

# Novel Classifier Fusion Approaches for Fault Diagnosis in Automotive Systems

Kihoon Choi, *Member, IEEE*, Satnam Singh, Anuradha Kodali, Krishna R. Pattipati, *Fellow, IEEE*, John W. Sheppard, *Fellow, IEEE*, Setu Madhavi Namburu, *Member, IEEE*, Shunsuke Chigusa, *Member, IEEE*, Danil V. Prokhorov, *Senior Member, IEEE*, and Liu Qiao, *Senior Member, IEEE*

**Abstract**—Faulty automotive systems significantly degrade the performance and efficiency of vehicles and are often major contributors of vehicle breakdown; they result in large expenditures for repair and maintenance. Therefore, intelligent vehicle health-monitoring schemes are needed for effective fault diagnosis in automotive systems. Previously, we developed a data-driven approach using a data-reduction technique, coupled with a variety of classifiers, for fault diagnosis in automotive systems. In this paper, we consider the problem of fusing classifier decisions to reduce diagnostic errors. Specifically, we develop three novel classifier fusion approaches: 1) class-specific Bayesian fusion; 2) joint optimization of the fusion center and individual classifiers; and 3) dynamic fusion. We evaluate the efficacies of these fusion approaches on an automotive engine data. The results demonstrate that the proposed fusion techniques outperform traditional fusion approaches. We also show that learning the parameters of individual classifiers as part of the fusion architecture can provide better classification performance.

**Index Terms**—Classifier fusion, data reduction, fault diagnosis, hidden Markov models (HMMs),  $k$ -nearest neighbor (KNN), multiway partial least squares (MPLS), parameter optimization, principal components analysis (PCA), probabilistic neural network (PNN), support vector machines (SVMs).

## NOMENCLATURE

FDD	Fault detection and diagnosis.
CRAMAS	ComputeR-Aided Multi-Analysis System.
MPLS	Multiway partial least squares.
SVM	Support vector machine.

Manuscript received November 11, 2007; revised June 27, 2008. First published September 19, 2008; current version published February 9, 2009. This work first appeared in its original form in the Proceedings of the IEEE Automatic Test Conference, Baltimore, MD, September 2007, pp. 260–269. This work was supported by the Toyota Technical Center, University of Connecticut, Storrs, under Contract AG030699-02. The Associate Editor coordinating the review process for this paper was S. Demidenko.

K. Choi, A. Kodali, and K. R. Pattipati are with the Department of the Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269 USA (e-mail: kihoon@engr.uconn.edu; anuradha@engr.uconn.edu; krishna@engr.uconn.edu).

S. Singh was with the Department of the Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269 USA. He is now with the General Motors India Science Laboratory, Bangalore 560 066, India (e-mail: satnam.singh@gm.com).

J. W. Sheppard is with the Department of Computer Science, The Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: jsheppa2@jhu.edu).

S. M. Namburu, D. V. Prokhorov, and L. Qiao are with the Toyota Technical Center, Ann Arbor, MI 48105 USA (e-mail: setumadhavi.namburu@tema.toyota.com; danil.prokhorov; liu.qiao@tema.toyota.com).

S. Chigusa was with the Toyota Technical Center, Ann Arbor, MI 48105 USA. He is now with Isuzu Motors America, Plymouth, MI 48170 USA.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIM.2008.2004340

PNN	Probabilistic neural network.
KNN	$k$ -nearest neighbor.
PCA	Principal components analysis.
LDA	Linear discriminant analysis.
QDA	Quadratic discriminant analysis.
ECU	Electronic control unit.
AdaBoost	Adaptive boosting.
HMM	Hidden Markov model.
FHMM	Factorial HMM.
DMFD	Dynamic multiple fault diagnosis.
ECC	Error-correcting code.

## I. INTRODUCTION

MODERN automobiles are being equipped with increasingly sophisticated electronic systems. Operational problems associated with degraded components, failed sensors, improper installation, poor maintenance, and improperly implemented controls affect the efficiency, safety, and reliability of the vehicles. Failure frequency increases with age and leads to loss of comfort, degraded operational efficiency, and increased wear and tear of vehicle components. An intelligent on-board FDD system can ensure uninterrupted and reliable operation of vehicular systems and aid in vehicle health management.

In our previous research, we considered a data-driven approach to fault diagnosis in an automotive engine system [1]. A data-driven approach to FDD has a close relationship with pattern recognition, wherein one learns classification rules directly from the data, rather than by using analytical models or a knowledge-based approach. The data-driven approach is attractive when one has difficulty in developing an accurate system model. To diagnose the faults of interest in the engine system, we employed several classifiers for fault isolation. These include multivariate statistical techniques exemplified by the PCA [2] and LDA/QDA [4] and pattern classification techniques epitomized by the SVM [5], PNN, and the KNN classifier [4]. We employed MPLS<sup>1</sup> as a data reduction technique to accommodate the processed information (i.e., transformed data) within the limited memory space available in the ECUs for control and diagnosis. AdaBoost [6] was used to improve the classifier performance. We validated and compared the accuracies and memory requirements of various fault-diagnosis schemes. We successfully applied the FDD scheme to an automotive engine and showed that it resulted in significant

<sup>1</sup>This algorithm was used as a data-reduction technique to convert the 3-D matrix data (samples  $\times$  measurements  $\times$  time), which we collected from CRAMAS [20] into a 2-D matrix (samples  $\times$  features).

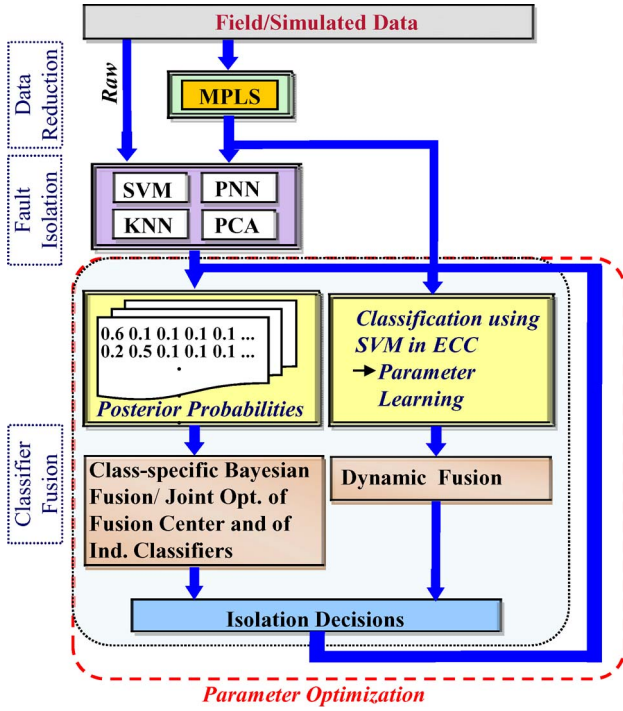


Fig. 1. Block diagram of the proposed FDD fusion.

reductions in computation time and data size without loss of classification accuracy.

It has been well recognized that typical pattern recognition techniques, which focus on finding the best single classifier, have one major drawback [7], [26]. Any complementary discriminatory information that other classifiers may provide is not tapped. Classifier fusion appears to be a natural step when a critical mass of knowledge for a single classifier has been accumulated [8]. The objective of classifier fusion is to reduce the diagnostic errors by combining the results of individual classifiers. The fusion process also allows analysts to use the strengths and weaknesses of each algorithm to reduce the overall diagnostic errors.

In this paper, we propose three novel approaches to classifier fusion: 1) class-specific Bayesian fusion; 2) joint optimization of the fusion center and individual classifiers; and 3) dynamic fusion. These were motivated by our previous work on multi-target tracking and distributed  $M$ -ary hypothesis testing [10], [11] and on DMFD [12], [13], respectively. Our primary focus in this paper is on evaluating how effectively our proposed classifier fusion approaches can reduce the diagnostic errors, as compared with traditional fusion methods and the individual classifiers.

This paper is organized as follows. An overview of diagnostic and fusion process, including our proposed approaches, is provided in Section II. Hardware-in-the-loop simulations (HILSs) and results are discussed in Sections III and IV, respectively. We conclude this paper in Section V with a summary and directions for future research.

## II. DIAGNOSTIC AND FUSION PROCESS OVERVIEW

A block diagram of the diagnostic and classifier fusion schemes in our proposed approach is shown in Fig. 1. The pro-

posed fusion scheme is a three-step process: 1) data reduction to convert 3-D tensors to 2-D matrices; 2) fault isolation via individual classifiers; and 3) classifier fusion with and without parameter optimization as part of the fusion architecture.

### A. Data Reduction

Due to memory-constrained ECUs of automotive systems, intelligent data reduction techniques are needed for on-board implementation of data-driven classification techniques. Traditional methods of data collection and storage capabilities often become untenable because of the increase in the number of observations (measurements) but mainly because of the increase in the number of variables associated with each observation (“dimension of the data”) [15]. Using data reduction techniques, the entire data set is projected onto a low-dimensional space, and the reduced space often gives information about the important structure of the high-dimensional data space. Among widely used dimension-reduction techniques [15], an MPLS-based data-reduction technique was examined in this paper. The advantage of this technique is that it provides good classification accuracy on high-dimensional data sets and is also computationally efficient. The goal of MPLS algorithm is to reduce the dimensionality of the input and output spaces to find latent variables that are most highly correlated, i.e., those that explain not only the variation in  $\underline{X}$ , but also the variations that are most predictive of  $Y$ . The 3-D tensor  $\underline{X}$  is decomposed into one set of score vectors (latent variables)  $\underline{t}$  and two sets of weight vectors  $\underline{w}_j$  and  $\underline{v}_k$  in the second and third dimensions, respectively [3]. The  $Y$  matrix is decomposed into score vectors  $\underline{t}$  and loading vectors  $\underline{q}$ . We have

$$\begin{aligned}\underline{X} &= \sum_{f=1}^U \underline{t}_f (\underline{w}_{jf} \otimes \underline{v}_{kf}) + \underline{E} \\ Y &= \sum_{f=1}^U \underline{t}_f \underline{q}_f^T + B.\end{aligned}\quad (1)$$

Here,  $U$  is the number of factors,  $j$  is the number of measurement index,  $k$  is the time index, the symbol  $\otimes$  is the Kronecker product, and  $\underline{E}$  and  $B$  are residual matrices. The problem of finding  $\underline{t}$ ,  $\underline{w}_j$ ,  $\underline{v}_k$ , and  $\underline{q}^T$  is accomplished by the nonlinear iterative partial least squares (NIPALS) algorithm [18]. These reduced data (arranged as a score matrix) were applied to the classifiers considered in this paper, and we found that it is very efficient in reducing the data size (by a factor of 2000 from the original size). Consequently, the classification algorithm could be embedded in existing ECUs with limited memory [26].

### B. Fault Isolation

The following classification techniques are used as individual classifiers for fault isolation:

- 1) pattern classification techniques: SVM, PNN, and KNN;
- 2) multivariate statistical technique: PCA.

1) *SVMs*: SVMs transform the data to a higher dimensional feature space and find an optimal hyperplane that maximizes the

margin between the classes [16]. There are two distinct features of an SVM. 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 number of data samples, called support vectors, to implement the classifier. A kernel function is used for fitting nonlinearly separable models by transforming the data into a higher dimensional feature space before finding the optimal hyperplane. In this paper, we employ a radial basis function to transform the data into the feature space.

2) *PNNs*: The PNN is a supervised method to estimate the probability distribution function of each class [4]. The likelihood of an input vector being part of a learned category or class is estimated from these functions. *A priori* probabilities and misclassification costs can be used to weight the learned patterns to determine the most likely class for a given input vector. If the *a priori* probability (relative frequency) of the classes is not known, then all the classes can be assumed to be equally likely. In this case, the determination of the class of an input vector is solely based on the closeness to the distribution function of a class.

3) *KNN*: The KNN classifier is a simple nonparametric method for classification. Despite the simplicity of the algorithm, it performs very well and is an important benchmark method [17]. The KNN classifier requires a metric  $d$  and a positive integer  $k$ . Classification of the input vector  $\underline{x}$  is accomplished using the subset of  $k$ -feature vectors that are closest to  $\underline{x}$  with respect to the given metric  $d$ . The input vector  $\underline{x}$  is then assigned to the class that appears most frequently within the  $k$ -subset. Ties can be broken by choosing an odd number for  $k$  (e.g., 1, 3, 5, etc.). Mathematically, this can be viewed as computing the *a posteriori* class probabilities  $P(c_j|\underline{x})$  via

$$P(c_j|\underline{x}) = \frac{k_j}{k} P(c_j) \quad (2)$$

where  $k_j$  is the number of vectors belonging to class  $c_j$  within the subset of  $k$  vectors, and  $P(c_j)$  is the prior probability of class  $c_j$ . A new input vector  $\underline{x}$  is assigned to the class  $c_j$  with the highest *a posteriori* class probability  $P(c_j|\underline{x})$ .

4) *PCA*: PCA transforms correlated variables into a smaller number of uncorrelated variables, called principal components. PCA calculates the covariance matrix of the training data and the corresponding eigenvalues and eigenvectors. The eigenvalues are then sorted, and the vectors (called scores) with the highest values are selected to represent the data in a smaller dimensional space. The number of principal components is determined by cross-validation [18]. The model of PCA is given by

$$X = \sum_{f=1}^U \underline{t}_f \underline{p}_f^T + E \quad (3)$$

where  $E$  is the residual matrix, and  $U$  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 of the objects in the reduced space. The NIPALS algorithm [18] is used to perform the PCA. 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 data, 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 is measured by the Hotelling statistic [19].

### C. Fusion Approaches

In the following, we discuss three novel approaches to fusion: 1) class-specific Bayesian fusion; 2) joint optimization of the fusion center and individual classifiers; and 3) dynamic fusion.

1) *Class-Specific Bayesian Fusion*: The class-specific Bayesian approach to classifier fusion exploits the fact that different classifiers can be good at classifying different fault classes. We let  $\{d_k\}_{k=1}^L$  be the set of models (classifiers) with  $C$  classes, and we let the targets be  $\{t_i\} = \{[t_{i1}, t_{i2}, \dots, t_{iC}]\}$ ,  $t_{ij} \in \{0, 1\}$ ;  $\sum_{j=1}^C t_{ij} = 1$  ( $i = 1, 2, \dots, N$ ). Let  $Z = \{z_i\}_{i=1}^N$  and  $\underline{x}$  be the set of training patterns and the test pattern, respectively. The *posterior* probability of class  $j$  from classifier  $k$  for the test pattern  $\underline{x}$  is denoted by  $e_{jk}(\underline{x})$ . Formally, the fused posterior probability of class  $c_j$  is

$$\begin{aligned} P(c_j|\underline{x}, Z) &= \sum_{k=1}^L P(c_j, d_k(c_j)|\underline{x}, Z) \\ &= \sum_{k=1}^L P(c_j|\underline{x}, Z, d_k(c_j)) P(d_k(c_j)|Z) \\ &= \sum_{k=1}^L e_{jk}(\underline{x}) P(d_k(c_j)|Z). \end{aligned} \quad (4)$$

Note that

$$\begin{aligned} P(d_k(c_j)|Z) &= \frac{P(Z|d_k(c_j)) P(d_k(c_j))}{P(Z)} \\ &= \frac{P(Z|d_k(c_j)) P(d_k(c_j))}{\sum_{l=1}^L P(Z|d_l(c_j)) P(d_l(c_j))}. \end{aligned} \quad (5)$$

We can initialize  $P(d_k(c_j)) = 1/L$ , assuming that all classifiers perform equally on class  $c_j$ , or make it proportional to the overall accuracies of classifiers on class  $c_j$ , i.e.,

$$P(d_k(c_j)) = \frac{a_k(c_j)}{\sum_{l=1}^L a_l(c_j)} \quad (6)$$

where  $a_k(c_j)$  is the accuracy of classifier  $k$  on the  $j$ th class. Evidently, we have

$$\ln P(Z|d_k(c_j)) = \sum_{i=1}^N t_{ij} \ln e_{jk}(z_i). \quad (7)$$

This simply sums the logs of *posterior* probabilities for the target class  $t_{ij} \in \{0, 1\}$  over all the samples corresponding to class  $c_j$ . The larger this number is (closer to zero because the

sum is negative), the better the classifier will be. Therefore, the numerator of  $P(d_k(c_j)|Z)$  can be evaluated in log form as

$$\begin{aligned} n_{kj} &= \ln P(Z|d_k(c_j)) + \ln P(d_k(c_j)) \\ &= \sum_{i=1}^N t_{ij} \ln e_{jk}(z_i) + \ln P(d_k(c_j)). \end{aligned} \quad (8)$$

Then, we have

$$P(d_k(c_j)|Z) = \frac{\exp(n_{kj})}{\sum_{l=1}^L \exp(n_{lj})}. \quad (9)$$

Using this equation, (4) is used to compute the posterior probability of class  $c_j$  for a given test pattern  $\underline{x}$ .

2) *Joint Optimization of the Fusion Center and Individual Classifiers*: The second approach, which is motivated by the fact that the decision rules of the fusion center and individual classifiers are coupled [11], involves the joint optimization of the fusion center and individual classifier decision rules. Given the classifiers, we develop the necessary conditions for the optimal fusion rule, taking into account the costs of decisions. Assume that there are  $L$  classifiers, a fusion center,  $C$  fault classes, and  $N$  training targets  $\{t_i\}$ . The decisions of individual classifiers are denoted by  $\{u_k\}_{k=1}^L$ , while that of the fusion center is  $u_0$ . The classification rule of the  $k$ th classifier is  $u_k \in \{1, 2, \dots, C\} = \gamma_k(\underline{x})$ , and that of the fusion center is  $u_0 \in \{1, 2, \dots, C\} = \gamma_0(u_1, u_2, \dots, u_L)$ . The fusion center must decide which of the classes has occurred based on the evidence from the  $L$  classifiers. The prior probabilities of each class  $c_j$  is  $P_j = N_j/N$ , where  $N_j$  is the number of training patterns of class  $c_j$ . We let  $J(u_0, c_j)$  be the cost of decision  $u_0$  by the committee of classifiers when the true class is  $c_j$ . The problem is to find the joint committee strategy  $\gamma_c = (\gamma_0, \gamma_1, \gamma_2, \dots, \gamma_L)$  such that the expected cost  $E\{J(u_0, c_j)\}$  is a minimum, where  $E$  denotes expectation over  $\{\underline{x}, u_k, 0 \leq k \leq L\}$  and  $\{c_j, 1 \leq j \leq C\}$ . The necessary conditions of optimality for the optimal decision rule is given by [11]

$$\begin{aligned} \gamma_0 : u_0 &= \arg \min_{d_0 \in \{1, 2, \dots, C\}} \\ &\times \sum_{j=1}^C \{P(u_1, u_2, \dots, u_L | c_j) P_j J(d_0, c_j)\}. \end{aligned} \quad (10)$$

The key problem here is obtaining  $P(u_1, u_2, \dots, u_L | c_j)$ . For  $C$  classes, this would involve building a table of  $C^L$  entries, which is clearly intractable in practice. The probability computation is simplified by correlating each classifier with respect to only the best classifier from the training data. The best classifier may change during iterations. Formally, we have

$$\begin{aligned} &P(u_1, u_2, \dots, u_L | c_j) \\ &= \prod_{k=1}^L P(u_k | c_j, u_1, u_2, \dots, u_{L-1}) \\ &\approx P(u_{\text{best}} | c_j) \prod_{\substack{k=1 \\ k \neq \text{best}}}^L P(u_k | c_j, u_{\text{best}}). \end{aligned} \quad (11)$$

We can estimate  $P(u_k | c_j, u_{\text{best}})$  by

$$P(u_k = d_k | c_j, u_{\text{best}} = h) = e_{d_k j h} \quad (12)$$

where  $e_{d_k j h}$  is the proportion of time that classifier  $k$  makes decision  $d_k$  when the true class is  $c_j$ , and the best classifier makes decision  $h$  for  $d_k = 1, 2, \dots, C$ ,  $j = 1, 2, \dots, C$  and  $h = 1, 2, \dots, C$ . Given the classifiers, one can also fuse the classifier decisions, taking into account the costs of decisions [10]. The key here is that the decision rules of the fusion center and individual classifiers are coupled. Given that the fusion rule is fixed, and that the decision strategies of other classifiers are fixed, the  $k$ th classifier makes its decision as follows:

$$\begin{aligned} \gamma_k : u_k &= \arg \min_{d_k \in \{1, 2, \dots, C\}} \\ &\times \sum_{j=1}^C \sum_{u_0=1}^C P(u_0 | \underline{x}, u_k = d_k, c_j) p(\underline{x} | c_j) P_j J(u_0, c_j). \end{aligned} \quad (13)$$

Using Bayes' rule and neglecting the data-dependent term  $p(\underline{x})$ , we have

$$\begin{aligned} u_k &= \arg \min_{d_k \in \{1, 2, \dots, C\}} \\ &\times \sum_{j=1}^C \sum_{u_0=1}^C P(u_0 | \underline{x}, u_k = d_k, c_j) P(c_j | \underline{x}) J(u_0, c_j). \end{aligned} \quad (14)$$

Equation (14) can be simplified as

$$u_k = \arg \min_{d_k \in \{1, 2, \dots, C\}} \sum_{j=1}^C P(c_j | \underline{x}) \hat{J}(u_0, c_j) \quad (15)$$

where

$$\begin{aligned} \hat{J}(u_0, c_j) &= \sum_{u_0=1}^C P(u_0 | \underline{x}, u_k = d_k, c_j) J(u_0, c_j) \\ &\approx \sum_{u_0=1}^C P(u_0 | u_k = d_k, c_j) J(u_0, c_j). \end{aligned} \quad (16)$$

Here,  $P(u_0 | u_k = d_k, c_j)$  is computed from the training data by counting the fraction of times that the fusion center made the decision  $u_0$  when the local classifier  $k$  made the decision  $d_k$  and that the true class is  $c_j$  in the previous iteration of the fusion rule. Furthermore,  $P(c_j | \underline{x})$  is the estimate of the *posterior* probability of class  $c_j$  for a given test pattern  $\underline{x}$ .

3) *Dynamic Fusion of Classifiers*: The dynamic fusion process is based on an optimization framework that computes the most likely fault sequence over time. This problem is formulated as one of finding the maximum *a posteriori* (MAP) configuration in an FHMM. The dynamic fusion problem is a specific formulation of the DMFD problem [12]–[14]. In the DMFD problem, the objective is to isolate multiple faults based on test (classifier) outcomes observed over time. The dynamic fusion problem consists of a set of possible fault states in a system and a set of binary classifier outcomes that are

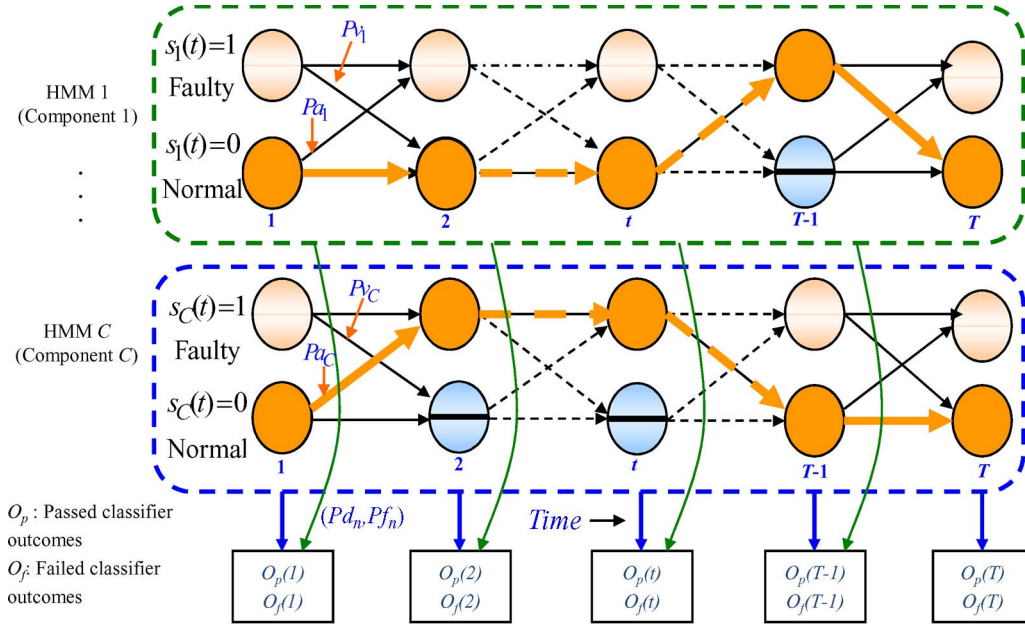


Fig. 2. Dynamic fusion problem viewed as an FHMM.

observed at each time (observation, decision) epoch. Evolution of fault states is independent, i.e., there is no direct coupling among the component states. Each classifier outcome provides information on a subset of the fault states (the entries with ones in the corresponding column of the ECC matrix). Thus, the component states are coupled via classifier outcomes. At each sample epoch, a subset of classifier outcomes is available. Classifiers are imperfect in the sense that the outcomes of some of the classifiers could be missing, and classifiers have missed-detection and false-alarm probabilities associated with them.

Formally, we represent the dynamic fusion problem as  $DF = \{S, \kappa, D, O, ECC, P, A\}$ , where  $S = \{s_1, s_2, \dots, s_C\}$  is a finite set of  $C$  components (failure sources, classes) associated with the system. The state of a component  $m$  is denoted by  $s_m(t)$  at discrete time epoch  $t$ , where  $s_m(t) = 1$  if failure source  $s_m$  is present, and  $s_m(t) = 0$ , otherwise. Here,  $\kappa = \{0, 1, \dots, t, \dots, T\}$  is the set of discretized observation epochs. The status of all component states at epoch  $t$  is denoted by  $\underline{s}(t) = \{s_1(t), s_2(t), \dots, s_C(t)\}$ . We assume that the probability distribution of initial state is known. The observations at each epoch are subsets of binary outcomes of classifiers  $O = \{o_1, o_2, \dots, o_L\}$ , i.e.,  $o_n(t) \in \{pass, fail\} = \{0, 1\}$ . Let the set of passed classifier outcomes be  $O_p$  and that of failed classifiers be  $O_f$ . Fig. 2 shows the DMFD problem viewed as an FHMM. The hidden system fault state of the  $m$ th HMM at discrete time epoch  $t$  is denoted by  $s_m(t)$ . Each fault state  $s_m(t)$  is modeled as a two-state HMM. Here, the true states of the component states and classifiers are hidden.

We also define the ECC matrix  $ECC = [e_{mn}]$  as the diagnostic matrix ( $D$ -matrix) of size  $C$  by  $L$ , which represents the full-order dependency among failure sources and classifiers. For each component state, e.g., for component  $s_i$  at epoch  $t$ ,  $A = \{Pa_m(t), Pv_m(t)\}$  denotes the set of fault appearance probability  $Pa_m(t)$  and fault disappearance prob-

ability  $Pv_m(t)$  defined as  $Pa_m(t) = P(s_m(t) = 1 | s_m(t-1) = 0)$  and  $Pv_m(t) = P(s_m(t) = 0 | s_m(t-1) = 1)$ . For static classes,  $Pa_m(t) = Pv_m(t) = 0$ . When these probabilities are nonzero, the model provides the capability of handling intermittent faults. Here,  $D = \{d_1, d_2, \dots, d_L\}$  is a finite set of  $L$  available binary classifiers, corresponding to the  $L$  columns of the ECC matrix, where the integrity of the system can be ascertained. The set of parameters  $P = \{Pd_n, Pf_n\}$  represents pairs of probabilities of detection and false alarm associated with each classifier  $n$ . Formally,  $Pd_n = P(o_n(t) = 1 | d_n(t) = 1)$ , and  $Pf_n = P(o_n(t) = 1 | d_n(t) = 0)$ . The dynamic fusion problem is one of finding, at each decision epoch  $t$ , the most likely fault state candidates  $\underline{s}(t) \in \{0, 1\}^m$ , i.e., the fault state evolution over time  $S^T = \{\underline{s}(1), \underline{s}(2), \dots, \underline{s}(T)\}$  that best explains the observed classifier outcome sequence  $O^T = \{O(1), O(2), \dots, O(T)\}$ . We formulate this as one of finding the maximum *a posteriori* (MAP) configuration

$$\hat{S}^T = \arg \max_{S^T} P(S^T | O^T). \quad (17)$$

This problem is computationally intractable. A near-optimal polynomial time algorithm based on Lagrangian relaxation and Viterbi decoding was developed. The details of this technique, termed a primal-dual optimization framework, may be found in [12]–[14].

Since the dynamic fusion algorithm is a multiple fault diagnosis algorithm, we may have both missed classifications as well as spurious faults. We used the following metrics from [12] and [27] to evaluate the performance of the dynamic fusion algorithm.

**Missed classification rate (MC):** MC is the percentage of faults not inferred by the algorithm at epoch  $t$ . Let  $\hat{\underline{s}}(t)$  be the inferred states of all failure sources and  $\underline{s}(t)$  be the true states.



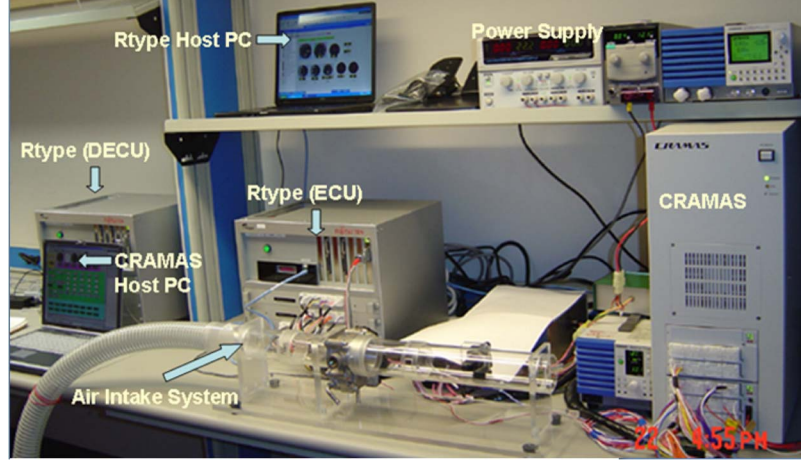


Fig. 3. HILS platform.

Then, MC and the average  $\overline{MC}$  over all epochs are obtained as follows:

$$MC(t) = 1 - \frac{|\hat{s}(t) \cap \underline{s}(t)|}{|\underline{s}(t)|} \quad (18)$$

$$\overline{MC} = \frac{\sum_{t=1}^T MC(t)}{T}. \quad (19)$$

*False (spurious) classification rate (FC):* FC is the percentage of fault states that are falsely inferred by the algorithm as fault states at epoch  $t$ . FC and average  $\overline{FC}$  are computed as

$$FC(t) = \frac{|\hat{s}(t) \cap \neg \underline{s}(t)|}{C - |\underline{s}(t)|} \quad (20)$$

$$\overline{FC} = \frac{\sum_{t=1}^T FC(t)}{T}. \quad (21)$$

Under a single fault assumption,  $\overline{FC}$  can be equal to zero at the cost of a higher  $\overline{MC}$ .

### III. HILSS

The proposed schemes are evaluated on an engine data set. A realistic automotive engine model is employed as an HILS, which simulates the behavior of the given system to achieve real-time diagnostic performance. The HILS platform is shown in Fig. 3, and the engine model is simulated under various fault conditions in a custom-built CRAMAS [20]. CRAMAS, which is a vehicle engine simulator, which is used to develop vehicular ECUs, is a high-speed multipurpose expandable system. The system is implemented as an approximate model of the Toyota Camry four-cylinder engine and consists of five subsystems: 1) air dynamics; 2) fuel dynamics; 3) rotational dynamics; 4) torque generation; and 5) exhaust system.

The faults considered in this paper are representative of real engine faults, and they are injected into the HILS platform by the CRAMAS. The engine system is subject to the following eight faults: 1) air flow sensor fault (misreading the air flow mass); 2) leakage in air intake manifold (a hole in the intake tube); 3) blockage of air filter (blocking the incoming air); 4) throttle angle sensor fault (misreading the throttle angle);

5) air/fuel ratio sensor fault (misreading the A/F ratio); 6) engine speed sensor fault (misreading the engine speed); 7) less fuel injection (delivering less fuel from the fuel pump); and 8) added friction (increasing the friction in cylinders). It also contains the following five sensors: 1) air flow meter reading; 2) air/fuel ratio; 3) vehicle speed; 4) turbine speed; and 5) engine speed. We collected observations of the five sensor readings (measurements) from the CRAMAS hardware-in-the-loop simulation. The model was simulated under a steady-state condition, and the operating condition for the simulation was as follows: 2485 r/min (engine speed); 18° (pedal angle); and 86° (water temperature). For each fault class, we performed simulations for 40 different severity levels (0.5%–20%); each run is sampled at 2000 time points with a 0.005-s sampling interval (10 s of data). The generalized likelihood ratio test (GLRT)-based fault detection tests [25] for this data set were perfect, i.e., each test had unity probability and zero false alarm rate. Thus, data consisted of faulty cases only. Consequently, our focus is on fault isolation only in this paper. We applied our proposed fusion techniques to the data set and compared them with individual classifier performance measures. Here, ten randomized data sets of twofold cross-validation were used to assess the classification performance.

### IV. RESULTS

We implemented and experimented with the three proposed fusion approaches on the CRAMAS engine data. Widely used fusion techniques, namely, the majority voting and the naive Bayes, were also evaluated and compared with our approaches. The diagnostic results, which are measured in terms of missed classification rates and testing times for the eight faults, are shown in Table I. Testing time was computed using Matlab software on a 2.3-GHz Intel Pentium 4 processor with 1 GB of RAM. We expect that the time shown could further be reduced by a factor of at least ten when implemented in the C language. As shown in Table I, we not only achieved lower missed classification rates but also obtained significant data reduction (from 25.6 MB to 12.8 KB)—a factor of 2000. Since data reduction improved classifier performance, the proposed fusion approaches were mainly evaluated on the reduced data set.

TABLE I  
COMPARISON OF INDIVIDUAL AND FUSION CLASSIFICATION ON RAW AND REDUCED DATA

CRAMAS®	Method	Missed Classification Rate $\pm$ Std Dev in % [Training Time $\pm$ Std Dev / Testing Time $\pm$ Std Dev in sec]				
		SVM	KNN ( $k=1$ )	KNN ( $k=3$ )	PNN	PCA
Raw Data (25.6 MB)	Individual Classification	8.76 $\pm$ 2.54 [3.17 $\pm$ 0.19/ 3.21 $\pm$ 0.23]	12.94 $\pm$ 2.23 [20.68 $\pm$ 0.28/ 19.05 $\pm$ 2.41]	15.01 $\pm$ 2.42 [20.73 $\pm$ 0.11/ 19.13 $\pm$ 2.25]	14.83 $\pm$ 2.19 [23.53 $\pm$ 0.11/ 23.40 $\pm$ 3.15]	22.5 $\pm$ 2.32 [9.66 $\pm$ 0.35/ 8.32 $\pm$ 1.73]
Reduced Data via MPLS (12.8 KB)	Individual Classification	8.19 $\pm$ 2.53 [0.02 $\pm$ 0.04/ 0.02 $\pm$ 0.01]	12.75 $\pm$ 2.07 [0.05 $\pm$ 0.01/ 0.05 $\pm$ 0.02]	14.13 $\pm$ 2.54 [0.05 $\pm$ 0.02/ 0.05 $\pm$ 0.02]	14.06 $\pm$ 2.13 [0.35 $\pm$ 0.04/ 0.25 $\pm$ 0.02]	21.06 $\pm$ 3.73 [0.95 $\pm$ 0.03/ 0.77 $\pm$ 0.06]
	Majority Voting	12.06 $\pm$ 1.89 [NA / 0.03 $\pm$ 0.01]				
	Naïve Bayes	11.81 $\pm$ 1.96 [0.01 $\pm$ 0.01 / 0.19 $\pm$ 0.02]				
	Bayesian Fusion	8.62 $\pm$ 2.85 [0.20 $\pm$ 0.04 / 0.22 $\pm$ 0.01]				
	Joint Optimization	8.39 $\pm$ 2.02 [4.22 $\pm$ 0.38 / 3.38 $\pm$ 0.22]				
	Dynamic Fusion	9.9 $\pm$ 1.9 [8635.69 $\pm$ 2949.88 / 0.12 $\pm$ 0.01] *False (spurious) classification rate: 2.49 $\pm$ 1.84 [8635.69 $\pm$ 2949.88 / 0.12 $\pm$ 0.01]				
	Bayesian Fusion with Classifier Selection	6.25 $\pm$ 2.29 [0.18 $\pm$ 0.01 / 0.19 $\pm$ 0.01]				
	Joint Optimization with Majority Voting	5.87 $\pm$ 2.04 [3.80 $\pm$ 0.4 / 3.4 $\pm$ 0.24]				
	Dynamic Fusion with Parameter Optimization	4.5 $\pm$ 1.6 [8635.69 $\pm$ 2949.88 / 0.12 $\pm$ 0.01] *False (spurious) classification rate: 4.8 $\pm$ 0.6 [8635.69 $\pm$ 2949.88 / 0.12 $\pm$ 0.01]				

Although the majority voting, the naive Bayes, the proposed Bayesian fusion, and the dynamic fusion (without parameter optimization) approaches for this problem helped marginal classifiers (i.e., PNN, KNNs, and PCA) in reducing the missed classification rates, they were unable to outperform the best single classifier, namely, SVM. For the CRAMAS data, the SVM algorithm resulted in 59 and 44 support vectors for the raw and reduced data, respectively. However, the proposed Bayesian and dynamic fusion approaches are comparable with the single best classifier and outperformed the majority voting and naive Bayes fusion approaches. Initially, the results from the joint optimization approach and class-specific Bayesian fusion were quite similar to that from the single best classifier (virtually tied statistically to SVM based on McNemar's test [25]). However, we were able to reduce the missed classification rates using the joint optimization approach on marginal classifiers [PNN, KNN ( $k=3$ ), and PCA] and then applying majority voting on the fused decision and those from SVM and KNN ( $k=1$ ). Specifically, the *posterior* probabilities from PNN, KNN ( $k=3$ ), and PCA were fed to the joint optimization algorithm, and then, the SVM result, KNN ( $k=1$ ) result, and the fused decision from the joint optimization algorithm were used as inputs to a majority voting fusion algorithm. Use of majority voting alone provided poor classification results; this implies that the joint optimization approach was definitely a contributor

to the reduced error. We believe that this is because the joint optimization of the fusion center and individual classifiers increases the diversity of the classifier outputs, which is a vital requirement for reducing the missed classification rates using fusion. The Bayesian fusion result was also improved by optimally selecting classifiers. For the result shown in Table I, three classifiers, SVM, and KNNs ( $k=1, 3$ ) were selected and fused for the Bayesian fusion approach. The criterion for optimally selecting classifiers in class-specific Bayesian fusion and joint optimization was based on both cross-validation and coarse optimization. For the dynamic fusion approach, we ran the fusion algorithm with a sampling interval of 0.5 s to suppress the noise in the data. Thus, we used a down sampling rate of 100 and obtained 20 time epochs for the dynamic fusion process. The results in Table I were obtained using 15 SVM classifiers, which are represented by the columns of the ECC matrix. The ECC matrix was generated using the Hamming code generation method [21]. The dynamic fusion achieved lower missed classification rate results as compared with any single classifier results. We experimented with two different approaches for  $P_d$  and  $P_f$  in the dynamic fusion process. The first approach used  $P_d$  and  $P_f$  learned from the *training data of individual classifiers*, while coarse optimization was applied to learn  $P_d$  and  $P_f$ , and the optimal parameters were  $P_d = 0.5-0.6$  and  $P_f = 0-0.02$  when they are part

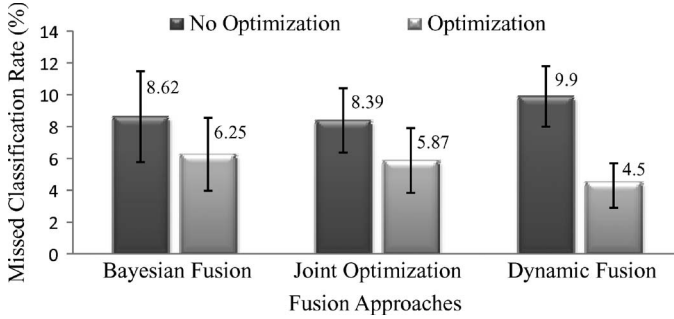


Fig. 4. Comparison of fusion approaches with no parameter optimization and with optimization.

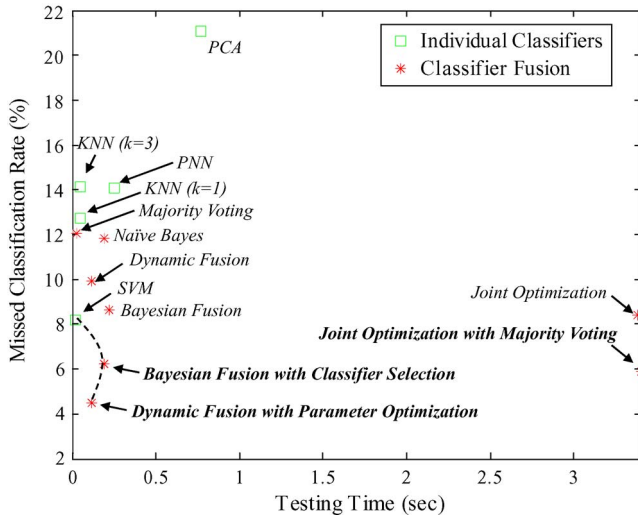


Fig. 5. Comparison of individual and fusion classification on reduced data.

of the dynamic fusion. We found that the dynamic fusion approach with parameter optimization significantly reduces the diagnostic error by 45.1% (as compared with SVM, which is the single best classifier on the reduced data).

Fig. 4 shows that dynamic fusion with parameter optimization provided the most significant improvement and was also the best in classification accuracy. Note that the training time for the dynamic fusion method depends on how many classifiers (ECC columns) are used. The higher the number of classifiers is, the larger the training time will be.

Fig. 5 provides a plot of missed classification rate versus testing time per pattern for all the approaches considered in this paper, as well as the Pareto efficiency (dashed line) [22]. The Pareto efficiency curve indicates all of the potentially optimal approaches to the problem in that analysts can trade off missed classification rate and testing time in an informed way, rather than considering the full range of fusion approaches. Since our primary focus is on diagnostic error, the lower values of this measure are preferred over the higher values. The figure clearly shows that our dynamic fusion with parameter optimization is superior to all other approaches. However, if one's focus is on training time or on the need for a faster response, Bayesian fusion with classifier selection would be adequate, while joint optimization with majority voting would also be a good choice, if the analyst is concerned about training time.

## V. CONCLUSION

In this paper, we have proposed and developed three new approaches to classifier fusion. In addition to individual classifiers, such as the SVM, PNN, KNN, and PCA for fault isolation, *posterior* probabilities from these classifiers were fused by class-specific Bayesian fusion, joint optimization of the fusion center and individual classifiers, and dynamic fusion. All these approaches were validated on the CRAMAS engine data (raw and reduced data sets). Although, in terms of missed classification rate, the results of the proposed approaches before applying optimization were quite close to the single best classifier performance, they are generally better than the majority voting and the naive Bayes-based fusion approaches. It confirms again that fusing marginal classifiers can increase the diagnostic performance substantially. We showed that classifier selection in the context of class-specific Bayesian fusion, majority voting among the best classifiers and fused marginal classifiers, and parameter optimization in dynamic fusion significantly reduced the overall diagnostic errors. The key empirical result here is that one needs to learn parameters as part of the fusion architecture (not standalone) to obtain the best classification performance from a team of classifiers. This is consistent with the finding in distributed detection theory that the individual sensors (classifiers in our case) operate at different operating points when they are part of a team (fusion in our case) than when they operate alone [23].

Our future research will focus on evaluating the proposed classifier fusion techniques on various real-world data sets, such as automotive field data, UCI repository, etc. Furthermore, we plan to explore and compare other fusion techniques that can be applied to fault diagnosis in automotive systems. For the dynamic fusion research, we also plan to explore relaxation of the independence assumption and solve the dynamic fusion problem when faults are dependent. Coupled HMMs offer a promising mathematical framework for the solution of this problem [24]. We also plan to extend the joint optimization approach to correlated faults via the Bayesian network/influence diagram framework.

## REFERENCES

- [1] K. Choi, J. Luo, K. R. Pattipati, S. M. Namburu, L. Qiao, and S. Chigusa, "Data reduction techniques for intelligent fault diagnosis in automotive systems," in *Proc. IEEE Autotestcon*, Anaheim, CA, Sep. 2006.
- [2] P. Nomikos and K. F. MacGregor, "Monitoring batch processes using multiway principal component analysis," *AIChE J.*, vol. 40, no. 8, pp. 1361–1375, 1994.
- [3] B. Rasmus, "Multiway calibration. Multilinear PLS," *J. Chemom.*, vol. 10, no. 1, pp. 259–266, 1996.
- [4] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley-Interscience, 2001.
- [5] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, Mar. 2002.
- [6] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.
- [7] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 226–239, Mar. 1998.
- [8] L. I. Kuncheva, *Combining Pattern Classifiers*. Hoboken, NJ: Wiley, 2004.



- [9] D. Ruta and B. Gabrys, "An overview of classifier fusion methods," *J. Comput. Inf. Syst.*, vol. 7, no. 1, pp. 1–10, 2000.
- [10] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction*. Hoboken, NJ: Wiley, 2001.
- [11] Z. B. Tang, K. R. Pattipati, and D. L. Kleinman, "A distributed  $M$ -ary hypothesis testing problem with correlated observations," *IEEE Trans. Autom. Control*, vol. 37, no. 7, pp. 1042–1046, Jul. 1992.
- [12] S. Singh, K. Choi, A. Kodali, K. R. Pattipati, S. M. Namburu, S. Chigusa, D. V. Prokhorov, and L. Qiao, "Dynamic multiple fault diagnosis: Mathematical formulations and solution techniques," *IEEE Trans. SMC: Part A*, Jan. 2008, to be published.
- [13] S. Singh, S. Ruan, K. Choi, K. R. Pattipati, P. Willett, S. M. Namburu, S. Chigusa, D. V. Prokhorov, and L. Qiao, "An optimization-based method for dynamic multiple fault diagnosis problem," in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, Mar. 2007, pp. 1–13.
- [14] S. Singh, K. Choi, A. Kodali, K. R. Pattipati, S. M. Namburu, S. Chigusa, D. V. Prokhorov, and L. Qiao, "Dynamic fusion of classifiers for fault diagnosis," in *Proc. IEEE SMC Conf.*, Montreal, QC, Canada, Oct. 2007, pp. 2467–2472.
- [15] I. K. Fodor and C. Kamath, "Dimension reduction techniques and the classification of bent double galaxies," *Comput. Stat. Data Anal. J.*, vol. 41, no. 1, pp. 91–122, Nov. 2002.
- [16] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, Jun. 1998.
- [17] J. K. Shah, *Sequential k-NN Pattern Recognition for Usable Speech Classification—A Revised Report*. Philadelphia, PA: Speech Process. Lab, Temple Univ., 2004.
- [18] S. Wold, P. Geladi, K. Esbensen, and J. Ohman, "Principal component analysis," *Chemometr. Intell. Lab. Syst.*, vol. 2, no. 1–3, pp. 37–52, 1987.
- [19] P. Nomikos, "Detection and diagnosis of abnormal batch operations based on multi-way principal component analysis," *ISA Trans.*, vol. 35, no. 3, pp. 259–266, 1996.
- [20] F. Takeshi, Y. Norio, Y. Takeshi, and K. Naoya, "Development of PC-based HIL simulator CRAMAS 2001," *Fujitsu Ten Tech. J.*, vol. 19, no. 1, pp. 12–21, 2001.
- [21] R. W. Hamming, "Error detecting and error correcting codes," *Bell Syst. Tech. J.*, vol. 26, no. 2, pp. 147–150, Apr. 1950.
- [22] M. J. Osborne and A. Rubenstein, *A Course in Game Theory*. Cambridge, MA: MIT Press, 1994.
- [23] A. Pete, K. R. Pattipati, and D. L. Kleinman, "Optimal team and individual decision rules in uncertain dichotomous situations," *Public Choice*, vol. 75, no. 3, pp. 205–230, 1993.
- [24] L. Xie and Z.-Q. Liu, "A coupled HMM approach to video-realistic speech animation," *Pattern Recognit.*, vol. 40, no. 8, pp. 2325–2340, Aug. 2007.
- [25] K. Choi, M. Azam, J. Luo, S. M. Namburu, and K. R. Pattipati, "Fault diagnosis in HVAC chillers using data-driven techniques," *IEEE Instrum. Meas. Mag.*, vol. 8, no. 3, pp. 24–32, Aug. 2005.
- [26] W. Donat, K. Choi, W. An, S. Singh, and K. R. Pattipati, "Data visualization, data reduction and classifier fusion for intelligent fault detection and diagnosis in gas turbine engines," *Trans. ASME J. Eng. Gas Turbines Power*, vol. 130, no. 4, p. 041 602, Jul. 2008.
- [27] F. Yu, F. Tu, H. Tu, and K. R. Pattipati, "A Lagrangian relaxation algorithm for finding the MAP configuration in QMR-DT," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 37, no. 5, pp. 746–757, Sep. 2007.



**Satnam Singh** received the B.E. degree from the Indian Institute of Technology (IIT), Roorkee, India, in 1996, the M.S. degree in electrical engineering from the University of Wyoming, Laramie, in 2003, and the Ph.D. degree in electrical engineering from the University of Connecticut, Storrs, in 2007.

He is currently a Senior Researcher with the General Motors India Science Laboratory, Bangalore, India. Prior to his graduate studies, he was a Design Engineer with Engineers India Limited, New Delhi, India, for four years. His research interests

include data mining, graphical models, pattern recognition, fault detection and diagnosis, and optimization theory.

Dr. Singh is a Reviewer for the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS and the IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT.



**Anuradha Kodali** received the B.E. degree in electronics and communications engineering from Andhra University, Visakhapatnam, India, in 2006. She is currently working toward the Ph.D. degree in electrical and computer engineering with the University of Connecticut, Storrs.

Her research interests include data mining, pattern recognition, fault detection and diagnosis, and optimization theory.



**Krishna R. Pattipati** (S'77–M'80–SM'91–F'95) received the B.Tech. degree in electrical engineering (with highest honors) from the Indian Institute of Technology, Kharagpur, India, in 1975 and the M.S. and Ph.D. degrees in systems engineering from the University of Connecticut (UConn), Storrs, in 1977 and 1980, respectively.

From 1980 to 1986, he was with Alphatech, Inc., Burlington, MA. Since 1986, he has been with UConn, where he is currently a Professor of electrical and computer engineering with the Department

of the Electrical and Computer Engineering. He has served as a consultant to Alphatech, Inc. and IBM Research and Development and is a cofounder of Qualtech Systems, Inc., which is a small business enterprise specializing in intelligent diagnostic software tools. He has authored over 370 published articles primarily focused on the application of systems theory and optimization (continuous and discrete) techniques to large-scale systems. His current research interests are adaptive organizations for dynamic and uncertain environments, multiuser detection in wireless communications, signal processing and diagnosis techniques for complex system monitoring, and multiobject tracking.

Dr. Pattipati served as the Editor-in-Chief of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: PART B—CYBERNETICS from 1998 to 2001, as the Vice President for Technical Activities of the IEEE Systems, Man, and Cybernetics Society during 1998–1999, and as the Vice President for Conferences and Meetings of the IEEE Systems, Man, and Cybernetics Society during 2000–2001. He was selected by the IEEE Systems, Man, and Cybernetics Society as the Outstanding Young Engineer of 1984 and received the Centennial Key to the Future Award. He was a coreipient of the Andrew P. Sage Award for the Best SMC Transactions Paper for 1999, the Barry Carlton Award for the Best AES Transactions Paper for 2000, the 2002 and 2008 NASA Space Act Awards for "A comprehensive toolset for model-based health monitoring and diagnosis," the 2003 AAUP Research Excellence Award, and the 2005 School of Engineering Teaching Excellence Award from the University of Connecticut. He also received the Best Technical Paper Awards at the 1985, 1990, 1994, 2002, 2004, and 2005 IEEE AutoTest Conferences and at the 1997 and 2004 Command and Control Conferences.



**Kihoon Choi** (M'07) received the M.S. degree in electrical and computer engineering from the University of Connecticut, Storrs, in 2005, where he is currently working toward the Ph.D. degree.

His research interests are fault diagnosis of complex systems, system real-time fault diagnosis, analysis and optimization of large-scale systems, and data mining.

Mr. Choi won the Best Technical Paper Award at the 2004 IEEE AutoTest Conference.



**John W. Sheppard** (M'86–SM'97–F'07) received the B.S. degree in computer science from Southern Methodist University, Dallas, TX, and the M.S. and Ph.D. degrees in computer science from Johns Hopkins University, Baltimore, MD.

He was recently named the Right Now Technologies Distinguished Professor in computer science at Montana State University, Bozeman. He holds an appointment as an Associate Research Professor in computer science at The Johns Hopkins University. He started his career as a research computer scientist for ARINC and attained the rank of Fellow. He has over 100 publications in artificial intelligence and diagnostics, including an authored book and an edited book. His research interests include algorithms for diagnostic and prognostic reasoning, machine learning and data mining in temporal systems, and reinforcement learning.

Dr. Sheppard currently serves as the Vice Chair of the IEEE Standards Coordinating Committee 20 (SCC20) on Test and Diagnosis for Electronic Systems. He is a former Chair of the Diagnostic and Maintenance Control subcommittee of SCC20, in which he is currently the Secretary, and the Official Liaison of the Computer Society Standards Activities Board to SCC20.



**Danil V. Prokhorov** (S'95–M'97–SM'02) received the M.S. degree (with honors) from the State Academy of Aerospace Engineering [formerly Leningradsky Institut Aviatsionnovo Priborostroyeniya (LIAP)], St. Petersburg, Russia, in 1992 and the Ph.D. degree in electrical engineering from Texas Tech University, Lubbock, in 1997.

He was with the Ford Research Laboratory, Dearborn, MI, from 1997 to 2005. He has been engaged in application-driven studies of neural networks and training algorithms. He is currently a

Research Manager with the Toyota Technical Center, Ann Arbor, MI. He has been a reviewer for numerous journals and conferences, a Program Committee member for many conferences, and a panel expert for the National Science Foundation every year since 1995. He has authored 80 technical publications, including several patents. His research interest is machine learning algorithms and their applications to decision making under uncertainty.

Dr. Prokhorov received the International Neural Network Society Young Investigator Award in 1999. He was the International Joint Conference on Neural Networks (IJCNN) 2005 General Chair and the IJCNN 2001 Program Chair.



**Setu Madhavi Namburu** (M'05) received the B.Tech. degree from Jawaharlal Nehru Technological University, Hyderabad, India, in 2002 and the M.S. degree from the University of Connecticut, Storrs, in 2006, both in electrical engineering.

She is currently a Research Engineer with the Toyota Technical Center, Ann Arbor, MI. Her research interests include fault diagnostics and prognostics of engineering systems, data mining, text mining, knowledge discovery, and artificial intelligence algorithms.



**Shunsuke Chigusa** (M'06) received the M.S. degree in electrical engineering from Osaka University, Osaka, Japan.

He was a Senior Principal Research Scientist with the Technical Research Department, Toyota Motors Engineering and Manufacturing North America, Inc., Ann Arbor, MI. He started his automotive career as an Autonomous Vehicle Control System Scientist and later as an Intelligent Diagnosis Scientist with Toyota Motors Corporation, Japan. He is currently with Isuzu Motors America, Plymouth, MI.



**Liu Qiao** (M'02–SM'08) received the B.Eng. degree from the Beijing University of Technology, Beijing, China, and the M.S. and Ph.D. degrees in electrical engineering from Tohoku University, Sendai, Japan.

He is the General Manager and the Chief Technologist of the Toyota Technical Center, Ann Arbor, MI. Switching from a university faculty position, he started his automotive career as an advanced automotive control system expert, advanced technology manager, eBusiness manager, and research manager. He successfully led the Canadian hybrid vehicle

project and its market introduction.