This article can be cited as S. Babu Devasenapati and K. I. Ramachandran, Hybrid Fuzzy Model Based Expert System for Misfire Detection in Automobile Engines, International Journal of Artificial Intelligence, vol. 7, no. A11, pp. 47-62, 2011.

Hybrid Fuzzy Model Based Expert System for Misfire Detection in Automobile Engines

S. Babu Devasenapati and K. I. Ramachandran

Department of Mechanical Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India.

s_babu@cb.amrita.edu, ki_ram@cb.amrita.edu

ABSTRACT

This paper evaluates the use of fuzzy unordered rule induction algorithm (FURIA) with correlation based feature selection (CFS) embedded feature subset selection as a tool for misfire detection. The vibration data of the automobile engine contains the engine performance data along with multitudes of other information. The decoding of engine misfire condition was achieved by processing the statistical features of the signals. The quantum of information available at a given instant is enormous and hence suitable techniques are adopted to reduce the computational load due to excess information. The effect of recursive entropy discretiser as feature size reduction tool and CFS based feature subset selection is analysed for performance improvement in the FURIA model. The FURIA based model is found to have a consistent high classification accuracy of around 88% when designed as a multi class problem and approaches 100% when the system is modeled as a two-class problem. From the results obtained the authors conclude that the combination of statistical features and FURIA algorithm is suitable for detection of misfire in spark ignition engines.

Keywords — *engine condition monitoring, misfire detection, fuzzy classifier, FURIA, data discretisation, IC engine.*

Mathematics Subject Classification: 65J05, 62G04 and 62H07

Computing Classification System: I.2 and 1.5

1.INTRODUCTION

Growth in global economy has brought along insurmountable environmental challenges threatening the very source of existence. A major part of this pollution can be attributed to transportation systems using internal combustion (IC) engines. Many countries have made it mandatory to rein in pollution due to IC engines using a combination of technology and monitoring systems. The engine diagnostic system of the vehicle should be designed to monitor misfire continuously because even with a small number of misfiring cycles, engine performance degrades, hydrocarbon emissions increase, and drivability will suffer (Lee & Rizzoni, 1995). The cylinder misfire cycle also results in a large quantity of unburned fuel being sent through the catalytic converter, which causes a reduction in its service life due to high temperature exposures (Klenk, et al., 1993) and also contributes to significant air pollution. The California Air Resources Board (CARB) regulations (California Air Resources Board, 1991) defines engine misfire as, "lack of combustion in the cylinder due to absence of spark, poor fuel metering, poor compression, or any other cause". Misfire detection in an internal combustion engine is very crucial to maintain optimum performance throughout its service life and to reduce emissions.

The use of Harr and Daubechies wavelets for signal processing by approximation (Yajnik & Mohan.S, 2009) is a workable idea but the use of signal approximation techniques could lead to loss of information, hampering the possibility of growing this model in to a full vehicle monitoring system. The use of wavelet based clustering techniques (Palanisamy & Selvan, 2009) for handling high dimension data is encouraging. Similarly the use of pattern recognition techniques including wavelets for structural health monitoring reported by Navarro and Mejia (Navarro & Mejia, 2010) can be reliably extended for detection of misfire. A detailed work is reported by Jinseok (Chang, et al., 2002) using a combination of engine block vibration and wavelet transform to detect engine misfire and knock in a spark ignition engine. The use of engine block vibration is encouraging since it requires minimum instrumentation. The use of wavelets in all the cited work has one common challenge; the requirement of increased computational capability to deal with the additional load induced into the model due to wavelets. The use of support vector machines (SVM) for pattern recognition when compared to neural network, method of least squares or linear discriminant analysis produces less misclassification rate hence more suitable for real time applications (Kalyani & Swarup, 2010). Misfire detection using SVM reported by (Devasenapati, et al., 2010a) demonstrates good classification efficiency but the main concern here is the computational complexity of SVM which could pose a serious challenge for implementation in an online model.

Extensive studies have been done using measurement of instantaneous crank angle speed (Tinaut, et al., 2007) and diverse other techniques have been developed on similar lines to predict misfire (Lee & Rizzoni, 1995). These methods call for a high resolution crank angle encoder and associated infrastructure capable of identifying minor changes in angular velocity due to misfire. The application of these techniques becomes more challenging due to continuously varying operating conditions involving random variation in acceleration coupled with the effect of flywheel, which tries to smoothen out minor variations in angular velocity at higher speeds. Fluctuating torque experienced by the crankshaft through the drive train poses additional hurdles in decoding the misfire signals. In-cylinder pressure monitoring is very reliable and accurate as individual cylinder instantaneous mean effective pressure could be calculated in real time. However, the cost of fitting each cylinder with a pressure transducer is prohibitively high. This initiated the quest for identifying a low cost model capable of competing with the existing solutions.

The idea of using imprecise knowledge for developing a system that could relate to real time quantifiable information has been presented by (Johanyák, 2010). The development of fuzzy logic controllers for managing chaotic and non-linear systems have been presented by Precup et.al., (Precup, et al., 2008) and (Precup, et al., 2010) which demonstrates the capabilities of fuzzy logic controllers in control critical applications. The prospect of using fuzzy logic in engines has been explored by (Liu, et al., 2000) using crank shaft angular velocity as the base and the same authors have reported the use of multiple feature fusion techniques (Liu, et al., 2002). The first technique uses a crank angle encoder and the second technique uses multiple features from various domains and hence both these techniques have cost and computation challenges, prompting to investigate the possibility of alternative low cost solutions but have validated the use of fuzzy algorithms for misfire detection.

The specific contribution of this work involves the development of a low cost, reconfigurable, automated machine learning model capable of working as an expert system for detection of misfire in IC engines using a mono axial piezoelectric accelerometer. The model is future proofed for the engine with the use of various data pre-processing techniques to avoid performance deviations due to deviation in signal pattern, which is expected with advancement in age of the engine. The expert system stretches itself to identify the exact cylinder in which the misfire occurs which is very useful in identifying the fault location.

The present study proposes a non-intrusive engine block acceleration measurement using a piezoelectric accelerometer connected to a computer through a signal conditioner. The acquired analog vibration signals are converted to digital signals using an analog to digital converter and the discrete data files are stored in the computer for further processing. Feature extraction, feature reduction and feature subset selection techniques are employed and the classification results obtained are presented in the ensuing discussion.

The section 2 describes the experimental setup, the data acquisition methodology using accelerometer and the signal conditioning unit while section 3 describes the experimental procedure in detail. The methods involved in data preprocessing like feature extraction, feature reduction and feature subset extraction are presented in section 4 and the detailed working of the FURIA and various stages of work by the algorithm is presented in section 5. The results and discussion are presented in detail under section 6 followed by conclusion in section 7. This study establishes that the combination of statistical features and FURIA algorithm is well suited for detection of misfire in spark ignition engines.

2. EXPERIMENTAL SETUP

The misfire simulator consists of two subsystems namely, IC engine test rig and data acquisition system. They are discussed in the following subsections. The model building process is presented in Figure 1.

2.1 IC Engine test rig

The experimental setup of the engine misfire simulator consists of a four stroke vertical four cylinder gasoline (petrol) engine. Misfire in the cylinder is simulated by switching off the high voltage electrical supply to individual spark plugs. The engine accelerator is manually controlled using a screw and nut mechanism that can be locked in any desired position. The engine speed is monitored using an optical interference tachometer.

2.2 Data acquisition system

Accelerometers have a wide operating range enabling them to detect very small and large vibrations. The vibration sensed is a reflection of the internal engine condition. The voltage output of the accelerometers is directly proportional to the vibration. A mono axial piezoelectric accelerometer and its accessories form the core equipment for vibration measurement and recording. The accelerometer is directly mounted on the center of the engine block, in between cylinder two and three, using adhesive mounting as shown in Figure 2. The output of the accelerometer is connected to the signal conditioning unit that converts the analogue signal into digital form. The digitized vibration signal (in time domain) is stored in the computer for further processing.

3. EXPERIMENTAL PROCEDURE

The engine is started by electrical cranking at no load and warmed up for 15 minutes. The signal conditioner is switched on, the accelerometer is initialized and the data is recorded after the engine speed stabilizes at 1500 rpm. A sampling frequency of 24 kHz and sample length of 8192 is maintained for all conditions. The highest frequency was found to be 10 kHz. The Nyquist–Shannon sampling theorem recommends that the sampling frequency must be at least twice that of the highest measured frequency or higher, hence the sampling frequency was chosen to be 24 kHz.

Extensive trials were taken and discrete vibration signals were stored in the files. Five cases were considered - normal running (without any misfire), engine with any one-cylinder misfire individually (*i.e.* first, second, third or fourth denoted by C1m, C2m, C3m and C4m respectively). All the misfire events were simulated at 1500 rpm, the rated speed of the engine electrical generator set. A sample plot of misfire and no-misfire recorded at 1500 rpm is presented in Figures 3a and 3b respectively.

4. FEATURE EXTRACTION

Statistical Features: Statistical analysis of vibration signals yields different parameters. The statistical parameters taken for this study are mean, standard error, median, standard deviation, sample variance, kurtosis, skewness, range, minimum, maximum and sum. These features were extracted from the vibration signals. The definitions for these features are commonly available and hence not presented.

4.1 Feature reduction

The wealth of information available in the extracted features is abundant and at times overwhelmingly large enough to distract the machine learning system leading to inferior performance. Data granulation as a means of feature reduction has many advantages since it reduces the content volume and makes it easy to handle lot of information without challenging the system resources. But the technique to discretise or compress data without loss of valuable information is the key challenge. There are many techniques reported in the literature but an algorithms that can suit the given condition needs to be validated by using the transformed data for developing the model and comparing it with the performance of the model built without data pre processing to establish performance improvements.

The Fayyad and Irani (FI) model (Fayyad, 1993) uses a supervised hierarchical split method where multiple ranges are created instead of binary ranges to form a tree. Multi-way splits of the numeric attribute at the same node are performed to produce discrete bins. The number of cut points is determined using the Minimum Description Length (MDL) principle. Here class information entropy is a measure of purity and it measures the amount of information which would be needed to specify the class to which an instance belongs (Dougherty, et al., 1995). Information entropy minimization heuristic is used to select threshold boundaries by finding a single threshold that minimizes the entropy function over all possible thresholds (Michael & Ciesielsk, 2003). This entropy function is then recursively applied to both of the partitions induced. Thresholds are placed half way between the two delimiting instances. At this point the MDL stopping criterion is applied to determine when to stop subdividing discrete intervals, (Fayyad, 1993).

4.2 Feature subset selection

Including all the features may improve the classification accuracy but the probability of over fitting the model saddled with additional computational load outweighs their consideration.

It is observed from the computations that there are significant differences in some of the feature values for different types of faults. Selecting those features is crucial for effective classification and doing it manually demands more expertise; however, the effectiveness of the manually selected features is not guaranteed. Selecting the most relevant features through suitable algorithm will yield better classification results. Here feature subset selection (FSS) is performed using Correlation based Feature Selection (CFS). CFS is an algorithm for selecting features that are highly correlated with the class but uncorrelated with each other (Hall, 2000). CFS has the ability to identify irrelevant, redundant, and noisy features from relevant features as long as their relevance does not strongly depend on other features. This method is adapted for building the model since signal corruption due to noise is more predominant in IC engines. The effect of using CFS on the developed model is studied.

From a list of 11 statistical features presented the CFS has recommended the following features as most prominent ones to be used for model building. They are standard error, standard deviation, sample variance, skewness, range and minimum.

5. CLASSIFIER

Fuzzy logic based classifier is used to build this expert system for misfire detection. Fuzzy logic is a system of knowledge that provides a simple method to draw definite conclusions from vague, ambiguous or imprecise information (Zadeh, 1965). The fuzzy logic mimics the human decision making process with its ability to work using approximate data to find precise solutions. Unlike classical logic which requires a deep understanding of a system, exact equations, and precise numeric values, fuzzy logic incorporates an alternative way of thinking, which allows modeling complex systems using a higher level of abstraction originating from user knowledge and experience.

Fuzzy Logic finds numerous applications in the wake of continuously increasing system complexity which challenges the ability of existing techniques to make a precise statement about its behavior. The working process of the fuzzy logic analysis and control method can be summarized as follows (Sowell, 2008):

1. Receiving measurement or other assessment of conditions existing in the system
2. Processing these inputs using fuzzy "If-Then" rules. These rules can be expressed in plain linguistic terms.
3. Averaging and weighting the resulting outputs from all the individual rules into one single output decision or signal. The output signal eventually arrived at is a precise appearing, defuzzified, "crisp" value.

The basic principle of fuzzy logic is to map an input space to an output space, and the primary mechanism for doing this is a list of ‘if-then’ statements called rules. Rules are the inputs for building a fuzzy inference engine. All rules are evaluated in parallel; hence, the order of the rules is unimportant. The rules are generated based on a membership function. A membership function is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1 as represented in Figure 4a. A set of ‘if-then’ rules defined using Membership Functions (MF) form the knowledge base for the fuzzy inference engine and is used for classification.

5.1 Working of FURIA

FURIA uses Ripper; an implementation of Cohen's Ripper (JRip) (Cohen, 1995) as the base classifier and learns fuzzy rules instead of conventional rules. FURIA proposes to learn a rule set for every single class, using a one versus rest decomposition and learns to separate each class from all other classes. This implies that no default rule is used and builds unordered rule sets instead of rule lists; hence the order in which the classes are presented is irrelevant (Hühn & Hüllermeier, 2009b). The FURIA algorithm can be explained by describing FURIAs working in a polychotomous classification problem involving *m* classes. It can be represented as

$$L = \{Z_1 \, . \, . \, . \, . \, . \, . \, . \, . Z_m\} \qquad (1)$$

Where

L is the set of all classes in the classification problem and

Z represents the individual class, in short ($Z \in L$).

These instances are represented in terms of numerical attributes A_i and D_i denotes the corresponding domains, where I = 1 to n. Using this formulation any given instance can be represented as an *n* dimensional attribute vector as follows

$$x = (x_1 \ldots . x_n) \in D \quad (2)$$

Where

$$D = D_1 \times \ldots . \times D_n$$

In the current work the n-dimensional attribute vector is the set of statistical features that are used to develop the classification system and each class i.e. misfire in each cylinder is represented by a combination of the selected statistical features. The rule building process of FURIA delves extensively into RIPPER algorithm where a single rule is represented in the following form

$$r = \langle r_A | r_C \rangle \quad (3)$$

Where

r_A is a combination of predicate or the features

r_C is the consequent part as a result of r_A

The premise part is a combination of features which is of the form (A_i θ V_i) where θ ∈ {<,=,>=} and $V_i \in D_i$. The consequent part r_C represents the class label assigned in the form class = Z where Z ∈ L. This essentially represents a rule of the form if . . . . then . . . . else. The stated rule format represented by Equation (3) is said to cover an instance represented by x = (x_1 . . . . x_n) if the attribute values x_i satisfy all the predicates in r_A.

The FURIA learns rules using a greedy approach implementing a separate and conquer strategy as presents by Fürnkranz (Fürnkranz, 1999). Rules are learnt for the first m-1 classes, starting with the smallest rule. The instances covered by the formed rules are removed from the training data once the rule is learnt and this format is adapted until no instances from the target class are left. This procedure is repeated for all the classes.

5.1.1 Rule growing

The process of growing a rule is achieved using a propositional version of the First Order Inductive Learner (FOIL) algorithm by Quinlan and Cameron (Quinlan & Cameron, 1993). The rule is initiated with an empty conjunction and adds features or selectors until the rule covers no more negative instances, i.e., instances not belonging to the target class. The next prospective feature is chosen in such a manner that it maximizes FOIL's information gain criterion (IG), which is a measure of improvement of the rule in comparison with the default rule for the target class and is given by

$$IG = p_r \left(\log_2 \left(\frac{p_r}{p_r + n_r} \right) - \log_2 \left(\frac{p}{p + n} \right) \right) \quad (4)$$

Where,

P_r and n_r represents the number of positive and negative instances covered by the rule under growing phase while

P and n represents the number of positive and negative instances covered by the default rule

Some rules over fitting the training data is observed in this process and is effectively mitigated by rule simplification or optimization. The pruning is done after rule learning for all classes are completed and not at each stage as done in the RIPPER algorithm. For the pruning procedure, the antecedents are considered in the order in which they were learned, and pruning actually means finding a position at which the list of antecedents can be cut without compromising the classification capability of the rule. This also reduces the system complexity and helps in maintaining the minimum description length for all the rules. The criterion to find the truncation point in the rule-value metric is as described by Hühn and Hüllermeier (Huhn & Hullermeier, 2009a).

5.1.2 Rule fuzzification

The fuzzification of the rule is achieved by replacing the intervals with fuzzy intervals. The fuzzy intervals in turn represent fuzzy sets which are formed using trapezoidal membership function as shown in Figure 4b. The process of rule fuzzification as presented by Huhn (Hühn & Hüllermeier, 2009b) describes that "A selector constraining a numerical attribute A_i (with domain D_i = R) in a RIPPER rule can be expressed in the form $(A_i \in I)$, where I is contained in R, is an interval: $I = (-\infty, v]$ if the rule contains a selector $(A_i \leq v)$, $I = [u, \infty)$ if it contains a selector $(A_i \geq u)$, and $I = [u, v]$ if it contains both". A fuzzy interval is specified by four parameters and will be written as

$$I^F = (\Phi^{s,L}, \Phi^{c,L}, \Phi^{c,U}, \Phi^{s,U}) \qquad (5)$$

$$I^F(v) = \begin{cases} 1 & \Phi^{c,L} \leq v \leq \Phi^{c,U} \\ \frac{v - \Phi^{s,L}}{\Phi^{c,L} - \Phi^{s,L}} & \Phi^{s,L} < v < \Phi^{c,L} \\ \frac{\Phi^{s,U} - v}{\Phi^{s,U} - \Phi^{c,U}} & \Phi^{c,U} < v < \Phi^{s,U} \\ 0 & \text{else} \end{cases} \qquad (6)$$

FURIA uses a trapezoidal membership function as shown in Figure 4b where I^F represents the class interval, $\Phi^{s,L}$ and $\Phi^{s,U}$ represent the lower and upper support levels for the interval while $\Phi^{c,L}$ and $\Phi^{c,U}$ represent the lower and upper bounds for the core set. FURIA replaces the crisp boundaries with trapezoidal functions according to the class information available at the training phase, also called rule building phase. The function is not constrained by requirements for symmetry or being in a closed interval which is depicted in Figure 4a and 4b respectively. Figure 4a is open ended whereas 4b is closed and not symmetric. FURIA is less complex in the sense that the membership function for each class is generated by the algorithm and the inference is presented in the form of rules which can be decoded directly; similar to that of any rule based classifier.

Figure 4a Membership function with open interval in upper bound

Figure 4b Non-symmetric trapezoidal membership function

Taking the intervals I_i of the original rules as the cores $[\Phi_i^{c,L}, \Phi_i^{c,U}]$ of the sought fuzzy Intervals I_i^F the problem is to find optimal bounds for the respective supports, i.e., to determine $\Phi_i^{s,L}$, $\Phi_i^{s,U}$

The process of fuzzification of a single antecedent represented by ($A_i \in I_i$) it is imperative to consider only the relevant training data D_i^T which means the system should ignore those instances that are excluded by any other antecedent $(A_j \in I_j^F), j \neq 1$

$$D_i^T = \{x = (x_1, \ldots, x_k) \in D_T \mid I_j^F(x_j) > 0 \ for \ all \ j \neq i\} \subseteq D_T \quad (7)$$

For an elaborate description of the algorithm it is recommended to refer the work by Huhn (Hühn & Hüllermeier, 2009b). Since the rules are based on fuzzy sets a post de-fuzzification of the result is not necessary in this system. The rules directly deliver the class value as output.

5.1.3 FURIA rules and inference

1. (kur in [-inf, -inf, 0.183662, 1.102526]) => state=good (CF = 0.99)
2. (min in [-inf, -inf, -0.77791, -0.770133]) and (se in [0.001914, 0.001916, inf, inf]) and (skew in [-inf, -inf, 3.627241, 3.663284]) => state=c1m (CF = 0.99)
3. (min in [-inf, -inf, -0.715099, -0.705441]) and (se in [0.001952, 0.001955, inf, inf]) and (kur in [-inf, -inf, 46.674071, 47.360043]) => state=c1m (CF = 0.92)
4. (min in [-inf, -inf, -0.688596, -0.664114]) and (kur in [-inf, -inf, 39.67244, 39.696621]) and (se in [0.001835, 0.001857, inf, inf]) and (skew in [-inf, -inf, 2.965135, 2.978808]) and (kur in [25.983998, 27.09646, inf, inf]) => state=c1m (CF = 0.98)
5. (min in [-inf, -inf, -0.688596, -0.68826]) and (se in [0.001917, 0.001925, inf, inf]) and (kur in [-inf, -inf, 43.618984, 43.701093]) and (skew in [-inf, -inf, 3.39066, 3.39729]) => state=c1m (CF = 0.98)
6. (se in [0.001718, 0.001739, inf, inf]) and (skew in [-inf, -inf, 2.330569, 2.399919]) => state=c1m (CF = 0.65)
7. (se in [0.001947, 0.001953, inf, inf]) and (skew in [-inf, -inf, 3.687214, 3.710682]) => state=c1m (CF = 0.92)
8. (min in [-inf, -inf, -0.790249, -0.647234]) and (skew in [-inf, -inf, 3.17084, 3.178481]) and (se in [0.001871, 0.001878, inf, inf]) and (kur in [34.028453, 35.344428, inf, inf]) => state=c1m (CF = 0.98)
9. (se in [0.002026, 0.002042, inf, inf]) => state=c1m (CF = 0.88)
10. (se in [-inf, -inf, 0.001401, 0.001571]) => state=c2m (CF = 0.99)
11. (min in [-inf, -inf, -0.677865, -0.670902]) and (se in [-inf, -inf, 0.001822, 0.001822]) and (skew in [1.766063, 1.789447, inf, inf]) => state=c3m (CF = 0.98)
12. (kur in [52.17887, 52.256831, inf, inf]) => state=c3m (CF = 0.84)
13. (kur in [41.163852, 41.872439, inf, inf]) and (se in [-inf, -inf, 0.001862, 0.001862]) => state=c3m (CF = 0.81)
14. (skew in [1.266158, 2.220344, inf, inf]) and (se in [-inf, -inf, 0.001846, 0.001848]) => state=c3m (CF = 0.87)
15. (kur in [46.674071, 48.041141, inf, inf]) and (min in [-inf, -inf, -0.851113, -0.843388]) => state=c3m (CF = 0.75)
16. (skew in [1.060682, 2.978808, inf, inf]) and (se in [-inf, -inf, 0.001876, 0.001876]) and (kur in [-inf, -inf, 40.526977, 40.946652]) and (min in [-0.811254, -0.780305, inf, inf]) => state=c3m (CF = 0.84)

17. (kur in [37.168119, 39.494076, inf, inf]) and (min in [-0.677865, -0.675765, inf, inf]) and (se in [0.001913, 0.001914, inf, inf]) and (max in [-inf, -inf, 3.072211, 3.077719]) => state=c4m (CF = 0.97)
18. (skew in [2.986058, 2.990408, inf, inf]) and (se in [-inf, -inf, 0.001909, 0.001961]) and (sd in [0.168523, 0.168523, inf, inf]) => state=c4m (CF = 0.65)

The inferences from the rules directly deliver the classification. Consider equation (1) where the rule says that for all instances of kurtosis equal to or greater than 0.183662 and less than 1.102526 will belong to Good i.e. no misfire and the rule has a certainty factor or confidence factor of 0.99. Any value lying outside this boundary does not belong to the class "Good". The confidence factor of 0.99 signifies very high rule strength. If multiple features are involved then the features are linked using the operator "and" only as observed in all the rules except rule 1. Other operators like "or, not" etc., are not required since the rules are pruned using the minimum possible description length. They will not be grown further if the smallest rule itself can effectively accomplish the classification. This observation is generally valid for all rule based classifiers. Discussion of the rules and their impact on the classifier is presented in the following section.

6. RESULTS AND DISCUSSION

The development of the expert system for misfire detection using Fayyad and Irani discretisation and decision tree based feature subset evaluator embedded in to a fuzzy round robin classifier algorithm is discussed with the implications of the following factors

- Classification accuracy of the classifier without data preprocessing
- Dimensionality reduction or feature reduction using Fayyad and Irani's algorithm
- Features subset selection using CFS

From the experimental setup through data acquisition, 200 signals have been acquired for each condition. The conditions are mentioned in section 3 and the features were extracted as mentioned in section 4. These features are pre-processed using feature reduction and features subset selection techniques and the effect of these techniques on the FURIA model is thoroughly investigated.

6.1 Evaluation of classifier

Evaluation of the FURIA classifier is performed using the standard tenfold cross validation process. The misclassifications details pertaining to FURIA without any data pre-processing is presented in the form of a confusion matrix in Table 1. *C1m* represents misfire in cylinder 1, C2m, C3m and C4m represents misfire in cylinder 2, 3 and 4 respectively. *Good* represents no misfire in any cylinder. The diagonal elements shown in the confusion matrix represents the correctly classified points and non-diagonal elements are the misclassified instances. Referring to Table 1, it is evident that the misclassification among the faulty conditions and '*good*' condition is minimal. However there are misclassifications among the faulty conditions which do not compromise the prediction accuracy but the converse is undesirable. For example consider row C1m in which 182 conditions are correctly identified as misfire in C1 but 8 are wrongly identified as misfire in C3, 8 in C4 and 2 in good. A specific setback at this point indicated in row C1m is that, two instances of misfire are wrongly misclassified as good, which is undesirable. We can infer that, adequate data preprocessing and model fine tuning are essential to avoid misclassification of good as misfire or vice versa. This model in the current form is not robust enough for real time application. Data preprocessing using FI model and FSS using CFS are embedded in to the model and evaluated for performance and robustness.

Table 1 Confusion matrix – FURIA with all features considered

STATE	Good	C1m	C2m	C3m	C4m
Good	200	0	0	0	0
C1m	**2**	182	0	8	8
C2m	0	0	200	0	0
C3m	0	23	0	135	42
C4m	0	16	0	17	167

Table 2 FURIA Classifier performance evaluation chart

	Without data preprocessing	With FI discretisation	With CFS based FSS	With CFS based FSS and FI discretisation
Model performance	88.4	87.6	87.3	86.9
Processing time taken in seconds	47.3	36.3	31.3	35.9
Number of rules generated by model	22	36	18	38
Model performance in two-class mode	99	98.5	99	97

Table 3 Decision tree Classifier performance chart

	Without data preprocessing	With FI discretisation	With CFS based FSS	With CFS based FSS and FI discretisation
Model performance	87.6	87.5	89.3	87.2
Model performance in two-class mode	99	100	99	100

The performance values depicted in Table 2 clearly portrays the performance of the developed model when subjected to various data preprocessing techniques. From the table it is evident that including all the data gives better performance but there is a risk of performance reduction due to model over fitting the data. In a later date when the engine noise increases due to wear, there are possibilities of the model suffering setbacks due to increased misclassifications. Detailed investigations reported by (Devasenapati, et al., 2010) on similar systems using decision trees are prone to instabilities when change in signal pattern appears due to ageing of the engines. The reported work has not used data preprocessing techniques which are proven to generalize the model for robust performance. The data has been used to build a decision tree classifier with data preprocessing for the sake of comparing performance. This rule based classifier is considered a bench mark classifier to evaluate other new algorithms. Table 3 depicts the performance of decision tree with similar conditions and it can be observed that all variations in decision tree are very much comparable to FURIA and achieve comparable classification efficiency. The model using CFS based features is selected since it has a higher performance in multi class mode and also performs well in two class mode but if only two class mode is essential then the option with CFS based FSS and FI discretisation can be chosen. However a judicious decision has to be taken among the available alternatives to freeze the best among the developed models.

On closely observing the rules presented in section 5.1.3 it is observed that conditions which were defined using a single rule (i.e. good and C2m) had the highest classification efficiency as depicted in (1,1) and (3,3) in table 1. The number of rules pertaining to C1m and C3m are the highest but when confidence factor (CF) is considered the condition C1m has rules with an average CF of 0.93 hence a higher classification accuracy is achieved but when condition C3m is evaluated the confidence factor of almost all the rules are less than 90% and the average CF is 0.85 leading to loss of classification accuracy. The loss of classification accuracy may be due to physical variations in components specific to cylinder 3. The system is aiming at forming more rules to capture the information for classifying C1m and C3m due to inherent noise in the signal but the strength of the rule is reflected by the CF value assigned to it. The best fuzzy system is the one which can classify the classes with minimum number of rules with a CF value as close to 1 as possible and capable of achieving 100% classification accuracy.

Table 3 Analysis of rules per class

STATE	**Good**	**C1m**	**C2m**	**C3m**	**C4m**
Number of rules	1	8	1	6	2

The fuzzy rule based classifier by virtue of its operation will avoid features that do not have appreciable information for classifications and hence those features will not form part of the classifier. The algorithm by itself performs data granulation however the effect of external data granulation (Fayyad, 1993) and feature subset selection (Hall, 2000) techniques have been analysed.

The results in Table 2 clearly confirms the findings and establishes the fact that the standalone FURIA classifier takes longer time in arriving at a decision compared to a hybrid expert system having additional data discretisation or feature subset selection algorithm supporting FURIA. This validates the need for using FI discretiser or CFS based FSS, both with comparable classification accuracy but with significantly reduced computation time required for arriving at a decision when compared to the stand alone FURIA model. The main advantage of proposing the use of FURIA is due to the fact that it has a novel rule stretching algorithm which is invoked when an unseen instance appears in the system. This is very attractive given the possibility of change in engine signature due to wear and tear.

7. CONCLUSION

In a condition monitoring activity fault identification forms the major objective and fault classification comes second in priority. In this context, the present algorithm performs fault identification (differentiating between good and faulty conditions) sufficiently well since it has not misclassified any instances out of 1000 samples supplied. This is calculated by considering good as one class and all defects as the second class. This assumption is logically valid since misfire detection is crucial and the identification of the exact cylinder where misfire happens is not critical.

From the results presented it is encouraging to conclude that FURIA based model is suitable for detection of misfire in IC engines. A relationship between the number of rules and the classifier accuracy or the computation time could not be inferred since the size and complexity of the rule could not be evaluated. Specifically focusing on the two-class model result that is presented in the fourth row of Table 2, one is able to infer that data preprocessing is absolutely necessary for improving the performance of the expert system and to reduce computational time required to arrive at a decision. The authors conclude that the model based on FI data discretisation is the best since it has 99% classification accuracy combined with minimum processing time of 31.3 seconds.

It should be noted that these results are specific to this application and cannot be generalized to other similar applications. Further studies are to be conducted on different engines at different operating conditions in order to generalize this finding.

REFERENCES

California Air Resources Board. (1991). Technical status Update and Proposed Revisions to Malfunction and Diagnostic System Requirements Applicable to 1994 and Subsequent California Passenger Cars, Light-Duty Trucks, and Medium-Duty Vehicles – (OBDII). *CARB staff report*.

Chang, J., Kim, M., & Min, K. (2002). Detection of misfire and knock in spark ignition engines by wavelet transform of engine block vibration signals. *Measurement science & technology, 13* (7), pp. 1108-1114.

Cohen, W. W. (1995). Fast Effective Rule Induction. *Proceedings of the Twelfth International Conference on Machine Learning. 1*, pp. 115-123. Lake Tahoe, California, USA.: Morgan Kaufmann.

Devasenapati, S. B., Ramachandran.K.I, & Sugumaran.V. (2010a). Misfire detection in a spark ignition engine using Support Vector Machines. *International Journal of Computer Applications, 5* (6), 25-29.

Devasenapati, S. B., Sugumaran, V., & Ramachandran, K. (2010). Misfire identification in a four-stroke four-cylinder petrol engine using decision tree. *Expert Systems with Applications, 37* (3), 2150-2160.

Dougherty, J., Kohavi, R., & Sahami, M. (1995). Supervised and unsupervised discretization of continuous features. *Proceedings 12th International Conference on Machine Learning. 1*, pp. 194-202. San Francisco, CA, USA: Morgan and Kaufmann.

Fayyad, U. I. (1993). Multi-interval discretization of continuous valued attributes for classification learning. *Proceedings of the 13th International Joint Conference on Artificial Intelligence* (pp. 1022-1027). Los Altos, California, USA: Morgan Kaufmann.

Fürnkranz, J. (1999). Separate-and-conquer rule learning. *Artificial Intelligence Review, 13*, 3-54.

Hall, M. A. (2000). Correlation-based feature selection for discrete and numeric class machine learning. *Proceedings of the Seventeenth International Conference on Machine Learning. 1*, pp. 359 - 366. San Francisco, CA, USA.: Morgan Kaufmann Publishers.

Huhn, J., & Hullermeier, E. (2009a). FR3: A fuzzy rule learner for inducing reliable classifiers, *IEEE Transactions on Fuzzy Systems, 17* (1), 138-149.

Hühn, J., & Hüllermeier, E. (2009b). FURIA: an algorithm for unordered fuzzy rule induction. *Data mining and knowledge discovery, 19* (3), 293-319.

Johanyák, Z. C. (2010). Student evaluation based on fuzzy rule interpolation. *International Journal of Artificial Intelligence, 5* (A10), 37-55.

Kalyani, S., & Swarup, K. S. (2010). Support vector machine based pattern recognition approach for static security assessment. *International Journal of Artificial Intelligence, 5* (A10), 17-36.

Klenk, M., Moser, W., Mueller, W., & Wimmer, W. (1993). *Misfire detection by evaluating crankshaft speed – A means to comply with OBDII.* SAE Paper 93099.

Lee, D., & Rizzoni, G. (1995). *Detection of partial misfire in IC engines using measurement of crankshaft angular velocity.* SAE Paper 951070.

Liu, S., Du, R. S., & Yang, S. (2000). Fuzzy pattern recognition for online detection of engine misfire by measurement of crankshaft angular velocity,. In D. Jiang, & A. Wang (Ed.), *International Conference on Sensors and Control Techniques (ICSC 2000), 1*, pp. 432-436. Wuhan, China: SPIE, WA, USA.

Liu, S., Gu, F., & Ball, A. (2002). The on-line detection of engine misfire at low speed using multiple feature fusion with fuzzy pattern recognition. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 216* (5), 391-402.

Michael, K. I., & Ciesielsk, V. (2003). An empirical investigation of the impact of discretization on common data distributions. *Proceedings of Third International Conference on Hybrid Intelligent Systems (HIS'03)* (pp. 692-701). Melbourne, Australia: IOS Press.

Navarro, J. C., & Mejia, R. V. (2010). Structural health monitoring by using pattern recognition,. *International Journal of Artificial Intelligence, 5* (A10), 24-31.

Palanisamy, C., & Selvan, S. (2009). Wavelet based fuzzy clustering of higher dimensional data. *International Journal of Artificial Intelligence, 2* (S09), 17-23.

Precup, R.-E., Tomescu, M. L., Preitl, S., & Petriu, E. M. (2008). Stable fuzzy logic control solution for Lorenz chaotic system stabilization. *International Journal of Artificial Intelligence, 1* (A08), 23-33.

Precup, R.-E., Tomescu, M. L., Preitl, S., & Petriu, E. M. (2010). Fuzzy Logic-based stabilization of a magnetic ball suspension system. *International Journal of Artificial Intelligence, 5* (A10), 56-66.

Quinlan, J., & Cameron, J. R. (1993). Foil: A midterm report. *Proceedings of the 6th European Conference on Machine Learning* (pp. 3-20). Berlin, Germany: Springer-Verlag.

Sowell, T. E. (2008, Sep). Fuzzy Logic for Just Plain Folks. *Web source* . www.fuzzy-logic.com/.

Tinaut, F. V., Andres, M., Laget, H., & Jose, D. L. (2007). Misfire and compression fault detection through the energy model. *Mechanical Systems and Signal Processing, 21* (3), 1521-1535.

Yajnik, A., & Mohan.S, R. (2009). Approximation of a signal using discrete wavelets transform. *International Journal of Artificial Intelligence, 2* (S09), 32-39.

Zadeh, L.A. (1965). Fuzzy sets. *Information and Control, 8* (3), 338-353.

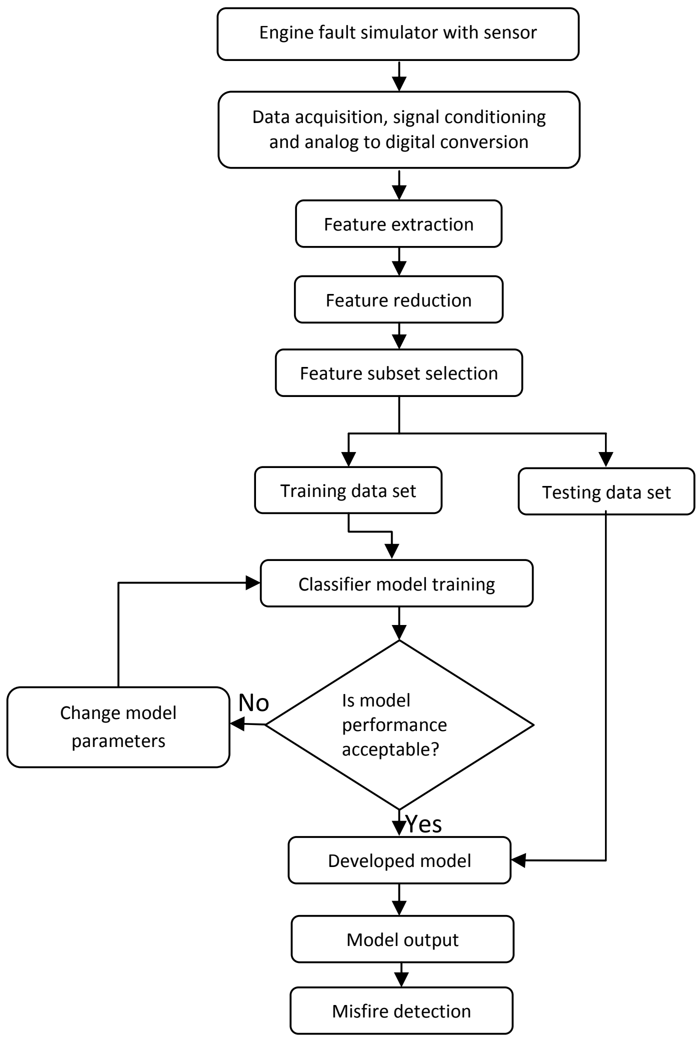
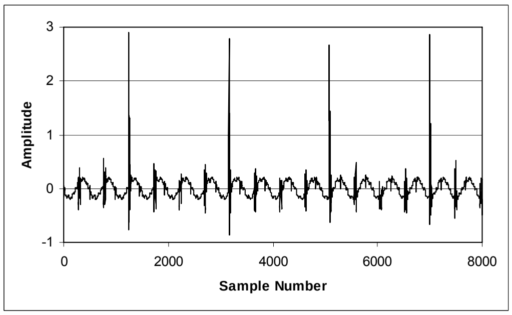


Figure 1 Flowchart for fault diagnosis system.

Figure 2 Experimental setup



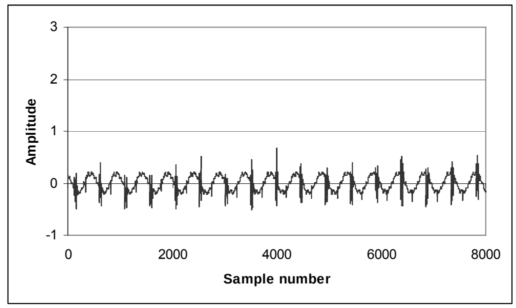


Figure 3a Amplitude plot-cylinder1 misfire

Figure 3b Amplitude plot- no misfire

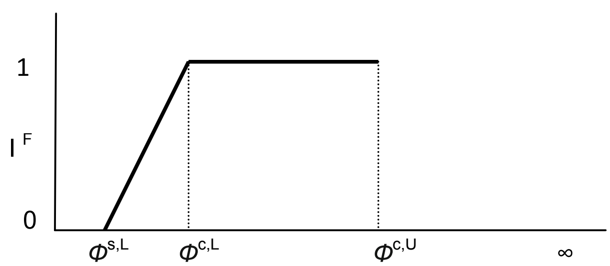


Figure 4a Membership function with open interval in upper bound

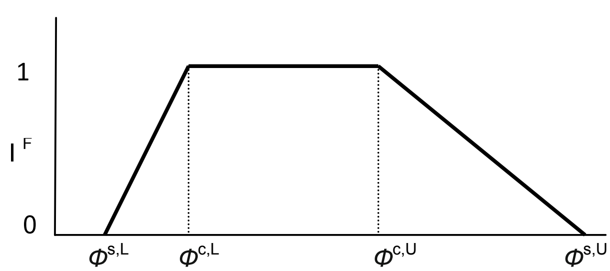


Figure 4b Non-symmetric trapezoidal membership function