Event-driven Data Mining Techniques for Automotive Fault Diagnosis

Chaitanya Sankavaram[1], Anuradha Kodali[1], Diego Fernando Martinez Ayala[1], Krishna Pattipati[1], Satnam Singh[2], and Pulak Bandyopadhyay[2]

[1] *Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269 USA*
E-mail: krishna@engr.uconn.edu

[2] *Diagnosis & Prognosis Group, India Science Lab, General Motors Global Research and Development,*
GM Technical Centre India Pvt Ltd, Bangalore, INDIA
E-mail: satnam.singh@gm.com

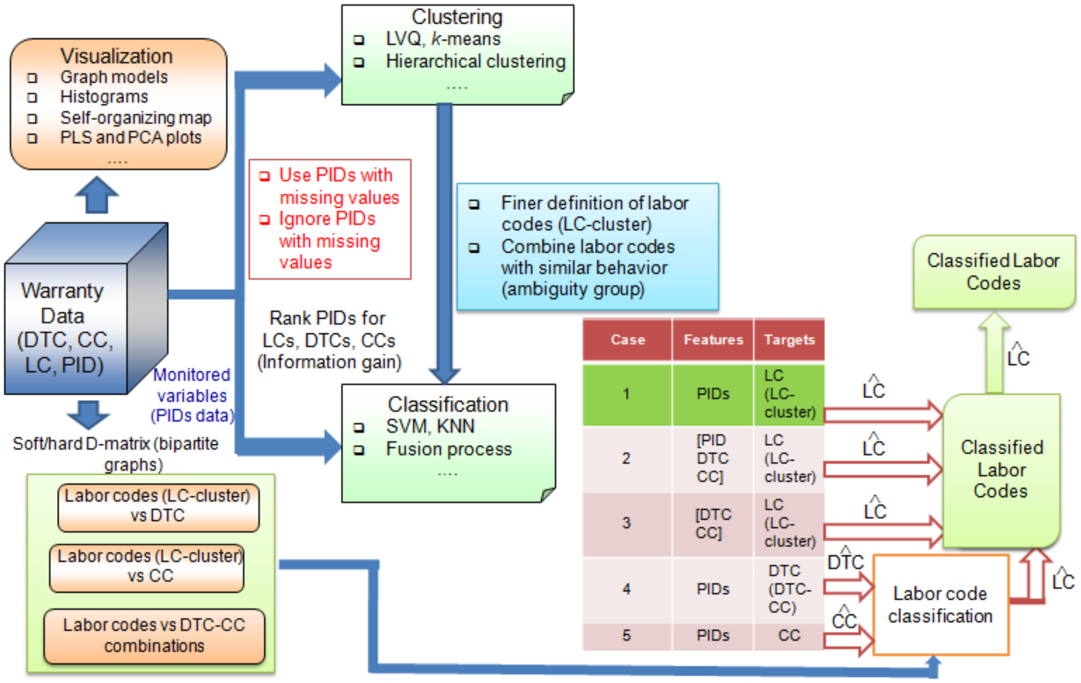
ABSTRACT

The increasing sophistication of electronics in vehicular systems is providing the necessary information to perform data-driven diagnostics. Specifically, the advances in automobiles enable periodic acquisition of data from telematics services and the associated dealer diagnostic data from vehicles; this requires a data-driven framework that can detect component degradations and isolate the root causes of failures. The event-driven data consists of diagnostic trouble codes (DTCs) and the concomitant parameter identifiers (PIDs) collected from various sensors, customer complaints (CCs), and labor codes (LCs) associated with the repair. In this paper, we discuss a systematic data-driven diagnostic framework featuring data pre-processing, data visualization, clustering, classification, and fusion techniques and apply it to field failure datasets. The results demonstrated that the support vector machine (SVM) classifier with DTCs and customer complaints as features provides the best accuracy (74.3%) compared to any other classifier and that a tree-structured classifier with SVM as the base classifier at each node achieves approximately 75.2% diagnostic accuracy.

1 INTRODUCTION

The relentless competition among automotive OEMs, increased demands from customers for dynamically-controlled safety systems and growing dependence on electronics are creating the need for a continuous monitoring system that tracks and identifies the trends and sources of component degradations prior to failure. Automotive OEMs collect a variety of on-board vehicle health data via telematics and off-board data via dealer diagnostics services. These data sources acquire different types of vehicle data at different sampling rates. For example, dealer diagnostic data is collected when a vehicle comes for repair at a dealer shop; the warranty data, collected infrequently, includes the diagnostic trouble codes (DTCs), freeze frame data (engineering variables/PIDs), repairs/replacement actions, and structured/unstructured text in the form of customer verbatim. The fleet data is collected at a much higher sampling frequency (e.g., every few ignition cycles) for overall health of vehicle subsystems, such as the engine and/or transmission system, emission system, airbag system, anti-lock brake system, tire pressure; this data is gathered even when the vehicle is functioning normally. However, what is needed is an early warning capability that continuously monitors the data, detects, isolates and estimates the severity of faults (viz., fault detection and diagnosis) based on models that includes cross-subsystem fault propagation effects, and relates the detected degradations in vehicles to accurate remaining life-time predictions (viz., prognosis) of replaceable components.

Methods for fault diagnosis can be classified as being associated with one or more of the following three approaches: model-based, knowledge-based, or data-driven. What if a mathematical model (for model-based diagnosis) or cause-effect graph model of system failures and their manifestations (for knowledge-based approach to diagnosis) is not available? The Data-driven approach to fault diagnosis is an alternative, provided that system monitoring data is available. A data-driven approach to fault diagnosis has close relationship with pattern recognition, wherein one learns classification rules directly from the data, rather than using mathematical models or a knowledge-based approach. Due to its simplicity and adaptability,


Visualization
□ Graph models
□ Histograms
□ Self-organizing map
□ PLS and PCA plots
....
Clustering
□ LVQ, *k*-means
□ Hierarchical clustering
....
□ Use PIDs with missing values
□ Ignore PIDs with missing values
Finer definition of labor codes (LC-cluster)
□ Combine labor codes with similar behavior (ambiguity group)
Warranty Data (DTC, CC, LC, PID)
Monitored variables (PIDs data)
Rank PIDs for LCs, DTCs, CCs (Information gain)
Soft/hard D-matrix (bipartite graphs)
Labor codes (LC-cluster) vs DTC
Labor codes (LC-cluster) vs CC
Labor codes vs DTC-CC combinations
Classification
□ SVM, KNN
□ Fusion process
....
Classified Labor Codes
$\widehat{LC}$
$\widehat{DTC}$
$\widehat{CC}$
Labor code classification

Case	Features	Targets
1	PIDs	LC (LC-cluster)
2	[PID DTC CC]	LC (LC-cluster)
3	[DTC CC]	LC (LC-cluster)
4	PIDs	DTC (DTC-CC)
5	PIDs	CC

Fig. 1. Diagnostic Process Overview

customization of a data-driven approach does not require an in-depth knowledge of the system. However, data-driven techniques provide no information on unobserved faults, even though they may be anticipated (Sankavaram et. al., 2009).

In this paper, we propose a systematic data-driven diagnostic framework for fault diagnosis. The key features of the proposed framework include data pre-processing, visualization, clustering, classification, and fusion algorithms to detect and isolate faults and to reduce no-trouble found rates and warranty costs. Often, the data acquired via onboard diagnostic systems (telematics), dealer diagnostics etc., require data pre-processing techniques in order to process the DTCs, engineering variables and the vehicle information. Hence, the data-preprocessing technique plays a crucial role in fault diagnosis. Furthermore, in order to diagnose the faults of interest, a number of classifier techniques are employed, viz., support vector machines (SVM), probabilistic neural network (PNN), Gaussian mixture models (GMM), *k*-nearest neighbor (KNN) classifier and so on (Bishop, 2006; Duda *et al.*, 2001). The diagnostic process also exploits clustering techniques, such as Principal Component Analysis (PCA), Partial Least Squares (PLS) and Linear Vector Quantization (LVQ) especially to cluster/ group the ambiguous faults in order to improve the classification accuracy. Furthermore, by examining the confusion matrices, we suggest a tree-structured classifier, where the ambiguous labor codes (LCs) are grouped together and a classifier is applied on the subset of LCs. We also illustrate a major limitation of data-driven approaches, viz., its inability to diagnose anticipated, but unobserved faults. We recommend that a recursive classifier should be developed that can incrementally adapt with the observed cases for unanticipated faults. The proposed diagnostic process is generic; it can be applied to a wide range of subsystems across vehicle types and models. We demonstrate the process on two datasets (dataset 1 and dataset 2) gleaned from field failure databases.

The paper is organized as follows: In Section 2, we describe our data-driven diagnostic framework. In Section 3, we describe the characteristics of the datasets, demonstrate our proposed framework and discuss the classification results. Finally, the paper concludes with a summary in Section 4.

2 DATA-DRIVEN DIAGNOSTIC FRAMEWORK

Our data-driven diagnostic framework, shown in Fig. 1, consists of visualization, feature selection, clustering, classification, and fusion techniques. Each process is explained in detail in the following subsections.

2.1 Graphical Visualization of Data

Data visualization helps in assessing the level of difficulty of classification problem and also helps in selecting data pre-processing techniques for better diagnosis (e.g., grouping or subdividing labor codes, etc.). We employed a variety of data visualization techniques viz., Self Organizing Map (SOM), PCA,

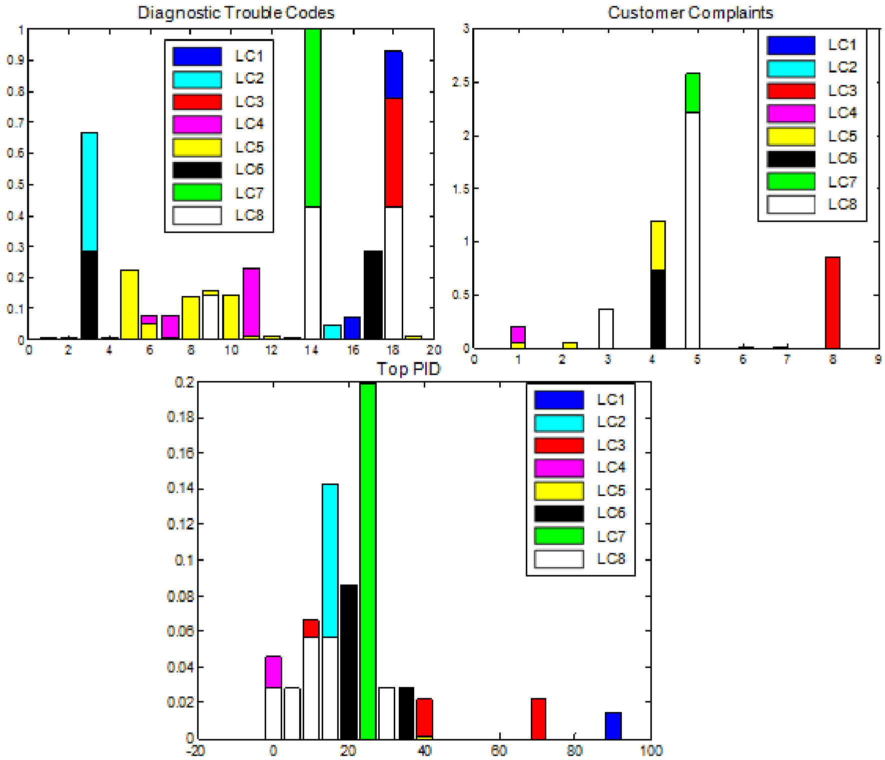


Fig. 2. Histograms of Top PID, DTCs and Customer Complaints

PLS, histograms, etc., as part of our diagnostic process. For instance, Fig. 2 shows the histograms of the top PID and DTCs conditioned on the LCs. We could observe that the DTCs (also CCs) may provide better isolation of labor codes when compared to the top PID alone (according to information gain criterion) because there is no significant overlap of labor codes with DTCs and CCs. This is borne out by our classification analysis later.

2.2 Bi-partite Fault models

Fault model is a probabilistic dependency matrix which depicts the cause-effect relationships between the failure modes (LCs) and the tests (DTCs, CCs). We have derived the fault models between (*i*) LCs and DTCs, (*ii*) LCs and CCs, and (*iii*) LCs and DTC-CC combinations via maximum likelihood estimation of probabilities given by,

$$\hat{p}_{ij} = \frac{n_{ij}}{n_i} \quad \Rightarrow \hat{P}(O_j / LC_i) = \frac{n(O_j, LC_i)}{n(LC_i)} \qquad (1)$$

where O_j is DTC_j or CC_j or $(DTC,CC)_j$, n_{ij} is the number of times O_j is associated with LC_i and n_i is the total number of observed cases with LC_i. In order to avoid the problem with ML estimate i.e., the possibility of having a zero probability because of an unseen combination of $(O_{j,}\ LC_j)$ in the training data, we use Laplacian smoothing (Metzler *et al.*, 2004) given by,

$$\hat{P}(O_j / LC_i) = \frac{n(O_j, LC_i)+1}{n(LC_i)+|LC|} \qquad (2)$$

where |LC| is the number of labor codes. The dependency matrix can be either binary (0 or 1 (hard)) or probabilistic (soft). These fault models provide an alternative method for inferring the LCs from the classified DTCs or CCs (see Fig1).

2.3 Feature Selection

The main idea of feature selection is to choose a subset of PIDs by eliminating those with little or no predictive information. We rank ordered PIDs via mutual information gain algorithm and selected the minimum number of PIDs required for classification based on the diagnostic accuracy on test case data. The mutual information (MI) (Cover *et. al.,* 1991) between a feature, F and a class, C is given by,

$$MI(F,C) = \sum_{C} P(C)[\int_{F} p(F \mid C)\log\frac{p(F \mid C)}{p(F)}dF] \qquad (3)$$

Histograms of relevant probability density functions from data are used to compute (3). We seek minimum number of PIDs to decrease the implementation complexity of classification approaches.

2.4 Classification

Our framework features a number of statistical classifiers, exemplified by SVM, decision trees, PNN,KNN, PCA, PLS, and GMM. A brief explanation of these techniques is given in the following subsections.

2.4.1 Support Vector Machine

Support vector machines transform the data to a higher dimensional feature space and find an optimal hyperplane that maximizes the margin between the classes (Burges, 1998). SVM has two distinct features. One is that it is often associated with the physical meaning of data, so that it is easy to interpret, and the other one is that it requires only a small amount of training data. A kernel function is used for fitting non-linear models by transforming the data into a higher dimension before finding the optimal hyperplane.

2.4.2 Decision Trees

Decision tree classifier predicts the class (LC, DTC, CC) based on selected features. The interior nodes of the tree correspond to one of the feature variables, while the edges correspond to the discrete outcomes of the feature variable. Each leaf represents a class given the values of the feature variables represented by the path from the root to the leaf. The decision tree is constructed using a variety of techniques, including AND/OR graphs, information gain, and information gain coupled with rollout (Tu *et al*., 2003).

2.4.3 Probabilistic Neural Network

The PNN is a supervised method that computes the likelihood of an input vector belonging to a specific class based on the learned probability distributions of each class. The learned patterns can also be weighted with *a priori* probability (relative frequency) of each category and misclassification costs to determine the most likely class for a given input vector. If the relative frequency of the categories is unknown, then all the categories can be assumed to be equally likely and the determination of category is solely based on the closeness of the input feature vector to the distribution function of a class (Duda *et. al.,* 2001).

2.4.4 *k*-Nearest Neighbor

The KNN classifier is a simple non-parametric method for classification (Duda *et. al.,* 2001). The KNN classifier calculates the distance of the input vector using *k*-nearest points from the training data and the class with the maximum a posteriori probability from those points is declared as the most-likely class. Normally, *k* is chosen as an odd number to avoid ties. Mathematically this can be viewed as computing the *a posteriori* class probabilities $P(c_i|\underline{x}_{new})$ as,

$$P\left(c_i \mid \underline{x}_{new}\right) = \frac{k_i}{k} p\left(c_i\right) \tag{4}$$

where k_i is the number of vectors belonging to class c_i within the *k*-nearest points. A new input vector $\underline{x}_{new}$ is assigned to the class c_i with the highest *a posteriori* class probability $P(c_i|\underline{x}_{new})$.

2.4.5 Principal Component Analysis

PCA is a multivariate statistical procedure that transforms the training data into a lower-dimensional space by transforming a number of correlated variables into a smaller number of uncorrelated new variables called principal components. These components represent the selection of a new coordinate system obtained by rotating the original variables and projecting them onto the reduced space defined by the first few principal components, where the first one describes the largest amount of variation in the data, the second one the second largest amount of variation in the data and so on. Each principal component is represented as a linear combination of the columns (*J*), and has a specific numerical value for each of the rows (*I*). In matrix form, the PCA model is:

$$X_s\left(i, j\right) = \sum_{f=1}^{L} \underline{t}_f \underline{p}_f^T + E \tag{5}$$

Here L is the number of principal components. The loading vectors ($\underline{p}_f$) are orthonormal and provide the directions with maximum variability. The score vectors ($\underline{t}_f$) from the different principal components are the coordinates for the objects in the reduced space. A classification of a new test pattern is done by obtaining its predicted scores and residuals. If the test pattern is similar to a specific class in the training network, the scores will be located near the origin of the reduced space, and the residual should be small. The distance of test data from the origin of the reduced space can be measured by Hotelling statistic (Nomikos, 1996; Wold *et. al.,* 1987).

2.4.6 Partial Least Squares

PLS is similar to principal component analysis (PCA). In PCA, the scores are calculated to give an optimal summary of input *X*, while in PLS the optimality is relaxed to make scores better predictors of the dependent (response) matrix *Y*. The PLS algorithm reduces the dimensionality of the input and output spaces to find highly correlated latent variables (score vectors), i.e., those that not only explain the variation in the input data *X*, but their variations that are most predictive of the output data *Y*. Once the latent variables are extracted, a least squares regression is performed to estimate the fault class. The scores are determined using nonlinear iterative partial least squares (NIPALS) algorithm. (Geladi, *et al*., 1986)

2.4.7 Gaussian Mixture Models

GMM is a multimodal distribution resulting from a number of component Gaussian functions (Duda *et. al.* 2001). It is characterized by a vector Θ of mean μ, variance σ^2 and weights of its components C given by,

$$p(x\,|\,\Theta) = \sum_{k=1}^{C} w_k N(x; \mu_k, \sigma_k) \tag{6}$$

where $$N(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{7}$$

and $$\Theta = \{\mu_1, \sigma_1, w_1, ...\mu_C, \sigma_C, w_C\}. \tag{8}$$

GMM constructs a probability density function of data for each class, and the mixture parameters for each class are learned via expectation-maximization (EM) algorithm. In deployment phase, an input test vector ($\underline{x}_{new}$) is categorized as class c_i if the maximum posterior probability $P(c_i|\underline{x}_{new})$ is attained by c_i.

2.5 Clustering

Clustering techniques can be used to perform 3 tasks, viz., visualization, classification, and grouping of data. Here, we use clustering to group data for each LC, thereby generating finer labor codes (failure modes) for classification. This division or decomposition of LCs into failure modes may improve classification accuracy in some cases. We employed *k*-means, GMM, and LVQ techniques to decompose labor codes. The *k*-means algorithm divides the data into '*k*' clusters based on the minimum distance criterion, i.e., a data point is assigned to a cluster '*i*' if its distance to the centroid of cluster *i* is minimum. LVQ is a supervised clustering algorithm that determines weight vectors (codebook) representing each output category and a data pattern is assigned to a cluster '*i*' if it is close to the weight vectors corresponding to cluster '*i*' (Kohonen, 1995).

2.6 Fusion Process

Another approach to increase the classification accuracy is to employ fusion of different classifiers. The labor code inferences generated from a PIDs-based classifier, a DTC-CC based classifier, and combined PIDs-DTC-CC classifier are fused with the estimated labor codes from the classification of DTCs and CCs based on the PIDs data (via the fault model). The LC which was classified majority times is declared as the final fused labor code. Thus, our framework facilitates the fusion of labor code estimates obtained from different classifiers to improve the classification accuracy.

3 EXPERIMENTAL RESULTS OF DATA-DRIVEN FRAMEWORK ON TWO DATASETS

We implemented a MATLAB data-driven toolbox to implement the diagnostic process and the experimental results are described in the following subsections.

3.1 Data Preparation

As mentioned earlier, the two datasets (dataset 1 and dataset 2) considered for the experimental validation of our diagnostic process had two types of data: the repair data and the DTC data[*]. Hence, for each of the datasets, the two types of data are matched and integrated into a single dataset via an automated database query program; our analysis has mainly focused on the "combined dataset". We also analyzed dataset 1 and compared the results of dataset 1 with that of the combined dataset (i.e., dataset 1 + dataset 2).

The major task here is to convert the PIDs data of the DTCs into a matrix, where each column corresponds to a PID and each row corresponds to an observed DTC. Each of these cases were matched with the corresponding claim (or labor code) data.

3.2 Data Characteristics

The combined data has 469 distinct PIDs associated with the observed DTCs. However, not all the PIDs are associated with every DTC. In some cases, for the same DTC, all the PIDs pertaining to that DTC are not recorded. This results in a "missing value" problem for classification. There are two approaches to address the missing value problem: ignore missing valued PIDs (i.e., consider only those PIDs that are recorded for every DTC) or employ classification techniques (e.g., SVM) that handles missing valued PIDs. We experimented with both of these approaches and found that ignoring the missing valued PIDs was the better approach for the datasets provided (more details on this provided under classification analysis).

The histograms of LCs are shown in Fig. 3; these plots provide insights into the most frequent labor codes. In both datasets, LC5 is the most frequently replaced component (63% of claims in dataset 1 and 75% of claims in dataset 2). Unfortunately, this component also has the highest replacement cost. Consequently, the manufacturer of this component should take corrective measures to reduce its frequency of failures (i.e., improve its reliability). Also, the most frequent DTCs are different in the two datasets; and although the number of customer complaint types has increased from 3 to 7, they are consistent across the two datasets. For example, CC2 and CC3 are major complaints in both the datasets. This also illustrates a major problem with data-driven approaches, viz., their inability to diagnose anticipated, but unobserved faults. This is because the number of DTCs fired, the number

[*] We cannot identify the subsystem for proprietary and competitive reasons.

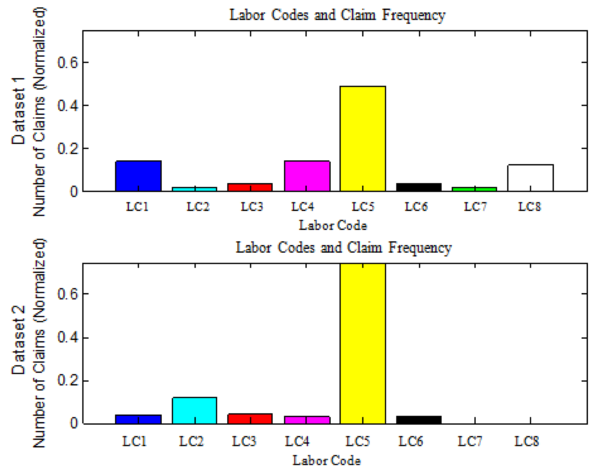


Fig. 3. Histograms of Labor Codes for datasets 1 and 2

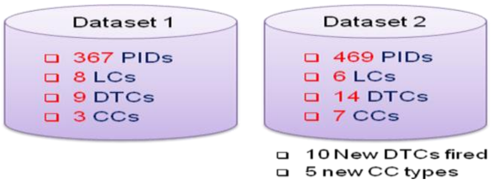


Fig. 4. Details of the Datasets

of PIDs, the number of customer complaint types, and the number of labor codes is different in the two datasets. For example, dataset 1 has 367 PIDs, 8 LCs, 9 DTCs and 3 CCs where as dataset 2 has 469 PIDs, 6 LCs, 14 DTCs and 7 CCs (see Fig. 4.). A way out of this problem is to create a recursive classifier that can adapt to unanticipated faults with the observed cases.

3.3 Classification and Clustering Analysis

We experimented with a variety of classifiers and with different feature sets. Typical setup includes either a subset of top-ranked PIDs alone or PIDs combined with DTCs and customer complaints or DTCs and customer complaints alone as the features (see Table I). Our extensive experimentation revealed that only the top PID is adequate for the combined dataset.

As shown in Table I, for the combined dataset, the SVM classifier with DTCs and customer complaints as features provided the best accuracy (74.3%) compared to any other classifier we experimented with. As already stated, PIDs data has missing values, which can be handled by SVM implicitly. PIDs with missing values could increase the diagnostic accuracy by 2% (69.7% → 71.6%); it classifies every case as LC5, except that 2 cases of LC5 are classified as LC4 and one case as LC1. Since 67% of the total claims are associated with the labor code LC5, this is not surprising. However, since all the other labor codes are classified as LC5, diagnostic *precision* is severely impacted. Also, note that applying clustering techniques to decompose the labor codes increases the diagnostic accuracy from 69.7% to 71.2% when the top PID is used. In all the other experimental runs, there was no advantage gained by clustering.

TABLE I. Comparison of (SVM) classifiers for the combined dataset and the dataset 1 alone

Approaches: Features [Targets]	**Average classification rate (%)**				
	Dataset 1		Combined Dataset		
	Raw	Clustered	Raw	Clustered	Raw
PIDs [Labor codes]	58.6 (9 PIDs)	74.2 (16 PIDs)	69.7 (1PID)	71.2 (1 PID)	71.6 (all PIDs)
PIDs +DTC+CC [Labor codes]	58.6 (9 PIDs)	74.2 (16 PIDs)	73.4 (1PID)	71.2 (1 PID)	71.6 (all PIDs)
DTC+CC [Labor codes]	20.7	56.1	74.3	69.7	N/A

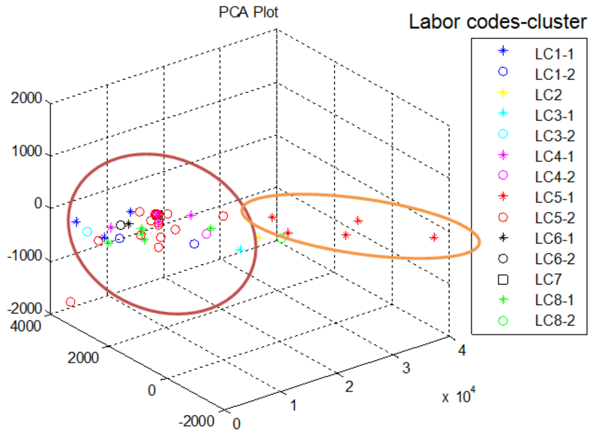


Fig 5. PCA clustering plot of dataset 1

TABLE II. Comparison of different classifier performances for the combined raw dataset

Approaches: Features [Targets]	**Average classification rate (%)**			
	KNN	RBF Network	Naïve Bayes	Random Forest
PIDs [Labor codes]	69.7 (1PID)	60.5 (1 PID)	69.7 (1 PID)	61.5 (1 PID)
PIDs +DTC+CC [Labor codes]	73.4 (1 PID)	66 (1 PID)	72.5 (1 PID)	72.5 (1 PID)
DTC+CC [Labor codes]	68.8	71.6	73.4	70.6

In Table I, we also included the results obtained by the analysis on dataset 1. Using dataset 1 alone, we obtained the best diagnostic accuracy of 74.2% by employing clustering on each labor code and performing classification using 16 top-ranked PIDs. Fig. 5 shows the PCA plot of the three principal components and the associated labor code clusters. It can be easily seen that the cases corresponding to LC5 cluster (i.e., LC5-1) are well separated from other cases suggesting that these can be classified easily. However, the DTCs and customer complaints did not improve accuracy in this case.

We also validated the classification model on dataset 2, i.e., trained the model on dataset 1 and tested it on dataset 2 using 9 top-ranked PIDs and by ignoring the missing values. The classification accuracy was only about 58%. This again points to the need to recursive classifiers that can adapt to unanticipated faults with the observed cases.

In Table II, we have also included the performance of different classifiers on the combined dataset. It shows that SVM classifier (see Table I) consistently performs well compared to any other classifier.

3.4 Rules for Classification Analysis

By examining the confusion matrices of various experimental runs (e.g., Table III), we developed a tree-structured classifier, where the internal nodes of the tree correspond to an SVM classifier on subsets of LCs. Here, the ambiguous labor codes are grouped into a single LC and the rest into another group. Then, a binary classifier is employed to classify the two sets of grouped LCs. For example, as shown in Fig. 6, at the root node C_1, labor codes (LC3, LC4, LC6, LC7, LC8) are grouped together, while the other group consists of (LC1, LC5, and LC2). After classification is performed at the C_1 node, further classification is carried out within each group until each labor code is classified at the leaf nodes. The tree-structured classifier increased the diagnostic accuracy from 74.3% to 75.2%.

If the leaf node is diagnosed incorrectly, a sequential replacement strategy is employed to isolate the labor code based on increasing order of the ratio of cost of each component to its prior probability. Fig. 7 shows the tree- structured classifier with the sequential replacement strategy. For example, if the classifier at node C'_2 categorizes a case as LC2, we replace LC2 and then test whether the problem is solved or not. If the problem is not solved, then we replace the next component in the list, LC5.

Assuming that the components are ordered in increasing order of (c_i/p_i), the expected cost of the tree-structured classifier is calculated via:

$$\sum_{i=1}^{m} p_i \sum_{j=1}^{i} c_j = \sum_{j=1}^{m} c_j \sum_{i=j}^{m} p_i \qquad (9)$$

where p_i is the probability of labor code i and c_j is the cost of replacing component j (see Table IV). Raghavan *et al.*, 1999 prove the optimality of the ratio strategy for sequential replacement.

In order to evaluate the benefits of employing the tree-structured classifier, we compare the expected cost of the tree-structured classifier with that of a knowledge management (KM) system that rank-orders component replacement decisions based on (c_i/p_i) only. This tree (shown in Fig. 8) depends only on the prior

TABLE III. Confusion matrix of tree structured classification analysis

	LC1	LC2	LC3	LC4	LC5	LC6	LC7	LC8
LC1	7	0	0	0	0	0	0	0
LC2	0	3	0	0	7	0	0	0
LC3	2	0	2	0	0	0	0	0
LC4	2	0	0	0	4	0	0	0
LC5	3	2	0	0	70	0	0	0
LC6	1	0	0	0	2	0	0	0
LC7	0	0	0	0	1	0	0	0
LC8	0	0	0	0	3	0	0	0

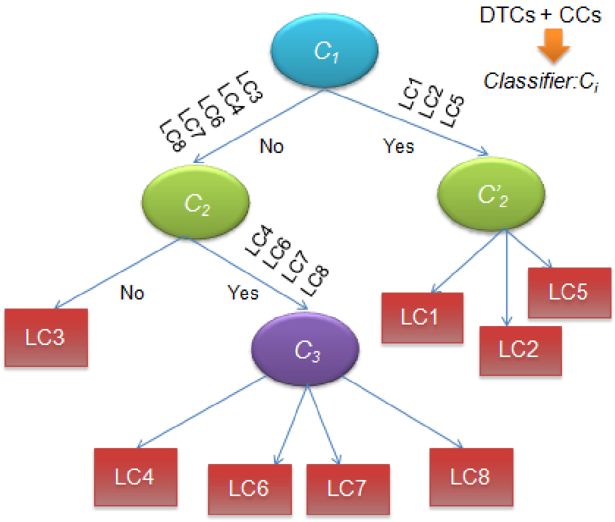


Fig. 6. Tree-structured classifier

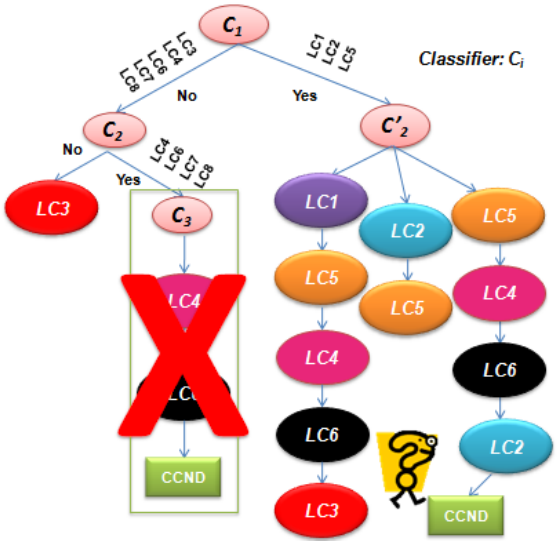
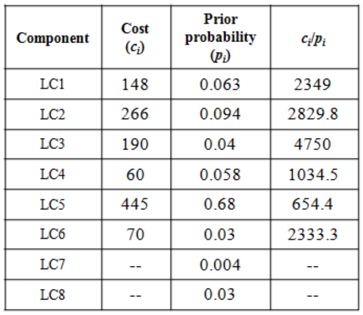


Fig. 7. Repair strategy developed from the tree-structured classification analysis

TABLE IV. Replacement Costs and Prior Probabilities of Components



Component	Cost (c_i)	Prior probability (p_i)	c_i/p_i
LC1	148	0.063	2349
LC2	266	0.094	2829.8
LC3	190	0.04	4750
LC4	60	0.058	1034.5
LC5	445	0.68	654.4
LC6	70	0.03	2333.3
LC7	--	0.004	--
LC8	--	0.03	--

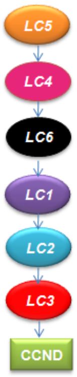


Fig. 8. Knowledge management tree that rank orders component replacements in increasing order of (c_i/p_i)

probability and cost information and does not perform classification. The results showed that average savings of 15.67% are possible with the tree-structured classifier (details are not shown here).

4 CONCLUSIONS

In this paper, we briefly discussed a systematic data-driven diagnostic framework to improve the first time fix rate, enhance vehicle availability and reduce warranty costs. We applied our framework on two datasets from a field failure database. The monitored data on DTCs and the associated PIDs, together with CCs, enable the application of sophisticated classifier and fusion techniques to isolate LCs. We have shown that DTCs and CCs provide better classification of LCs (74.3%) over a PIDs-based classifier (69.7%). Besides, we employed a tree-structured classifier with an SVM at each node to sequentially test the related components for the failure cause and to improve diagnostic accuracy (75.2%). Major advantage of these classification-based diagnosis over a KM system that considers priors only accrues when the entropy of probability mass function of labor codes is large (i.e., it is not a skewed distribution).

We have also observed the limitations of data-driven approach in handling unobserved LCs and DTCs. We recommend that a recursive classifier be created at the outset and adapt this classifier *incrementally* for unanticipated faults with the observed cases. The development of incremental data-driven learning techniques is an area of our ongoing research.

ACKNOWLEDGMENT

The work reported in this paper was supported, in part, by the GM India Science Lab, Bangalore, India and the National science foundation (Cyber-Physical systems) under grant no. ECCS-0931956. We thank GM ISL for their helpful suggestions and support in this work. Any opinions expressed in this paper are solely those of the authors and do not represent those of the sponsor.

REFERENCES

(Bishop, 2006) C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

(Burges, 1998) C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121-167, 1998.

(Cover *et. al.,* 1991) Cover, T. M., and J. A. Thomas, *Elements of Information Theory*, Wiley, 1991.

(Duda *et al*., 2001) R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification : 2nd edition*, John Wiley and Sons, 2001.

(Jackson, 1991) J. E. Jackson, *A User's Guide to Principal Components*, John Wiley & Sons, New York, 1991.

(Kohonen, 1995) T. Kohonen, *Self-organizing maps*, Springer, Berlin, 1995.

(Metzler *et al*., 2004) D. Metzler, V. Lavrenko and W. B. Croft, "Formal Multiple Bernoulli Models for Language Modeling," In. proc. 27th annual international ACM conference on Research and development in information retrieval, SIGIR' 04, Sheffield, UK, 2004, pp. 540-541.

(Nomikos, 1996) P. Nomikos, "Detection and Diagnosis of Abnormal Batch Operations Based on Multi-way Principal Component Analysis," *ISA Tran.*, vol. 35, no. 3, pp. 259-266, 1996.

(Raghavan *et al*., 1999) V.Raghavan, M. Shakeri, and K. Pattipati, "Test Sequencing Algorithms with Unreliable Tests," *IEEE Transactions on Systems, Man and Cybernetics: Part A - Systems and Humans*, vol. 29, no. 4, July 1999, pp. 347-357.

(Sankavaram *et al.,* 2009) C. Sankavaram, B. Pattipati, A. Kodali, K. Pattipati, M. Azam, and S. Kumar, "Model-based and Data-driven Prognosis of Automotive and Electronic Systems", 5th Annual IEEE Conference on Automation Science and Engineering, Bangalore, India, August 22-25, 2009.

(Tu *et al*., 2003) F. Tu and K.R. Pattipati, "Rollout Strategies for Sequential Fault Diagnosis," *IEEE Transactions on Systems, Man, and Cybernetics: Part A: Systems and Humans*, Vol. 33, No. 1, January 2003, pp. 86-99.

(Wold, *et al*., 1987) Wold, S., P. Geladi, K. Esbensen, and J, Ohman, "Principal Component Analysis", *Chemometrics and Intelligent Laboratory Systems*, vol. 2:37-52, 1987.

(Geladi, *et al*., 1986) Geladi, P., and B. R. Kowalski, "Partial Least-Squares Regression: A Tutorial", *Analytica, Chemica, Acta*, 1986.