltage and Vi is the input voltage. The components of the circuit under test are R1, R2, R3, R4, R5, R6, C1 and C2. The nominal values of the components are R1=R3=2.7 KΩ, R2=1.5 KΩ, R4=12 KΩ, R5=1 KΩ, R6=10 KΩ, C1=C2=10 nF.

3.4 FAULT DICTIONARY

Fault dictionary is the practical approach belonging to the simulation before test (SBT) technique. Fault dictionary is constructed by injecting faults to each component of the filter with ±50% deviation from the nominal value under faulty condition. Frequency response of an analog circuit is the graph which shows the variation in the gain of the analog circuit with respect to the frequency of operation. When the frequency of the input stimulus is varied the output voltage and hence the gain of the circuit varies. By plotting this variation in the gain with respect to frequency, a frequency response of the circuit is obtained. Mathematically to implement this, the transfer function of the analog circuit is required to be calculated depending upon its required response and the bandwidth requirement.

Apart from mathematically obtaining the frequency response of the analog circuit, the frequency response graph is obtained by simulating the analog circuit using the bode plotter tool available in MATLAB. Bode plotter is very useful for the analysis of filter circuits. It produces a

graph of circuit's frequency response. In this the gain of the circuit under test is plotted with respect to the frequency. Bode plotter is available as bode diagram of frequency response in the control system tool box of MATLAB software. This is used to compute the magnitude and phase of the frequency response of the linear time invariant models. In the bode diagram the magnitude is plotted in db and the phase in degrees.

The frequency response graph thus obtained will contain several values, so while using all these values will result in more execution time and will reduce the accuracy. In order to increase the accuracy and reduce the execution time, two types of feature extractions are used.

They are:

1. **Polynomial Coefficients**
2. **V-Transform Coefficients**

3.5 POLYNOMIAL COEFFICIENTS

For the fault injected on each component, the frequency response graph obtained will be different. There will be different frequency response graphs indicating the different parametric variation faults in each of the components present in the circuit. The parametric fault introduced in each component values are with the variation of ±50% (i.e.) two-hundred faults are there for each component keeping all other components at their nominal values within the tolerance limit. It has been observed that for each fault in the circuit component value a unique frequency response graph is obtained.

The collected graphs are applied to the preprocessor to get the proper and distinguishable features. This is done using polynomial curve fitting. Polynomials are one of the most commonly used types of curves in regression. The Polynomial Curve Fitting uses the method of least squares when fitting data. The frequency response of the circuit is curve fitted with 9th order polynomial curve fitting tool to yield 10 polynomial coefficients for the fault scenario of each component. Polynomial curve fitting is done using the curve fitting toolbox of MATLAB software. Since the frequency response graphs obtained are different for different fault components, polynomial coefficients obtained are also different and unique for different faults. These polynomial coefficients are used to prepare the fault dictionary, for further classification

of the faults. A feature set comprising of the components with values under fault condition and its corresponding polynomial coefficients will be constructed.

The following steps are followed in the curve fitting process:

1. The frequency response graph is first transferred and stored in a MICROSOFT office excel work sheet.

2. These values of the graph from excel worksheet are transferred to the MATLAB workspace.

3. The data stored in the MATLAB workspace is imported in the curve fitting tool box.

4. The imported graph in the curve fitting tool box is curve fitted using polynomial curve fitting.

5. A ninth order polynomial is used to fit output frequency response of CUT yielding ten polynomial coefficients.

6. This process is repeated for every output frequency response related to each fault scenario of each component of CUT and ten polynomial coefficients are recorded for each curve fit. These coefficients are further used to build the fault dictionary.

3.5.1 FAULT DICTIONARY USING POLYNOMIAL COEFFICIENTS

Fault dictionary created by using the polynomial coefficients obtained by curve fitting the frequency response of the circuit with the polynomial curve fitting tool of MATLAB are tabulated. The table shows the sample fault dictionary for the faults injected on R1 component.

TABLE 3.1: SAMPLE FAULT DICTIONARY OF POLYNOMIAL COEFFICIENTS FOR R1 COMPONENT

FAULT INDEX	POLYNOMIAL COEFFICIENTS										
	a0	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10
1	-0.9957	0.9918	-0.85	-0.9815	0.9612	-0.9257	0.8413	-0.6676	0.3726	0.0697	-0.6217
1	-0.9951	0.9906	-0.85	-0.9847	0.9678	-0.9159	0.8246	-0.7262	0.4712	-0.0801	-0.465
1	-0.9957	0.9919	-0.85	-0.9809	0.9604	-0.9385	0.8674	-0.6592	0.3587	0.0892	-0.6386
1	-0.996	0.9922	-0.85	-0.9829	0.9643	-0.9311	0.8526	-0.6901	0.4001	0.0455	-0.6167
1	-0.9957	0.9916	-0.85	-0.982	0.9626	-0.9373	0.8668	-0.6989	0.4234	0.0026	-0.5717
1	-0.9962	0.9926	-0.85	-0.985	0.968	-0.9255	0.8393	-0.7319	0.4822	-0.0979	-0.4475
1	-0.9958	0.9918	-0.85	-0.9817	0.962	-0.9329	0.8539	-0.6479	0.337	0.1216	-0.6629
1	-0.9954	0.9911	-0.85	-0.9849	0.9677	-0.9248	0.841	-0.6434	0.3341	0.1211	-0.6622
0	-0.9956	0.9915	-0.85	-0.984	0.9664	-0.9212	0.8312	-0.7213	0.464	-0.0723	-0.4683
1	-0.9962	0.9927	-0.85	-0.9845	0.9671	-0.9236	0.8383	-0.723	0.4717	-0.0878	-0.4536
1	-0.9957	0.9917	-0.85	-0.9855	0.9691	-0.9428	0.876	-0.6731	0.3679	0.0939	-0.6559
1	-0.9953	0.9911	-0.85	-0.9868	0.9717	-0.9172	0.8255	-0.6258	0.3014	0.169	-0.6972
0	-0.9949	0.99	-0.85	-0.9874	0.9731	-0.9256	0.8437	-0.6849	0.3919	0.0554	-0.6222
1	-0.9955	0.9915	-0.85	-0.9863	0.9709	-0.932	0.8576	-0.6689	0.3713	0.0763	-0.6325
1	-0.9958	0.992	-0.85	-0.9844	0.967	-0.9331	0.8578	-0.7154	0.4577	-0.0691	-0.4649

Fault dictionary is created for all the other components in a similar manner. Randomly 75% of the samples are selected as training samples and the remaining 25% are selected as testing samples from the fault dictionary created for all the faulty components.

3.6 V-TRANSFORM COEFFICIENTS

V-transform is the transformation of the polynomial coefficients. V-transform is a non-linear transform that increases the sensitivity of polynomial coefficients with respect to circuit component variations by three to five times. In addition, it makes the original polynomial coefficients monotonic. The V-transform acts on the polynomial expansion of the circuit's function. The main properties of the V-transform are:

1. It makes the original polynomial coefficients monotonic.
2. It reduces the masking of parametric faults due to process variation.

3. It increases the sensitivity of polynomial coefficients to the circuit parameter variation, thereby enhancing diagnostic resolution.

The sensitivity of V-Transform Coefficients (VTC) with respect to circuit parameter variation is up to 3 to 5 times greater than the sensitivity of polynomial coefficients.

V-Transform coefficients are defined as follows: if $C_1, C_2 \dots C_n$ are polynomial coefficients of CUT then their V-Transform coefficients $V_{c1}, V_{c2} \dots V_{cn}$ are:

$$V_{ci} = e^{\gamma C_i'} \; \forall \, 0 \le i \le n \qquad (4)$$

Where C_i' are the modified polynomial coefficients defined by

$$\frac{dC_i'}{dp_j} = \left|\frac{dC_i}{dp_j}\right| \; \forall \, 0 \le i \le n \qquad (5)$$

The modification ensures that the modified polynomial coefficients are monotonic with the polynomial coefficients. The V-Transform coefficients (VTC) are exponential functions of the modified polynomial coefficients and γ is a sensitivity parameter chosen according to the desired sensitivity. Choices of $\gamma = 3$, for instance, results in a 3 times more sensitive coefficient to circuit parameters.

The sensitivity increase is due to enhancement of correlation of the V-transform coefficient to specific components, where each coefficient is multiplied by a factor. The transformation of the polynomial coefficients (i.e.) the V- Transform coefficients obtained is used to create the fault dictionary, for further classification of the faults.

3.6.1 FAULT DICTIONARY USING V-TRANSFORM COEFFICIENTS

Fault dictionary created by using the V-Transform coefficients (i.e.) the transformation of the polynomial coefficients are tabulated. The table shows the sample fault dictionary for the faults injected on R1 component.

TABLE 3.2: SAMPLE FAULT DICTIONARY OF V-TRANSFORM COEFFICIENTS FOR R1 COMPONENT

FAULT INDEX	V-TRANSFORM COEFFICIENTS										
	a0	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10
1	388.5804	1.2443	0.0518	15.1318	1.0873	0.8354	1.4728	0.4951	3.0221	0.2250	0.1257
1	386.4877	1.2432	0.0517	15.2824	1.1087	0.8067	1.3292	0.5885	2.6357	0.2455	0.3152
1	388.6970	1.2443	0.0519	15.1137	1.0657	0.8698	1.5204	0.4662	3.0728	0.2240	0.1127
1	389.3972	1.2448	0.0517	15.2023	1.0808	0.8457	1.4355	0.5073	3.0144	0.2176	0.1372
1	388.3473	1.2443	0.0518	15.1636	1.0693	0.8661	1.4299	0.5142	2.8641	0.2248	0.1785
1	390.0988	1.2452	0.0515	15.2870	1.0934	0.8244	1.3370	0.5853	2.5883	0.2479	0.3504
1	388.6970	1.2445	0.0518	15.1500	1.0759	0.8503	1.5334	0.4605	3.1716	0.2232	0.0950
1	387.4164	1.2437	0.0516	15.2801	1.0943	0.8269	1.5252	0.4675	3.1479	0.2244	0.0954
0	388.1143	1.2441	0.0517	15.2503	1.0988	0.8164	1.3497	0.5765	2.6472	0.2470	0.3048
1	390.2158	1.2452	0.0515	15.2663	1.0957	0.8243	1.3511	0.5770	2.5930	0.2496	0.3337
1	388.4638	1.2443	0.0515	15.3122	1.0661	0.8697	1.4986	0.4667	3.1598	0.2153	0.1055
1	387.3002	1.2435	0.0515	15.3720	1.1100	0.8031	1.5482	0.4556	3.2943	0.2236	0.0744
0	385.5612	1.2428	0.0515	15.4044	1.0971	0.8236	1.4348	0.5078	3.0357	0.2185	0.1310
1	387.9979	1.2439	0.0515	15.3536	1.0849	0.8437	1.4839	0.4822	3.0581	0.2219	0.1193
1	388.9302	1.2445	0.0516	15.2640	1.0800	0.8489	1.3862	0.5487	2.6365	0.2506	0.3050

Fault dictionary using V-transform coefficients are created for all the other components in a similar manner.

CHAPTER 4

SUPPORT VECTOR MACHINES (SVM)

4.1 SVM CLASSIFICATION

Support Vector Machine (SVM) is one of the most efficient supervised machine learning algorithms which is mostly used for classification problems. SVM usually deals with pattern classification that means this algorithm is used mostly for classifying the different types of patterns. SVM approach has some advantages compared to other classifiers. They are robust, accurate and very effective even in cases where the number of training samples is very small. SVMs are essentially binary classifiers, but they can even handle multi-class problems.

The aim of support vector classification is to device a computationally efficient way of learning good separating hyperplanes in a high dimensional feature space. Support vector machine is a Learning machine which finds an optimal separating hyperplane. It uses a linear hyperplane to create a classifier with a maximum margin. The algorithm aims to find support vectors and their corresponding coefficients to construct an optimal separating surface by the use of kernel functions in high dimensional feature space.

SVM classifies data by finding the best hyperplane that separates all data points of one class from those of the other class. The best hyperplane for a SVM means the one with the largest margin between the two classes. Margin means the maximum width of the slab parallel to the hyperplane that has no interior data points. The maximal margin hyperplane will be more accurate in classifying the data than the smaller margin. The separating hyperplane can be written as

$$\mathbf{w} \cdot \mathbf{x} + \mathbf{b} = 0 \qquad \textbf{(6)}$$

where, w is a weight vector and b is a bias (scalar). The maximal margin is denoted mathematically by the formula as

$$\mathbf{M} = 2 \,/\, ||\mathbf{W}|| \qquad \textbf{(7)}$$

where, $||W||$ is the Euclidean norm of w.

The optimal separating hyperplane maximizes the margin of the training data.

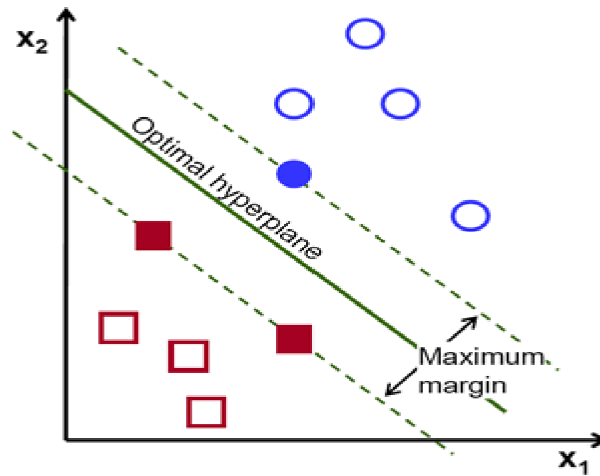


FIG 4.1: SUPPORT VECTORS

The notation to define a hyperplane is given by

$$\boldsymbol{f(x) = \beta_0 + \beta^T x} \qquad \textbf{(8)}$$

where β is known as the weight vector and β_0 as the bias. The optimal hyperplane can be represented in an infinite number of ways by scaling of β and β_0 . Among all the possible representations of the hyperplane the one chosen is

$$\boldsymbol{|\beta_0 + \beta^T x| = 1} \qquad \textbf{(9)}$$

x represents the training samples that are closest to the hyperplane. The training samples that are closest to the hyperplane are called the Support Vectors. The distance between the point x and a hyperplane is given by

$$\boldsymbol{distance = \frac{|\beta_0 + \beta^T x|}{||\beta||}} \qquad \textbf{(10)}$$

4.1.1 MULTI-CLASS CLASSIFICATION

A multi-class classifier is constructed by combining several binary classifiers. In multi-class classification, each training point belongs to one of N different classes. The goal is to construct a function which, given a new data point will correctly predict the class to which the new point belongs. There are two approaches for multi-class classification:

- **One-Vs-One**
- **One-Vs-All**

The earliest method is the One-Against-All (OVA) which constructs K classifiers, where K is the number of classes. The k th classifier is trained by labeling all the examples in the k th class as positive and the remainder as negative. The final hypothesis is given by the formula:

$$f_{ova}(x)=\arg\max_{i=1,\ldots,k}(f_i(x)) \qquad (11)$$

Another popular paradigm, called One-Against-One (OVO), proceeds by training k(k-1)/2 binary classifiers corresponding to all the pairs of classes. From the two approaches the One-Vs-All is more faster and memory efficient and is widely used for multiclass classification problems. It requires $O(N^2)$ classifiers instead of O(N), but each classifier is (on average) much smaller. If the time to build a classifier is super-linear in the number of data points, OVA is a better choice for SVM multi-class problems.

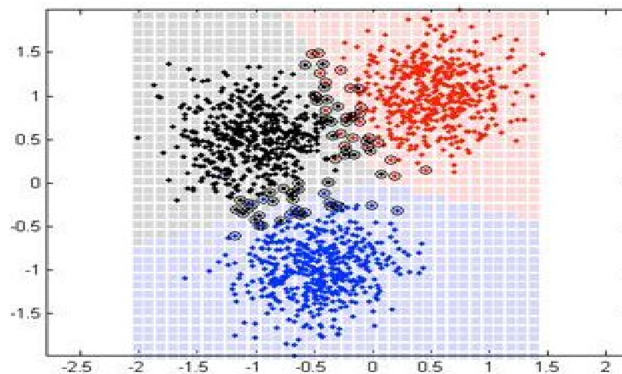


FIG 4.2: SVM – MULTI-CLASS CLASSIFICATION

4.2 FEATURE SUBSET EXTRACTION

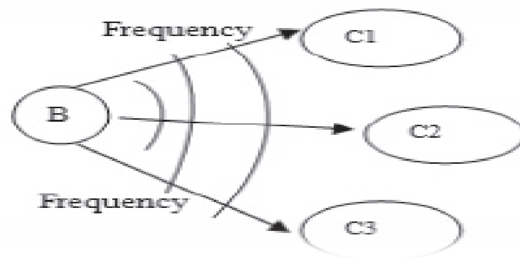
Feature selection is a process of selecting subset of features from a large data set. The best subset of features contains least number of dimensions that contribute to the accuracy of the model by removing irrelevant features. The advantages of feature subset selection are:

- It creates a less complex dataset which is easily interpretable.
- It enhances the performance of the classifier.
- Accuracy is increased.
- It reduces the execution time.
- Processing cost is reduced in terms of storage requirements.

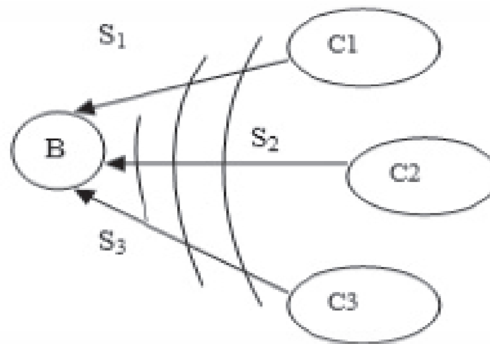
There are various evolutionary techniques available for obtaining the best feature subset such as Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Ant Colony Optimization (ACO), Bat algorithm (BA) etc., the evolutionary algorithm used for obtaining the best feature subset in the proposed method is the Binary Bat Algorithm (BBA).

4.2.1 BINARY BAT ALGORITHM

BBA is the binary version of the bat algorithm which is used to deal with the feature selection problem. The characteristics of bat for finding its prey are being used in binary bat algorithm. The BBA will have artificial bats navigating and hunting in binary search spaces by changing their positions from “0” to “1” and vice-versa. The BBA was inspired by the echolocation behavior of microbats with varying pulse rate of emission and loudness. As the bat approaches the prey, the bat’s pulse emission rate increases and the loudness decreases which are used for selecting the optimum points in the bat algorithm. Bats emit sonar signals in order to locate potential prey. These signals bounce back if they hit an object. Bats are able to interpret the signals to see if the object is large or small and if it is moving toward or away from them.



Bat send sound signal with frequency f



Echo signal used to calculate the distance s

FIG 4.3: ECHOLOCATION BEHAVIOR OF BATS

The standard BA is based on 3 idealized rules:

- All bats use echolocation to sense distance, and they also 'know' the difference between food/prey and background barriers in some magical way.
- Bats fly randomly with velocity v_i at position x_i with a fixed frequency fmin, varying wavelength λ and loudness Ao to search for prey. They can automatically adjust the wavelength of their emitted pulses and adjust the rate of pulse emission,$r \in [0,1]$ depending on the proximity of their tatget;
- Although the loudness can vary in many ways, we assume that the loudness varies from a large (positive) A_0 to a minimum constant value A_{min}.

The bat algorithm is computed with a swarm of bats that work together to find the best solutions. Each bat is identified with the velocity v_i and position x_i in a d-dimensional sollution space which are updated during the iterations. At the beginning, each bat is assigned a frequency in the interval $[f_{min}, f_{max}]$.

The new velocity and position at time step t are calculated by

$$f_i = f_{min} + (f_{max} - f_{min})\beta \qquad (18)$$

$$v_i^t = v_i^{t-1} + \left(x_i^{t-1} - x_{Gbest}\right)f_i \qquad (19)$$

$$x_i^t = x_i^{t-1} + v_i^t \qquad (20)$$

where $\beta \epsilon [0,1]$ is a random vector drawn from a uniform distribution, x_{Gbest} is the current global best solution which is located after comparing all the solutions among all the bats. As all these equations guarantee the exploitability of BA, a new solution for each bat is updated as follows:

$$X_{new} = X_{old} + \varepsilon A^t \qquad (21)$$

where $\varepsilon \epsilon [-1,1]$ is a random number, A^t is the average loudness of all the bats at this time step. The two parameters, the loudness A_i and the pulse rate r_i are updated as follows:

$$A_i^{t+1} = \alpha A_i^t \;\; ; \;\; r_i^{t+1} = r_i^o[1 - exp(-\gamma t)] \qquad (22)$$

where α and γ are constants. The loudness and the pulse rate are updated when the new solutions are improved to guarantee that the bats are moving toward the best solutions.

To enhance the generalization capability of the analog fault diagnosis method, we proposed to construct a big feature set that contains different kinds of features and use the BBA to select the optimal feature subset. A binary version of BA is needed for the feature selection problem, as each bat moves in the search space towards continuous-valued positions in the BA. In BBA the bat's position are represented by binary vectors. V-shaped transfer function is used in the binary version of BA. The v-shaped transfer function and the position updating rule are given as follows:

$$V\left(v_{ik}^{t+1}\right) = \left|\frac{2}{\pi}\arctan\left(\frac{2}{\pi}v_{ik}^t\right)\right| \qquad (23)$$

$$x_{ik}^{t+1} = \begin{cases} (x_{ik}^{t})^{-1} & \mathit{If\ rand} < V(v_{ik}^{t+1}) \\ x_{ik}^{t} & \mathit{rand} \geq V(v_{ik}^{t+1}) \end{cases} \qquad (24)$$

where v_{ik}^{t} and x_{ik}^{t} indicate the position and velocity of the i-th bat at iteration t in the k-th dimension, and $(x_{ik}^{t})^{-1}$ is the complement of x_{ik}^{t} . The figure shows the pseudo code of BBA.

Binary bat algorithm

Initialize the bat population x_i (i=1,2,...,n)=rand(0 or 1)and v_i =0

Define pulse frequency f_i at x_i

Initialize pulse rates $r_i$and the loudness A_i

***While** (t<Maximum number of iterations)*

Adjusting frequency and updating velocities

Calculate transfer function value using equation (6)

Update locations/solutions using equations (7)

***if** (rand > r_i)*

Select a solution (x_G) among the best solutions

Change some of the dimensions of position vector with some of the dimensions of x_G

end if

Generate a new solution by flying randomly

***if** (rand < A_i & $f(x_i) < f(x_G)$)*

Accept the new solutions

Increase r_i and reduce A_i

end if

Rank the bats and find the current best x_G

end while

FIG 4.4: PSEUDO CODE OF BBA

4.2.2 FEATURE SUBSET CONSTRUCTION

The polynomial coefficients and the V-transform coefficients are used to construct the two big features set F in the proposed method.

$$F = [\boldsymbol{f_1}, \boldsymbol{f_2}, \dots \boldsymbol{f_n}]$$

where n is the total number of features in big feature set F. In BBA, the bat's position is represented by binary vectors. The bat's positions are binary coded and the binary coded string is given to the feature set. A fitness function is used in order to select the best feature subset by using the BBA. At each iteration of the bat's motion, the fitness value is calculated and the value that is closer to the fitness value is determined as the best value and is stored. Finally the feature subset is created consisting of all the best values. SVM then classifies the feature subset.

4.2.3 FITNESS FUNCTION

The fitness function is constructed by considering the important factors such as the SVM classification accuracy, the number of selected features and the feature cost. The fitness value is calculated in order to achieve higher accuracy solution with smaller number of features and with reduced total feature cost. The fitness function is given by

$$\boldsymbol{fitness = W_1 \times Ac + W_2 \times \frac{N_a - N_s}{N_a \times Ct}} \qquad (25)$$

where Ac is the classification accuracy, W_1 is the weight of the classification accuracy, N_a is the total number of all the features, N_s is the number of selected features Ct is the total cost of selected features, $W_2 = 1 - W_1$is the weight of selected features. Parameters W_1 and W_2 are set according to the degree of importance between classification accuracy and the number of selected features and they provide better tradeoff between fault diagnosis accuracy and the total cost.

4.3 KERNEL FUNCTIONS OF SVM

In machine learning, kernel methods are a class of algorithms for pattern analysis, whose best known member is the support vector machine (SVM). The general task of pattern analysis is to find and study general types of relations in datasets. For many algorithms that solve these tasks,

the data in raw representation have to be explicitly transformed into feature vector representations via a user-specified feature map. In contrast, kernel methods require only a user-specified kernel, i.e., a similarity function over pairs of data points in raw representation. Kernel functions have been introduced for sequence data, graphs, text, images as well as vectors.

Algorithms capable of operating with kernels include the kernel perceptron, support vector machines (SVM), Gaussian process, Principle component analysis (PCA), canonical correlation analysis, ridge regression, spectral clustering, linear adaptive filters and many others. Any linear model can be turned into a non-linear model by applying the kernel trick to the model replacing its features by a kernel function.

Support vector machines along with kernel-based algorithms provide good classification results than Artificial Neural Networks (ANN's) for most of the benchmark problems. Kernel methods used in SVM were applied to a variety of problems such as classification and regression. There are different kernel methods of SVM such as Polykernel, RBF kernel, Puk kernel etc., which are introduced to obtain higher classification accuracy by varying the kernel parameters.

There are various kernel functions used with SVMs, but the choice of a particular kernel function to map the non-linear input space into a linear feature space depends highly on the nature of the data. As the nature of the data is unknown, the finest mapping function must be resolved experimentally by applying and validating various kernel functions producing the highest generalization performance. Therefore by adjusting the kernel parameters, the best kernel function can be determined.

4.3.1 POLYNOMIAL KERNEL (POLY KERNEL)

Polynomial kernels are commonly used with support vector machines which represents the similarity of vectors in the feature space over polynomials of the original variables. In a POLY kernel, K corresponds to an inner product in a feature space based on the mapping φ:

$$\boldsymbol{K}\,(\boldsymbol{x},\boldsymbol{y}) = \langle \boldsymbol{\varphi}(\boldsymbol{x}), \boldsymbol{\varphi}(\boldsymbol{y}) \rangle \qquad (\mathbf{26})$$

where x and y are inputs in the vector space.

4.3.2 RADIAL BASIS KERNEL (RBF)

RBF is a popular kernel function used commonly with SVM classification. The RBF kernel on two samples $\times$ and $\times'$, represented as feature vectors in some input space is defined as

$$\boldsymbol{K}(\times,\times') = \boldsymbol{exp}\left(-\frac{\|\times - \times'\|^2}{2\sigma^2}\right) \qquad (27)$$

where $\|\times - \times'\|^2$ is the squared Euclidean distance between the two feature vectors and σ is a free parameter.

4.3.3 PEARSON VII KERNEL (PUK)

PUK kernel is a Universal kernel function generally applied to SVM. PUK is very flexible and has possibility to change easily by adapting its parameters. Therefore it is possible to use Pearson VII kernel function as a general kernel which can replace the other kernel functions.

4.4 SIMULATION RESULTS

4.4.1 SIMULATION RESULTS FOR THE FEATURE SET

The set of features obtained from the fault dictionary are split into training and testing samples. A set of 1870 samples are obtained from the fault dictionary from which 1470 samples are used for training and 400 samples are used for testing. The training and testing samples are given as input to the SVM for classification. The SVM classification is performed for the Polynomial coefficients and the V-transform coefficients. Using confusion matrix the performance measures such as accuracy, error, precision, specificity and sensitivity are computed.

The confusion matrix consists of True Positive (TP), True Negative (TN), False Positive (FP) and the False Negative (FN) values from which the performance measures can be computed by using the formulas mentioned.

4.4.1.1 Accuracy

Accuracy is the proportion of the correctly identified samples to the total number of samples.

$$\boldsymbol{Accuracy = \frac{TP + TN}{TP + TN + FP + FN}} \qquad (12)$$

4.4.1.2 Error

Error is the deviation from accuracy or correctness. It is the difference between the observed or approximately determined value and the true value of a quantity in statistics.

$$\boldsymbol{Error = \frac{FP + FN}{TP + TN + FP + FN}} \qquad (13)$$

4.4.1.3 Sensitivity

Sensitivity measures the proportion of positives that are correctly identified. Recall is the True Positive Rate also referred to as Sensitivity.

$$\boldsymbol{Sensitivity = \frac{TP}{TP + FN}} \qquad (14)$$

4.4.1.4 Specificity

True Negative Rate is also called Specificity which is the same as the FP Rate. Specificity measures the proportion of negatives that are correctly identified.

$$\boldsymbol{Specificity = \frac{TN}{TN + FP}} \qquad (15)$$

4.4.1.5 Precision

Precision refers to the closeness of two or more measurements to each other. Precision is also referred to as the Positive Predictive Value (PPV).

$$\boldsymbol{Precision = \frac{TP}{TP + FP}} \qquad (16)$$

4.4.1.6 F-Measure

F-measure is a measure of a test's accuracy and is defines as the weighted harmonic mean of the precision and recall of he test.

$$F - Measure = \frac{2 \times precision \times sensitivity}{precision + sensitivity} \qquad (17)$$

The results obtained after the SVM classification with Polynomial coefficients and V-transform coefficients are compared and tabulated.

TABLE 4.1: COMPARISON OF RESULTS FOR THE FEATURE SET

Performance Measures	Polynomial Coefficients	V-Transform Coefficients
Accuracy	90.75%	81.75%
Error	9.25%	18.25%
Execution time	4.0938 s	3.875 s
Specificity	0.8664	0.8990
Sensitivity	0.0185	0.0950
Precision	0.0625	0.1825
F-measure	0.0285	0.1249

It is observed that the accuracy has improved for the Polynomial coefficients than the V-transform coefficients. By varying the sensitivity factor γ of VTC, there might be an increase in accuracy of V-transform coefficients.

4.4.2 SIMULATION RESULTS FOR THE FEATURE SUBSET

SVM classification is performed for the feature subsets obtained by using the Binary Bat algorithm. Using confusion matrix different performance measures are computed for the feature subsets and are compared with the feature sets.

TABLE 4.2: COMPARISON OF RESULTS FOR THE FEATURE SUBSETS USING BBA

Performance Measures	Polynomial Coefficients	V-Transform Coefficients
Accuracy	93.25%	87.50%
Error	6.75%	12.50%
Execution time	2.3906 s	2.0156 s
Specificity	0.8862	0.9001
Sensitivity	0.0812	0.1065
Precision	0.1800	0.1900
F-measure	0.1119	0.1365

It is observed that by SVM classification the accuracy and the various performance measures have improved for the feature subsets obtained by using Binary Bat Algorithm.

4.4.3 SIMULATION RESULTS FOR THE DIFFERENT KERNEL FUNCTIONS

The set of 1870 samples from the fault dictionary are used from which 1470 are used as training samples and the remaining 400 are used as testing samples. SVM classification is performed for the two feature sets consisting of Polynomial and V-transform coefficients using different kernel functions by varying the kernel parameters. The value of the exponent in the Polynomial kernel is chosen to be 1. The value of gamma in the Radial basis function is set as 0.01 and the values of omega and sigma in the PUK kernel function are chosen as 1. Therefore by varying the complexity parameter C in the ranges between 10^{-6} to 10^{6} and choosing the values for the insensitive loss function ε as 0.1, 0.001 and 10^{-12}, the accuracies for the kernel functions

are measured. Confusion matrix is used from which different performance measures between the kernel functions are compared. The following tables show the improved results for the varied parameters.

TABLE 4.3: SIMULATION RESULTS FOR THE POLYNOMIAL KERNEL FUNCTION WITH $\varepsilon = 0.1$

Performance Measures	$\varepsilon = 0.1$					
	$C = 0.001$		$C = 1$		$C = 10^3$	
	PC	VC	PC	VC	PC	VC
Accuracy	12.5%	12.5%	82.5%	84.25%	93%	91.25%
Time	0.01s	0.11s	0.02s	0.001s	0.01s	0.01s
Sensitivity	0.125	0.125	0.825	0.843	0.93	0.913
Specificity	0.125	0.125	0.025	0.023	0.01	0.013
Precision	0.016	0.016	0.844	0.881	0.94	0.928
F-measure	0.028	0.028	0.816	0.834	0.93	0.911

TABLE 4.4: SIMULATION RESULTS FOR THE POLYNOMIAL KERNEL FUNCTION WITH $\varepsilon = 0.001$

Performance Measures	$\varepsilon = 0.001$					
	$C = 0.001$		$C = 1$		$C = 10^3$	
	PC	VC	PC	VC	PC	VC
Accuracy	29%	31%	79.5%	83.25%	88%	93.25%
Time	0.01 s	0.01 s	0.01 s	0.01 s	0.02 s	0.02 s
Sensitivity	0.290	0.310	0.795	0.833	0.880	0.933
Specificity	0.101	0.099	0.029	0.024	0.017	0.010
Precision	0.196	0.235	0.868	0.879	0.915	0.946
F-measure	0.213	0.247	0.787	0.824	0.871	0.931

TABLE 4.5: SIMULATION RESULTS FOR THE POLYNOMIAL KERNEL FUNCTION WITH $\varepsilon = 10^{-12}$

Performance Measures	$\varepsilon = 10^{-12}$					
	C = 0.001		*C = 1*		*C =10^3^*	
	PC	**VC**	**PC**	**VC**	**PC**	**VC**
Accuracy	29%	31%	59%	84.5%	**95.5%**	**93.75%**
Time	0.05s	0.01s	0.01s	0.02s	0.02s	0.01s
Sensitivity	0.290	0.310	0.59	0.845	0.955	0.938
Specificity	0.101	0.099	0.059	0.022	0.006	0.009
Precision	0.196	0.235	0.597	0.891	0.967	0.951
F-measure	0.213	0.247	0.558	0.835	0.953	0.936

TABLE 4.6: SIMULATION RESULTS FOR THE RBF KERNEL FUNCTION WITH $\varepsilon = 0.1$

Performance Measures	$\varepsilon = 0.1$					
	C = 0.001		*C = 1*		*C =10^3^*	
	PC	**VC**	**PC**	**VC**	**PC**	**VC**
Accuracy	12.5%	12.5%	41.25%	39%	82.75%	91.75%
Time	0.01s	0.02s	0.39s	0.38s	0.09s	0.06s
Sensitivity	0.125	0.125	0.413	0.39	0.828	0.918
Specificity	0.125	0.125	0.084	0.087	0.025	0.012
Precision	0.016	0.016	0.519	0.440	0.857	0.934
F-measure	0.028	0.028	0.406	0.333	0.816	0.916

TABLE 4.7: SIMULATION RESULTS FOR THE RBF KERNEL FUNCTION WITH $\varepsilon = 0.001$

Performance Measures	$\varepsilon = 0.001$					
	$C = 0.001$		$C = 1$		$C = 10^3$	
	PC	VC	PC	VC	PC	VC
Accuracy	29%	31%	49%	60.25%	91.75%	92%
Time	0.45 s	0.45 s	0.39 s	0.38 s	0.13 s	0.09 s
Sensitivity	0.290	0.310	0.490	0.603	0.918	0.920
Specificity	0.101	0.099	0.073	0.057	0.012	0.011
Precision	0.196	0.235	0.552	0.600	0.941	0.937
F-measure	0.213	0.247	0.453	0.565	0.911	0.919

TABLE 4.8: SIMULATION RESULTS FOR THE RBF KERNEL FUNCTION WITH $\varepsilon = 10^{-12}$

Performance Measures	$\varepsilon = 10^{-12}$					
	$C = 0.001$		$C = 1$		$C = 10^3$	
	PC	VC	PC	VC	PC	VC
Accuracy	29%	31%	36.5%	38.75%	**92.25%**	**93.25%**
Time	0.63s	0.44s	0.39s	0.38s	0.06s	0.06s
Sensitivity	0.290	0.310	0.365	0.388	0.923	0.933
Specificity	0.101	0.099	0.091	0.088	0.011	0.010
Precision	0.196	0.235	0.336	0.322	0.946	0.936
F-measure	0.213	0.247	0.319	0.327	0.916	0.932

TABLE 4.9: SIMULATION RESULTS FOR HE PUK KERNEL FUNCTION WITH $\varepsilon = 0.1$

Performance Measures	$\varepsilon = 0.1$					
	$C = 0.001$		$C = 1$		$C = 10^3$	
	PC	VC	PC	VC	PC	VC
Accuracy	12.5%	12.5%	96.25%	96.75%	98.5%	97.5%
Time	0.03s	0.02s	0.23s	0.14s	0.13s	0.11s
Sensitivity	0.125	0.125	0.963	0.968	0.985	0.975
Specificity	0.125	0.125	0.005	0.005	0	0.003
Precision	0.016	0.016	0.966	0.973	1	0.982
F-measure	0.028	0.028	0.962	0.967	0.992	0.977

TABLE 4.10: SIMULATION RESULTS FOR THE PUK KERNEL FUNCTION WITH $\varepsilon = 0.001$

Performance Measures	$\varepsilon = 0.001$					
	$C = 0.001$		$C = 1$		$C = 10^3$	
	PC	VC	PC	VC	PC	VC
Accuracy	41.5%	48.5%	92.75%	96%	99.25%	98%
Time	0.59 s	0.61 s	0.2 s	0.14 s	0.16 s	0.14 s
Sensitivity	0.415	0.485	0.928	0.960	0.993	0.980
Specificity	0.084	0.074	0.010	0.006	0.001	0.001
Precision	0.328	0.358	0.952	0.967	0.993	0.991
F-measure	0.328	0.380	0.923	0.959	0.992	0.984

TABLE 4.11: SIMULATION RESULTS FOR THE PUK KERNEL FUNCTION WITH $\varepsilon = 10^{-12}$

Performance Measures	$\varepsilon = 10^{-12}$					
	C = 0.001		*C = 1*		*C =10³*	
	PC	**VC**	**PC**	**VC**	**PC**	**VC**
Accuracy	41.5%	48.5%	93%	97%	**100%**	**97.75%**
Time	1s	0.59s	0.13s	0.08s	0.05s	0.03s
Sensitivity	0.415	0.485	0.93	0.97	1	0.978
Specificity	0.084	0.074	0.010	0.004	0	0.003
Precision	0.328	0.358	0.953	0.974	1	0.981
F-measure	0.328	0.38	0.925	0.969	1	0.977

The following figures show the variations in accuracies obtained for the different kernel functions for the values of C equal to 0.001, 1 and 1000 and for the ε values equal to 0.1, 0.001 and 10^{-12} between the two feature sets consisting of Polynomial coefficients and the V-transform coefficients.

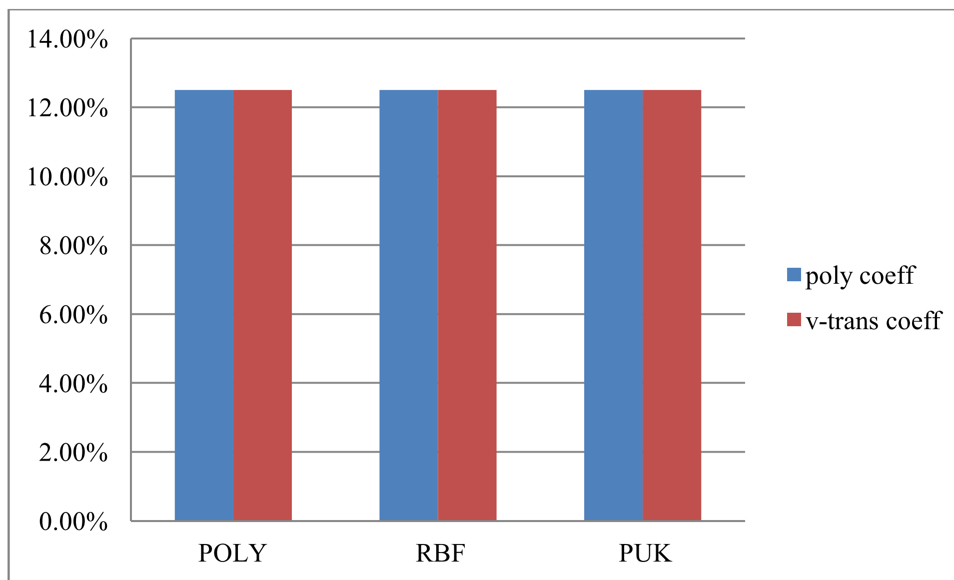


FIG 4.5: ACCURACY PLOT FOR POLYNOMIAL VS V-TRANSFORM COEFFICIENTS WITH C = 0.001 AND $\varepsilon = 0.1$

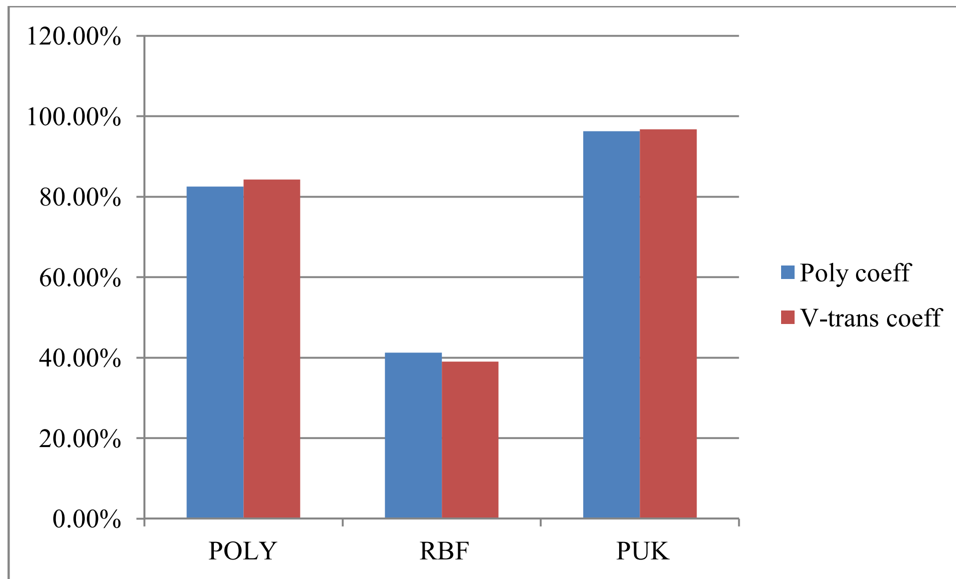


FIG 4.6: ACCURACY PLOT FOR POLYNOMIAL VS V-TRANSFORM COEFFICIENTS WITH C = 1 AND $\varepsilon = 0.1$

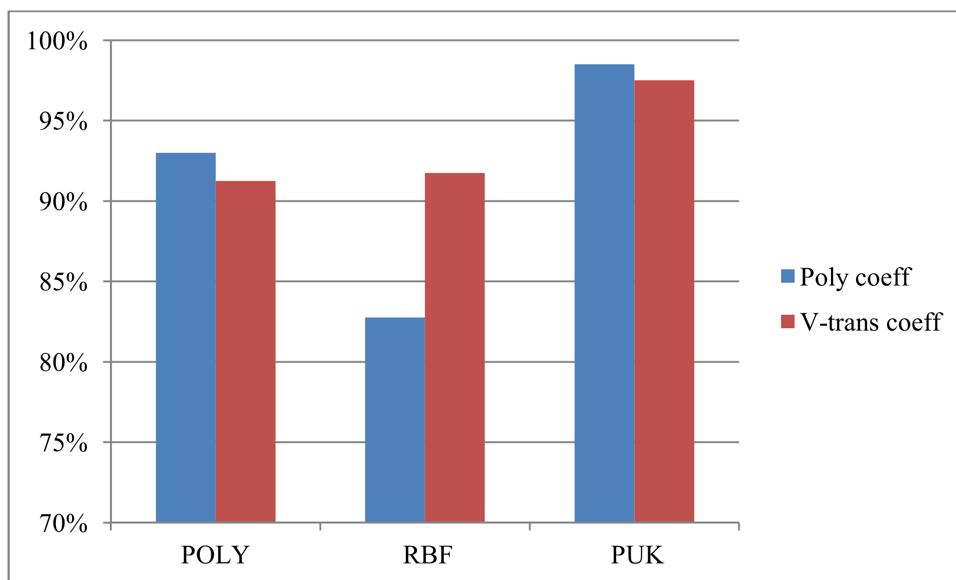


FIG 4.7: ACCURACY PLOT FOR POLYNOMIAL VS V-TRANSFORM COEFFICIENTS WITH C = 1000 AND $\varepsilon = 0.1$

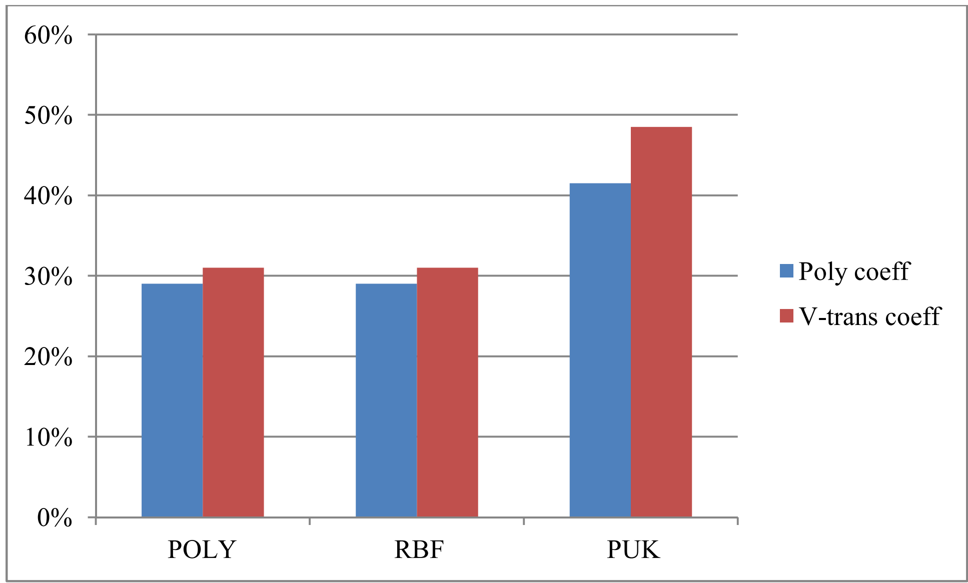


FIG 4.8: ACCURACY PLOT FOR POLYNOMIAL VS V-TRANSFORM COEFFICIENTS WITH C = 0.001 AND $\varepsilon = 0.001$

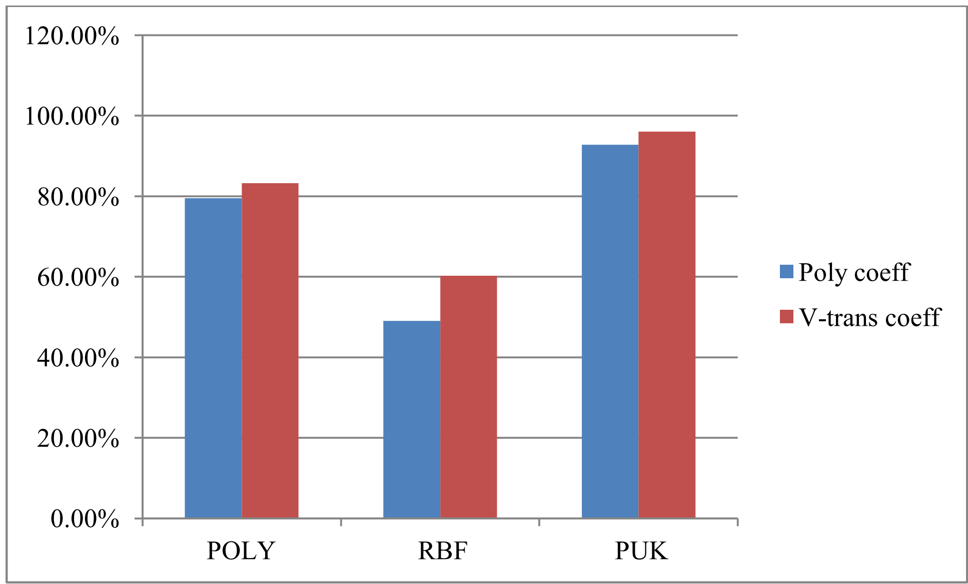


FIG 4.9: ACCURACY PLOT FOR POLYNOMIAL VS V-TRANSFORM COEFFICIENTS WITH C = 1 AND $\varepsilon = 0.001$

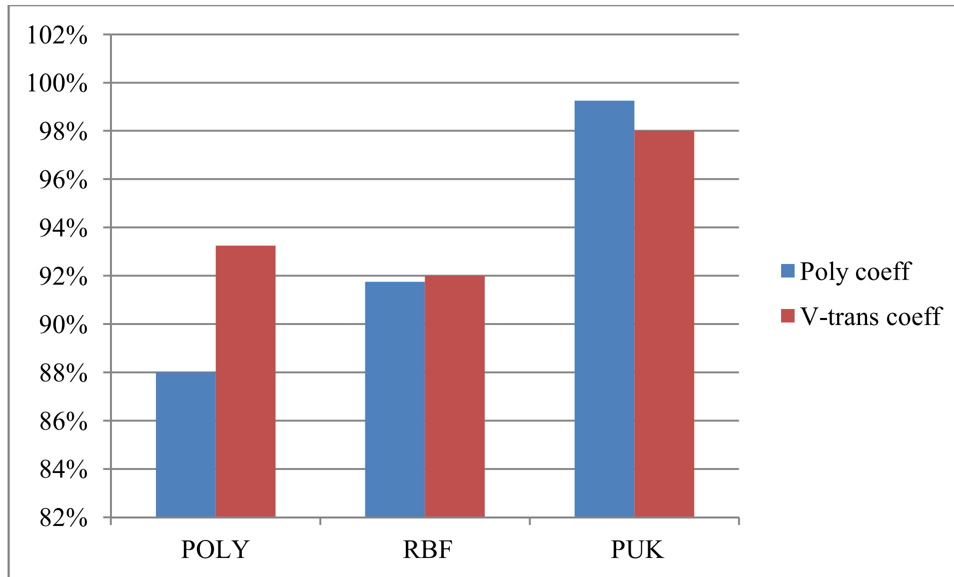


FIG 4.10: ACCURACY PLOT FOR POLYNOMIAL VS V-TRANSFORM COEFFICIENTS WITH C = 10^3 AND $\varepsilon = 0.001$

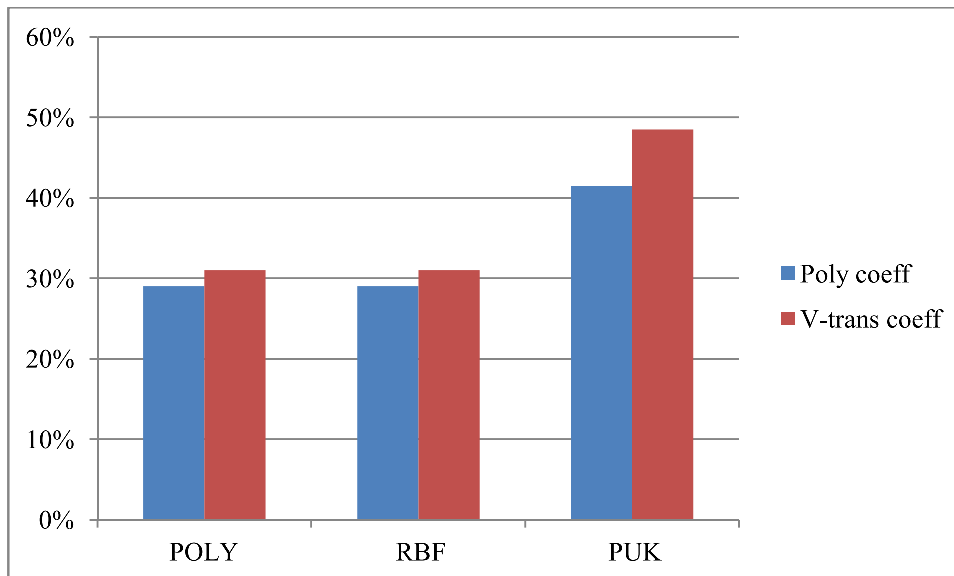


FIG 4.11: ACCURACY PLOT FOR POLYNOMIAL VS V-TRANSFORM COEFFICIENTS WITH C = 0.001 AND $\varepsilon = 10^{-12}$

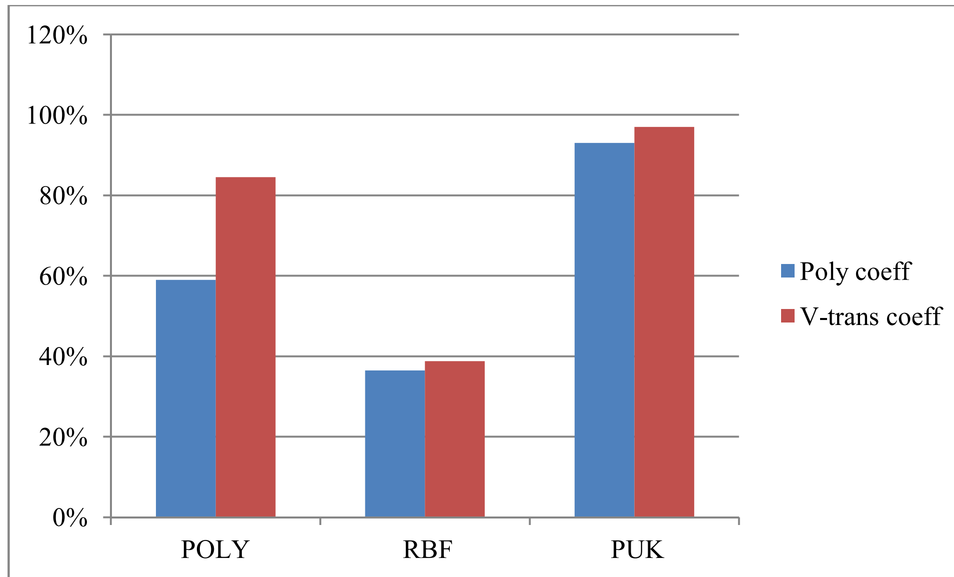


FIG 4.12: ACCURACY PLOT FOR POLYNOMIAL VS V-TRANSFORM COEFFICIENTS WITH C = 1 AND $\varepsilon = 10^{-12}$

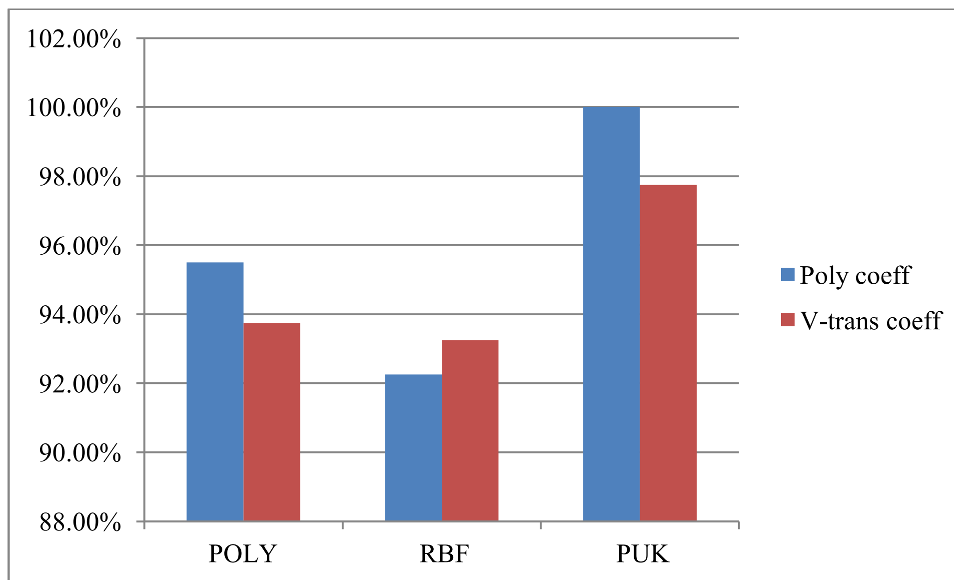


FIG 4.13: ACCURACY PLOT FOR POLYNOMIAL VS V-TRANSFORM COEFFICIENTS WITH C = 10^3 AND $\varepsilon = 10^{-12}$

It is observed that the SVM based on Polynomial kernel, RBF kernel and PUK kernel shows the similar performance on mapping the relation between the input and the output data. A comparison of simulation results for SVM classification based on POLY, RBF and PUK kernels show that the accuracy increases for both the Polynomial coefficients and the V-transform coefficients by choosing the complexity parameter value as 10^3 and the insensitive loss function ε value as 10^{-12}. This shows that for a maximum value of C and a minimum value of ε, the support vectors obtained will be maximum. Also from the comparison, PUK kernel produces higher classification accuracy compared to the Polynomial kernel and RBF kernel functions.

CHAPTER 5
HIERARCHICAL SUPPORT VECTOR MACHINES

5.1 OVERVIEW

A Hierarchical Support Vector Machine (HSVM) is used for multi-class classification problems. HSVM is a tree-structured network of support vector machine (SVM). HSVM is a decision tree with a SVM at each node. At the root node of the decision tree, all classes will be available for prediction. The number of classes available for prediction keeps decreasing as the tree gets descended.

Before learning the SVM classifier at a node, the classes available at that node are partitioned into two using a max-cut unsupervised decomposition. The classification model to distinguish between the two class partitions is then learnt using the bipartite decomposition as two-class input to train the SVM at that node.

During classification, one traverses the decision tree from the root and keeps applying the SVM classifier at each visited node until one reaches a leaf node indicating the output class. It has been shown that HSVM uses distance measures to exploit the natural class groupings, the hierarchical structure results in a fast and intuitive SVM training process that requires little running and gives high classification accuracy and good generalization.

The speed and accuracy of HSVM depends on its tree structure. HSVM overcomes the disadvantages of one-against-one and one-against-all classifiers by using a binary hierarchical classification structure.

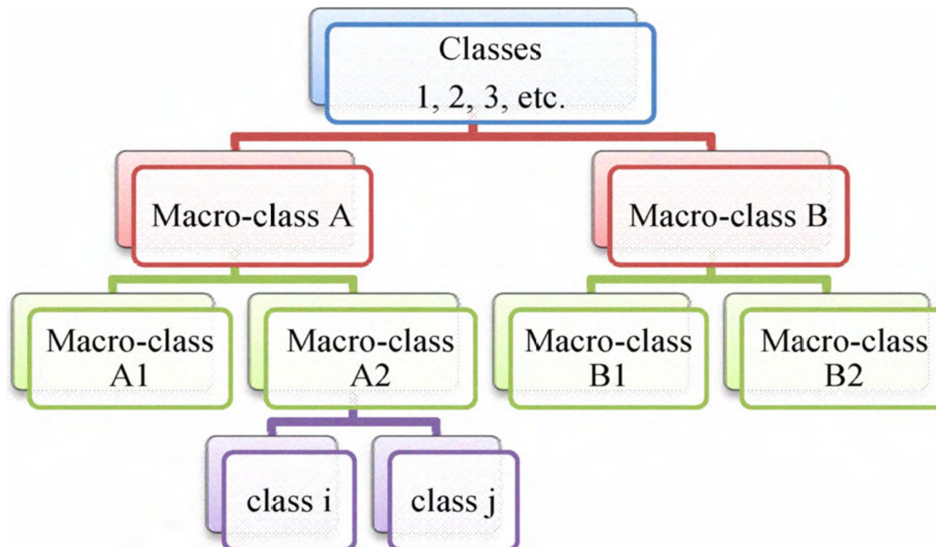


FIG 5.1: AN EXAMPLE OF HIERARCHICAL CLASSIFICATION STRUCTURE

The structure makes a coarse separation between classes at the upper levels and finer decisions are made at the lower levels. At the top node, the original k classes are divided into two smaller groups of classes called macro-classes. This procedure is repeated in subsequent levels, until there is only one class in the final sub-group. This method decomposes the original problem into k-1 binary sub-problems. In testing, a hard decision is made at each node, i.e. the test input is always assigned to one of the macro-classes at each node. If both outputs are smaller than some threshold, the test input is rejected as a false class. Thus only log2K classifiers are required to traverse a path from top to bottom.

The drawback of a hard-decision hierarchical classifier is that if a miss or misclassification error occurs at some internal node, it cannot be corrected in the subsequent levels, since its correct path will not be visited. Therefore a soft-decision hierarchical classification strategy is normally used to address this problem.

5.2 HIERARCHICAL CLASSIFIER DESIGN

The hierarchical clustering (i.e. the macro-class selection) at each node in the hierarchy should not be done arbitrarily or by intuition. There are two different design approaches for the macro-class selection at each node.

They are:

- **Agglomerative (bottom-up) approach**
- **Divisive (top-down) approach**

Of the two approaches, top-down is normally used for hierarchical classifiers, as bottom-up design does not provide a good separation of classes at each node.

5.3 CLUSTERING RULES

In HSVM, a set of classes is generally separated into two macro-classes at each node, where each class is present in only one of the two macro-classes. There are three general clustering rules in HSVM.

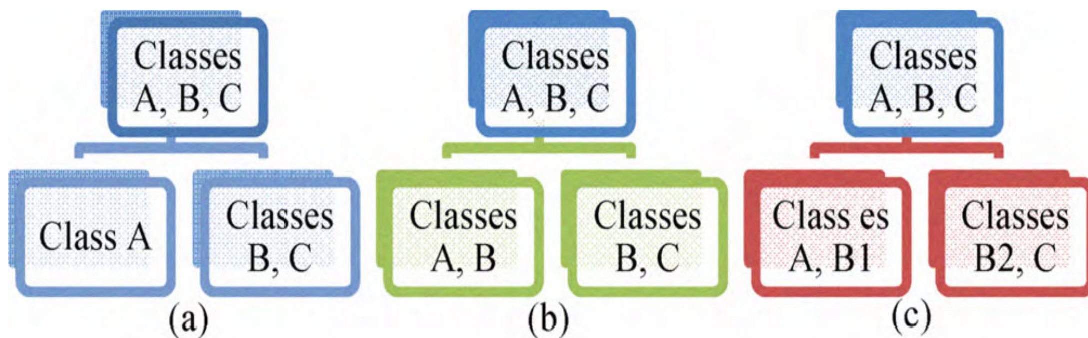


FIG 5.2: ILLUSTRATION OF THE THREE CLUSTERING RULES

In fig (a), each class always presents in only one macro-class in each level. This method decomposes the original k class problem into k-1 binary sub-class problems.

In fig (b), each class may be entirely present in more than one macro-class in each level. This method results in overlapping macro-class pairs at many nodes in a hierarchy.

In fig (c), each class may be divided into several macro-classes in each level (i.e) class B splits into B1 and B2. This method provides better clustering and classification results, but it will increase the width and depth of the tree (i.e) number of levels.

From the three clustering rules, the rule in fig (a) is generally chosen, as it involves only k-1 classifiers for k class problem.

5.4 SIMULATION RESULTS

Hierarchical SVM classification is performed for the two feature sets consisting of the polynomial coefficients and the v-transform coefficients. HSVM splits the training samples into training and the testing samples. Out of 1870 samples from the fault dictionary, 1470 samples are selected as training samples and the remaining 400 samples are chosen as testing samples. From the 1470 training samples, a maximum of 100 rows of samples are chosen for each class, thereby choosing 800 rows of samples for eight classes.

From 800 samples, 75% of the samples are chosen as training samples and the remaining 25% of the samples are chosen as testing samples. HSVM classification is then performed for the two feature sets consisting of 800 rows of samples.

TABLE 5.1: HSVM CLASSIFICATION

Feature set	Training Accuracy	Testing Accuracy
Polynomial coefficients	99.75%	99.5%
V-transform coefficients	99.83%	100%

CHAPTER 6

SUMMARY

In the proposed work, analog fault diagnosis in filter circuit is performed based on the Polynomial coefficients and the V-transform coefficients. Two feature sets are extracted from the three op-amp biquad filter used as the circuit under test. Support Vector Machine (SVM) classification is performed for the two feature sets. By using Binary Bat Algorithm, feature subset selection is obtained for the two feature sets. SVM classification is performed for the feature subset and the results are compared with the feature set. Three kernel functions such as the Polynomial kernel, Radial Basis kernel and the Pearson VII kernel functions of SVM are then utilized. Choosing the default values of the kernel functions such as 1 for exponent value in Polykernel, 1 for omega and sigma values in PUK kernels and choosing 0.01 for gamma in RBF kernels and by varying the kernel parameters of SVM such as c and epsilon, the classification accuracy is increased for the feature sets. Different performance measures are computed using the confusion matrix and are compared between the feature sets, feature subsets and for the feature sets using the different kernel functions. Hierarchical SVM classification is then performed for the feature sets to attain better classification accuracy.

The table shows the comparison of accuracies obtained between the feature sets and the subsets.

TABLE 6.1: ACCURACY COMPARISON BETWEEN THE FEATURE SETS AND THE SUBSETS USING BBA

Features	Feature Set	Feature subset using BBA
Polynomial coefficients	90.75%	93.25%
V-transform coefficients	81.75%	87.50%

It is noticed that the accuracy has increased for the feature subsets obtained by using BBA than the feature sets.

By varying the kernel parameters c and epsilon of SVM for the varied kernel functions, it is observed that the accuracy increases for all the kernel functions by choosing the value of $c = 10^3$ and $\varepsilon = 10^{-12}$. The table shows the comparison of results between the different kernel functions.

TABLE 6.2: ACCURACY COMPARISON BETWEEN THE KERNEL FUNCTIONS

Features	POLY kernel	RBF kernel	PUK kernel
Polynomial coefficients	95.5%	92.25%	100%
V-transform Coefficients	93.75%	93.25%	97.75%

From the comparison, it is examined that the accuracy has increased for the Pearson VII kernel function.

The increased classification accuracy obtained by using Hierarchical SVM for the feature sets with 800 rows of training and testing samples is shown in the table.

TABLE 6.3: TRAINING AND TESTING ACCURACY OF HSVM

Feature set	Training Accuracy	Testing Accuracy
Polynomial coefficients	99.75%	99.5%
V-transform coefficients	99.83%	100%

It is observed that the accuracy increases for the training and the testing samples of V-transform coefficients by using HSVM.

CHAPTER 7

CONCLUSION

Two analog circuit test schemes are proposed for high resolution fault detection. The first scheme uses Polynomial coefficients of the circuit's frequency response for fault detection. The second scheme uses a transformation on the polynomial coefficients (i.e.) V-Transform coefficients for fault detection. An experimental validation of the test scheme on the biquad filter and based on the SVM classification shows that polynomial coefficient provides better accuracy than the V-Transform coefficient. By changing the sensitivity factor of VTC, there might be an increase in accuracy of V-Transform coefficient. Feature subset selection is obtained using an evolutionary algorithm known as the Binary Bat Algorithm (BBA). SVM classification performed for the feature subset yields better classification results than that obtained for the feature sets. By varying the kernel parameters C and epsilon for the three kernel functions such as Polykernel, RBF kernel and the PUK kernel functions, the classification accuracy for the feature sets are increased. It is also observed that the PUK kernel function yields better accuracy compared to the other two kernel functions. The classification accuracy is further increased by using the Hierarchical SVM for the feature sets containing Polynomial and V-transform coefficients.

REFERENCES

[1] Sindia, S., Singh, V., & Agrawal, V. D. (2009, November). Multi-tone testing of linear and nonlinear analog circuits using polynomial coefficients. In *Asian Test Symposium, 2009. ATS'09.* (pp. 63-68). IEEE.

[2] Sindia, S., Singh, V., & Agrawal, V. D. (2010, January). Parametric fault diagnosis of nonlinear analog circuits using polynomial coefficients. In *VLSI Design, 2010. VLSID'10. 23rd International Conference on* (pp. 288-293). IEEE.

[3] Kumar, A., & Singh, A. P. (2013). Neural network based fault diagnosis in analog electronic circuit using polynomial curve fitting. *International Journal of Computer Applications*, *61*(16).

[4] Sindia, S., Singh, V., & Agrawal, V. D. (2009, May). Polynomial coefficient based DC testing of non-linear analog circuits. In *Proceedings of the 19th ACM Great Lakes symposium on VLSI* (pp. 69-74). ACM.

[5] Sindia, S., Agrawal, V. D., & Singh, V. (2011, May). Non-linear analog circuit test and diagnosis under process variation using V-transform coefficients. In *VLSI Test Symposium (VTS), 2011 IEEE 29th* (pp. 64-69). IEEE.

[6] Sindia, S., Agrawal, V. D., & Singh, V. (2012). Parametric fault testing of non-linear analog circuits based on polynomial and V-transform coefficients. *Journal of Electronic Testing*, *28*(5), 757-771.

[7] Anis, S., El-mahlawy, M. H., Gadallah, M. E., & El-samahy, E. A. (2014). Parametric Fault Detection of Analogue Circuits. *International Journal of Computer Applications*, *96*(9).

[8] Yi, Z., Xueye, W., & Haifeng, J. (2008). One-class classifier based on SBT for analog circuit fault diagnosis. *Measurement*, *41*(4), 371-380.

[9] Taha, A. M., Mustapha, A., & Chen, S. D. (2013). Naive bayes-guided bat algorithm for feature selection. *The Scientific World Journal*, *2013*.

[10] Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. *Nature inspired cooperative strategies for optimization (NICSO 2010)*, 65-74.

[11] Nakamura, R. Y., Pereira, L. A., Costa, K. A., Rodrigues, D., Papa, J. P., & Yang, X. S. (2012, August). BBA: a binary bat algorithm for feature selection. In *Graphics, Patterns and Images (SIBGRAPI), 2012 25th SIBGRAPI Conference on* (pp. 291-297). IEEE.

[12] Cui, J., & Wang, Y. (2011). A novel approach of analog circuit fault diagnosis using support vector machines classifier. *Measurement*, *44*(1), 281-289.

[13] Grzechca, D., & Rutkowski, J. (2009). Fault diagnosis in analog electronic circuits-the SVM approach. *Metrol. Meas. Syst*, *16*(4), 583-598.

[14] Zhao, M., Fu, C., Ji, L., Tang, K., & Zhou, M. (2011). Feature selection and parameter optimization for support vector machines: A new approach based on genetic algorithm with feature chromosomes. *Expert Systems with Applications*, *38*(5), 5197-5204.

[15] Frohlich, H., Chapelle, O., & Scholkopf, B. (2003, November). Feature selection for support vector machines by means of genetic algorithm. In *Tools with Artificial Intelligence, 2003. Proceedings. 15th IEEE International Conference on* (pp. 142-148). IEEE.

[16] Ardjani, F., Sadouni, K., & Benyettou, M. (2010, November). Optimization of SVM multiclass by particle swarm (PSO-SVM). In *Database Technology and Applications (DBTA), 2010 2nd International Workshop on* (pp. 1-4). IEEE.

[17] Li, G. Z., Yang, J., Liu, G. P., & Xue, L. (2004). Feature selection for multi-class problems using support vector machines. *PRICAI 2004: Trends in Artificial Intelligence*, 292-300.

[18] Abakar, K. A., & Yu, C. (2014). Performance of SVM based on PUK kernel in comparison to SVM based on RBF kernel in prediction of yarn tenacity.

[19] Wang, Y. C. F., & Casasent, D. (2009, June). A support vector hierarchical method for multi-class classification and rejection. In *Neural Networks, 2009. IJCNN 2009. International Joint Conference on* (pp. 3281-3288). IEEE.

[20] Hsu, C. W., & Lin, C. J. (2002). A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, *13*(2), 415-425.

[21] Sahbi, H., & Geman, D. (2006). A hierarchy of support vector machines for pattern detection. *Journal of Machine Learning Research*, *7*(Oct), 2087-2123.

[22] Baudat, G., & Anouar, F. (2001). Kernel-based methods and function approximation. In *Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on* (Vol. 2, pp. 1244-1249). IEEE.

[23] Huang, T. M., Kecman, V., & Kopriva, I. (2006). *Kernel based algorithms for mining huge data sets* (Vol. 1). Heidelberg: Springer.

[24] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273-297.

[25] Mohammad, M. N., Sulaiman, N., & Khalaf, E. T. (2011). A novel local network intrusion detection system based on support vector machine. *Journal of Computer Science*, *7*(10), 1560.

[26] Maji, S., Berg, A. C., & Malik, J. (2008, June). Classification using intersection kernel support vector machines is efficient. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (pp. 1-8). IEEE.

[27] Haasdonk, B. (2005). Feature space interpretation of SVMs with indefinite kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(4), 482-492.

[28] Ektefa, M., Sidi, F., Ibrahim, H., Jabar, M. A., & Memar, S. (2011). A comparative study in classification techniques for unsupervised record linkage model. *Journal of Computer Science*, *7*(3), 341.

[29] Üstün, B., Melssen, W. J., & Buydens, L. M. (2006). Facilitating the application of Support Vector Regression by using a universal Pearson VII function based kernel. *Chemometrics and Intelligent Laboratory Systems*, *81*(1), 29-40.

[30] Tsang, I. W., Kwok, J. T., & Cheung, P. M. (2005). Core vector machines: Fast SVM training on very large data sets. *Journal of Machine Learning Research*, *6*(Apr), 363-392.

[31] Liu, X. Y., Jiang, H. Y., & Tang, F. Z. (2010). Parameters optimization in SVM based-on ant colony optimization algorithm. In *Advanced Materials Research* (Vol. 121, pp. 470-475). Trans Tech Publications.

[32] Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine learning*, *46*(1), 389-422.

[33] Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., & Vapnik, V. (2000, January). Feature selection for SVMs. In *Proceedings of the 13th International Conference on Neural Information Processing Systems* (pp. 647-653). MIT Press.

[34] Sindia, S., Singh, V., & Agrawal, V. D. (2009, September). V-transform: an enhanced polynomial coefficient based DC test for non-linear analog circuits. In *Proc. of 8th IEEE East West Design & Test Symposium* (pp. 283-286).

PUBLICATIONS

- Published a paper titled **"A comparative Study of SVM Kernel Functions Based on Polynomial Coefficients and V-Transform Coefficients,"** in the International Journal of Engineering and Computer Science in Volume 6 Issue 3 on March 2017.

- Presented a paper titled **"Analog Fault Detection in Filter Circuit Using Binary Bat Algorithm and SVM Principle,"** in the **IEEE International Conference on Science, Technology, Engineering and Management (ICSTEM'17)** at Kalaignarkarunanidhi Institute of Technology held on 3rd and 4th March 2017.

International Journal of Engineering and Computer Science
ISSN : 2319-7242
www.ijecs.in

Certificate Of Publication

This to Certify That the Research Paper entitled

A Comparative Study of SVM Kernel Functions Based on Polynomial Coefficients and V-Transform Coefficients

Author By

D.Shanthini

Have published in ***Volume 6 Issue 3 March 2017***

In

International Journal of Engineering And Computer Science

The Mentioned Research paper is measured up to the required standard

Editor In chief
Dr Dinesh Soulnki
IJECS PUBLICATION
WWW.IJECS.IN

KIT - Kalaignarkarunanidhi Institute of Technology

(Accredited with 'A' Grade by NAAC)
(Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai)
KIT Global Institute for Advanced Studies & Research
Coimbatore-641 402, Tamilnadu, India.

Certificate of Appreciation

This is to certify that ~~Dr./Mr.~~/Ms./~~Prof~~ SHANTHINI. D

of KUMARAGURU COLLEGE OF TECHNOLOGY *has presented a paper titled* ANALOG FAULT DETECTION IN FILTER CIRCUIT USING BINARY BAT ALGORITHM AND SVM PRINCIPLE *in ICMR sponsored* ***IEEE International Conference on Science, Technology, Engineering and Management (ICSTEM'17)*** *held on 3rd and 4th March, 2017 at KIT - Kalaignarkarunanidhi Institute of Technology, Coimbatore, Tamilnadu, India.*

Dr. S. Santhi	**Dr. P.Anbalagan**	**Dr. N.Mohan Das Gandhi**
Organizing Chair	*Director*	*Principal*

In Association with