

Anomaly Detection in Nuclear Power Plants via Symbolic Dynamic Filtering

Xin Jin, *Student Member, IEEE*, Yin Guo, Soumik Sarkar, *Student Member, IEEE*, Asok Ray, *Fellow, IEEE*, and Robert M. Edwards

Abstract—Tools of sensor-data-driven anomaly detection facilitate condition monitoring of dynamical systems especially if the physics-based models are either inadequate or unavailable. Along this line, symbolic dynamic filtering (SDF) has been reported in literature as a real-time data-driven tool of feature extraction for pattern identification from sensor time series. However, an inherent difficulty for a data-driven tool is that the quality of detection may drastically suffer in the event of sensor degradation. This paper proposes an anomaly detection algorithm for condition monitoring of nuclear power plants, where symbolic feature extraction and the associated pattern classification are optimized by appropriate partitioning of (possibly noise-contaminated) sensor time series. In this process, the system anomaly signatures are identified by masking the sensor degradation signatures. The proposed anomaly detection methodology is validated on the International Reactor Innovative & Secure (IRIS) simulator of nuclear power plants, and its performance is evaluated by comparison with that of principal component analysis (PCA).

Index Terms—Data-driven fault detection, feature extraction, pattern classification, symbolic dynamics, time series analysis.

I. INTRODUCTION

CONDITION monitoring and timely detection of incipient faults are critical for operational safety and performance enhancement of nuclear power plants. There are various sources of anomalous behavior (i.e., deviation from the nominal condition) in plant operations, which could be the consequence of a fault in a single component or simultaneous faults in multiple components. Often it is difficult for the plant operator to detect the anomaly and locate the associated anomalous component(s), especially if the anomaly is small and evolve slowly. Upon occurrence of an anomalous event and subsequent pervasion of its effects, the operator could be overwhelmed by the sheer volume of information, generated simultaneously from various sources.

Manuscript received August 17, 2010; revised October 01, 2010; accepted October 10, 2010. Date of publication December 03, 2010; date of current version February 09, 2011. This work was supported in part by the U.S. Department of Energy under NERI-C Grant DE-FG07-07ID14895 and by NASA under Cooperative Agreement NNX07AK49A. Any opinions, findings and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

The authors are with the Department of Mechanical and Nuclear Engineering, Pennsylvania State University, University Park, PA 16802 USA (e-mail: xuj103@psu.edu; yxg141@psu.edu; szs200@psu.edu; axr2@psu.edu; rmenuc@engr.psu.edu).

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/TNS.2010.2088138

Therefore, it would be beneficial to develop an automated condition monitoring system to assist the plant operator to detect the anomalies and isolate the anomalous components.

Condition monitoring algorithms are primarily divided into two different categories, namely, model-based and data-driven. Both model-based and data-driven techniques have been reported in literature for condition monitoring of nuclear power plants. Examples of model-based condition monitoring can be found in [1], [2]. Among data-driven tools, neural networks (NN) and principal component analysis (PCA)-based tools [3]–[6] are most popular.

Although model-based techniques have their advantages in terms of physical interpretation, their reliability and computational efficiency for condition monitoring often decrease as the system complexity increases. On the other hand, data-driven techniques are expected to remain largely reliable and computationally efficient in spite of increased system complexity if the goal is to monitor the input-output information from an ensemble of (appropriately calibrated) sensors while considering the entire system as a black-box. However, unless the ensemble of acquired information is systematically handled, data-driven techniques may become computationally intensive and the performance of condition monitoring may deteriorate due to sensor degradation. Furthermore, data-driven techniques would require high volume of training data (e.g., component malfunction data in the present context).

A problem with handling time series data is its volume and the associated computational complexity; therefore, the available information must be appropriately compressed via transformation of high-dimensional data sets into low-dimensional features with minimal loss of class separability. In our previous work [7], we reported Symbolic Dynamic Filtering (SDF) for detection of anomalies (i.e., deviations from the nominal condition) in dynamical systems. The SDF method is shown to be useful for feature extraction from time series and has been experimentally validated for real-time execution in different applications (e.g., electronic circuits [8] and fatigue damage monitoring in polycrystalline alloys [9]).

A major challenge in any sensor-data-driven detection tool is to identify the actual anomaly in the system in the presence of sensor degradation (e.g., drift and noise) without succumbing to a large number of false alarms or missed detections. The situation becomes even more critical if the control system uses observations from the degraded sensors as feed-back signals and thereby distorts the control inputs. Traditionally, redundant sensors along with methods based on analytic redundancy have been used for sensor anomaly identification [10], [11].

The present paper proposes an alternative approach to the above problem, where different class labels are assigned to data sets that are generated from different plant health conditions. The same class labels are assigned to similar health conditions and the associated sensor data are subjected to different noise variance at respective sensor degradation levels. To this end, data-driven anomaly detection is posed as a multi-class pattern classification problem, where the tasks of class assignment and the SDF-based feature extraction are optimized in a supervised manner to enhance the classification performance. A major step in SDF-based feature extraction is partitioning of time-series data to generate symbol blocks that are subsequently converted to feature vectors by use of the probabilistic finite state automata (PFSA) concept [7], [12]. In this context, the major contributions of this paper are delineated below.

- Application of symbolic dynamic filtering (SDF) and probabilistic finite state automata (PFSA) concepts for anomaly detection and condition monitoring in nuclear power plants;
- Optimization of data partitioning in an SDF-based feature-extraction setting;
- Construction of a data-driven anomaly detection method at the component level and evaluation of its robustness to sensor noise;
- Evaluation of the proposed SDF-based anomaly detection algorithm relative to principal component analysis (PCA) that is a commonly used tool for anomaly detection.

The paper is organized into seven sections including the present one. Section II describes the International Reactor Innovative & Secure (IRIS) test-bed of nuclear power plants on which the problem of plant condition monitoring has been formulated and validated. Section III poses data-driven anomaly detection as a multi-class pattern classification problem. Section IV presents the background and formulation of SDF-based feature extraction for anomaly detection, and describes the partitioning optimization methodology. Section V explains how anomaly detection algorithms are constructed. Section VI compares the results of the proposed method with those of a PCA-based method on the IRIS test-bed. Section VII summarizes the paper and makes major conclusions along with recommendations for future research.

II. DESCRIPTION OF THE IRIS TEST-BED

The International Reactor Innovative & Secure (IRIS) nuclear power plants is based on the design of a next-generation nuclear reactor. It is a modular pressurized water reactor (PWR) with an integral configuration of all primary system components. Fig. 1 shows the layout of the primary side of the IRIS system that is offered in configurations of single or multiple modules, each having a power rating of 1000MWt (about 335MWe) [13]. The nominal reactor core inlet and outlet temperatures are 557.6°F (292°C) and 626°F (330°C), respectively. The pressurizer, eight steam generators, and the control rod mechanism are integrated into the pressure vessel with the reactor core. There is no huge pipe used to connect these components. This design avoids the large loss of coolant accident (LOCA) [13]. The entire control rod mechanism is

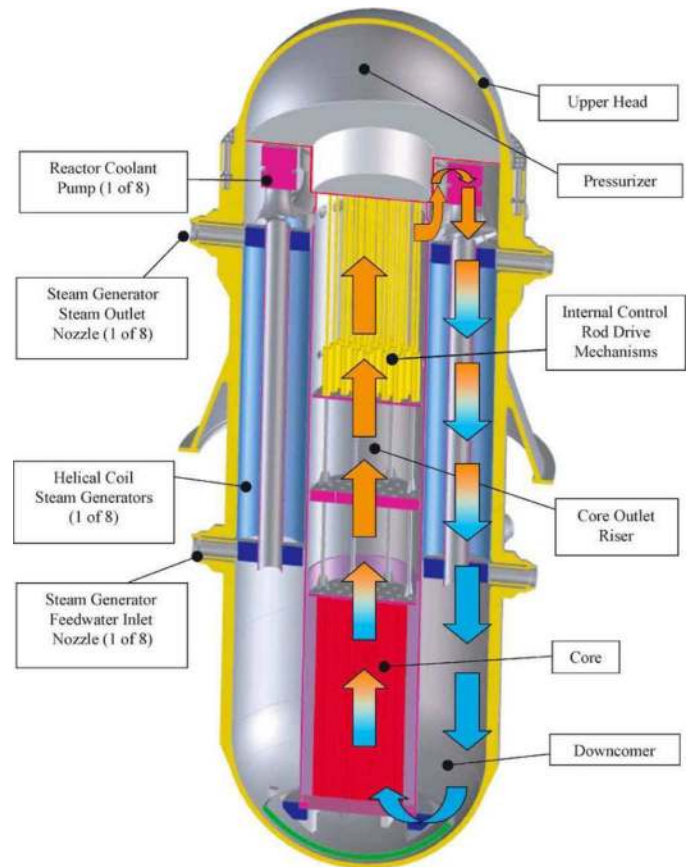


Fig. 1. Layout of the primary side of the IRIS system [13].

mounted inside the pressure vessel to avoid the problem of penetrating the the pressure vessel head by the control rod.

In order to develop an automated anomaly detection algorithm, it is necessary to “teach” or “train” a pattern classifier so that it captures the process dynamics and produces the desired output for the given inputs. For systems where this cannot be done experimentally, high-fidelity simulators should be constructed for developing anomaly detection algorithms. The IRIS simulator, used in this paper, has been built by researchers at North Carolina State University [14]–[16]. In the normal operating range, reactor power is determined through a point kinetics model, with rod position controlled through a user-specified program. Feedwater control is based upon steam demand, similar to control strategies employed in B&W once-through steam generators [17]. In the very low power range, a constant heat input is assumed to simulate decay heat, and the average moderator temperature is allowed to float.

The test-bed is built using FORTRAN programming language. This FORTRAN model includes a reactor core model, a helical coil steam generator (HCSG) model. The test-bed is implemented on a Quad Core 2.83 GHz CPU 8 GB RAM Workstation in the laboratory of Penn State. The IRIS simulator is operated in the *integrated control* mode through built-in PID controllers, which operates all three subsystems (i.e., turbine, feedwater flow and control rods) to combine the rapid response of a reactor-following system with the stability of a turbine-following system [17].

The IRIS test-bed is capable of simulating both normal conditions at different operational modes (e.g., normal power maneuvers, reactor start-up, and turbine loading) [15]) and a variety of anomalous scenarios that include:

- Actuator anomalies, e.g., feedwater pump trip, malfunctions of reactor coolant pump and control rod mechanism;
- Sensor failures, e.g., malfunctions of temperature, pressure, and flow-rate sensors;
- Internal anomalies, e.g., uncertainties in the fuel temperature coefficient of reactivity, coolant heat capacity, and feedwater heat capacity.

In the IRIS test-bed, sensor degradations are realized as injected noise and disturbances. Depending on the location and modality of a sensor, there could be several different degradation levels. For example, the degradation levels in a pressure sensor have different characteristics from those of a temperature sensor. Furthermore, depending on the location and operating environment, even sensors of the same modality could have different degradation characteristics. In general, sensor degradation is categorized as the following [18]:

- Constant bias and drift (i.e., slowly-varying bias);
- Change in sensor response time due to aging; and
- Change in the variance of sensor noise (e.g., due to large external electromagnetic radiation from electric motors).

Amongst the above sensor degradation types, only sensor degradation due to changes in the noise variance are investigated in this paper. The rationale is that the sensors are assumed to be periodically tested and calibrated; hence, sensor degradation due to aging, bias, and drift is much less likely.

A case study has been presented in this paper to validate the anomaly detection methodology, in which the reactor coolant pump (RCP) is chosen to be the location of a component-level degradation and the primary coolant temperature (T_{HL}) sensor in the hot leg piping is chosen for anomaly detection in the RCP.

Since the plant controller receives feedback signals from the T_{HL} sensor, any degradation in this sensor could pervade through the plant, which will potentially affect the outputs of the remaining components due to the inherent electro-mechanical and control feedback. Component-level anomaly detection under different sensor noise variance is posed as a multi-class classification problem in the sequel.

III. DATA-DRIVEN ANOMALY DETECTION POSED AS A MULTI-CLASS PATTERN CLASSIFICATION PROBLEM

Component-level anomaly detection in nuclear power plants involves identification of the anomaly type and location & quantification of the anomaly level. Although the model configuration in the IRIS test-bed can be easily extended to simultaneous anomalies in multiple components, this paper deals with a single component, namely, the reactor coolant pump (RCP), where the task is to detect an anomaly and identify its level for (possibly unknown) sensor noise variance. The RCP overspeed percentage (ψ_{RCP}) is chosen as a health parameter. Table I shows the approximate ranges of ψ_{RCP} under different anomaly levels. Here, the low anomaly level indicate very minimal overspeed in RCP and hence also includes the absolute nominal

TABLE I
ANOMALY TYPES IN REACTOR COOLANT PUMP (RCP)

| Anomaly Type | RCP Overspeed ψ_{RCP} (%) |
|--------------|--------------------------------|
| Low | 0.00 to 0.30 |
| Medium | 0.30 to 0.60 |
| High | 0.60 to 0.90 |

TABLE II
NOISE LEVEL AND SNR IN THE T_{HL} SENSOR

| Noise Level | Signal to Noise Ratio Range |
|-------------|-----------------------------|
| Level 1 | 30 dB to 15 dB |
| Level 2 | 15 dB to 10 dB |
| Level 3 | 10 dB to 7 dB |

health condition ($\psi_{RCP} = 0$). Similarly, depending on the standard deviation of the noise in the hot-leg coolant temperature sensor T_{HL} , three sensor noise levels are selected. Table II shows the ranges of signal-to-noise ratio (SNR) for different noise levels of the hot-leg coolant temperature sensor (T_{HL}). The SNR in decibels (dB) is defined as

$$\begin{aligned} SNR_{dB} &= 10 \log_{10} \left(\frac{P_{signal}}{P_{noise}} \right) \\ &= 20 \log_{10} \left(\frac{A_{signal}}{A_{noise}} \right) \end{aligned} \quad (1)$$

where P_{signal} is the power of the signal, P_{noise} is the power of the noise, A_{signal} is the root mean square (RMS) amplitude of the signal, and A_{noise} is the RMS value of the noise. Since the T_{HL} responses are close to sine waves (as shown in Fig. 2), $A_{signal} \simeq A_{pp}/2\sqrt{2}$, where A_{pp} is the peak to peak amplitude and $A_{noise} = \sigma_{T_{HL}}$.

In the above context, $(3 \times 3) = 9$ classes of data sets are chosen to define each class by an RCP anomaly level and a noise level of the T_{HL} sensor. One hundred simulation runs are performed on the IRIS system for each class to generate time series data, among which 50 samples were chosen for training and the remaining samples for testing. The parameters, RCP overspeed percentage ψ_{RCP} and standard deviation of T_{HL} are chosen randomly from independent uniform distribution such that all of the parameter values are within the prescribed ranges given in Tables I and II. Fig. 3 plots the generated samples in the two-dimensional parameter space, where different classes of samples are shown in different colors and are marked within the class number. Note that the standard deviation ($\sigma_{T_{HL}}$) in Fig. 3 shows values of actual standard deviation, not as percents.

Often component degradations in nuclear plants occur on a slow-time scale, i.e., the anomaly gradually evolves over a relatively long time span. Sudden failures in plant components are disastrous and may occur only if a component accumulates sufficient amount of anomaly over a prolonged period and reaches the critical point, which justify early detection of an anomaly for enhancement of plant operational safety. However, component anomaly at the early stages of a malfunction is usually very

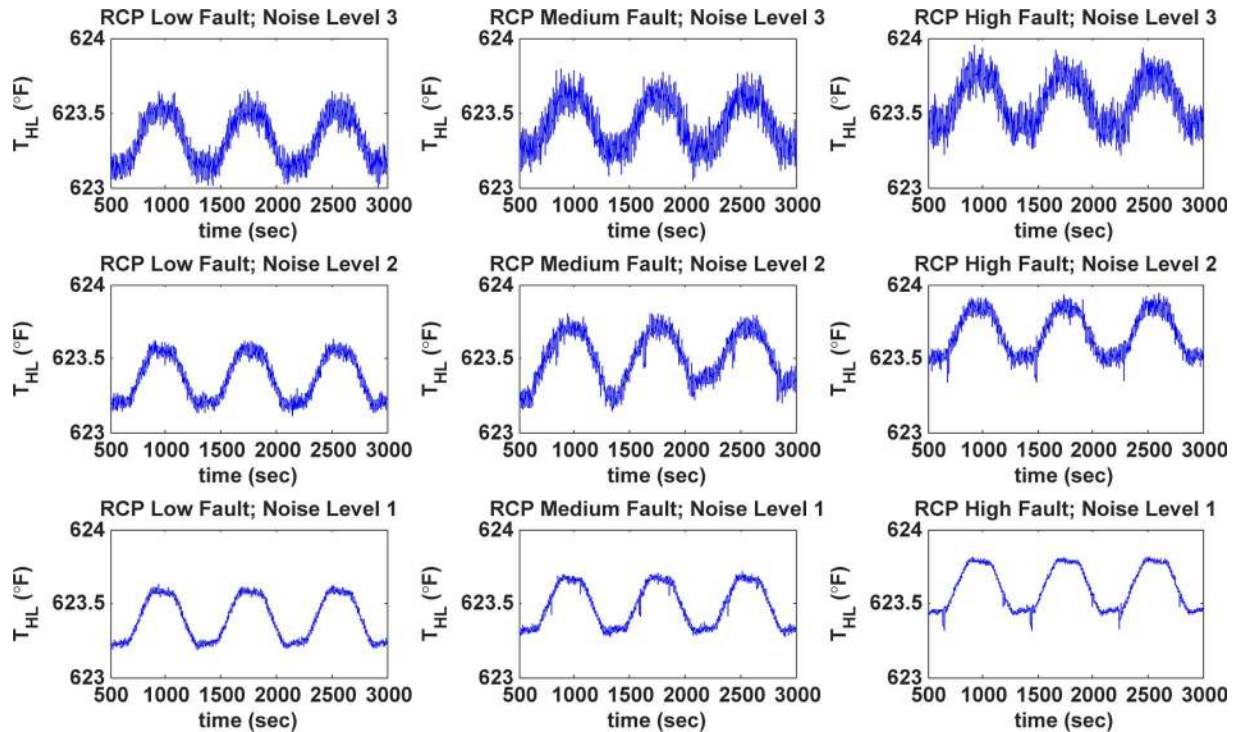


Fig. 2. Representative time series data for RCP anomaly and T_{HL} degradation conditions.

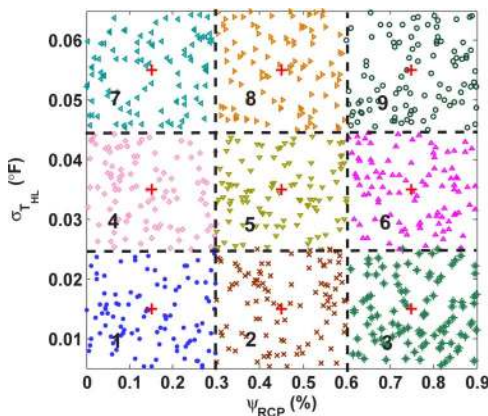


Fig. 3. Original class labels for data collection.

small and is difficult to detect, especially under steady-state operations. One way to “amplify” the anomaly signature is to perturb the plant by (quasi-)periodically perturbing the turbine load such that the plant is in a transient state for a period that is sufficiently long for anomaly detection and insignificant from the perspectives of plant operation.

For each sample point in the parameter space, a time series is collected for T_{HL} sensor under persistent excitation of turbine load inputs that have truncated triangular profiles with the mean value of 99% of nominal output power, fluctuations within $\pm 1\%$ and frequency of 0.00125 Hz (period $T = 800$ sec). In other words, the turbine load is fluctuating between nominal power (i.e., 335 MW) and 98% of nominal power (i.e., 328.3 MW). Fig. 4 shows the profile of turbine load and a typical example of turbine output power as a result of fluctuation in turbine load.

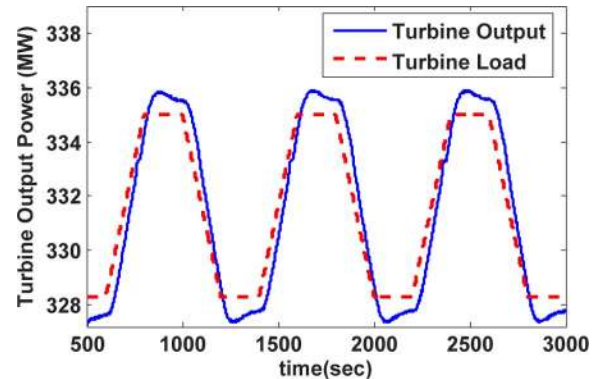


Fig. 4. Profile of turbine load and turbine output response.

For each experiment, the sampling frequency for data collection is 1 Hz (i.e., the inter-sample time of 1 sec) and the length of the perturbation time window is 2,400 seconds, which generate 2,400 data points. The total perturbation period for each experiment is 3,000 seconds. In this paper, to eliminate the possible transients, the data for the first 600 seconds have not been used so that quasi-stationarity is achieved. Fig. 2 shows representative examples of T_{HL} time series data from each of nine classes. Reduction of the perturbation period needs to be investigated for in-plant operations, which is a topic of future research.

With the objective of building a data-driven diagnostic algorithm that is robust to sensor noise (within an allowable range), a data class is defined to be only dependent on the RCP over-speed parameters. Thus, the 9 original classes are reduced to 3 classes as shown in Fig. 5. This is the final class assignment for the data set, where each class has $(50 \times 3) = 150$ training samples and $(50 \times 3) = 150$ test samples. Thus, the problem of

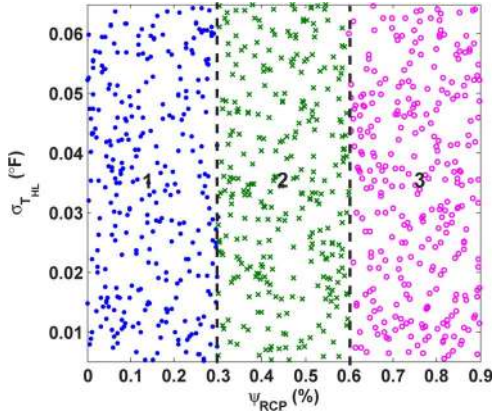
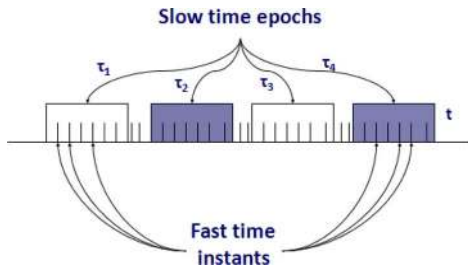


Fig. 5. Revised class assignment for anomaly detection.


 Fig. 6. Pictorial view of the two time scales: (i) *Slow time scale* of anomaly evolution and (ii) *Fast time instants* of data acquisition.

component level anomaly detection is formulated in presence of sensor noise as a multi-class classification problem; in the present scenario, number of classes is 3.

IV. SYMBOLIC DYNAMIC FILTERING (SDF)-BASED FEATURE EXTRACTION

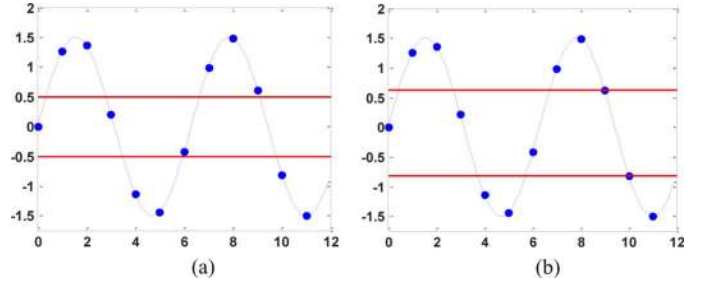
This section briefly reviews the concepts of symbolic dynamic filtering (SDF) [7], [8] for feature extraction, followed by optimization of partitioning.

A. Review of Symbolic Dynamic Filtering (SDF)

Symbolic feature extraction from time series data is posed as a two-time-scale problem. The *fast scale* is related to the response time of the process dynamics. Over the span of data acquisition, dynamic behavior of the system is assumed to remain invariant, i.e., the process is quasi-stationary at the fast scale. On the other hand, the *slow scale* is related to the time span over which non-stationary evolution of the system dynamics may occur. It is expected that the features extracted from the fast-scale data will depict statistical non-stationarity between two different slow-scale epochs if the underlying system has undergone a change. The concept of two time scales is illustrated in Fig. 6.

The method of extracting features from stationary time series data is comprised of the following steps:

- Sensor time series data, generated from a physical system or its dynamical model, are collected at a slow-scale epoch and let it be denoted as \mathbf{q} . A compact (i.e., closed and


 Fig. 7. Illustration of classical partition schemes (alphabet size $|\Sigma| = 3$).

bounded) region $\Omega \in \mathbb{R}^n$, where $n \in \mathbb{N}$, within which the stationary time series is circumscribed, is identified. Let the space of time series data sets be represented as $\mathcal{Q} \subseteq \mathbb{R}^{n \times N}$, where $N \in \mathbb{N}$ is sufficiently large for convergence of statistical properties within a specified threshold. While n represents the dimensionality of the time-series, N is the number of data points in the time series. Then, $\{\mathbf{q}\} \in \mathcal{Q}$ denotes a time series at the slow-scale epoch of data collection.

- Encoding of Ω is accomplished by introducing a partition $\mathbb{B} \triangleq \{B_0, \dots, B_{(|\Sigma|-1)}\}$ consisting of $|\Sigma|$ mutually exclusive (i.e., $B_j \cap B_k = \emptyset \forall j \neq k$), and exhaustive (i.e., $\bigcup_{j=0}^{|\Sigma|-1} B_j = \Omega$) cells, where each cell is labeled by symbols $\sigma_j \in \Sigma$ and $\Sigma = \{\sigma_0, \dots, \sigma_{|\Sigma|-1}\}$ is called the alphabet. This process of coarse graining can be executed by uniform, maximum entropy [19], or any other scheme of partitioning. Then, the time series data points that visit the cell B_j are denoted as $\sigma_j \forall j = 0, 1, \dots, |\Sigma| - 1$. This step enables transformation of the time series data $\{\mathbf{q}\}$ to a symbol sequence $\{\mathbf{s}\}$, consisting of the symbols σ_j in the alphabet Σ .
- A probabilistic finite state machine (PFSA) of the Markov dynamical process, consisting of r states, is constructed and the symbol sequence $\{\mathbf{s}\}$ is run through the PFSA. The (irreducible stochastic) $(r \times r)$ state transition matrix $\Pi \equiv [\pi_{jk}]$ is obtained at the slow-scale epoch. [Note: π_{jk} is the transition probability from state j to state k of the PFSA with $\pi_{jk} \geq 0$ and $\sum_k \pi_{jk} = 1$.] The $(1 \times r)$ state probability vector $\mathbf{p} = [p_1 \ \dots \ p_r]$, which is the left eigenvector corresponding to the unique unity eigenvalue of Π , could be treated as the extracted feature vector that is a low-dimensional representation of the dynamical system at the slow-scale epoch.

For anomaly detection using SDF, the nominal time-series is partitioned by one of the classical schemes (e.g., uniform partitioning (UP) or maximum entropy partitioning (MEP)) [7], [8], [20], as illustrated in Fig. 7(a) and 7(b), respectively. Then, using the steps described earlier, a low-dimensional feature vector \mathbf{p} is constructed from each data set.

Upon completion of feature extraction, a classifier is used to distinguish the set of feature vectors of class Cl_i , i.e., $\{\mathbf{p}^{Cl_i}\}$, from the feature vectors of the other classes, where $i = 1, 2, \dots, K$, and K is total number of classes. There are plenty of choices available for design of both parametric and

non-parametric classifiers in literature [21], [22]. Among the parametric type of classifiers, one of the most common techniques is to consider up to two orders of statistics in the feature space. In other words, the mean feature is calculated for every class along with the variance of the feature space distribution in the training set. Then, a test feature vector is classified by using the Mahalanobis distance [23] or the Bhattacharya distance [24] of the test vector from the mean feature vector of each class. However, these methods lack in computational efficiency if the feature space distribution cannot be described by second order statistics (i.e., non-Gaussian in nature). In the present context, Gaussian feature space distribution cannot be ensured due to the nonlinear nature of the partitioning feature extraction technique. Therefore, a non-parametric classifier, such as the k -NN classifier may a better candidate for this study [21], [25]; however, in general, any other suitable classifier, such as the support vector machines (SVM) or the Gaussian mixture models (GMM) [22] may also be used.

B. Optimization of Partitioning

Properties and variations of transformation from the symbol space to the feature space have been extensively studied in the Mathematics, Computer Science and especially Data Mining literature. Apparently, similar efforts have not been expended to investigate partitioning of time series data to optimally generate symbol blocks for pattern classification and anomaly detection. Symbolic false nearest neighbor partitioning (SFNNP) [26] optimizes a generating partition by avoiding topological degeneracy. However, a major shortcoming of SFNNP is that it may become extremely computationally intensive if the dimension of the phase space of the underlying dynamical system is large. Furthermore, if the time series data become noise-corrupted, the states of SFNNP rapidly grow in number and thus the partitioning may erroneously require a large number of symbols to capture pertinent information on the system dynamics [20]. This shortcoming could be largely alleviated by wavelet space partitioning (WSP) that is particularly effective for noisy data for large-dimensional dynamical systems [7]; maximum entropy partitioning (MEP) was used by Rajagopalan and Ray [8] to generate symbol blocks from time series data by WSP. Although WSP is significantly computationally faster than SFNNP and is suitable for real-time applications, WSP also has several shortcomings such as, requirements of good understanding of signal characteristics for selection of the wavelet basis, identification of appropriate scales, and lossy & non-unique conversion of the two-dimensional scale-shift domain into a single dimension. Subbu and Ray [20] introduced Hilbert-transform-based analytic signal space partitioning (ASSP) as an alternative to WSP, and Sarkar *et al.* [27] generalized ASSP for symbolic analysis of noisy signals.

The data partitioning techniques, described above, primarily provide a symbolic representation of the underlying dynamical system under a given quasi-stationary condition, rather than capturing the data-evolution characteristics as a consequence of an anomaly. Next we present a partitioning optimization

methodology that endeavors to overcome this shortcoming to make SDF a robust data-driven feature extraction tool for pattern classification and anomaly detection.

Optimization criteria that are reported for feature extraction in multi-class classification problems are broadly classified as:

- 1) Filtering method that makes use of the information content feedback (e.g., Fisher criteria, statistical dependence, and information-theoretic measures) as optimization criteria for feature extraction.
- 2) Wrapper method that includes the classifier inside the optimization loop to maximize the predictive accuracy (e.g., classification rate using statistical re-sampling or cross-validation [22]).

The wrapper method is adopted here to minimize the classification error on the training set. The rationale for this choice is the non-binary nature of the problem at hand and the possible non-Gaussian distribution of training samples in the feature space.

In a multi-class problem, ideally one should jointly minimize all the off-diagonal elements of the confusion matrix, while maximizing the diagonal elements. However, in that case, the dimension of the objective space blows up with increase in the number of classes, which is obviously impractical. Therefore, two cost functionals may be defined on the confusion matrix by using another penalty weighting matrix, elements of which denote the relative penalty values for different confusions in the classification process. Formally, let there be Cl_1, \dots, Cl_n classes of labeled time-series data given as the training set. A partitioning \mathbb{B} is employed to extract features from each sample and a k -NN classifier \mathbb{K} is used to classify them. After the classification process, the confusion matrix [28] \mathbf{C} is obtained, where the value of its element c_{ij} denotes the frequency of data from class Cl_i being classified as data from class Cl_j . Let \mathbf{W} be the weighting matrix, where the value of its element w_{ij} denotes the penalty incurred by the classification process for classifying a data set from class Cl_i as a data set from class Cl_j . With these definitions, the cost due to expected classification error, $Cost_E$ is defined as:

$$Cost_E \triangleq \frac{1}{N_s} \left(\sum_i \sum_j w_{ij} c_{ij} \right) \quad (2)$$

where N_s is the total number training samples including all classes. The outer sum in the above equation sums the total penalty values for misclassifying each class Cl_i . Thus $Cost_E$ is related to the expected classification error. Although, in the current formulation, the total penalty values are equally weighted for all classes, that can be changed based on prior knowledge about the data and the user requirements.

It is implicitly assumed in many supervised learning algorithms that the training data set is a statistically similar representation of the whole data set. However, this assumption may not be very accurate in practice. A natural solution to this problem is to choose a feature extractor that minimizes the worst-case

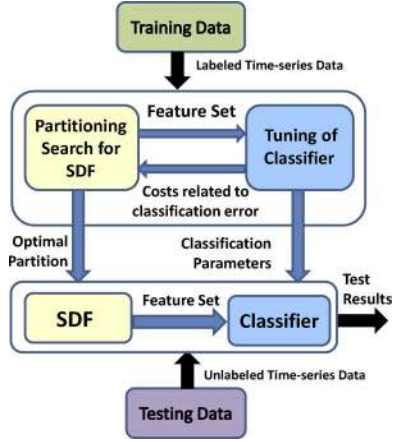


Fig. 8. General framework for optimization of feature extraction.

classification error [29] as well. In the present setting, that cost due to worst-case classification error, $Cost_W$ can be defined as:

$$Cost_W \triangleq \max_i \left(\frac{1}{N_i} \sum_j w_{ij} c_{ij} \right) \quad (3)$$

where N_i is the number of training samples in class Cl_i . With such construction, the dimension of the objective space is not a function of the number of classes, which makes it convenient for classification with large number classes. As described earlier, classification needs to be performed on the training data set to calculate the costs during optimization of the feature extractor, i.e., the partitioning.

Fig. 8 depicts the general outline of the classification process. Labeled time series data from the training set are partitioned and the generated low-dimensional feature vectors (via symbolization and PFSA construction) are fed to the classifier.

After classification, the two training error costs defined as above are computed and fed back to the feature extraction block. During classification, the classifier may be tuned to obtain better classification rates. For example, for k -NN classifiers [25], choice of neighbor size or the distance metric can be tuned. The partitioning is updated to reduce the cost based on the feedback. The iteration is continued until the set of optimal partitioning in a multi-objective scenario and the correspondingly tuned classifier are obtained. Choice of the optimal partitioning is done using the Neyman-Pearson criterion [30] as described later. After the choice is made, the optimal partitioning and the tuned classifier are used to classify the test data set. Although this is the general framework that is being proposed for the optimization methodology, tuning of the classifier has not been performed in this paper as the main focus here is to choose the optimal partitioning to minimize the classification error related cost.

Similar to the classical partitioning cases, the Euclidean distance is chosen as the metric. For partitioning optimization, at first, the number of cells $|\Sigma|$ of the partitioning \mathbb{B} is chosen to be 4 in this case. For computation purpose, a suitably fine grid

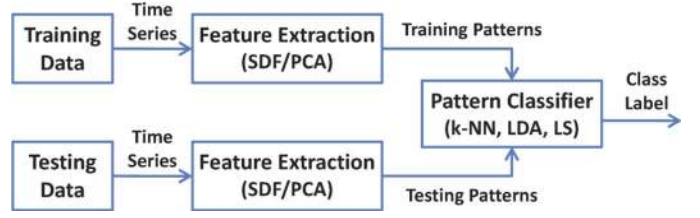


Fig. 9. Flow chart for data-driven anomaly detection.

size depending on the data characteristics is then assumed. It should be clear that each of the grid boundaries denote a possible position of a partitioning cell boundary. In this paper, the data space region Ω is divided into 40 grid cells, i.e., 39 grid boundaries excluding the boundaries of Ω and the alphabet size is selected to be $|\Sigma| = 4$. That is, there are 4 partitioning cells and 3 partitioning boundaries to choose. Hence, the number of elements (i.e., 4-dimensional partitioning vectors) in the space \mathcal{P} of all possible partitioning is: ${}^{39}C_3 = 9139$. Since the cardinality of \mathcal{P} is computationally tractable in this example, a direct search-based Pareto optimization procedure is followed in this paper. If the partitioning space \mathcal{P} happens to be significantly large such that usage of a direct search approach becomes infeasible for evaluation of all possible partitioning, then other searching schemes (e.g., those based on genetic algorithms [31]) should be applied.

By searching the partition space \mathcal{P} , the positions of its elements (i.e., the partitioning vectors) are located in the (two-dimensional) $Cost_E$ - $Cost_W$ objective space. The resulting Pareto front is generated by identifying the non-dominated points [30] in the objective space. In the present case, a non-dominated point (i.e., a partitioning vector) is such that no other partitioning has lower values of both $Cost_E$ and $Cost_W$. Finally, the Neyman-Pearson criterion [30] is applied to choose the optimal partitioning \mathbb{B}^* for minimization of the cost $Cost_E$, while not allowing the cost $Cost_W$ to exceed the allowable worst-case classification error. In other words, the optimal partitioning \mathbb{B}^* is the solution to the following constrained optimization problem:

$$\mathbb{B}^* = \arg \min_{\mathbb{B}} Cost_E(\mathbb{B}) \text{ such that } Cost_W(\mathbb{B}) \leq \epsilon \quad (4)$$

where ϵ is the allowable worst-case classification error that is denoted by the (positive scalar) threshold determined by the user.

V. ANOMALY DETECTION ALGORITHM

This section describes how the anomaly detection algorithm is formulated based on the concept of symbolic dynamic filtering (SDF). For comparative evaluation, another anomaly detection algorithm is constructed based on the principal component analysis (PCA), also known as proper orthogonal decomposition or *Karhunen-Loève transformation* [21], [22], [32], which is a linear feature extraction technique and computes the largest eigenvectors of the covariance matrix.

A flowchart of the anomaly detection algorithms is shown in Fig. 9 and the following steps summarize the procedure.

A. Symbolic Dynamic Filtering for Anomaly Detection

The following steps summarize the procedure of SDF for anomaly detection.

- **Time series data acquisition** on the fast scale from sensors and/or analytical measurements (i.e., outputs of a physics-based or an empirical model). Data sets of different plant and sensor conditions are collected at slow time epochs, and divided into partitioning, training and testing sets.
- **Partitioning the time series data.** Each segment of the partitioning is assigned a particular symbol from the symbol alphabet set Σ . This step enables transformation of time series data from the continuous domain to the symbolic domain [33]. The partitioning is generated from the partitioning data and remains fixed for subsequent training and testing stage.
- **Construction of probabilistic finite state automata** from each data set with alphabet size $|\Sigma|$ and window length D . Choices of $|\Sigma|$ and D depend on the characteristics of the data. In this paper, $D = 1$ and $|\Sigma| = 4$ are taken.
- **Generation of state probability vectors.** The state probability vectors $\{\mathbf{p}\}$ are recursively computed as an approximation of the natural invariant density of the dynamical system. For each data set, a state probabilistic vector is generated and serve as the feature vector in the subsequent pattern classification tasks.
- **Identification of the class information** The low-dimensional state probability vectors are input to pattern classifiers to identify the class labels. The classifier is trained in a supervised manner with the training patterns. The k -nearest neighbors (k -NN) algorithm [21], [22] has been used in this paper as the pattern classifier due to its simplicity.

B. Principal Component Analysis for Anomaly Detection

The time series data sets are organized into an $(M \times N)$ -dimensional data matrix, where M is the number of data sets, N is the length of each (one dimensional) time-series data set. Let \mathbf{X} be the centered version of the original $(M \times N)$ data matrix. For $M < N$, it is numerically efficient [22] to analyze the $(M \times M)$ matrix $\mathbf{S} \triangleq (1/M)\mathbf{X}\mathbf{X}^T$ that has the same nonzero eigenvalues as the $(N \times N)$ computed covariance matrix $(1/M)\mathbf{X}^T\mathbf{X}$.

Let $\{\mathbf{v}_i\}$ be the eigenvectors of the matrix \mathbf{S} corresponding to the eigenvalues $\{\lambda_i\}$ that are arranged in the decreasing order of magnitude. The m largest (real positive) eigenvalues, where $m \leq M$, are selected such that $\sum_{i=1}^m \lambda_i > \eta \sum_{i=1}^M \lambda_i$, where η is a real positive fraction close to 1. The corresponding normalized eigenvectors \mathbf{u}_i in the original data space are calculated as follows [22].

$$\mathbf{u}_i = \frac{1}{\sqrt{(M\lambda_i)}} \mathbf{X}^T \mathbf{v}_i. \quad (5)$$

These m eigenvectors obtained from (5) are then grouped to form an $(N \times m)$ projection matrix \mathbf{W}_{PCA} , where $\mathbf{W}_{PCA} \triangleq [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m]$. Thus, a new low-dimensional data matrix is generated as the $M \times m$ matrix $\mathbf{Y}_{train} \triangleq \mathbf{X}_{train} \mathbf{W}_{PCA}$, where

the training data set \mathbf{X}_{train} has the same structure as \mathbf{X} . Each of the m rows of \mathbf{Y}_{train} is considered as a feature vector.

Following the above procedure, we obtain the projection matrix \mathbf{W}_{PCA} and the low-dimensional data matrix \mathbf{Y}_{train} from the training data matrix \mathbf{X}_{train} . Similarly, in the testing stage, the low-dimensional data matrix is obtained as $\mathbf{Y}_{test} = \mathbf{X}_{test} \mathbf{W}_{PCA}$.

In summary, to apply PCA to high-dimensional data, we first evaluate $\mathbf{X}\mathbf{X}^T$ and then find its eigenvectors and eigenvalues and then compute the eigenvectors in the original data space [22].

So far we have obtained a low-dimensional representation of the original data matrix by using PCA as the feature extraction tool; the next step is to use a pattern classifier to determine the class labels of the testing data sets. For performance comparison of SDF with PCA, the k -NN algorithm is also used as the pattern classifier for PCA. Rows of the $M \times m$ matrices \mathbf{Y}_{train} and \mathbf{Y}_{test} serve as the training patterns and testing patterns, respectively. Other common pattern classifiers, namely, linear discriminant analysis (LDA) and least squares algorithm (LS) [21], [22] have also been used for comparative evaluation, as reported later in the paper.

VI. RESULTS AND DISCUSSION

This section presents pertinent results for the case study of anomaly detection in the reactor coolant pump (RCP) for comparative evaluation of the optimal partitioning-based SDF tool with those based on classical methods of partitioning as well as PCA.

At the beginning of the optimization procedure, a weighting matrix \mathbf{W} needs to be defined to calculate the cost functionals $Cost_E$ and $Cost_W$ from the confusion matrix for the training data set. In this case study, \mathbf{W} is defined according to the adjacency properties of classes in the parameter space, i.e., $w_{ii} = 0 \forall i \in \{1, 2, 3\}$, i.e., there is no penalty for correct classification. The weights are selected as: $w_{ij} = |i - j|$, $\forall i \in \{1, 2, 3\}$, i.e., given that a data sample originally from Cl_i is classified as a member of Cl_j , the penalty incurred by the classification process increases with increase in the separation of Cl_i and Cl_j in the parameter space. Then, it follows that:

$$\mathbf{W} = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix}.$$

The data space region Ω is divided into 40 grid cells, i.e., 39 grid boundaries excluding the boundaries of Ω . Each partitioning in the space \mathcal{P} is evaluated by calculating $Cost_E$ and $Cost_W$ to identify the optimal partitioning. Fig. 10 shows a relevant region of the (two-dimensional) $Cost_E$ - $Cost_W$ objective space, where the elements of the space \mathcal{P} are located. The Pareto front is also generated from this evaluation. The threshold ϵ , i.e., the maximum allowable $Cost_W$ is taken to be 0.041 in this case and the optimal partitioning (OptP) is chosen by the Neyman-Pearson criterion as discussed earlier. Thus, the user has the option of choosing a classification operating point with selected values of $Cost_E$ and $Cost_W$. The Pareto front [30]

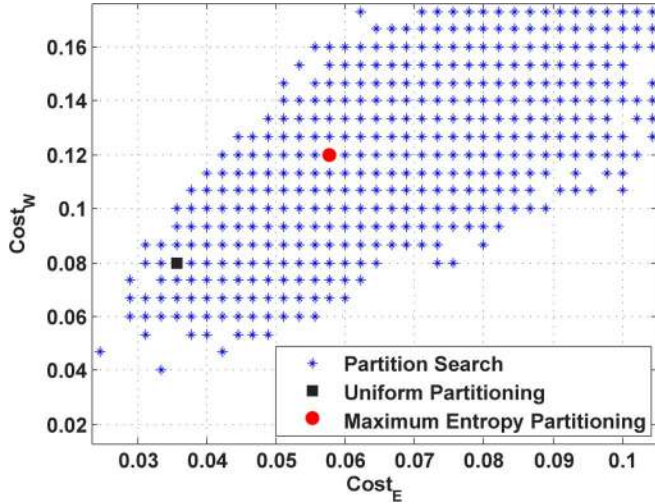


Fig. 10. Two dimensional objective space for partitioning optimization.

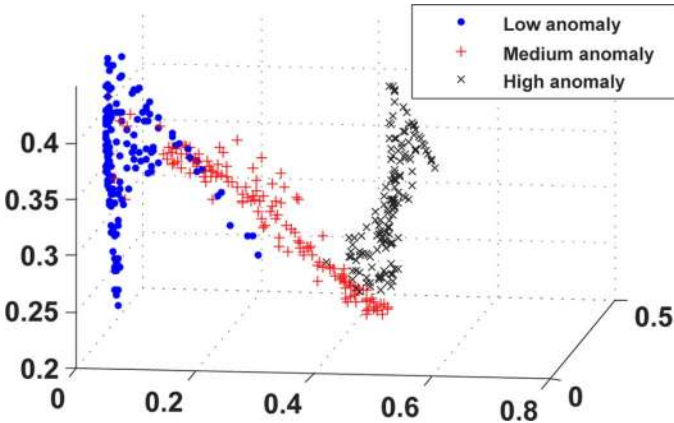


Fig. 11. Feature space of the training set using optimal partitioning.

is generated after the threshold ϵ is chosen. Locations of the classical partitioning (i.e., uniform (UP) and maximum entropy (MEP)) are also plotted along with the elements of \mathcal{P} in the figures for comparative evaluation.

For SDF analysis, the alphabet size is taken to be $|\Sigma| = 4$ and the depth for constructing PFSA is taken to be $D = 1$. Features are classified by a k -NN classifier using the Euclidean distance metric. Fig. 11 shows locations of the training features in the three-dimensional plot using first three linearly independent elements of the feature vectors obtained by using the chosen optimal partitioning, OptP. Note, only $(|\Sigma| - 1)$ out of its $|\Sigma|$ elements of a feature vector are linearly independent, because a training feature vector, \mathbf{p} is also a probability vector, i.e., the sum of its elements is constrained to be equal to 1. Levels of RCP anomalies are represented in different symbols: blue \bullet indicates low anomaly, red $+$ indicates medium anomaly, and black \times indicates high anomaly. The class separability is retained by the feature extraction (partitioning) process even after compressing a time series (with 2,400 data points) into 3 numbers.

For comparison purpose, classical partitioning schemes, such as, Uniform Partitioning (UP) and Maximum Entropy Partitioning (MEP) are also used with the same alphabet size, $|\Sigma| =$

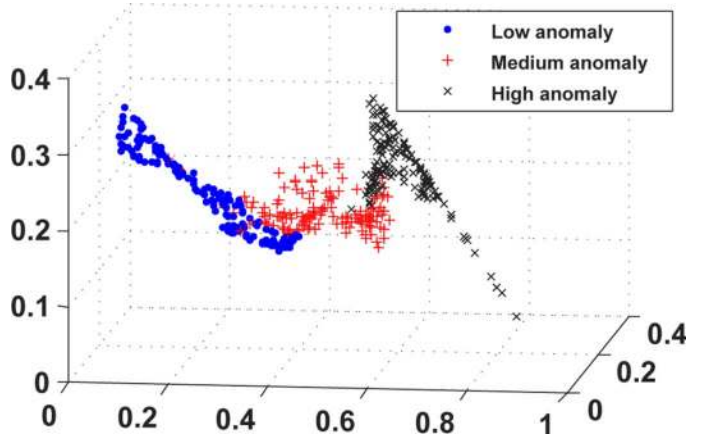


Fig. 12. Feature space of training set—uniform partitioning (UP).

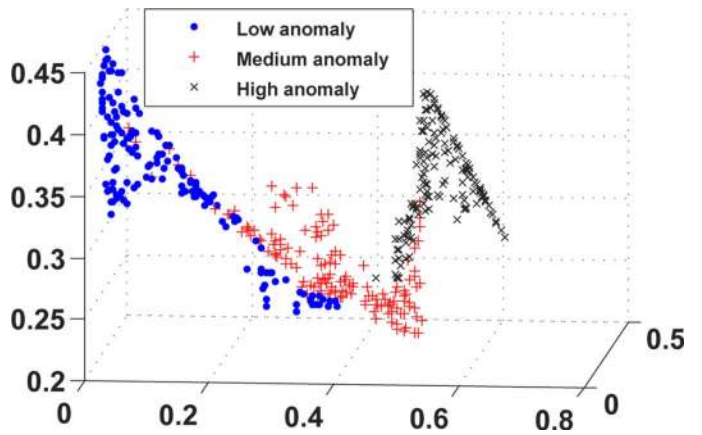


Fig. 13. Feature space of training set—maximum entropy partitioning (MEP).

4. Figs. 12 and 13 show the location of each training time series in the three dimensional (using first three linearly independent elements of the feature vectors) feature space plot using UP and MEP, respectively. It is observed that the class separation in Fig. 12 is not as good as that in Fig. 11, especially the separation between low/medium anomaly samples and high anomaly samples. The samples of the same class in Fig. 13 do not cluster as close as that in Fig. 11; some of the low anomaly samples (i.e., those with blue dots below the red $+$ cluster in Fig. 13) are closer to the medium anomaly cluster than their own cluster.

Finally, the confusion matrices [28] for the SDF-based methods (OptP, UP and MEP) with k -NN on the test data set are given by $C_{test}^{OptP+kNN}$, C_{test}^{UP+kNN} and $C_{test}^{MEP+kNN}$, respectively.

$$C_{test}^{OptP+kNN} = \begin{pmatrix} 142 & 8 & 0 \\ 14 & 136 & 0 \\ 0 & 4 & 146 \end{pmatrix}$$

$$C_{test}^{UP+kNN} = \begin{pmatrix} 145 & 5 & 0 \\ 17 & 132 & 1 \\ 0 & 6 & 144 \end{pmatrix}$$

$$C_{test}^{MEP+kNN} = \begin{pmatrix} 143 & 7 & 0 \\ 18 & 132 & 0 \\ 0 & 5 & 145 \end{pmatrix}.$$

In a confusion matrix, its elements c_{ij} denote the frequency of data in the class Cl_i being classified as data in class Cl_j . The diagonal elements c_{ii} represent the frequency of the data that are correctly classified. If all data are classified correctly, then all the off-diagonal elements of the confusion matrix must be zeros. There are fewer off-diagonal elements in $C_{test}^{OptP+kNN}$, which implies that the expected classification error of using OptP is smaller than those using MEP and UP. The minimum diagonal element in $C_{test}^{OptP+kNN}$ is larger than those in C_{test}^{UP+kNN} and $C_{test}^{MEP+kNN}$, which implies that the worst case error of using OptP is smaller than those using MEP and UP.

For comparative evaluation, the data sets are analyzed using other common pattern recognition tools [21], [22]. In this case study, principal component analysis (PCA) is used as the feature extraction tool, while k -NN, linear discriminant analysis (LDA) and least squares algorithm (LS) are used as the pattern classifiers. For PCA anomaly detection, $M = 450$, $N = 2400$, and by choosing $\eta = 0.90$, the corresponding number of largest eigenvalues turns out to be $m = 20$, as explained in Section V-B. The confusion matrices for PCA with k -NN, LDA and LS are given by $C_{test}^{PCA+kNN}$, $C_{test}^{PCA+LDA}$ and C_{test}^{PCA+LS} , respectively. An inspection of the confusion matrices reveals that PCA-based anomaly detection algorithm yields similar performance in identifying the low and high level anomalies as the SDF-based algorithm, but the performance in identifying the medium level anomaly is significantly worse.

$$C_{test}^{PCA+kNN} = \begin{pmatrix} 149 & 1 & 0 \\ 40 & 110 & 0 \\ 0 & 4 & 146 \end{pmatrix}$$

$$C_{test}^{PCA+LDA} = \begin{pmatrix} 136 & 14 & 0 \\ 18 & 126 & 6 \\ 0 & 3 & 147 \end{pmatrix}$$

$$C_{test}^{PCA+LS} = \begin{pmatrix} 145 & 5 & 0 \\ 27 & 115 & 8 \\ 0 & 0 & 150 \end{pmatrix}$$

Fig. 14 shows how the neighbor size in the k -NN classifiers affect the classification performance, and the performances of OptP, MEP, UP and PCA with different neighbor size k are compared. Only odd values of neighbor size are shown because when using an even value for k , it might be necessary to break a tie in the number of nearest neighbors by selecting a random tiebreaker or using the nearest neighbor among the tied groups. It is seen in Fig. 14 the classification performance of SDF-based methods is consistently better than that of PCA-based method and almost independent of the neighbor size, whereas the classification error of PCA increases with the number of neighbor size. This results show that the feature vectors extracted by SDF retain better separability among classes than those extracted by PCA.

Table III presents the comparison of the classification error related costs for SDF-based methods and PCA-based methods on the test data set. The observation made from these results indicate that the classification performance of SDF-based methods are superior to that of the PCA-based methods. The classification performance of MEP and UP are close, and UP performs

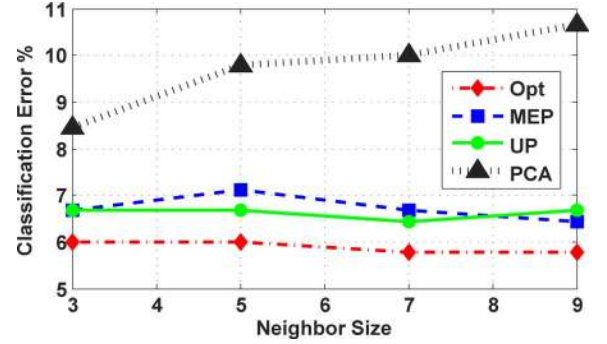


Fig. 14. Classification error vs. neighbor size in k -NN classifiers.

TABLE III
COMPARISON OF CLASSIFICATION PERFORMANCES OF DIFFERENT METHODS ON TEST-DATA SET (50×9 SAMPLES)

| Methods | $CostE$ | $CostW$ | Classification Error % |
|----------|---------|---------|------------------------|
| OptP+kNN | 0.0578 | 0.0933 | 5.78 |
| UP+kNN | 0.0644 | 0.1200 | 6.44 |
| MEP+kNN | 0.0667 | 0.1200 | 6.67 |
| PCA+kNN | 0.1000 | 0.2667 | 10.00 |
| PCA+LDA | 0.0911 | 0.1200 | 9.11 |
| PCA+LS | 0.0889 | 0.1800 | 8.89 |

slightly better in this case study. The costs due to worst-case classification error and expected classification error are reduced compared to that of the uniform partitioning scheme by optimizing the partitioning process over a representative training set.

It is noted that, for some problems, the classical partitioning schemes may perform similar to the optimal one. Therefore, the optimization procedure may also be used to evaluate the capability of any partitioning scheme towards achieving a better classification rate. The evaluation can be performed by using a part of the labeled training data set as the validation set. Although the construction of the cost functions allow solutions of problems with a large number of classes, its upper limit could be constrained by the alphabet size used for data partitioning that determines the dimension of the feature space.

VII. SUMMARY, CONCLUSIONS AND FUTURE WORK

This paper presents a data-driven method for detection of component-level anomalies in nuclear power plants with sensors that are subjected to different noise variance. The proposed anomaly detection method is based on Symbolic Dynamic Filtering (SDF) [7]–[9] that is recently reported in the literature as a real-time data-driven pattern recognition tool. The task of anomaly detection is viewed as a multi-class pattern classification problem, where feature extraction is optimized to enhance the classification rate. The proposed methodology efficiently (i.e., with low memory requirements and fast execution) compresses a high-volume database to execute anomaly detection for a large number of classes. It has been shown that the SDF-based anomaly detection algorithm has superior performance compared with the PCA-based anomaly detection algorithm.

This proposed method is tested on the IRIS simulator for the white Gaussian noise that is a common representation of sensor uncertainties. If exact specifications of the sensor noise are available, the anomaly detection algorithm should be trained based on these specifications. In the absence of this information, the best choice would be to use white Gaussian noise at different levels of signal-to-noise ratio (SNR), which is done in this paper.

Scalability is a critical issue for data-driven supervised methods of anomaly detection. A natural way to circumvent this problem is to perform anomaly isolation before estimating a particular anomaly level. Furthermore, although this paper has shown application of the developed feature extraction method to component anomaly detection problems, the technique is general enough to be used for detecting sensor and actuator anomalies under different architectures.

Potential applications of the proposed anomaly detection method include real-time detection of component-level anomalies in nuclear power plants with low false alarm and missed detection rates. While the training stage could be accomplished in a nuclear power plant simulator with high fidelity (e.g., scaled experimental reactor [34] or computer simulator [35]), the testing stage should be ideally performed in real time in an operating nuclear power plant. However, based on the simulation results only, it is found that the plant perturbation needed to execute the proposed anomaly detection method require small induced changes in the turbine load (e.g., less than 2% nominal load) over a short period, which does not have any significant adverse effects on stability and performance of the plant operation.

While there are research issues in this field [36] that are require in-depth research, the following areas are recommended for future research:

- Usage of other classifiers (e.g., Support Vector Machines (SVM) and Gaussian Mixture Models (GMM)) for comparison with the proposed method's performance;
- Inclusion of the step of tuning the classifier inside the optimization loop as described in the general framework shown in Fig. 8;
- Extension of the proposed anomaly detection method to accommodate other types of sensor degradation (e.g., bias and drifting);
- Investigation of anomalous situations with simultaneous anomalies in multiple plant components, actuators, and sensors;
- Reduction of the plant perturbation period for the testing phase without any significant compromise of detection performance.

ACKNOWLEDGMENT

The authors would like to thank Prof. J. Michael Doster for his technical support in implementing the simulation experiments using the IRIS simulator.

REFERENCES

[1] K. Zhao and B. R. Upadhyaya, "Model based approach for fault detection and isolation of helical coil steam generator systems using principal component analysis," *IEEE Trans. Nucl. Sci.*, vol. 53, no. 4, pp. 2343–2352, Aug. 2006.

[2] K. C. Gross, R. M. Singer, S. W. Wegerich, and J. P. Herzog, "Application of a model-based fault detection system to nuclear plant signals," in *Proc. 9th Int. Conf. Intelligent Systems Applications to Power Systems*, Seoul, Korea, Jul. 6–10, 1997.

[3] B. R. Upadhyaya, K. Zhao, and B. Lu, "Fault monitoring of nuclear power plant sensors and field devices," *Prog. Nucl. Energy*, vol. 43, no. 1–4, pp. 337–342, 2003.

[4] B. Lu and B. R. Upadhyaya, "Monitoring and fault diagnosis of the steam generator system of a nuclear power plant using data-driven modeling and residual space analysis," *Ann. Nucl. Energy*, vol. 32, pp. 897–912, 2005.

[5] K. Hadad, M. Mortazavi, M. Mastali, and A. Safavi, "Enhanced neural network based fault detection of a VVER nuclear power plant with the aid of principal component analysis," *IEEE Trans. Nucl. Sci.*, vol. 55, no. 6, pp. 3611–3619, Dec. 2008.

[6] P. F. Fantoni and A. Mazzola, "A pattern recognition-artificial neural networks based model for signal validation in nuclear power plants," *Ann. Nucl. Energy*, vol. 23, no. 13, pp. 1069–1076, 1996.

[7] A. Ray, "Symbolic dynamic analysis of complex systems for anomaly detection," *Signal Process.*, vol. 84, no. 7, pp. 1115–1130, 2004.

[8] V. Rajagopalan and A. Ray, "Symbolic time series analysis via wavelet-based partitioning," *Signal Process.*, vol. 86, no. 11, pp. 3309–3320, Nov. 2006.

[9] S. Gupta, A. Ray, and E. Keller, "Symbolic time series analysis of ultrasonic data for early detection of fatigue damage," *Mech. Syst. Signal Process.*, vol. 21, no. 2, pp. 866–884, 2007.

[10] A. Ray and S. Phoha, "Calibration and estimation of redundant signals for real-time monitoring and control," *Signal Process.*, vol. 83, no. 12, pp. 2593–2605, 2003.

[11] X. Jin, A. Ray, and R. M. Edwards, "Redundant sensor calibration and estimation for monitoring and control of nuclear power plants," *Trans. Amer. Nucl. Soc.*, vol. 101, pp. 307–308, Nov. 2009.

[12] E. Vidal, F. Thollard, C. Higuera, F. Casacuberta, and R. C. Carrasco, "Probabilistic finite-state machines-part I," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 7, pp. 1013–1025, Jul. 2005.

[13] M. D. Carelli, "The design and safety features of the IRIS reactor," *Nucl. Eng. Des.*, vol. 230, pp. 151–167, 2004.

[14] J. March-Leuba, J. Mullens, R. Wood, B. Upadhyaya, J. Doster, and C. Mayo, A New Paradigm for Automatic Development of Highly Reliable Control Architectures for Nuclear Power Plants, 2002, NERI Project 99-119, Final Rep., ORNL/TM-2002/193.

[15] H. Shen, "Advanced Feedwater Control for Next Generation Nuclear Power Systems," Ph.D. dissertation, Dept. Nucl. Eng., North Carolina State Univ., Raleigh, 2006.

[16] H. Shen and J. M. Doster, "Application of a neural network based feed-water controller to helical steam generators," *Nucl. Eng. Des.*, vol. 239, no. 6, pp. 1056–1065, 2009.

[17] S. C. Stultz and J. B. Kitto, *Steam: Its Generation and Use*, 41st ed. Barberton, OH: Babcock & Wilcox, 2005.

[18] F. Koushanfar, M. Potkonjak, and A. Sangiovanni-Vincentelli, "Online fault detection of sensor measurements," in *IEEE Sensors*, 2003, pp. 974–980.

[19] R. Steuer, L. Molgedey, W. Ebeling, and M. Jimenez-Montano, "Entropy and optimal partition for data analysis," *Euro. Phys. J. B*, vol. 19, pp. 265–269, 2001.

[20] A. Subbu and A. Ray, "Space partitioning via hilbert transform for symbolic time series analysis," *Appl. Phys. Lett.*, vol. 92, no. 8, pp. 084107-1–084107-3, Feb. 2008.

[21] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2001.

[22] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer-Verlag, 2006.

[23] G. J. Mclachlan, *Discriminant Analysis and Statistical Pattern Recognition*. New York: Wiley-Interscience, 2004.

[24] E. Choi and C. Lee, "Feature extraction based on the Bhattacharyya distance," *Pattern Recognit.*, vol. 36, pp. 1703–1709, Aug. 2003.

[25] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 21–27, 1967.

[26] M. Buhl and M. B. Kennel, "Statistically relaxing to generating partitions for observed time-series data," *Phys. Rev. E*, vol. 71, no. 4, pp. 046213–046213, Apr. 2005.

[27] S. Sarkar, K. Mukherjee, and A. Ray, "Generalization of Hilbert transform for symbolic analysis of noisy signals," *Signal Process.*, vol. 89, no. 6, pp. 1245–1251, 2009.

[28] R. Kohavi and F. Provost, "Glossary of terms," *Mach. Learn.*, vol. 30, no. 2/3, pp. 271–274, 1998.

- [29] R. Alaiz-Rodríguez, A. Guerrero-Curieses, and J. Cid-Sueiro, "Minimax classifiers based on neural networks," *Pattern Recognit.*, vol. 38, no. 1, pp. 29–39, 2005.
- [30] R. Steuer, *Multiple Criteria Optimization: Theory, Computations, and Application*. New York: Wiley, 1986.
- [31] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
- [32] K. Fukunaga, *Statistical Pattern Recognition*, 2nd ed. Boston, MA: Academic, 1990.
- [33] D. Lind and M. Marcus, *An Introduction to Symbolic Dynamics and Coding*. Cambridge, U.K.: Cambridge Univ. Press, 1995.
- [34] M. Carelli *et al.*, "The SPES3 experimental facility design for the IRIS reactor simulation," *Sci. Technol. Nucl. Install.*, vol. 2009, no. 579430, 2009.
- [35] T. Bajs, D. Grgić, V. Segon, L. Oriani, and L. Conway, "Development of RELAP5 nodalization for IRIS non-LOCA transient analyses," in *Proc. Amer. Nuclear Society Topical Meeting in Mathematics and Computations*, Gatlinburg, TN, Apr. 6–10, 2003.
- [36] S. Gupta and A. Ray, "Statistical mechanics of complex systems for pattern identification," *J. Stat. Phys.*, vol. 134, no. 2, pp. 337–364, 2009.