Decision Tree Based Approach for Fault Diagnosis in Process Control System

Dr. Tarun Chopra[a], Jayant Acharya[b]

***Abstract*----**Decision Tree is one of the most popular classification algorithms in current use in Data Mining and Machine Learning. Decision trees create an easily understandable structure for evaluating complex decisions. In this paper, the performance of the proposed approach based on Single Decision Tree based method is demonstrated on the DAMADICS benchmark problem. An attempt has been made to improve the performance of fault diagnosis task on DAMADICS benchmark**.**

***Keywords*---**Benchmark study, Decision Tree, Fault Diagnosis, Gini Splitting algorithm .

I. Introduction

Decision tree is a hierarchical tree structure which is used to classify data on the basis of a series of rules about the attributes of the underlying classes. These attributes can be any type of quantitative variables ranging from binary to decimal, hexadecimal, nominal and ordinal values, while the classes must be qualitative type (categorical, descriptive or ordinal).

Decision trees create an easily understandable structure for evaluating complex decisions and are particularly useful for managers and technocrats in making decisions related with selection of various strategies, projects or investment alternatives.

Decision trees are produced by algorithms that identify various ways of splitting a data set into branch-like segments. These segments form an inverted decision tree that originates with a root node at the top of the tree. The binary structure is commonly used for designing decision trees.

Rules about the data can easily be created from a decision tree. Using decision tree, the value of a target variable can be predicted from the values of a set of predictor variables and also, the classification of unseen records can easily be predicted [1].

Decision trees are one of the most popular multiple variable analysis methods because of their ease of use, robustness with a variety of data and accuracy levels, and high interpretability.

The aforementioned merits of decision trees motivated the authors to adopt this technique for decision making in relation to Fault diagnosis in a Complex Benchmark Process Control System, with multiple measured variables and overlapping fault classes.

II. State Of Art

One of the earliest uses of decision trees was in the study of television broadcasting by Belson in 1956.The first widely-used program for generating decision trees was ―AID (Automatic Interaction Detection) developed in 1963 by J. N. Morgan and J. A. Sonquist written in FORTRAN and limited by the hardware of the time; AID was suitable only for small to medium size data sets, and it could generate only regression trees. Nonetheless, this pioneering program was well received and widely used during the 1960's and 70's [2].

AID was followed by many other decision tree generators including THAID by Morgan and Messenger in 1973 and ID3 and later, C4.5 by J. Ross Quinlan [3-4].

The theoretical underpinning of decision tree analysis was greatly enhanced by the research done by Leo Breiman, Jerome Friedman, Richard Olshen and Charles Stone that was published in their book Classification and Regression Trees. Much of their research has been included in their algorithm called Classification and Regression tree (CART). [5-6].

Although decision trees have been in development and use for over 50 years, many new forms of decision trees are evolving that promise to provide exciting new capabilities in the areas of data mining and machine learning in the years to come.

III. Decision Tree Approach

A decision tree partitions the space of all joint predictor variable values x into J -disjoint regions {Rj}j . A lucid account of this approach has been presented by Friedman in 2003 [7], and will be followed here.

A response value yj is assigned to each corresponding region Rj . For a given set of joint predictor values x, the tree prediction y = T_J (x) assigns as the response estimate, the value assigned to the region containing x

$$\mathbf{x} \in R_j \Rightarrow T_J(\mathbf{x}) = \hat{y}_j \, .$$

Given a set of regions, the optimal response values associated with each one are easily obtained, namely the value that minimizes prediction risk in that region

[a] Associate Professor, Department of Electrical Engineering, Govt. Engineering College Bikaner (India)-334004

[b] System Analyst, Govt. Engineering College Bikaner (India)-334004

$$\hat{y}_j = \arg\min_{y'} E_y[L(y, y') \mid \mathbf{x} \in R_j].$$

The main difficulty in this problem is to find a good set of regions {Rj}j. . Although there are many ways to partition the predictor variable space, the vast majority of these provide poor predictive performance. In the context of decision trees, choice of a particular partition directly corresponds to the choice of a distance function d(x; x') and scale parameter in kernel methods. Unlike with kernel methods where this choice is the responsibility of the user, decision trees attempt to use the data to estimate a good partition[7]. Unfortunately, optimal partition calculation requires computation that grows exponentially with the number of regions J, making it feasible for very small values of J.

Generally, all tree based methods use a greedy top-down recursive partitioning strategy to induce a good set of regions on the basis of training data set. Beginning with a single region covering the entire space of all joint predictor variable values, it is partitioned into two regions by choosing an optimal splitting predictor variable xj and a corresponding optimal split point s. Points x for which xj ≤ s are defined to be in the left daughter region, and those for which xj > s comprise the right daughter region. Each of these two daughter regions is then itself optimally partitioned into two daughters of its own in the same manner, until misclassification error reduces to the desired threshold. Thus, recursive partitioning continues until all or a majority of the observations within each region have the same response value y. At this point a recursive recombination strategy (tree pruning) is employed in which sibling regions are in turn merged in a bottom-up manner until the number of regions J* that minimizes an estimate of future prediction risk is reached [8].

IV. PROBLEM STATEMENT

DAMADICS (Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems) benchmark has been developed as a benchmarking tool for fault diagnosis and isolation (FDI) methods . The core of this benchmark is a Simulink model of an electro-pneumatic valve actuator. This model includes three subsystems: a control valve, a spring-and-diaphragm pneumatic servomotor, and a positioner. The servomotor acts on the control valve plug which position controls the fluid flow passing through the pipelines. The stem of the servomotor is driven by compressed air, which acts on a flexible diaphragm and is balanced by a spring. A positioner is used to avoid miss-positions of the stem caused by internal and external factors like friction and change of supply pressure and provides digital I/O for the actuator.

The benchmark contains total 44 types of fault scenarios, but as reported in the literature [9], the misclassification occurs due to overlapping phenomenon among different fault classes.

Cosmin Danut Bocanialaet al et al [10] have applied novel fuzzy classifier using fuzzy subsets in DAMADICS benchmark problem. Fuzzy subsets are induced (built) on the basis of a point-to-set similarity measure between a point and a set of points in the measurements space. In the paper authors have remarked that the overlapping between the two mentioned groups of faults i.e. {F7, F10}, and {F11, F15, F16} is critical and there is also non-critical overlapping between {F1, F7}, {F2, F19} and {F13, F18}. They have reported that the isolation between different fault strengths improves by using their approach works but they have not quantified it and they have not considered {F8, F12, F14} as according to them it is not distinguishable from the normal behavior (N).

The work presented here is a sincere attempt for further improvement of fault diagnosis results obtained in the cited work on DAMADICS benchmark using single decision tree based method.

V. METHODOLOGY

The dataset used for this case study have been generated by employing the MATLAB-SIMULINK model of the actuator as shown in fig 1.

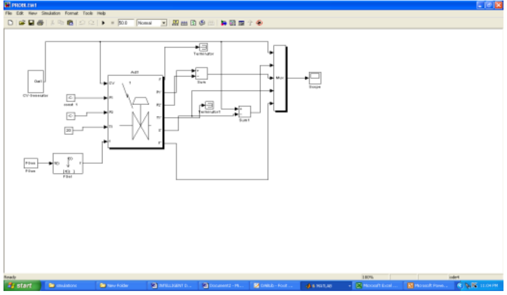

Fig :1

In accordance with the scope of the defined objective for this paper, only data related with Normal flow condition and fault categories F5 (External leakage: leaky bushing, covers, terminals), F9 (servomotor housing or terminal tightness), F12 (Electro-pneumatic transducer fault), F14 (Pressure sensor fault) and small and medium fault strength for fault F8 (Twisted piston rod) have been considered. The Single Decision tree model based on CART [6] has been used for this purpose, with Maximum splitting levels limited to 10. Classification analysis has been performed while using Gini Splitting algorithm and surrogate splitters for any missing values in dataset.

The category weights or priors were obtained from data file distribution and variable weights were set to be initially equal. Misclassification cost was also set to be equal or unitary. Cross validation method was used for tree pruning and validation. The tree pruning criterion was selected to be minimum cost complexity with the target standard error of 0.00.

The following parameters were set for this analysis:-

Minimum size node to split: 10

Max. categories for continuous predictors: 200

Number of cross-validation folds: 10

The format of various Data elements is now described:

Input Data

Number of variables (data columns): 7
Number of data rows: 80
Total weight for all rows: 80
Rows with missing target or wt. values: 0
Rows with missing predictor values: 0
Details of Variables are summarized in Table 1 .

TABLE: 1 -SUMMARY OF VARIABLES

Number	Variable	Class	Type
1	CV	Predictor	Continuous
2	P1	Predictor	Continuous
3	P2	Predictor	Continuous
4	T	Predictor	Continuous
5	X	Predictor	Continuous
6	F	Predictor	Continuous
7	Type of fault	Target	Categorical

VI. RESULTS

The investigations proved that the single decision tree based classifier generalizes reasonably well for both cases involving critical overlapping fault classes. The strength of this approach for other case involving one type from non critical overlapping fault classes has also been considered. Validation Statistics, Percentage Misclassification for Training Data and Validation Data for all the three cases considered in this work have been presented in tables 2-10.

Case: 1- Model Summary for Critical Overlapping fault group (Fault group F7, F10)

Maximum depth of the tree = 2
Total number of group splits = 1
The full tree has 2 terminal (leaf) nodes, as shown in fig 2.
The minimum validation relative error occurs with 2 nodes.
The relative error value is 0.1with a standard error of 0.04.

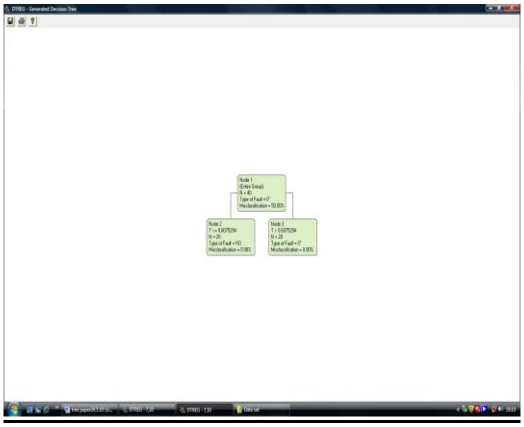

Fig: 2

TABLE: 2 -VALIDATION STATISTICS (FOR CASE1)

Nodes	Val cost	Val std. err.	RS cost	Complexity
2	0.1000	0.0400	0.0000	0.0000 <-- Min. validation error
1	1.0000	0.0000	1.0000	0.5000

TABLE: 3 -MISCLASSIFICATION TABLE FOR TRAINING DATA (FOR CASE1)

Actual			Misclassified		
Category	Count	Wt	Count	Wt	%
F7	20	20	0	0	0.000
F10	20	20	0	0	0.000
Total	40	40	0	0	0.000

TABLE: 4 -MISCLASSIFICATION TABLE FOR VALIDATION DATA (FOR CASE1)

Actual			Misclassified		
Category	Count	Wt	Count	Wt	%
F7	20	20	0	0	0.000
F10	20	20	2	2	10.000
Total	40	40	2	2	5.000

Case: 2- Model Summary for Critical Overlapping fault group (Fault group F11, F15, F16)

Maximum depth of the tree = 3
Total number of group splits = 2
The full tree has 3 terminal (leaf) nodes,
as shown in fig 3.
The minimum validation relative error occurs
with 3 nodes.
The relative error value is 0.0417 with a standard error of 0.0180

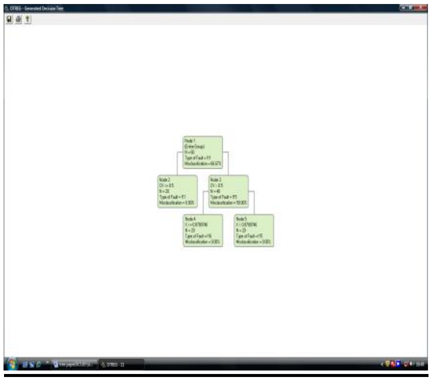

Fig:3

TABLE: 5-VALIDATION STATISTICS (FOR CASE-2)

Nodes	Val cost	Val std. err.	RS cost	Complexity
3	0.0417	0.0180	0.0000	0.0000 <-- Min. validation error
2	0.6667	0.0000	0.5000	0.333333
1	1.0000	0.0000	1.0000	0.333333

TABLE: 6 -MISCLASSIFICATION TABLE FOR TRAINING DATA (FOR CASE- 2)

Actual			Misclassified		
Category	Count	Wt	Count	Wt	%
F11	20	20	0	0	0.000
F15	20	20	0	0	0.000
F16	20	20	0	0	0.000
Total	60	60	0	0	0.000

https://doi.org/10.15242/IJCCIE.IAE0317001

TABLE: 7 -MISCLASSIFICATION TABLE FOR VALIDATION DATA (FOR CASE-2)

Actual			Misclassified		
Category	Count	Wt	Count	Wt	%
F11	20	20	1	1	5.000
F15	20	20	0	0	0.000
F16	20	20	1	1	5.000
Total	60	60	2	2	3.333

Case: 3 - Model Summary for Non Critical Overlapping fault group (Fault group F2, F19)

Maximum depth of the tree = 2

Total number of group splits = 1

The full tree has 2 terminal (leaf) nodes, as shown in fig 4.

The minimum validation relative error occurs with 2 nodes.

The relative error value is 0.05 with a standard error of 0.02

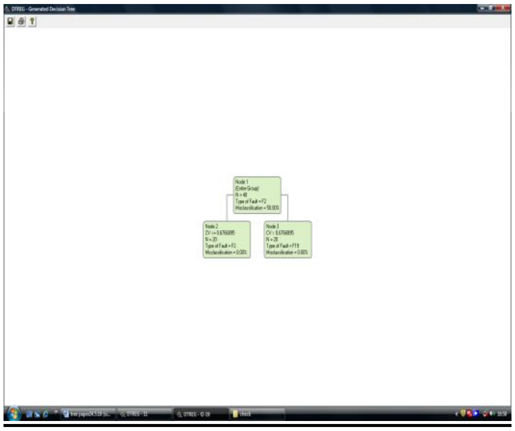

Fig: 4

TABLE: 8-VALIDATION STATISTICS (FOR CASE3)

Nodes	Val cost	Val std. err.	RS cost	Complexity
2	0.050	0.02000	0.0000	0.0000 <-- Min. validation error
1	1.00	0.000	1.0000	0.5000

TABLE: 9 -MISCLASSIFICATION TABLE FOR TRAINING DATA (FOR CASE3)

Actual			Misclassified		
Category	Count	Wt	Count	Wt	%
F2	20	20	0	0	0.000
F19	20	20	0	0	0.000
Total	40	40	0	0	0.000

TABLE: 10 -MISCLASSIFICATION TABLE FOR VALIDATION DATA (FOR CASE3)

Actual			Misclassified		
Category	Count	Wt	Count	Wt	%
F2	20	20	1	1	0.000
F19	20	20	0	0	5.000
Total	40	40	1	1	2.500

VII. DISCUSSIONS

The proposed approach shows great promise in handling the classification (discrimination) task of faults inside overlapping areas with fine precision.

REFERENCES

[1] Padraic G. Neville,"Decision Trees for Predictive Modeling", SAS Institute Inc., 1999.

[2] Morgan & Sonquist,"Problems in the analysis of survey data and a proposal", JASA, 58, 415-434. (Original AID), 1963.

[3] Morgan & Messenger THAID -- A sequential analysis program for the analysis of nominal scale dependent variables, Survey Research Center, U of Michigan, 1973.

[4] Quinlan, J.R., "C4.5: Programs for Machine Learning", 1993.

[5] Breiman, Leo, Jerome Friedman, Richard Olshen, and Charles Stone, "Classification and Regression Trees". Pacific Grove: Wadsworth, 1984.

[6] Phillip H. Sherrod, "DTREG Predictive Modeling Software", 2003

[7] Jerome H. Friedman, "Recent advances in Predictive (Machine) Learning", PHYSTAT2003, SLAC, Stanford, California, September 8-11, 2003

[8] Kardi Teknomo, "Tutorial on Decision Tree", 2009

[9] Michal Bartys et al, "Introduction to the DAMADICS actuator FDI benchmark study" Control Engineering Practice 14 (2006) 577–596

[10] Cosmin Danut Bocanialaet al "Application of a novel fuzzy classifier to fault detection and isolation of the DAMADICS benchmark problem", Control Engineering Practice 14 (2006) 653-669.

https://doi.org/10.15242/IJCCIE.IAE0317001