

# **Progress Report**

*(Nov04-Oct 05)*

**Project Title:**

**Modeling, Classification and Fault Detection  
of Sensors using Intelligent Methods**



**Principal Investigator**

**Prem K Kalra**

**Department of Electrical Engineering,  
Indian Institute of Technology, Kanpur  
INDIA**

## Executive Summary

The project report deals with various methodologies for testing and classification of sensors especially accelerometer and gyro. The objective the project is to develop platform so that data collected from the laboratory about sensors can be used by the software platform to make decisions about performance of a sensor. The techniques selected for classification of sensors are broadly in two classes i.e. time domain and frequency domain.

These techniques should demonstrate the qualities like robustness and reliability. The selection of these techniques has done based on study conducted during pre-project. The list of techniques short listed is the following:

### ? Time domain techniques

#### ✍ Statistical techniques

- ✚ Mean, Variance, Skew ness and Kurtosis
- ✚ Multivariate regression methods for classification
- ✚ Principal Component Approach(PCA)

#### ✍ Neural Network Based

##### ✚ Neural Network based classifier

- ? Multiplicative Neuron based
- ? Sigma-Pi Neuron based
- ? Conventional Neuron based

##### ✚ Neural Network based Principal Component Algorithms

- ? Linear PCA based
- ? Non-linear PCA based

- o Adaptive Principal Component Analysis Using Lateral Inhibition
- o Generalized Hebbian Learning algorithm (*Nonlinear PCA*)

##### ✚ Neural Network based Independent Component Algorithms

- ? Fast ICA

#### ✍ Synergistic Approach using neural network and fuzzy logic

- ✚ SANFIS
- ✚ ANFIS

#### ✍ Decision Tree

- ✚ ID3
- ✚ Probabilistic
- ✚ Possibilistic

## ✍ Frequency Domain Approaches

- ✚ Spectrum Analysis

- ✚ Cepstrum Analysis

In addition to above mentioned techniques, it was considered during the pre-project study that the software simulator must include graphic use interface (GUI), preprocessing ,post processing techniques , data management scheme and plotting of graphs.

In first few months the development of techniques mentioned above has been carried out in MATLAB so that the understanding of algorithms and to build confidence on the performance of the algorithms can be achieved. Later on development work has been started in JAVA so that its benefits like portability etc can be exploited and dependence on standard software can be avoided.

So far number of algorithms has been tested on the data provided by ISRO with good success rate. Some especially frequency-domain algorithms based have to developed and tested. It is understood that project would be completed in time. It may be possible to permit the extension of six-month time which would facilitate to give shape to simulator through rigorous testing. It is expected that the simulator would be comparable to any proficient tool.

# Progress Report

(Nov 2004-Oct 2005)

## Project Title: Modeling, Classification and Fault Detection of Sensors using Intelligent Methods

### 1. Objectives

The objectives of this project is to identify, understand and analyze the parameters affecting the system response this is done by modeling the adaptive systems using soft computing and intelligent methods such as ANN, Fuzzy Logic, SVM and Probabilistic techniques.

This report summarizes the work carried out so far. Progress Report is organized in the following manner. There are four sections. First section focuses on the brief details of tools developed using JAVA so far. Then the second section deals with the GUI-and-APPLETS part of the project and describes the basic architecture of the module. Results on sensor data with the help of developed algorithms have been discussed in the third section. Finally, fourth section deals with the future plans.

### 2. Algorithms Developed So Far

This section describes the progress of the work from *July 2004 to Nov 2004*. At the initial stage of the project the algorithms were implemented using MATLAB®. In this span of time the algorithms developed in MATLAB were implemented using JAVA.

#### 2.1 Brief introduction of first and second progress report

The details of algorithms developed in MATLAB are submitted with second progress report from *July 2004 to Nov 2004*. The Following algorithms were developed and tested in MATLAB.

✍ Statistical Tools and Indicators

✍ Decision Tree

✍ Artificial Neural Network]

1. *Conventional ANN*
2. *ANN with Multiplicative Neuron*
3. *ANN with Sigma-Pi-Sigma Neuron Model*
4. *ANN with Sigma-Pi-Pi Neuron Model*

✍ Neuro-Fuzzy Systems

1. *SANFIS*

✍ Principal Component Analysis (PCA)

✍ Support Vector Machines

✍ Inference System using Least Square Curve Fitting Technique in combination with ANN

## **2.2 Details of Third progress report**

The algorithms mentioned in first and second progress report were implemented using JAVA. A GUI similar to MATLAB GUI is also implemented using the JAVA based applet programming. Along with this few new algorithms were also developed and embedded in the form of a complete module. The implementation of algorithms using JAVA causes their faster operation and gives a platform independent environment.

In this report few new algorithms for decision tree analysis, Principal Component Analysis (PCA) and Independent Component Analysis (ICA) were specifically included that results in better solution.

The details of algorithms which are developed using JAVA are listed below:

✍ Statistical Tools and Indicators

✍ Decision Tree :

1. *Probabilistic decision trees*

2. *Possibilistic decision trees*

✍ Artificial Neural Network

1. *Conventional ANN*

2. *ANN with Multiplicative Neuron*

✍ Principal Component Analysis (PCA):

1. *APEX(Adaptive Principal Component Analysis Using Lateral Inhibition) (Nonlinear PCA)*

2. *GHA (Generalized Hebbian Learning algorithm) (Nonlinear PCA)*

✍ Support Vector Machines

- ✗ Independent Component Analysis: *This has been implemented using JAVA and working well for some classical problems. It will be tested for IISU application.*

### 3. Graphical-User-Interface (GUI)

A Graphical-User-Interface (GUI) has been developed using JAVA. The algorithms were run in background. This GUI supports a user friendly environment and it is as easy as developed in MATLAB. Similar to MATLAB based GUI there is a main GUI on which there are links for various tools. In following sub sections the screen shots and details of the GUI environment is described.

#### 3.1 Main GUI:

Main GUI consists of push buttons. Each item brings a new GUI corresponding to a particular tool. The options for various tools are to be selected from the main menu which looks like as shown below:

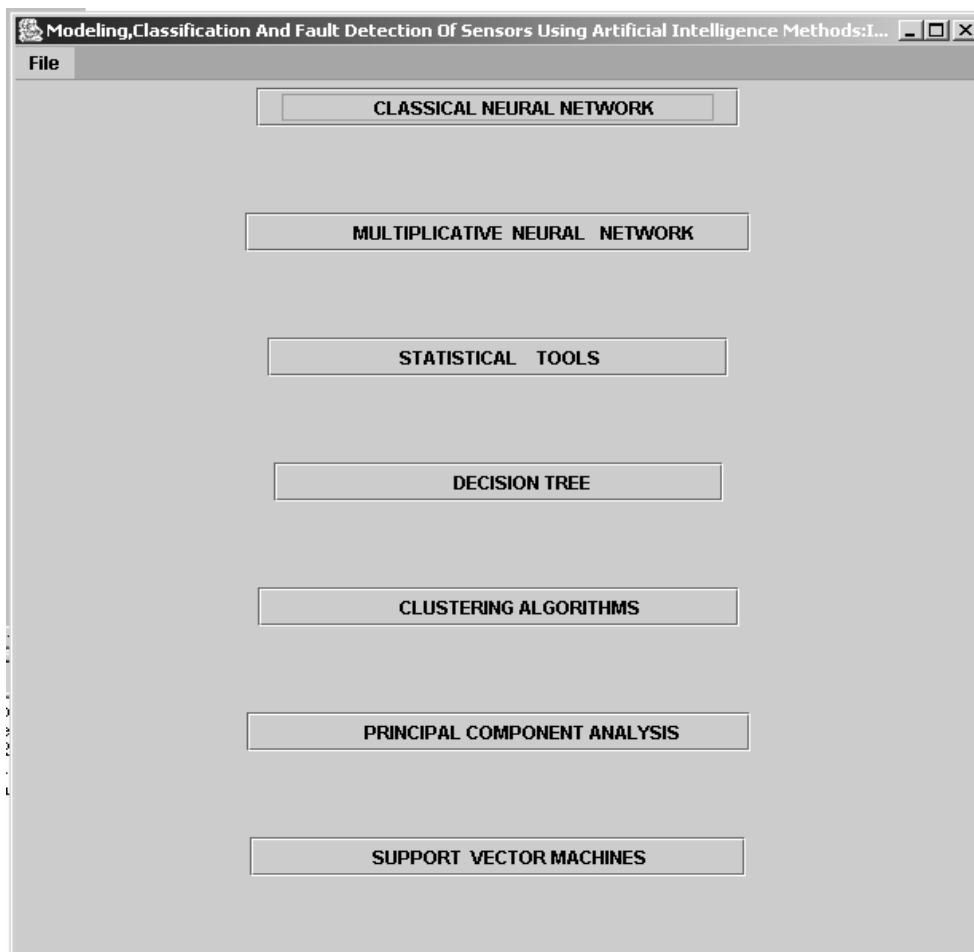


Fig. 1 Screenshot of the Main GUI

### 3.2 GUI for Classical Neural Network

As the user clicks on the Network Architecture the following window is displayed. Here we can enter the number of inputs ,outputs and number of hidden layers.

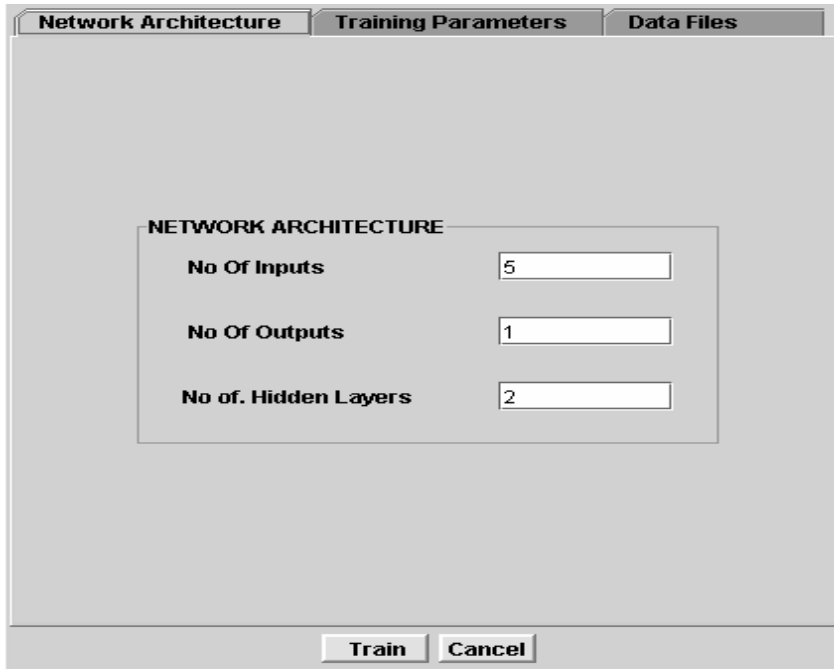


Fig. 2 (a) Screenshot of the GUI for classical neural network

Similarly various training parameters can also be provided through the GUI below:

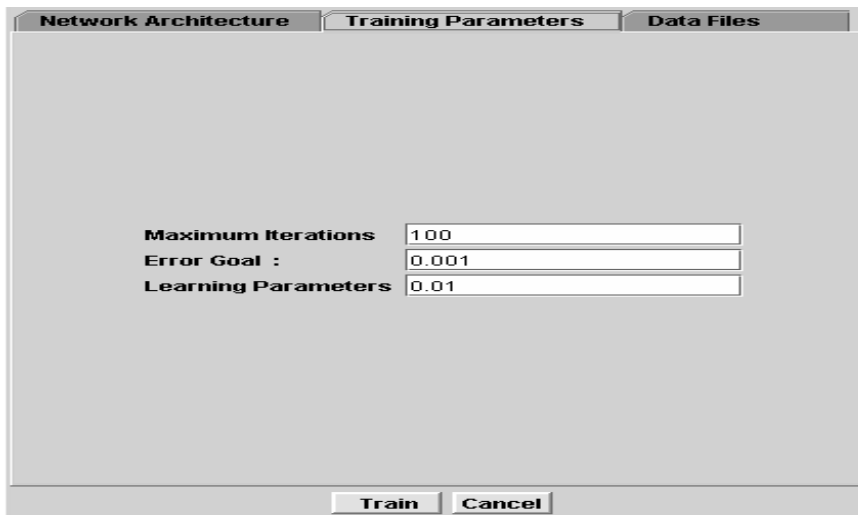


Fig. 2 (b) Screenshot of the GUI for classical neural network

### 3.3 GUI for Statistical Tools

The GUI provides the user with various statistical tools. The user can select the data file through Browse button and can select the desired rows and columns to be manipulated.

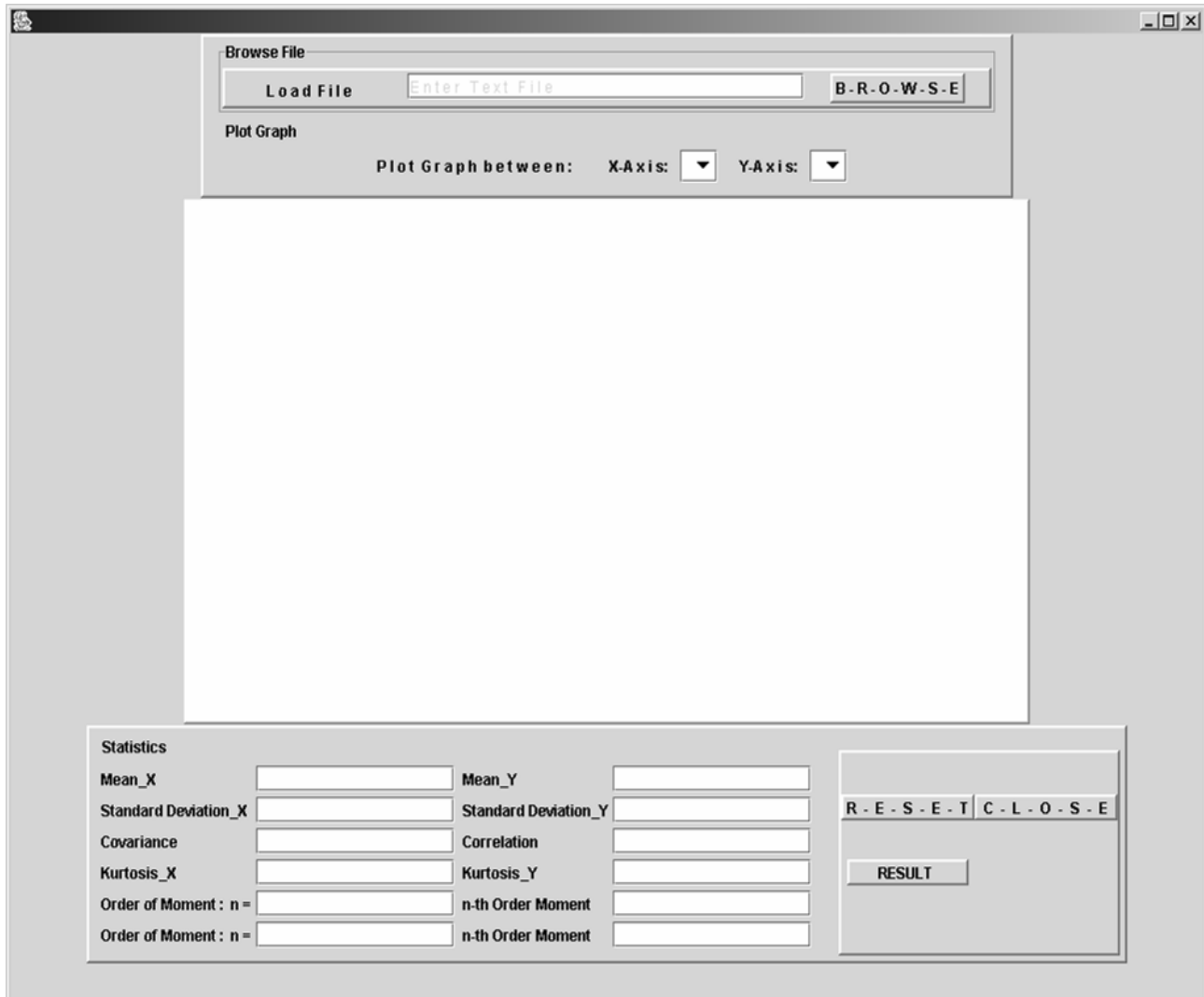


Fig. 3 Screenshot of the GUI for Statistical Tools



### 3.4 Decision Trees GUI

Here we can select the data file , define cut off values, choose different splitting criterions and classify them in the form of decision trees.

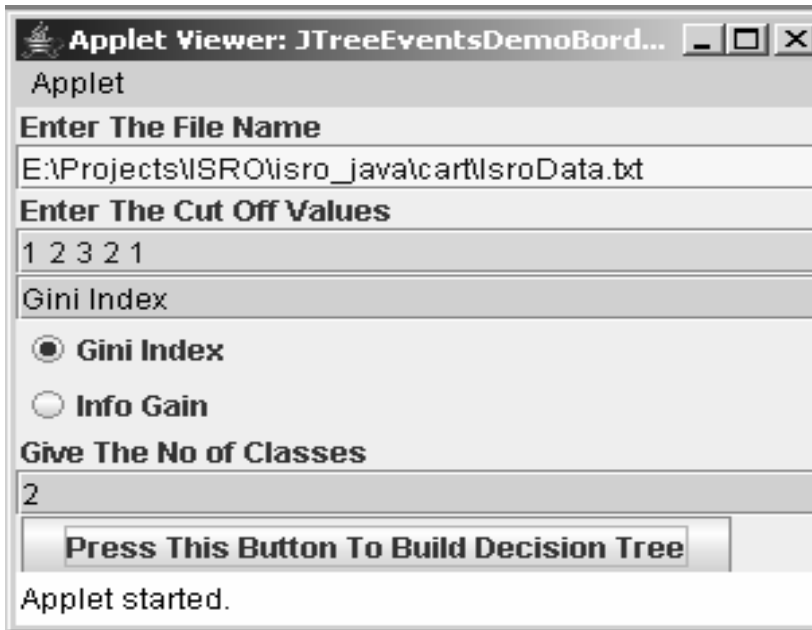


Fig. 4 Screenshot of the GUI for Decision Trees

### 3.5 Support Vector machines GUI

Here we can draw different patterns and even load the data file, and then classify that by clicking on the RUN button

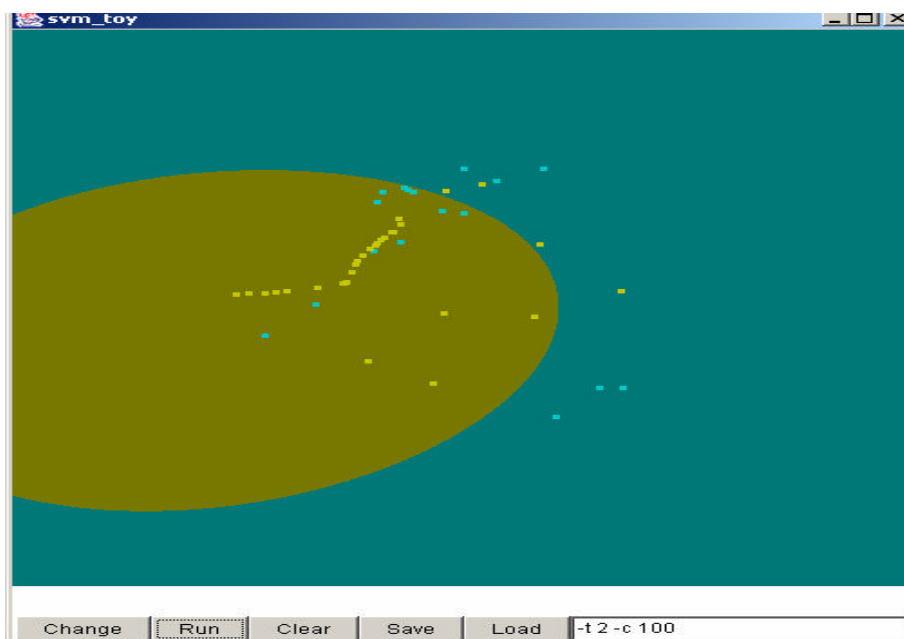


Fig. 5 Screenshot of the Support Vector Machines

## 4. Results:

The developed algorithms were tested on data given by IISU. The details of algorithms were tested on the Gyroscope data for classification into GOOD and BAD categories are as follows:

- ✍ Decision Trees
- ✍ Artificial Neural Network
  - ✍ *Conventional ANN*
  - ✍ *ANN with Multiplicative Neuron*
- ✍ Support vector machines

### 4.1 Decision Tree:

Decision tree analysis was made with the classification data for the gyroscope. It has also been tested on new data set given by IISU specifying success and failure data points as the output. Figure 6, shows the screen shot taken during decision tree analysis. The information about each node can be obtained by clicking on that particular node.

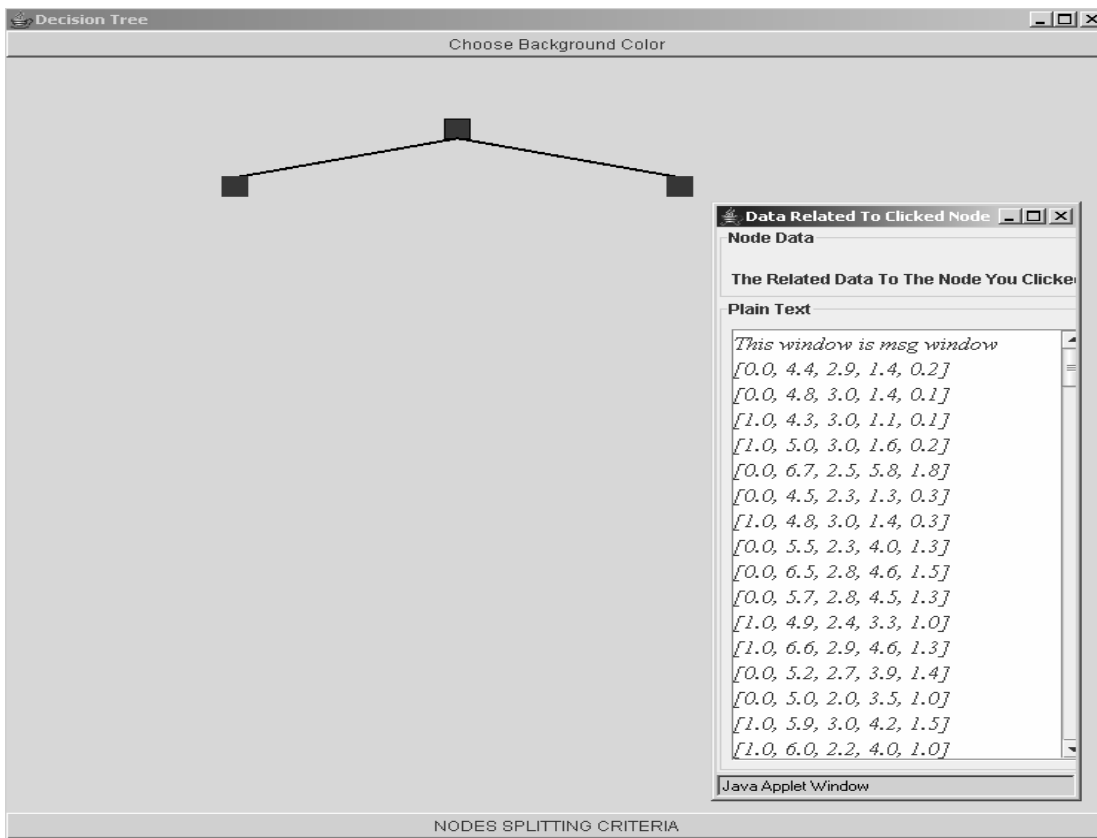


Fig. 6 Screen shot for the decision tree result using Gini Index

**Results using Information Gain:** Figure 7, shows the screen shot for decision tree formation when information gain is considered as the splitting criterion.

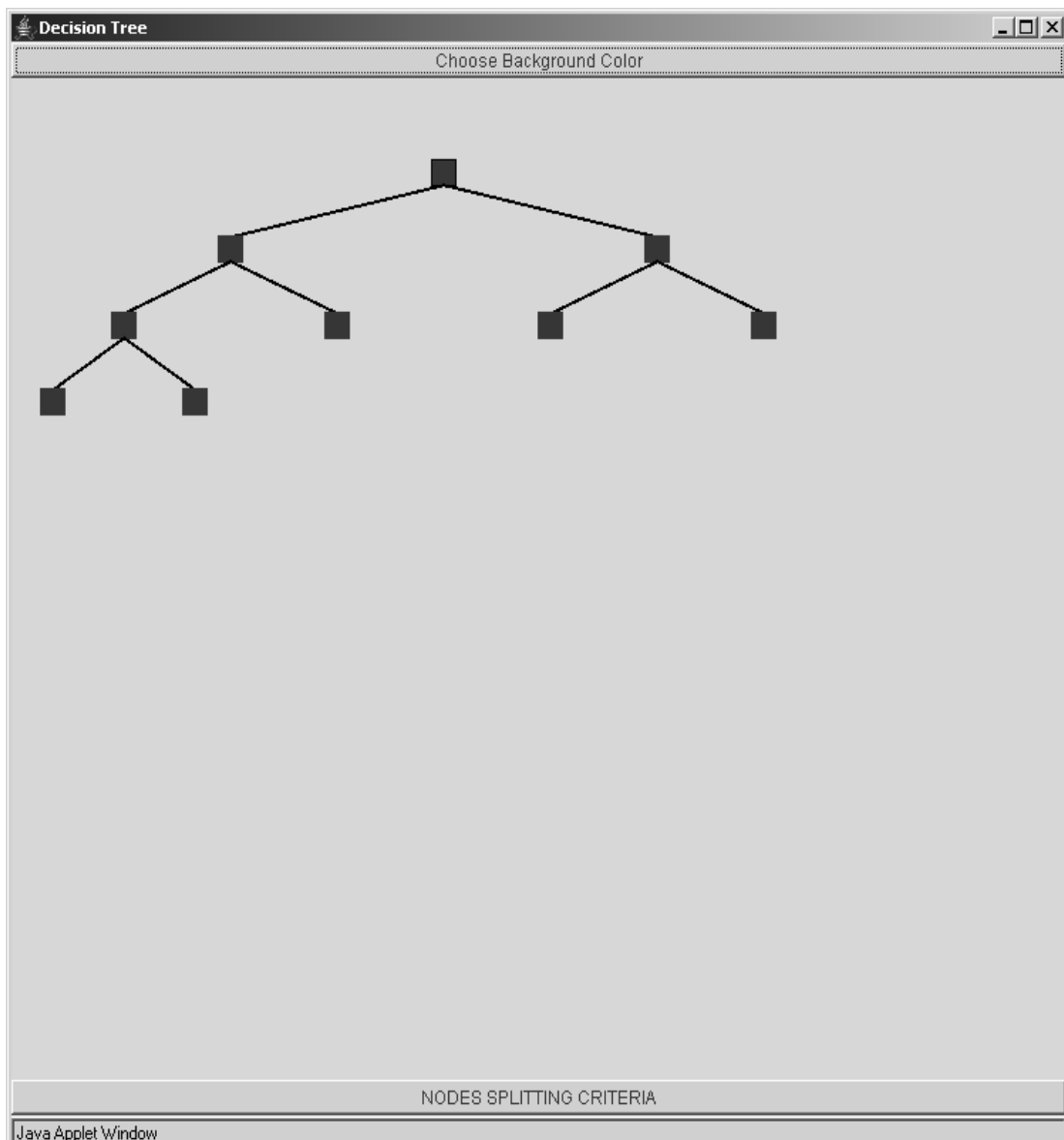


Fig. 7 Screen shot for the decision tree result using Information Gain as the splitting criterion

**Decision tree based on Probability, Possibility and Belief Theory:** New algorithms for formation of Decision trees using Probability, possibility and belief theory have also been developed. Figure 8 shows the screen shot taken during the analysis. Here a set of rules is displayed as an output.

```

Simulator Log
File
E:\ISRO_ALGO\New Folder (2)\Decision_Tree\Decision_tree_fuzzy\mtech_gui\prog_data
E:\ISRO_ALGO\New Folder (2)\Decision_Tree\Decision_tree_fuzzy\mtech_gui\prog_data
POSSIBILITY
Minimum Ambiguity :Temperature === 0.5611432809063756
S(Hot,Volleyball) 0.37704918032786877
S(Hot,Swimming) 0.6721311475409837
S(Hot,W_lifting) 0.1967213114754098
S(Mild,Volleyball) 0.5178571428571428
S(Mild,Swimming) 0.30357142857142855
S(Mild,W_lifting) 0.5357142857142857
S(Cool,,Volleyball) 0.2093023255813953
S(Cool,,Swimming) 0.0697674418604651
S(Cool,,W_lifting) 0.883720930232558
??? IF Temperature IS Cool,
THEN W_lifting = 0.883720930232558
IF Temperature IS Hot
AND Outlook IS Sunny
then Swimming = 0.8536585365853657
IF Temperature IS Hot AND
Outlook IS Cloudy then Swimming = 0.7222222222222221
IF Temperature IS
Hot AND Outlook IS Rain,
then W_lifting = 0.7272727272727272
IF Temperature IS Mild ANDWind IS Windy then W_lifting = 0.8125
IF Temperature IS Mild ANDWind IS Not_windy, then Volleyball = 0.783783783783783

```

Fig. 8 Screen shot for the Decision tree based on Probability, Possibility and Belief Theory

**4.2 Artificial Neural Network:** This tool consists of conventional artificial neural network algorithms and multiplicative type artificial neural network algorithms.

#### 4.2.1 Conventional ANN

We tested the conventional ANN which is able to classify the sensors into GOOD and BAD ones quite satisfactorily. In case of the old data provided by IISU during the pre-project, we used half of the data for training and another half for testing purposes. That means out of 320 data set, 160 were used for training and remaining 160 were used for testing purposes. In that case, we got 100 % correct classification results.

In the new data both train and test files were provided. The results are listed below:

These are the results for the new data set provided to us which are satisfactory.

### Training results for data set A:

- ✍ No. of Inputs = 53
- ✍ No. of Hidden Layer = 2
- ✍ No. of Outputs = 1
- ✍ Iteration = 2000
- ✍ MSE =  $3.61 \times 10^{-5}$

### Testing:

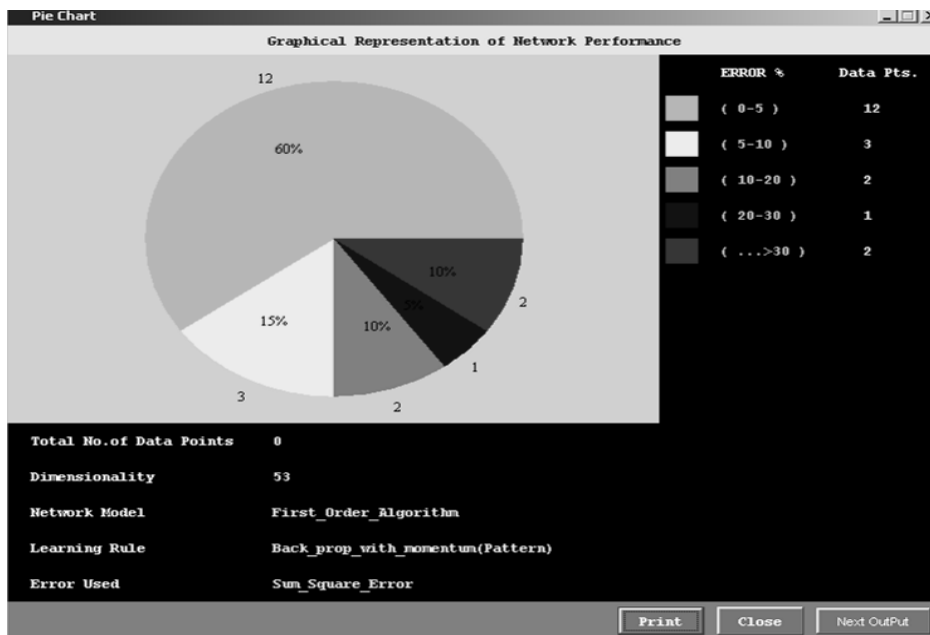


Fig.9 Screen shot after the analysis carried out on IISU data set by classical neural network.

### 4.2.2 Multiplicative ANN

We tested the Multiplicative ANN which is able to classify the sensors into GOOD and BAD ones quite satisfactorily. Data sets taken for analysis are identical as used with conventional neural network. The results obtained after analysis are listed below which are satisfactory.

### Training results for data set A:

- ✍ No. of Inputs = 53
- ✍ No. of Hidden Layer Neurons = 4
- ✍ No. of Outputs = 1
- ✍ Learning Rate = 0.001
- ✍ Iterations = 10000
- ✍ Mean Square Error (MSE) =  $8.006 \times 10^{-2}$

## Training results for data set B:

- ✍ No. of Inputs = 92
- ✍ No. of Hidden Layer Neurons = 4
- ✍ No. of Outputs = 1
- ✍ Iteration = 280
- ✍ MSE =  $3.611 \times 10^{-5}$

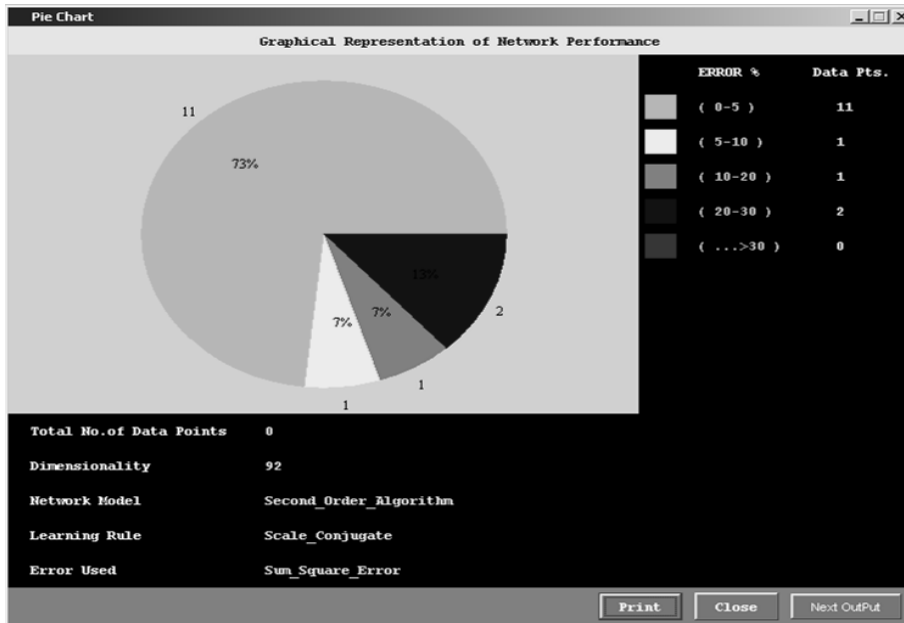


Fig. 10 Screen shot after the analysis carried out on IISU data set by multiplicative neural network.

## 4.3 Support Vector Machines(SVMs)

SVMs provide a new approach to the problem of pattern recognition (together with regression estimation and linear operator inversion) with clear connections to the underlying statistical learning theory. They differ radically from comparable approaches such as neural networks: SVM training always finds a global minimum, and their simple geometric interpretation provides fertile ground for further investigation. An SVM is largely characterized by the choice of its kernel, and SVMs thus link the problems they are designed for with a large body of existing work on kernel based methods.

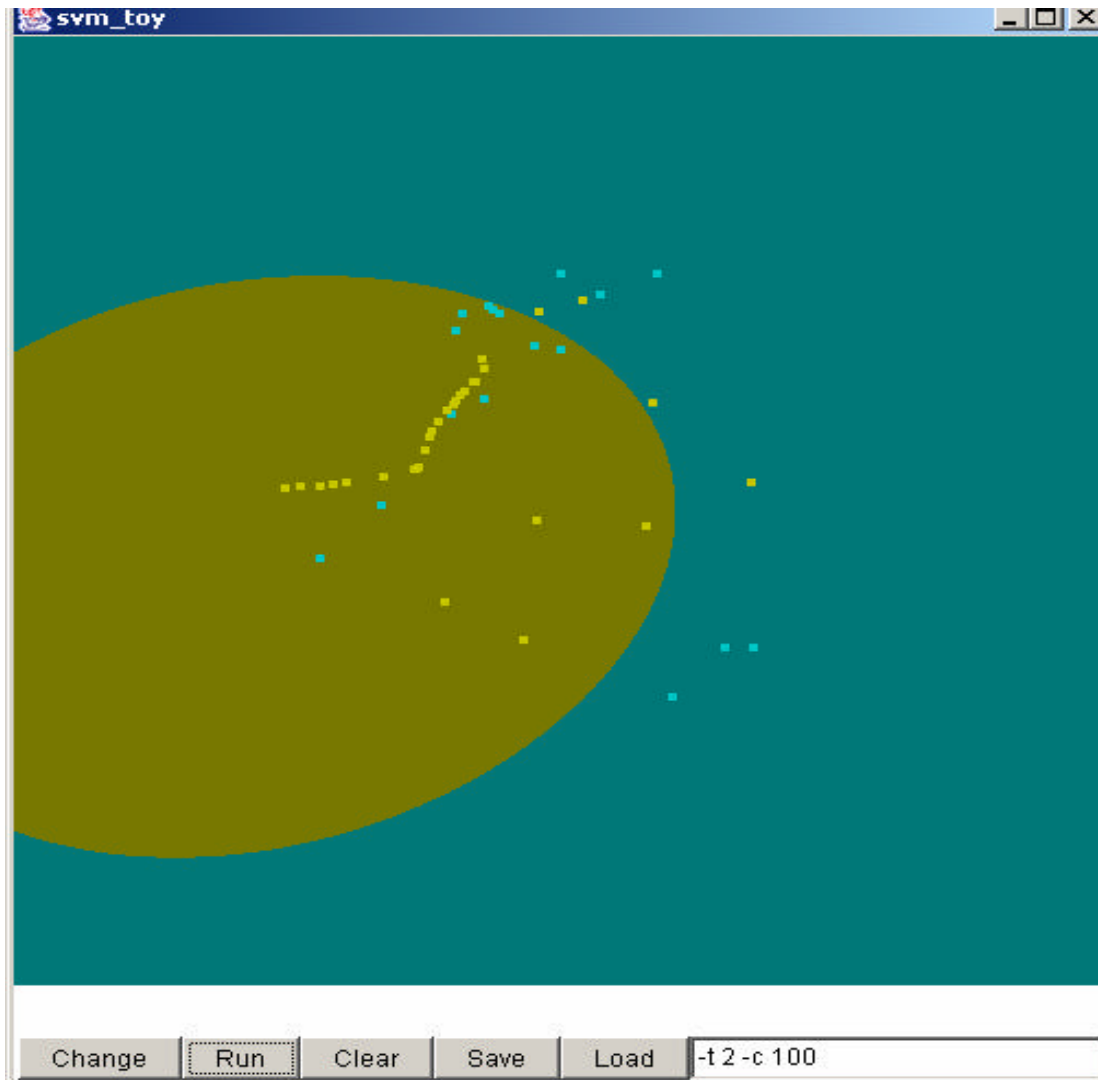


Fig. 11 Screen shot after the Support Vector machines analysis carried out on IISU data set.

## 5. Future Plan

In the present phase of project following New Algorithms were developed:

- ? Neural Principal Component
  1. GHA(Generalized Hebbian Learning algorithm)
  2. APEX(Adaptive Principal Component Analysis Using Lateral Inhibition )
- ? Support Vector Machine

In next phase of this project, we plan to test the IISU data set on following algorithms. In order to have robust operation and exhaustive testing of the developed algorithms on IISU data sets a six month extension in project is required without any extra financial support.

The following are the listed tasks that have to accomplish in the extended period which includes exhaustive testing of algorithms on IISU data sets.

- ✗ Independent Component Analysis (ICA)
- ✗ Non Linear PCA comprising of:
  - ✗ GHA and APEX algorithms is developed but not tested.
  - ✗ Implementation of SANFIS in Java
  - ✗ Fuzzy logic algorithms are to be implemented.
  - ✗ Database has been developed in MySql which needs to be integrated in final module of project.

## 6. Summary of the work carried out so far

### Conversion of Matlab codes to Java

<b>Algorithms developed in Matlab</b>	<b>Algorithms developed in java</b>
Classical Neural Network	Classical Neural Network
Neural-Network with sigma pi-sigma	Decision Tree 1. Fuzzy Decision Tree 2. Cart
Neural Network with pi-pi neuron model	Neural Principal Component Analysis
Decision Tree	Statistical tools
Principal Component Analysis	Support Vector machine
Statistical tools	
Self Adaptive Neural network	

### Tested Algorithms:

<b>Algorithms developed in Matlab</b>	<b>Algorithms developed in java</b>
Classical Neural Network	Classical Neural Network
Neural-Network with sigma pi-sigma	Decision Tree 1. Fuzzy Decision Tree 2. Cart
Neural Network with pi-pi neuron model	Neural Principal Component Analysis
Decision Tree	Statistical tools
Principal Component Analysis	Support Vector machine
Statistical tools	
Self Adaptive Neural network	

### NOTE:

**Blue** ---- Tested algorithms

**Black** ----- Not Tested



