

Robust Algorithm to Learn Rules for Classification - a Fault Diagnosis Case Study

P Arun Balaji

Research Scholar
Vellore Institute of Technology
Chennai - Campus
School of Mechanical Engineering (SMEC)
India

V Sugumaran

Professor
Vellore Institute of Technology
Chennai - Campus
School of Mechanical Engineering (SMEC)
India

Machine learning algorithms are used for building classifier models. The rule-based decision tree classifiers are popular ones. However, the performance of the decision tree classifier varies with hyperparameter tuning. The optimum hyperparameter values are obtained using either optimization algorithms or trial and error methods. The present study utilizes the MODLEM algorithm to overcome the drawbacks accounted for by decision tree algorithms. Eliminating hyperparameter tuning and producing results closer to standard decision tree algorithms makes MODLEM a robust classification algorithm. The robustness of the MODLEM algorithm is illustrated with the fault diagnosis case study. The case study is faults diagnosis of an automobile suspension system using vibration signals acquired at various fault conditions.

Keywords: *Fault diagnosis, Decision tree, MODLEM, Rough set, Machine learning, Classification*

1. INTRODUCTION

Machine learning algorithms are widely used to build classifiers. Amongst them, automation engineers are specifically interested in decision tree-based classifiers. Decision trees give simple (if-then rules) that can be easily implemented in real-time systems (Micro-controllers/processors). Decision tree algorithms are relatively simple to use. However, one has to tune its hyperparameters for better performance. As parameter tuning is a tricky task, especially when the parameters are many, it is challenging for classifier designers. Hence, there is a need for the new algorithm to build a decision tree that does not require hyperparameter tuning. This paper illustrates the use of the MODLEM algorithm for building a decision tree that can be used for classification problems.

To illustrate the capability of the MODLEM algorithm, a case study of the fault diagnosis of an automobile suspension system is taken up and presented. The fault diagnosis method involves acquiring raw signals, feature extraction, feature selection, and building a classifier model.

Researchers propose many decision tree algorithms using rule-based classifiers to understand the classification process [1]. As a result, plenty of rule-based algorithms are proposed, such as J48, and one R. JRip, PART, etc. [2,3]. Among these classifiers, the J48 classifier is used in many applications due to its classification ability with the limited dataset and high classification accuracy.

The J48 algorithm is used in various applications in the area of medical industries for disease identification due to the fact that generated tree is understandable by

the user and it is easy to derive rules from it [4]; for the same reason, it uses rule-based classifier J48 for identifying heart diseases is proposed by the author in [5]. The ability to create understandable rules is made way to use rule-based classifiers in mechanical industries, particularly in the area of fault diagnosis using vibration signals.

The authors, Subrata et al., proposed a method to identify faults in the gearbox (transmission device to increase torque) widely used in industries using the J48 decision tree classifier with an accuracy of 92%[6]. A method of gear fault diagnosis is proposed in automobile applications by author Praveen et al., where the J48 algorithm can distinguish faults with an accuracy of 99.70%.

The authors in [7] used the decision tree algorithm model to detect faults in centrifugal pumps using vibration signals with an accuracy of 100%. Similarly, the J48 algorithm is used in many other applications [8,9,10].

Based on the literature survey, the authors have identified the following research gap: limited numbers of research studies have been conducted on identifying faults in suspension components. The available research work has primarily focused on finding faults in specific components of the suspension system, such as the lower arm bush worn out, strut mount fault, and tie rod ball joint fault. However, these specific faults have received little attention. The existing research in this field has predominantly employed a data-driven approach using a vibrating platform to induce forced vibrations. Additionally, several studies have utilized standard algorithms to classify faults in rotary machinery like suspension systems. However, there needs to be more information regarding the tuning of hyperparameters in these algorithms. Moreover, these parameters will vary depending on the type of signal and the specific problem under consideration.

The authors propose a robust rule-based classifier called the MODLEM to address these challenges. This

Received: September 2022, Accepted: May 2023

Correspondence to: Dr V Sugumaran, School of Mechanical Engineering, Vellore Institute of Technology – Chennai Campus, India

E-mail: v_sugu@yahoo.com

doi: 10.5937/fme2303338B

© Faculty of Mechanical Engineering, Belgrade. All rights reserved

FME Transactions (2023) 51, 338-346 338

classifier aims to overcome the difficulties mentioned above associated with parameter tuning by providing a more effective approach for fault classification in suspension components.

The MODLEM classifier is a Meta classifier that uses an understanding of dataset characteristics to improve the algorithm performance and uses this acquired knowledge to assist in selecting a learning algorithm based on the dataset's characteristics.

The MODLEM classifier outperforms the other rule-based classifiers because it can generate automated rules that are easy to understand. Hence, the author in [11] used the MODLEM algorithm to identify liver diseases. Similarly, the authors in [12] conducted a comparative study of various rule-based classifier algorithms. In their study, MODLEM produced the second-highest classification accuracy. Hence, in the present study, the advantage and robustness of the MODLEM classification algorithm for fault diagnosis of rotary machinery are illustrated by benchmarking with the J48 algorithm with accuracy and confusion matrix of fault diagnosis of various suspension component faults. As only a few studies were conducted in condition monitoring of vehicle suspension systems [13,14,15], this paper illustrates the robustness of the MODLEM algorithm over the J48 algorithm for fault diagnosis of automobile suspension systems.

2. MODLEM ALGORITHM

In rule-based classifiers, when the same attributes are used again to describe some other object, the problem of inconsistency in deriving the rule arises [15] shown in Figure 1.

Stefanowski introduced the MODLEM (Modified Learning from Examples Module) algorithm to overcome the abovementioned drawback proposed by induct rules. It works through sequential covering and heuristically produces a small set of decision rules for every decision concept. Moreover, the rule can cover all the positive examples and omit the negative examples, as shown in Figure. 2.

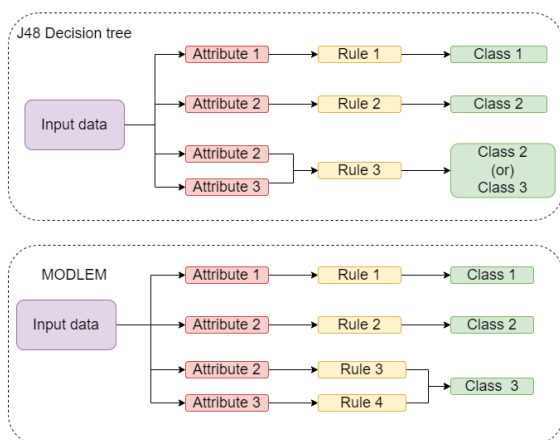


Figure 1. The function of rule-based classifiers and MODLEM

The author used a relabeling filter and MODLEM algorithm to improve the classification accuracy with the dataset having an imbalance class (or minority class

having less number of instances to learn from the data). Also, the author quoted that the advantage of the MODLEM classifier is that rules induced by the algorithm proved the best single classifier. Furthermore, it can handle various data properties and works with reasonable computational costs to generate rules. The algorithm can be used for two categories, one for descriptive analysis and another for predictive analysis. In the case of descriptive analysis, the algorithm describes a relationship between an attribute and an object by determining their dependency level. In the predictive analysis, the algorithm performs based on experience learning to identify an object for the given rules for the attributes [16,17].

The main procedure for rule induction order is shown in steps as follows:

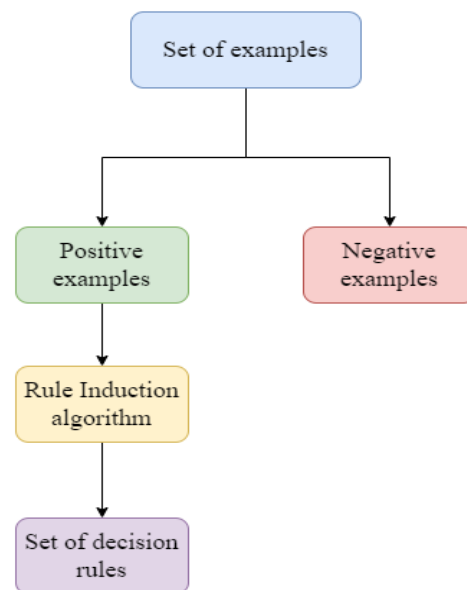


Figure 2. Rule induction algorithm for knowledge acquisition

The main procedure for rule induction order is shown in steps as follows:

Step 1: Formulating the first rule by the optimum selection of primary conditions based on selected criteria

Step 2: Store the rule

Step 3: Remove all the examples which learned positively from the stored rule

Step 4: Continue removing all the examples till all the decision concepts are identified

Step 5: If some of the positive examples are uncovered, then repeat the step 1 to 4 sequentially for the succeeding decision concept

In the MODLEM algorithm, the process of rule induction involves handling numerical attributes and generating elementary conditions for the rules. These conditions are represented as either $(ar < vl)$ or $(ar \geq vl)$; in this, 'ar' denotes the attribute, and 'vl' denotes the value. While building a rule, if the same attribute (ar) is selected twice, then it may be deducted as $(ar = [vl1, vl2])$ that results from an intersection of two conditions $(ar < vl2)$ and $(ar \geq vl1)$ such that $vl1 < vl2$. For nominal attributes, these conditions are $(ar = va)$ or could be extended to the set of values.

The algorithm for the MODLEM in given in Figure. 3 (Stefanowski, 2007).

Algorithm : RULE-based learning algorithm (MODLEM)

```

(B - input - a group of positive instances present in a
decision concept;
criteria - performance measure;
T - output single local covering of I, taken as rule
condition parts)
begin
G := B; {A instantaneous group of rules formulated
based on rules generated}
T := ∅;
While G ≠ ∅; do {searching for rules until the
presence
of unidentified examples}
  begin
    T := ∅; {a candidate for a rule condition part}
    S := U; {a group of objects hold by O}
    while (T = ∅) or (not(T ⊆ B)) do {end condition
for rule selection}
      begin
        t := ∅; {a candidate for an elementary
condition}
        for each attribute q ∈ C do {search of best
elementary condition}
          begin
            new_t := Find optimum condition(q; S);
            if Better(new_t; t; criteria)
            then t := new_t;
            {evaluate if a new conditions is better than
previous one
according to the chosen evaluation
measure}
          end;
          T := T ∪ {t}; {add the best condition to the
candidate rule}
          S := S ∩ {t}; {focus on examples covered
by the candidate}
        end; { while not(T ⊆ B) }
        for each elementary condition t ∈ T do
          if [T - t] ⊆ B then T := T - {t}; {test a rule
minimality}
          T := T ∪ {t}; {store a rule}
          G := B - ∪t ∈ T [t]; {remove already covered
examples}
        end; { while G ≠ ∅ }
      set;
    end; {procedure}
  Function to find best condition
  (input c - given attribute; S - set of examples;
output best_t - best condition)
  begin
    best_t := ∅;
    if c is a numerical attribute then
      begin
        H := list of selected values for
attribute c and objects from S;
        { H (i) - in distinct value in the list
        }
        for i := 1 to length(H)-1 do
          if object class assignments for H(i)
and H(i+1) are different then
            begin
              v := (H(i) + H(i+1)) / 2;
              create a new_t as either (c < v) or (c ≥
v);
            if Better(new_t; best_t; criteria)
            then best_t := new_t;
            end;
          end;
        else { attribute is nominal }
          begin
            for each value v of attribute c do
              if Better(c=v, best_t, criteria)
              then best_t := (c=v);
            end;
          end;
        end; {function}.
  
```

Figure 3. MODLEM algorithm

PROs

1. Union of rules made it a best single classifier
2. Ability to handle various data properties, such as numerical attributes, without its pre-discretization.
3. Ability to work as an ensemble classifier.
4. Ability to handle an imbalanced data set and provides optimum classification accuracy.
5. No need to tune numerical hyperparameters like other rule-based classifiers.
6. Consistent performance.
7. Easy to understand the classification process from the rule
8. Much more insights about the data set is possible

Cons.

1. Required additional computation time and resources compared to other rules-based classifiers
2. Less popular among the researchers

Hence, in order to test the performance of the MODLEM classifier, the following study was conducted to benchmark the MODLEM classifier's performance with the J48 algorithm.

Table 1: Data set description

S.No	Drive Speed	Load condition	Fault condition	Total no. of signals
1	780 rpm (≈ 60kmph)	No load	GOOD, LABWO,	800 (100 signals/condition)
2		200 psi	LABJF, STWO, STED,	800 (100 signals/condition)
3		400 psi	TRBJF, STMF, WLP	800 (100 signals/condition)

3. EXPERIMENTAL DESCRIPTION

The experiment is conducted to identify the fault at the earliest to prevent further damage, which is necessary to maintain the safety and reliability of the vehicle since the suspension is used to maintain uniform contact between wheels with the road surface [18,19].

The experiment was carried out on test setup, and the vibrational signals were acquired, simulating the suspension's real-time working at different load conditions. The experimental test setup used in the studies to acquire data is shown in Figure. 4a. The test setup consists of suspension system components (strut, lower arm, knuckle, tie rod wheel, and loading system). A piezoelectric sensor was deployed on the lower arm to measure the vibration signals.

The vibration signals were acquired with a sampling frequency of 20 kHz using a data acquisition system (National Instruments) and LabVIEW software. The acquired signals for each fault condition are portrayed in Figure. 4b. The collected data consist of eight states of the suspension system, namely, Good, lower arm bush worn out (Labwo) [20], strut mount fault (Stmf), lower arm ball joint fault (LABJF)[21], strut worn out (Stwo) [22], strut external damage (Sted), tie rod ball joint fault (Trbjf) [23] and low wheel pressure (Wlp) [24]. The image of the faulty components considered for the study is given in Figure 5.

A detailed data set description is given in Table 1.

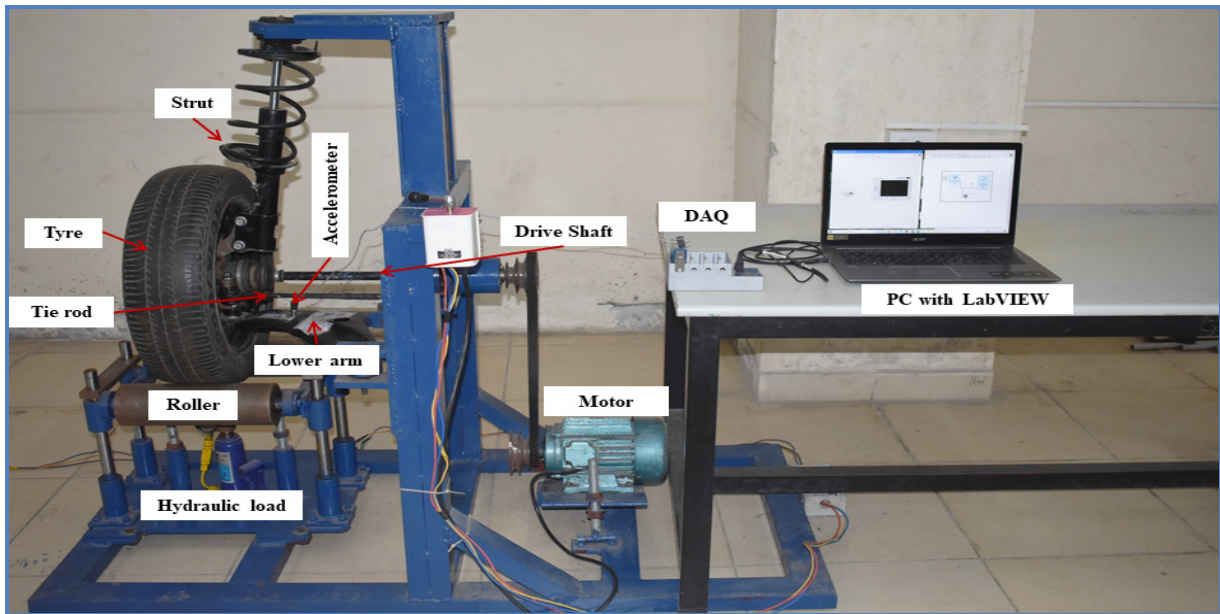


Figure 4a. Suspension fault experimental test rig

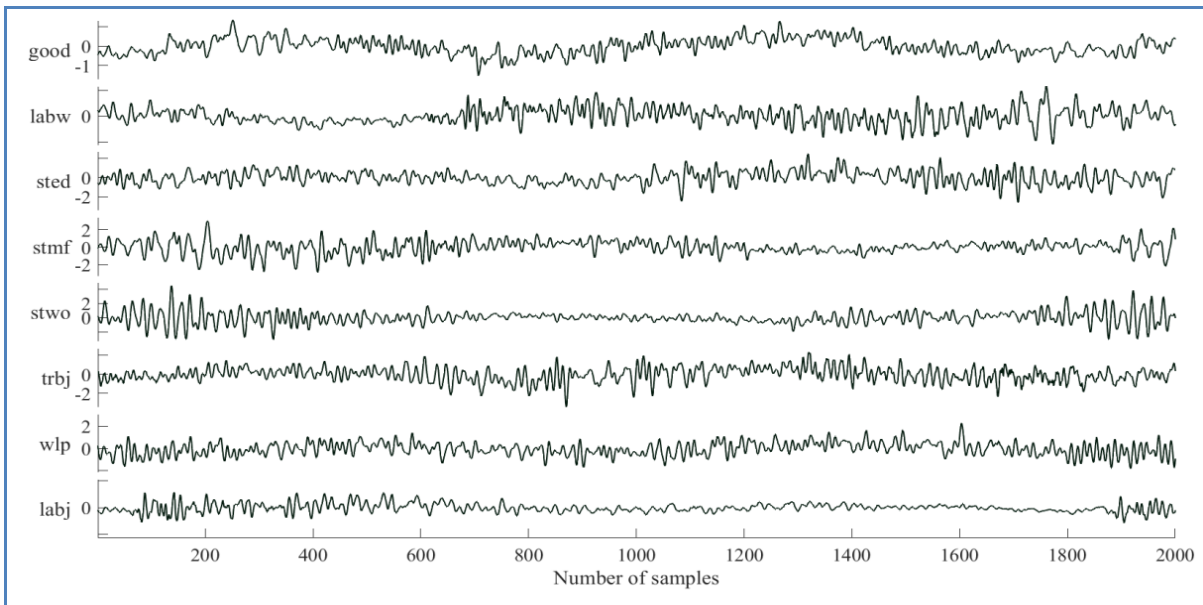


Figure 4b. Time domain plots of vibration signals.



Figure 5. a- Sted, b-Stmf, c- Labwo, 5d-Trbj, e-Labjf, f-Wlp

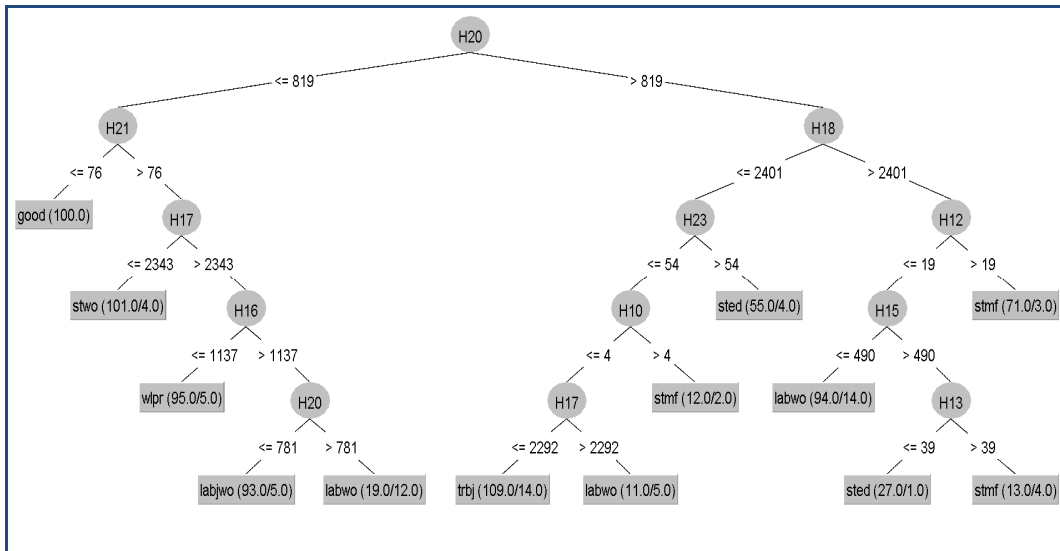


Figure 6: Statistical Feature selected for No-load condition(J48)

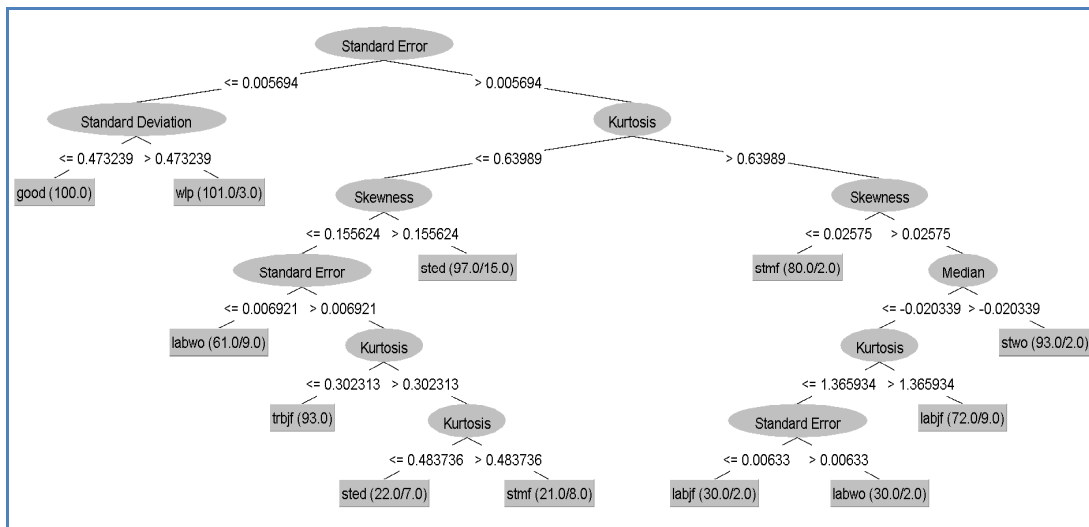


Figure 7: Histogram Feature selected for No-load condition(J48)

4. FEATURE EXTRACTION

The feature extraction was done to extract useful information by recognizing the pattern from the raw signal. In fault diagnosis studies, statistical features, histogram features, and discrete wavelet transform features ARMA features are widely used features by researchers [25–28]. For the current study, statistical and histogram features were considered. Statistical and histogram features can be used to deduce the unique pattern from the signals. In the case of statistical feature extraction, the following features are extracted from the signal using Microsoft statistical toolbox. They are sum, mean, median, mode, maximum, range, skewness, kurtosis, minimum, standard error, sample variance, and standard deviation[28,29].

Observing the time domain plot of all the fault signals implies that the acceleration amplitude values vary from class to class. Histogram plots can bring out such variations in amplitude and pattern in the vibration signals of various conditions. The bins of the histograms can be chosen such that the amplitude differences of vibration signals form a unique pattern for a particular condition.

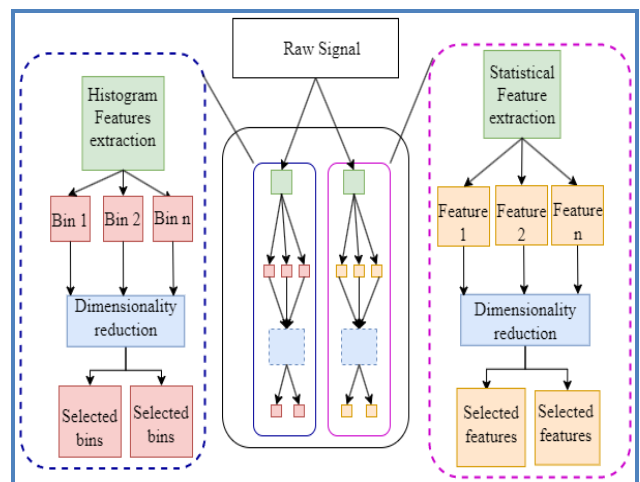


Figure 8: Feature extraction and selection process

5. DIMENSIONALITY REDUCTION

Feature selection is a crucial step in the classification process as it involves removing irrelevant features that have no impact on the classification outcome, thereby reducing computational load. This process is also

known as dimensionality reduction. Figure 5 illustrates the feature engineering process, which encompasses feature extraction and selection. During the feature engineering process, the J48 algorithm's decision tree was utilized to identify significant features [30-33]. The decision tree consists of multiple nodes that follow a top-to-bottom flow. The topmost node is known as the root node, and branching occurs until a leaf node (class) is reached. Figure 6 and Figure 7 depict sample decision trees generated by the J48 algorithm.

6. FAULT CLASSIFICATION USING J48 AND MODLEM CLASSIFIER

In this section, the performance of the MODLEM classification algorithm at various load conditions was benchmarked with the J48 algorithm to verify the advantage of MODLEM in the fault diagnosis process. For that, the selected statistical features and histogram features were used in the classification processes, and the performance measure, namely, classification accuracy and confusion matrix, is considered. The hyper-parameter setting for the J48 algorithm and MODLEM at the time of the fault classification process is described in Table 2.

Table 2 Parameters of two classifiers

Classifier	Hyperparameter	Range/Value
J48	Batch size	100
	C-confidence factor	varied from 0 to 1 in steps of 0.1 (shown in Figure 8)
	M- minimum number of object	varied from 5 to 100 in step of 5 (shown in Figure 9)
MODLEM	Classification strategy	m estimate
	Conditional measure	conditional entropy
	Rules type	lower approximation of certain rules

Figure 9 and Figure 10 display the plots drawn between the confidence factor and J48 classifier performance. From the plots, one can observe that the optimum confidence factor for each load condition varies. From Figure 8, the optimum confidence factor for statistical features is 0.1, 0.1, & 0.2 for no load, half load, and full load conditions, respectively. Similarly, for histogram features, the confidence factor of 0.1 produce maximum classifier performance in all three load conditions.

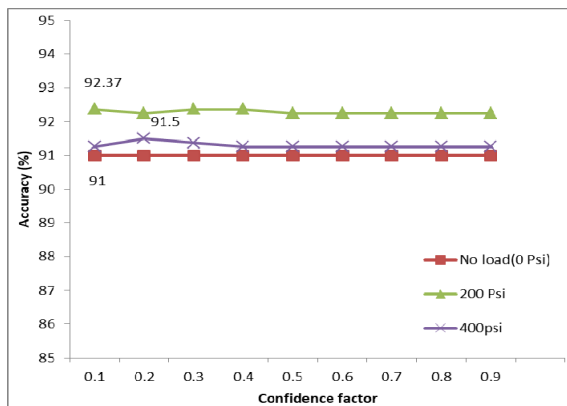


Figure 9. Parameter tuning of a confidence factor for statistical features

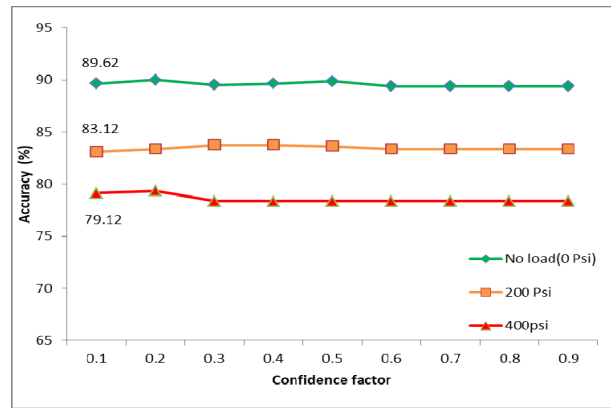


Figure 10. Parameter tuning of a confidence factor for histogram features

Figure 11 and Figure 12 illustrate the relationship between a minimum number of objects and classifier performance. From these plots, one can understand that the minimum number of objects required considering for rule-making for all load conditions is five for both statistical and histogram features, respectively.

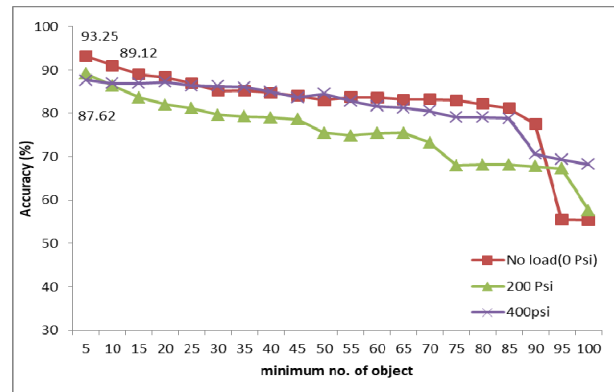


Figure 11. parameter tuning of minimum no. of the object for Statistical Features

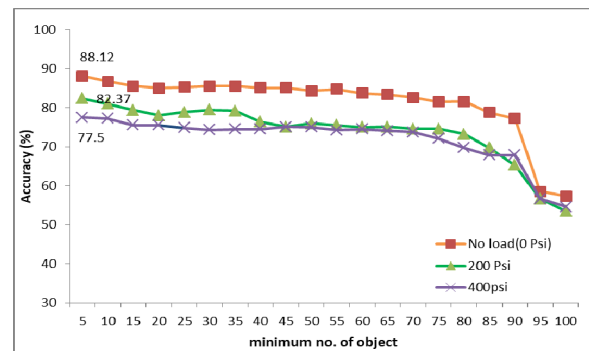


Figure 12. Parameter tuning of minimum no. of objects for histogram features

7. RESULT AND DISCUSSION

Verification of classification Performance of the MODLEM at different load conditions

The performance measure discussed here is classification accuracy from Figure.12. One can observe that the accuracy of MODLEM is significantly higher compared to the J48 algorithm in all load conditions. Also, the improved performance can be seen even in different data sets of histogram features. Similarly, the second performance measure, the confusion matrix, is

used in many applications to know the actual classification happened. The row-wise element in the confusion matrix indicates the actual class, and column-wise indicates the predicted class, as shown in Table 3.

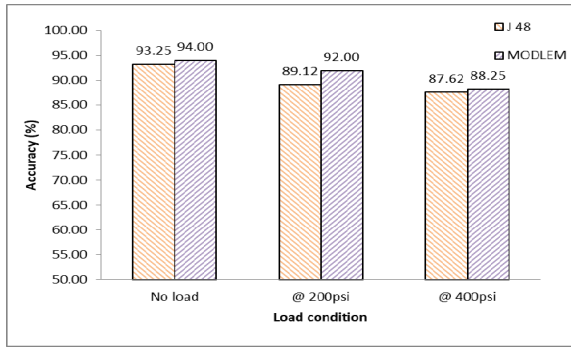


Figure 13: Comparison of results of J48 and MODLEM classifier with statistical features

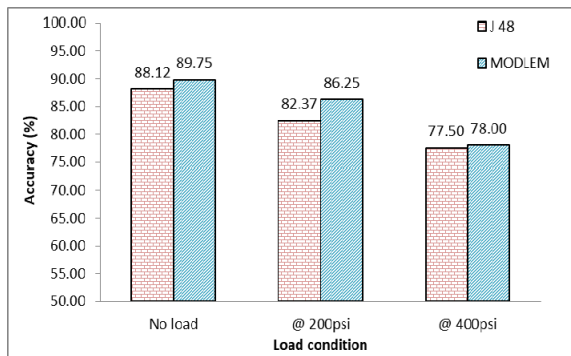


Figure 14: Comparison of results of J48 and MODLEM classifier with histogram features

From Figures 13 and 14, one can understand that the MODLEM algorithm has higher classification accuracy compared to J48 in both cases of statistical and histogram features. Also, from the confusion matrix given in Tables 3 and 4, the classification capability of the MODLEM is similar to that of the J48 algorithm.

Table 5 provides a breakdown of the class-wise accuracy of the MODLEM algorithm, representing the performance of the MODLEM classifier in terms of true positive rate (TP), false positive rate (FP), precision (Pr), recall, and F-measure. TP measures the proportion of instances correctly classified as "good," while FP represents mistakenly classified instances. In an ideal classifier, TP should be close to one, and FP should be zero.

From Table 5, the average TP value is greater than 0.8. Hence, the rule-based classifier will be best suitable for this type of specific problem. Precision (Pr) refers to the probability of correctly classifying retrieved instances for a specific class[32]. It is calculated as the ratio of true positive (TP) to the sum of true positive and false positive instances (TP+FP). Precision is also known as the positive predictive value and serves as a measure of accuracy or quality.

Table 5: Class-wise accuracy of MODLEM classifier under different loads

Feature type	Load condition	TP Rate	FP Rate	Precision	Recall	F-measure	ROC
Statistical features	No load	0.94	0.009	0.94	0.94	0.939	0.96
	200PSI	0.92	0.011	0.92	0.92	0.92	0.95
	400PSI	0.88	0.017	0.88	0.88	0.88	0.93
Histogram features	No load	0.89	0.015	0.89	0.89	0.89	0.94
	200PSI	0.86	0.020	0.86	0.86	0.86	0.92
	400PSI	0.78	0.031	0.77	0.78	0.77	0.87

Recall, also known as sensitivity, represents the ability of the classifier to correctly classify instances (TP) out of the total number of instances (TP+FN). False-negative (FN) instances are considered type 2 errors, indicating cases where the classifier misclassifies the actual category. The F-measure is defined as the harmonic mean of both recall and precision. It can be seen as an approximate average of recall and precision. When recall and precision values are close, the F-measure is generally the square of the geometric mean divided by the arithmetic mean. The f-measure is expressed as
$$2 * \frac{(\text{recall} * \text{precision})}{(\text{recall} + \text{precision})}$$

Table 3. J48 classifier confusion matrix at No-load condition

a ₁	a ₂	a ₃	a ₄	a ₅	a ₆	a ₇	a ₈	Class
99	0	0	0	0	0	0	1	a ₁ = GOOD
0	88	4	0	2	3	0	3	a ₂ = LABJF
0	6	86	2	1	1	4	0	a ₃ = LABWO
0	0	6	89	2	0	3	0	a ₄ = STED
0	0	4	3	92	0	1	0	a ₅ = STMF
0	1	0	0	0	99	0	0	a ₆ = STWO
0	0	0	2	0	0	98	0	a ₇ = TRBJF
0	4	1	0	0	0	0	95	a ₈ = WLP

Table 4. MODLEM classifier confusion matrix at No-load condition

a ₁	a ₂	a ₃	a ₄	a ₅	a ₆	a ₇	a ₈	Class
99	0	0	0	0	0	0	1	a ₁ = GOOD
0	93	1	0	2	2	0	2	a ₂ = LABJF
0	6	82	6	4	0	2	0	a ₃ = LABWO
0	0	2	91	2	0	5	0	a ₄ = STED
0	2	2	1	94	1	0	0	a ₅ = STMF
0	0	0	0	0	100	0	0	a ₆ = STWO
0	0	0	2	1	0	97	0	a ₇ = TRBJF
1	3	0	0	0	0	0	96	a ₈ = WLP

8. CONCLUSION

This study proposes the use of the MODLEM classifier to identify rules for classifier design. A comparative study was conducted using a dataset acquired from a specially designed suspension test setup, aiming to classify multiple faults in suspensions. The results demonstrate that the MODLEM algorithm outperforms the J48 algorithm, which required parameter tuning for each dataset. In the current case study, the MODLEM algorithm achieves an average classification accuracy of 91.42% for statistical features and 84.67% for histogram features when classifying faults. Furthermore, the experiment highlights the applicability of the MODLEM classifier in various industrial applications due to its robust performance compared to standard tree-based classifiers used for fault identification in rotary machinery based on vibration signals.

NOMENCLATURE

ARMA	Auto-regressive moving average
MODLEM	Modified learning from examples module
LABJF	lower arm ball joint fault
Pr	Precision
Sted	Strut external damage
Trbjf	Tie rod ball joint fault
TP	True positive rate
FP	False positive rate
TN	True negative rate
FN	False negative rate
Stwo	Strut worn out
Labwo	Lower arm bush worn out
Wlp	Wheel low pressure
Stmf	Strut mount fault

ACKNOWLEDGMENT

This research has not received any funding from any agencies.

REFERENCES

- [1] Colombo D, Gobbi M, Mastinu G, et al. (2019) Sensing methodologies to determine automotive damper condition under vehicle normal operation. *Mech Syst Signal Process* 1:1–6. <https://doi.org/10.1016/j.apenergy.2018.08.030>
- [2] Egaji OA, Chakhar S, Brown D (2019) An innovative decision rule approach to tire pressure monitoring. *Expert Syst Appl* 124:252–270. <https://doi.org/10.1016/j.eswa.2019.01.051>
- [3] Stefanowski J, Janicka M, Lango M, et al. (2014) A rule-based algorithm for liquid rocket combustion. *J Phys Conf Ser* 4:1–6. <https://doi.org/10.2514/6.2007-5528>
- [4] Mahesh JU, Naganjaneyulu K V, Likitha P, Aishwarya KNSS (2014) Analysis of J48 Algorithm in Classification-Ebola Virus. *Int J Emerg Trends Sci Technol* 1:1289–1292. <https://doi.org/10.13140/RG.2.2.17135.76961>
- [5] Vachiravel S (2015) Diagnosis of Heart Disease Using Decision Tree and SVM Classifier. *Int J Appl Eng Res* 10:598–602
- [6] Mukherjee S, Kaushal R, Kumar V, Sarangi S (2021) A Novel Approach of Gearbox Fault Diagnosis by Using Time Synchronous Averaging and J48 Algorithm. Springer Singapore
- [7] Muralidharan V, Sugumaran V, Sivanathan S (2010) Artificial Neural Network based Classification for Monoblock Centrifugal Pump using Wavelet Analysis. *Int J Mech Eng* 1:28–37
- [8] Ihya R, Namir A, El Filali S, et al. (2019) J48 algorithms of machine learning for predicting user's the acceptance of an E-orientation Systems. In: *ACM International Conference Proceeding Series*
- [9] Daud N, Mohd Noor NL, Aljunid SA, et al. (2019) Predictive Analytics: The Application of J48 Algorithm on Grocery Data to Predict Obesity. In: 2018 IEEE Conference on Big Data and Analytics, ICBDA 2018. pp 1–6
- [10] Joshua A, Sugumaran V (2019) Comparative study on tree classifiers for application to condition monitoring of wind turbine blade through histogram features using vibration signals: A data-mining approach. *SDHM Struct Durab Heal Monit* 13:399–416. <https://doi.org/10.32604/sdhm.2019.03014>
- [11] Abdelminaam DS, El-ashmawi WH, Ibraheem SA (2019) HMFC: Hybrid MODLEM – Fuzzy Classifier for Liver Diseases Diagnose
- [12] Joshua A, Vishnuvardhan R, Deenadayalan G, et al. (2019) Implementation of rule based classifiers for wind turbine blade fault diagnosis using vibration signals. *Int J Recent Technol Eng* 8:320–331. <https://doi.org/10.35940/ijrte.B1050.0982S1119>
- [13] Abdelfattah A, Ibrahim H (2021) Health Monitoring of Automotive Suspension System using Machine Learning. *Proc 7th Int Conf Veh Technol Intell Transp Syst (VEHITS 2021)* 325–332. <https://doi.org/10.5220/0010402503250332>
- [14] Weispfenning T, Isermann R (1997) Fault Detection of Vehicle Suspensions. *IFAC Proc Vol 30: 801–806*. [https://doi.org/10.1016/s1474-6670\(17\)42498-0](https://doi.org/10.1016/s1474-6670(17)42498-0)
- [15] Lozoya-Santos J, Tudón-Martínez JC, Morales-Menendez R, et al (2012) Fault detection for an automotive MR damper. In: *IFAC Proceedings Volumes (IFAC-PapersOnline)*. pp 1023–1028
- [16] Stefanowski J (2007) On combined classifiers, rule induction and rough sets. *Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics)* 4374 LNCS:329–350. https://doi.org/10.1007/978-3-540-71200-8_18
- [17] Stefanowski J (2004) An experimental evaluation of improving rule based classifiers with two approaches that change representations of learning examples. *Eng Appl Artif Intell* 17:439–445. <https://doi.org/10.1016/j.engappai.2004.04.017>
- [18] Wang G, Yin S (2014) Data-driven fault diagnosis for an automobile suspension system by using a clustering based method. *J Franklin Inst* 351:3231–3244. [https://doi.org/https://doi.org/10.1016/S1474-6670\(17\)34773-0](https://doi.org/https://doi.org/10.1016/S1474-6670(17)34773-0)
- [19] Khode SS (2017) A Review on Independent Suspension System of Light Commercial Vehicle. *IOSR J Mech Civ Eng* 17:14–19. <https://doi.org/10.9790/1684-17010061419>
- [20] Quebec UDU (2014) Vibration and force analysis of lower arm of suspension system. Thesis
- [21] Raes S, Devreese T, De Pauw J, De Baets P (2015) Design of a tribological ball joint tester. *Int J Sustain Constr Des* 6:5. <https://doi.org/10.21825/scad.v6i1.1137>
- [22] Nozaki H, Inagaki Y (1999) Technology for measuring and diagnosing the damping force of shock absorbers and the constant of coil springs when

- mounted on a vehicle. *JSAE Rev* 20:413–419. [https://doi.org/10.1016/S0389-4304\(99\)00024-7](https://doi.org/10.1016/S0389-4304(99)00024-7)
- [23]23. Ferreira C (2018) A New Methodology for Detection of a Loose or Worn ball joint used in vehicles suspension system. *Conf Theor Exp Mech Mater / 11th Natl Congr Exp Mech Porto/Portugal* 10:1–6
- [24]24. Isermann R, Wesemeier D (2009) Indirect Vehicle Tire Pressure Monitoring with Wheel and Suspension Sensors. *IFAC*
- [25]25. Alamelu Manghai TM, Jegadeeshwaran R (2019) Vibration based brake health monitoring using wavelet features: A machine learning approach. *JVC/Journal Vib Control* 25:2534–2550. <https://doi.org/10.1177/1077546319859704>
- [26]26. Sen A, Majumder MC, Mukhopadhyay S, Biswas RK (2017) Condition Monitoring of Rotating Equipment Considering the Cause and Effects of Vibration: A Brief Review. *Int J Mod Eng Res* 7:36–49
- [27]27. Madhusudana CK, Kumar H, Narendranath S (2016) Condition monitoring of face milling tool using K-star algorithm and histogram features of vibration signal. *Eng Sci Technol an Int J* 19:1543–1551. <https://doi.org/10.1016/j.jestch.2016.05.009>
- [28]28. Kumar DP, Muralidharan V, Hameed SS (2022) Multi-Point Tool Condition Monitoring System - A Comparative Study. *FME Trans* 50:193–201. <https://doi.org/10.5937/fme2201193K>
- [29]29. Jawad SM, Jaber AA (2021) Rolling Bearing Fault Detection Based on Vibration Signal Analysis and Cumulative Sum Control Chart. *FME Trans* 49:684–695. <https://doi.org/10.5937/fme2103684M>
- [30]30. Sugumaran V, Muralidharan V, Ramachandran KI (2007) Feature selection using Decision Tree and classification through Proximal Support Vector Machine for fault diagnostics of roller bearing. *Mech Syst Signal Process* 21:930–942. <https://doi.org/10.1016/j.ymssp.2006.05.004>
- [31]31. Sugumaran V, Ramachandran KI (2011) Effect of number of features on classification of roller bearing faults using SVM and PSVM. *Expert Syst Appl* 38:4088–4096. <https://doi.org/10.1016/j.eswa.2010.09.072>
- [32]32. Bhosale D, Ade R, R. Deshmukh P (2014) Feature Selection based Classification using Naive Bayes, J48 and Support Vector Machine. *Int J Comput Appl* 99:14–18. <https://doi.org/10.5120/17456-8202>
- [33]33. Dave V, Thakker H, Vakharia V (2022) Fault Identification of Ball Bearings using Fast Walsh Hadamard Transform, LASSO Feature Selection, and Random Forest Classifier. *FME Trans* 50:202–210. <https://doi.org/10.5937/fme2201202D>

**РОБУСТАН АЛГОРИТАМ ЗА УЧЕЊЕ
ПРАВИЛА ЗА КЛАСИФИКАЦИЈУ – СТУДИЈА
СЛУЧАЈА ДИЈАГНОЗЕ ГРЕШКЕ**

П.А. Балази, В. Сугумаран

Алгоритми машинског учења се користе за изградњу модела класификатора. Класификатори стабла одлучивања засновани на правилима су популарни. Међутим, перформансе класификатора стабла одлучивања варирају са подешавањем хиперпараметара. Оптималне вредности хиперпараметара се добијају коришћењем алгоритама оптимизације или метода покушаја и грешака. Ова студија користи МОДЛЕМ алгоритам да би се превазишли недостаци алгоритама стабла одлучивања. Елиминасање подешавања хиперпараметара и стварање резултата ближих стандардним алгоритмима стабла одлучивања чини МОДЛЕМ робусним класификационим алгоритмом. Робусност МОДЛЕМ алгоритма је илустрована студијом случаја дијагнозе грешке. Студија случаја је дијагностика кварова система овјеса аутомобила помоћу сигнала вибрација добијених у различитим условима квара.