

Comparative Evaluation of Symbolic Dynamic Filtering for Detection of Anomaly Patterns

Chinmay Rao Soumik Sarkar Asok Ray Murat Yasar
crr164@psu.edu szs200@psu.edu axr2@psu.edu yasarm@technosci.com
*Pennsylvania State University
University Park, PA-16802, USA*

Abstract—Symbolic Dynamic Filtering (*SDF*) has been recently reported in literature as a pattern recognition tool for early detection of anomalies (i.e., deviations from the nominal behavior) in complex dynamical systems. This paper presents a comparative evaluation of *SDF* relative to other classes of pattern recognition tools, such as Bayesian Filters and Artificial Neural Networks, from the perspectives of: (i) Anomaly detection capability, (ii) Decision making for failure mitigation and (iii) Computational efficiency. The evaluation is based on analysis of time series data generated from a nonlinear active electronic system.

Index Terms—Bayesian Filtering, Symbolic Dynamics, Neural Networks, Anomaly Detection

I. INTRODUCTION

ANOMALY is defined as deviation from nominal behavior of a dynamical system. For many human-engineered complex systems, early detection of anomalies with low false alarm rates mitigates the risk of forthcoming failures. Recently, a pattern identification technique, called Symbolic Dynamic Filtering (*SDF*), has been reported [1], [2] for early detection of anomaly patterns in dynamical systems, possibly due to parametric or non-parametric changes. While abrupt changes of large magnitude are not difficult to detect, *SDF* specifically meets the challenge of detecting slowly evolving anomalies at an early stage. The core concept of *SDF* is built on identification of statistical patterns from symbol sequences generated by coarse-graining of time series data [1], [3]. These statistical patterns represent behavior of the dynamical system, which may change with the evolution of anomaly(ies). The information contained in a set of time series data is compressed in the form of a probability histogram that may evolve with the anomaly progression.

The major objective of this paper is to evaluate *SDF* with other pattern recognition methods such as Bayesian Filtering (*BF*), which is both model-based and dynamic data-driven, and is capable of detecting parametric or non-parametric changes in the model. The Kalman (Extended Kalman) Filter is often adequate for linear (linearized) systems, but it may fail to capture the dynamics of a nonlinear system, specifically with non-additive uncertainties.

This work has been supported in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under Grant No. W911NF-07-1-0376 and by NASA under Grant No. NNX07AK49A.

Recent literature has reported Monte Carlo Markov Chain (*MCMC*) techniques, such as Particle Filtering [4] and Sigma Point techniques [5] that yield numerical solutions to Bayesian state estimation problems and have been applied to anomaly detection in nonlinear dynamical systems [6]. In addition to *BF*, this paper investigates other classes of well-known pattern recognition tools such as Artificial Neural Networks (*ANN*), Principal Component Analysis (*PCA*), [7] and Kernel Regression Analysis (*KRA*) [8] for pattern change detection [7]. In the class of *ANN*, multilayer perceptron and radial basis function configurations have been widely used for detection of anomalous patterns [7]. These pattern recognition tools have been evaluated for comparison with *SDF* from the following perspectives.

- Performance evaluation in terms of quality of anomaly detection (e.g., enhanced detection capability and reduced rate of false alarm)
- Decision making for mitigation of forthcoming failures
- Computational efficiency (e.g., execution time and memory requirements)

II. REVIEW OF SYMBOLIC DYNAMIC FILTERING

The theory of symbolic dynamic filtering (*SDF*) for time series data analysis is built upon the underlying principles of *Nonlinear Dynamics*, *Symbolic Dynamics*, *Information Theory*, and *Statistical Pattern Recognition*. While the details are reported in previous publications [1], [3], the essential concepts are succinctly explained in this section for completeness of this paper.

Detection of anomaly patterns is formulated as a two-time-scale problem, illustrated in Fig. 1. The *fast time scale* is related to response time of the process dynamics. Over the span of a given time series data sequence, dynamic behavior of the system is assumed to remain invariant. The *slow time scale* is related to the time span over which parametric or non-parametric changes may occur and exhibit non-stationary dynamics.

In general, a long time span in the fast time scale is several orders of magnitude smaller in the slow time scale. The continuously varying process of system dynamics is often modeled as a finite dimensional dynamical system in the setting of an initial value problem as:

$$\frac{d\mathbf{x}(t)}{dt} = f(\mathbf{x}(t), \theta(t_s)); \mathbf{x}(0) = \mathbf{x}_0, \quad (1)$$

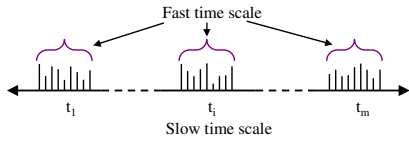


Fig. 1. Pictorial view of the two time scales: (i) *Slow time scale* of anomaly evolution and (ii) *Fast time scale* for data acquisition and signal conditioning

where $t \in [0, \infty)$ denotes the (fast-scale) time; $\mathbf{x} \in \mathbb{R}^n$ is the state vector in the phase space; and $\theta \in \mathbb{R}^\ell$ is the (possibly anomalous) parameter vector varying in (slow-scale) time t_s . A convenient way of learning the dynamical behavior is to rely on the additional information provided by (sensor-based and/or model-based) time series data [9].

A. Symbolic Dynamics, Encoding, and State Machine

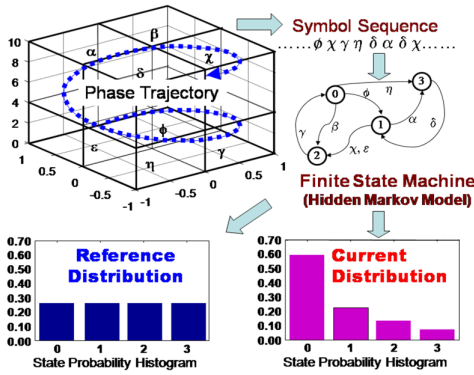


Fig. 2. Concept of Symbolic Dynamic Filtering

Let $\Omega \in \mathbb{R}^n$ be a compact region, within which the trajectory of the dynamical system, governed by Eq. (1), is circumscribed as illustrated in Fig. 2. The region Ω is partitioned into a finite number of (mutually exclusive and exhaustive) cells, so as to obtain a coordinate grid. Let the cell, visited by the trajectory at a time instant, be denoted as a random variable taking a symbol value from the alphabet Σ . An orbit of the dynamical system is described by the time series data as $\{x_0, x_1, \dots, x_k, \dots\}$ with $x_i \in \Omega$, which passes through or touches one of the cells of the partition. Each initial state $x_0 \in \Omega$ generates a sequence of symbols defined by a mapping from the phase space into the symbol space as:

$$x_0 \rightarrow s_0 s_1 s_2 \dots s_k \dots \quad (2)$$

where each s_i , $i = 0, 1, \dots$ takes a symbol from the alphabet Σ .

The mapping in Eq. (2) is called *Symbolic Dynamics* as it attributes a physically admissible sequence of symbols to the system dynamics starting from an initial state. Figure 2 elucidates the concepts of partitioning a finite region of the phase space and the mapping from the partitioned space into the symbol alphabet. Figure 2 also shows conversion of the symbol sequence into a finite-state machine.

B. Space Partitioning

Several partitioning techniques have been reported [10] for symbol generation from the phase space, primarily based

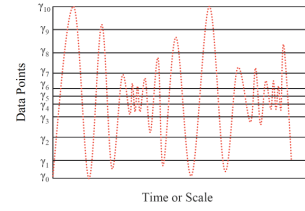


Fig. 3. An Example of Space Partitioning

on Symbolic False Nearest Neighbors (*SFNN*), which may become computationally intensive if the dimension of the phase space is large. The wavelet transform [11] largely alleviates these shortcomings and has been shown to yield comparable performance with several orders of magnitude smaller execution time [3].

In wavelet based partitioning, the time series data are first converted to the wavelet domain, where wavelet coefficients are generated at different time shifts. The wavelet space is then partitioned with *alphabet* size $|\Sigma|$ into segments of coefficients on the ordinate separated by horizontal lines. In the illustrative example of Fig. 3, the partitioning has been done to create $|\Sigma| = 10$ cells (i.e., intervals along the ordinate in this case). The choice of $|\Sigma|$ depends on specific experiments, noise level and also the available computation power. A large *alphabet* may be noise-sensitive while a small *alphabet* could miss the details of signal dynamics.

Once the partitioning is done with alphabet size $|\Sigma|$ at the nominal condition (time epoch t_0), it is kept constant for all (slow time) epochs $\{t_1, t_2, \dots, t_k, \dots\}$.

C. State Machine Construction

A finite state machine is now constructed, where the states of the machine are defined corresponding to the given *alphabet* set Σ and window length D . D is chosen as the length of consecutive symbol words to be considered [1]. Each state belongs to an equivalence class of symbol words of length D or more, which is characterized by a word of length D at the leading edge. Therefore, the number n of such equivalence classes is less than or equal to the total permutations of the alphabet symbols within words of length D . That is, $n \leq |\Sigma|^D$; some of the states may be forbidden with zero probability of occurrence.

Using the symbol sequence generated, the state machine is constructed on the principle of sliding block codes [12]. The window of length D on the symbol sequence $\dots \sigma_{i_1} \sigma_{i_2} \dots \sigma_{i_k} \dots$ is shifted to the right by one symbol, such that it retains the last $(D-1)$ symbols of the previous state and appends it with the new symbol σ_{i_ℓ} at the end. The machine constructed in this fashion is called the D -Markov machine [1].

The states of the machine are marked with the corresponding symbolic word permutation and the edges joining the states indicate the occurrence of a symbol σ_{i_ℓ} . The occurrence of a symbol at a state may keep the machine in the same state or move it to a new state. On a given symbol sequence $\dots \sigma_{i_1} \sigma_{i_2} \dots \sigma_{i_\ell} \dots$ generated from the time series data collected at a slow time epoch, a window of length D is moved by keeping a count of occurrences of word

sequences $\sigma_{i_1} \cdots \sigma_{i_D} \sigma_{i_{D+1}}$ and $\sigma_{i_1} \cdots \sigma_{i_D}$ which are respectively denoted by $N(\sigma_{i_1} \cdots \sigma_{i_D} \sigma_{i_{D+1}})$ and $N(\sigma_{i_1} \cdots \sigma_{i_D})$. For $N(\sigma_{i_1} \cdots \sigma_{i_D}) \neq 0$, the transitions probabilities are then obtained by these frequency counts as follows:

$$\begin{aligned} \pi_{jk} &\equiv P(q_k|q_j) = \frac{P(q_k, q_j)}{P(q_j)} = \frac{P(\sigma_{i_1} \cdots \sigma_{i_D} \sigma)}{P(\sigma_{i_1} \cdots \sigma_{i_D})} \\ &\Rightarrow \pi_{jk} \approx \frac{N(\sigma_{i_1} \cdots \sigma_{i_D} \sigma)}{N(\sigma_{i_1} \cdots \sigma_{i_D})} \end{aligned} \quad (3)$$

where the corresponding states are denoted by $q_j \equiv \sigma_{i_1} \sigma_{i_2} \cdots \sigma_{i_D}$ and $q_k \equiv \sigma_{i_2} \cdots \sigma_{i_D} \sigma$.

D. Anomaly Evolution and Pattern Identification

Behavioral pattern changes may take place in dynamical systems due to accumulation of faults and progression of anomalies. The pattern changes are quantified as deviations from the nominal pattern and are characterized by a scalar-valued function, called *Anomaly Measure* μ . The anomaly measures at slow time epochs $\{t_1, t_2, \dots\}$ are obtained as:

$$\mu^k \equiv d(\mathbf{p}^k, \mathbf{p}^0)$$

where the $d(\bullet, \bullet)$ is an appropriately defined distance function.

III. CONSTRUCTION OF ANOMALY DETECTION ALGORITHMS

This section explains how anomaly detection algorithms are constructed for different pattern recognition tools.

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. Data sets are collected at different slow time epochs $t_0, t_1, t_2, \dots, t_k, \dots$
- **Generation of wavelet transform coefficients** [11], obtained with an appropriate choice of the wavelet basis and scales [3].
- **Partitioning** [3] of the wavelet space at the nominal condition at time epoch t_0 . Each segment of the partitioning is assigned a symbol from the alphabet Σ .
- **Construction of a finite state automaton** at time epoch t_0 (nominal condition) from alphabet size $|\Sigma|$ and window length D . The structure of the finite state machine is fixed for subsequent slow time epochs $\{t_1, t_2, \dots, t_k, \dots\}$.
- **Calculation of the state probability vectors** $\mathbf{p}^0, \mathbf{p}^1, \mathbf{p}^2, \dots, \mathbf{p}^k, \dots$. The probability distribution \mathbf{p}^0 of damage patterns is recursively computed as an approximation of the natural invariant density of the dynamical system at the slow time epoch t_0 . Subsequently $\mathbf{p}^1, \mathbf{p}^2, \dots, \mathbf{p}^k, \dots$ at slow time epochs, $t_1, t_2, \dots, t_k, \dots$ are computed from the respective symbolic sequences using the finite state machine constructed at time epoch t_0 .
- **Computation of scalar anomaly measures** $\mu^1, \mu^2, \dots, \mu^k, \dots$ based on evolution of these probability

vectors and by defining an appropriate scalar distance function $\mu_k = d(\mathbf{p}^k, \mathbf{p}^0)$ with respect to the nominal condition [1].

B. Bayesian Filtering for Anomaly Detection

Bayesian filtering tracks the states more effectively if the system is closer to the nominal condition. The tracking error would be greater when the system is in an anomalous condition. To this effect, the innovation sequences are computed, and their histograms are obtained, where the innovation ϵ is defined as the difference between the true output y and the predictor output \hat{y}^- .

At the nominal condition, the histogram of the innovation sequence resembles an impulse, or approximately Gaussian sequence with very small variance. As the anomaly progresses, the model becomes less accurate and the estimation errors become higher. Thus, the histogram of the innovation sequence shows an increase in the variance and the distribution diverges from a Gaussian. This increase is characterized as a measure of the anomaly. To this effect, the probability density of the innovation sequences $\mathbf{p}^k(\epsilon)$ are generated at slow time epochs t_k and the anomaly measure μ_k at any epoch k is given by an appropriate distance function $d(\mathbf{p}^k(\epsilon), \mathbf{p}^0(\epsilon))$.

C. Neural Networks for Anomaly Detection

The training data set for both types of neural networks, namely, Radial Basis Function Neural Networks, (*RBFNN*) and Multi Layer Perceptron Neural Networks, (*MLPNN*), are prepared in the same manner. The neural networks are trained based on the *NARX* model from the input-output data sets at the nominal condition. After an error goal is achieved, the neural network is allowed to track the output signal of the system under both nominal and anomalous conditions. The neural network generates an output signal estimate \hat{y} . The innovation $\epsilon_k \triangleq (y_k - \hat{y}_k)$ serves as a measure for the tracking performance and a *pdf* is created for the innovation sequence. If at nominal condition the *pdf* is \mathbf{p}^0 and the *pdf* at slow time epoch t_k is \mathbf{p}^k , then the anomaly measure is given by the distance $d(\mathbf{p}^k, \mathbf{p}^0)$.

D. Statistical methods for Anomaly Detection

1) **Principal Component Analysis (PCA)**: *PCA* serves as a feature selector in the pattern analysis via dimension reduction from n to m . The $n \times n$ covariance matrix, obtained from the time series data, generates the orthonormal eigenvectors v^k and the corresponding non-negative real eigenvalues λ_k . The eigenvalues are arranged in the increasing order of magnitude. The m largest eigenvalues and associated eigenvectors are selected such that $\sum_{i=1}^m \lambda_i > \eta \sum_{i=1}^n \lambda_i$, where η is a real positive number close to 1 (e.g., $\eta = 0.95$). The principal feature matrix F is defined as:

$$F = \left[\sqrt{\frac{\lambda_1}{\sum_{i=1}^m \lambda_i}} v^1 \quad \dots \quad \sqrt{\frac{\lambda_d}{\sum_{i=1}^m \lambda_d}} v^d \right] \quad (4)$$

The feature matrix F^0 represents the status of the system derived from the time series data at the nominal condition

t_0 . Similarly, feature matrix F^k is obtained from time series data at slow time epoch t_k . Then, the anomaly measure at t_k is obtained as the distance $d(F^k, F^0)$.

2) *Kernel Regression Analysis (KRA)*: In *KRA* At the nominal condition, the kernel estimator is $\hat{f}_0(x)$. For different anomalous conditions, the regression parameters, (μ, θ_α) , are kept fixed; and the kernel estimator $\hat{f}_k(x)$ is evaluated from the data set under the (possibly anomalous) condition at the slow time epoch t_k . Then, the anomaly measure at the k^{th} epoch is obtained as the distance $d(\hat{f}_k, \hat{f}_0)$.

The distance function for all methods is chosen as the standard Euclidean norm.

IV. EXPERIMENTS AND RESULTS

This section presents results of anomaly detection experimentation on an active electronic circuit apparatus that implements a second order non-autonomous, forced Duffing equation [13]. The governing equation of the system with a cubic nonlinearity is given below:

$$\frac{d^2 y}{dt^2} + \beta(t_s) \frac{dy}{dt} + y(t) + y^3(t) = A \cos(\omega t) \quad (5)$$

The dissipation parameter $\beta(t_s)$, realized as a resistance in the circuit, varies in the slow time t_s and is treated as a constant in the fast time t at which the dynamical system is excited. The goal is to detect, at an early stage, changes in $\beta(t_s)$ that is associated with the anomaly. The effects of growth in $\beta(t_s)$ are presented as the response of a stimulus with amplitude $A = 22$ and frequency $\omega = 5$. The stationary behavior of the system response for this input stimulus is obtained for several values of β in the range of 0.10 to 0.40. The four plates, 4(a) to 4(d), in the first row of Fig. 4 exhibit four phase plots in for the values of the parameter at 0.10, 0.30, 0.32, and 0.34, respectively. Each plot relates the phase variable of electrical charge that is proportional to the voltage across one of the capacitors in the electronic circuit, with its time derivative (i.e., the instantaneous current). While a small difference between the phase plots for $\beta = 0.10$ and $\beta = 0.30$ is noticeable, there is no clearly visible difference between the plots for $\beta = 0.30$ and $\beta = 0.32$ in plates 4(b) and 4(c). However, the phase plots for $\beta = 0.32$ and $\beta = 0.34$ in the plates, 4(c) and 4(d), display a very large difference, indicating period doubling possibly due to onset of bifurcation.

The four plates, 4(e) to 4(h), in the second row of Fig. 4 exhibit four histograms that are the pattern vectors generated by *SDF* for β equal to 0.10, 0.30, 0.32, and 0.34, respectively. It is seen that, in the plate 4(h), a sudden change in the pattern vector occurs as only the first two states are visited frequently. This is indicative of a transition to significantly different dynamical behavior.

The four plates, 4(i) to 4(l), in the third row of Fig. 4 exhibit four probability density functions that are the pattern vectors generated by *RBFNN* for β equal to 0.10, 0.30, 0.32, and 0.34, respectively. It is also seen in the plate 4(l) that a sudden change in the shape of probability distribution

occurs, indicating a transition to a significantly different dynamical behavior.

The plates, 4(m) to 4(p) in the fourth row of Fig. 4 exhibit four probability density functions that are the pattern vectors generated by *PF* for β equal to 0.10, 0.30, 0.32, and 0.34, respectively. It is observed the variance of the (non-Gaussian) probability distribution increases with β . This is indicative of increasing state estimation error due to modeling error. The remarkable trait in Plate 4(p) is that the distribution abruptly changes from a unimodal to a bimodal structure, indicating a transition to a significantly different dynamical behavior.

Plots of the normalized anomaly measure μ versus the dissipation parameter β are exhibited in Fig. 5(a) for *SDF*, *PCA*, *RBFNN* and *MLPNN*, and in Fig. 5(b) for *SDF*, *PF*, *UKF* and *KRA*. The profiles in each of these two figures exhibit the growth of μ as β grows from 0.1 (considered as the nominal condition) to the completely faulty condition of 0.4. All profiles show gradual increase in μ with β .

Figure 5(a) compares *SDF* for detection of anomaly patterns to *MLPNN* and *RBFNN* neural networks as well as Principal Component Analysis (*PCA*) and Bayesian Filter-based methods (*PF* and *UKF*). The Multilayer Perceptron Neural Network (*MLPNN*) consists of three hidden layers with 50 neurons in each one of them and an output layer with one neuron (as the number of output is one). On the other hand, the Radial Basis Function Neural Network (*RBFNN*) uses only one hidden layer and one output layer (with one neuron) as described earlier. Optimal training was obtained using 100 neurons in the hidden layer. For training of the network, two thousand data points are chosen from the input-output time-series data set of the nominal system, i.e., with $\beta = 0.1$ at steady state.

Fig. 5(b) compares the performance of *SDF* with Bayesian filter-based methods (i.e., particle filter (*PF*) and unscented filter (*UKF*)). These filters are calibrated to the nominal condition of $\beta = 0.1$, and the filter is designed to track both states (e.g., $y(t)$ and $\dot{y}(t)$), where 50 particles are used for the particle filter, as a tradeoff between tracking performance in the nominal conditions and CPU execution time and memory requirements. For unscented filtering, the parameter κ is set equal to 3, which is reported to be optimal for Gaussian priors [5]. For both *PF* and *UKF*, the variance of the zero-mean Gaussian process noise is set to 0.01 and the variance for zero-mean Gaussian measurement noise is 0.05. The Monte Carlo Markov Chain (*MCMC*) analysis has been carried out on 10,000 data points, sampled at a rate of $T_s = 0.01 \text{ sec}$.

Figures 5(a) and Fig. 5(b) exhibit a family of normalized profiles of anomaly measure μ versus the dissipation parameter β , where each profile show gradual increase in μ until the bifurcation at $\beta \approx 0.33$. Changes in the value of μ , its slope (i.e., $\frac{\partial \mu}{\partial \beta}$), and its curvature (i.e., $\frac{\partial^2 \mu}{\partial \beta^2}$) provide early warnings for a forthcoming major change in the system dynamics. From this perspective, the performance of *SDF* is superior to that of Bayesian filtering, both types of Neural networks, and other statistical methods (i.e., *PCA* and *KRA*). It is also noted that the profile of *SDF* is smoother than

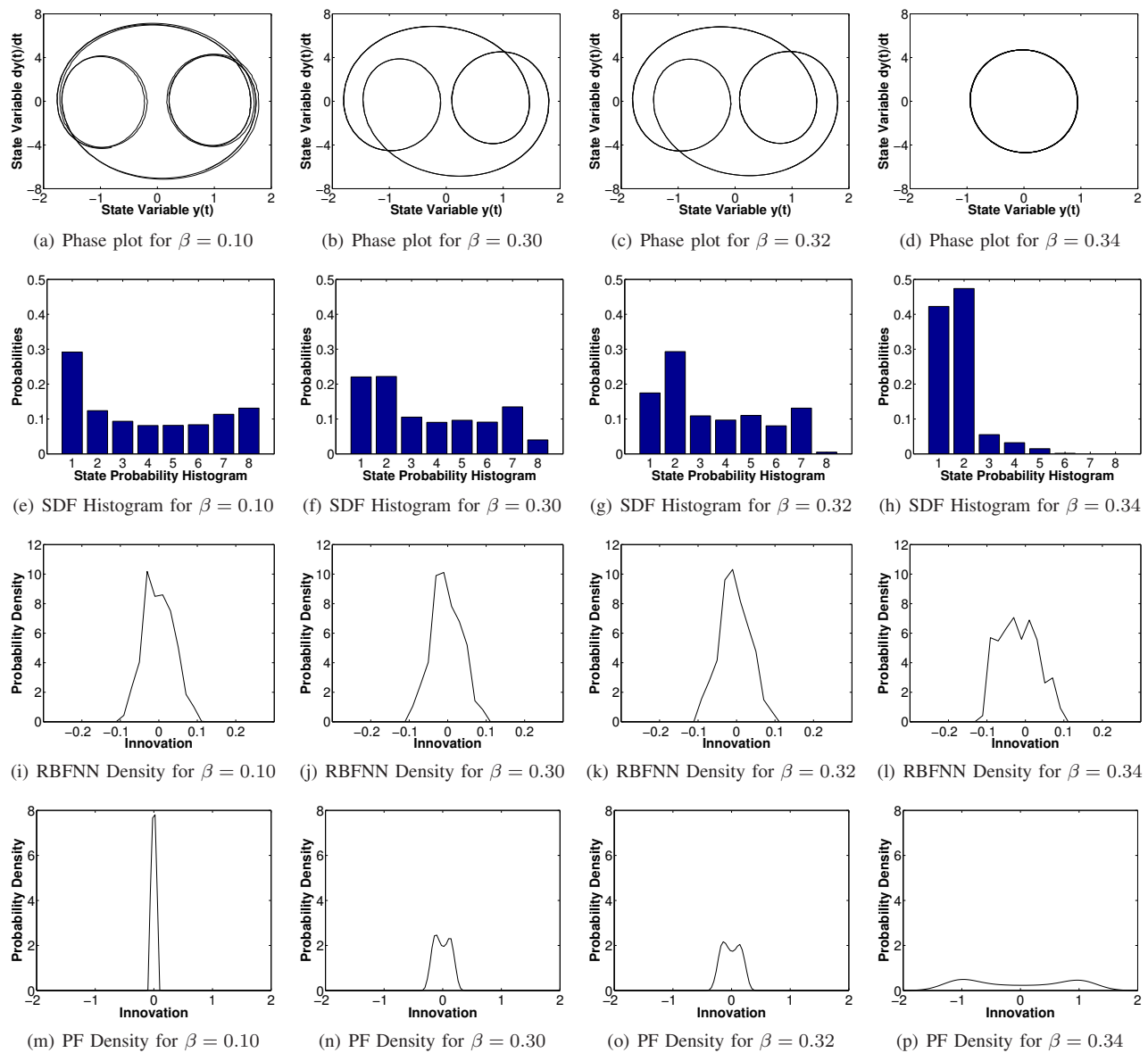


Fig. 4. Evolution of anomaly patterns for changes in system dynamics

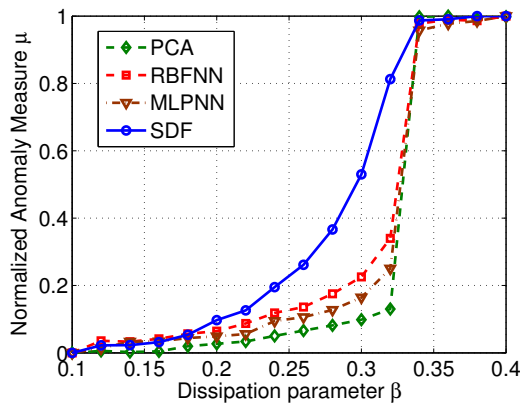
those of PF and UKF. The smoothness of SDF reduces false alarms particularly for small changes in β from the nominal condition. Similarly, SDF outperforms *RBFNN*, *MLPNN*, *PCA*, and *KRA*.

Table I provides a comparison of execution time and memory requirement of the afore-mentioned seven methods for computation of the anomaly measure μ . In each case, the CPU time for a single operation cycle at a time epoch, listed in Table I, is obtained from the average of execution time for operation cycles at 16 consecutive slow time epochs on a 3.40 GHz Pentium 4 processor in the Matlab 7.0.1 environment. As seen in Table I, the execution time varies from a fraction of millisecond for *KRA* to hundreds of seconds for *PF*. Execution time for Neural Network-based methods and SDF are comparable although *RBFNN* is faster than *MLPNN* and SDF. However, Bayesian filters *UKF* and *PF* are one and two orders of magnitude slower than SDF, respectively. The requirement of (random access) memory

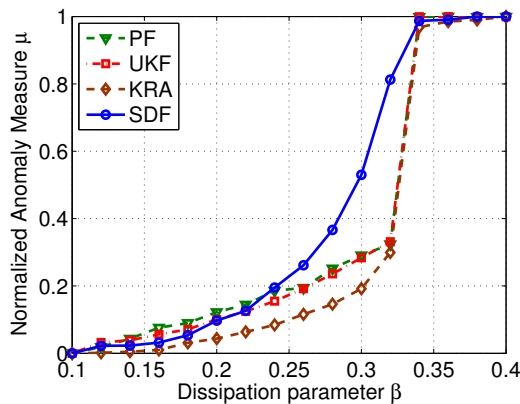
TABLE I
COMPARISON OF EXECUTION TIME

Anomaly detection method	Execution time	Memory requirement
<i>KRA</i>	2.23×10^{-3} sec	2.95 MB
<i>PCA</i>	4.30×10^{-2} sec	2.88 MB
<i>RBFNN</i>	8.09×10^{-1} sec	4.05 MB
<i>MLPNN</i>	4.60×10^0 sec	4.15 MB
SDF	4.65×10^0 sec	2.94 MB
<i>UKF</i>	5.10×10^1 sec	4.19 MB
<i>PF</i>	2.74×10^2 sec	4.69 MB

in each case is more or less similar (less than 5MB), which is insignificant for a commercially available laptop computer. However, for *RBFNN* and *MLPNN*, the training phase requires 45MB to 60 MB of memory, which is also reasonable.



(a) Comparison of *SDF*, *ANN*, and *PCA*



(b) Comparison of *SDF*, Bayesian filtering, and *KRA*

Fig. 5. Evaluation of Gradually Evolving Anomaly Patterns

V. SUMMARY, CONCLUSION AND FUTURE WORK

Symbolic Dynamic Filtering (*SDF*) is compared with three classes of pattern recognition techniques. The comparative evaluation is based on a non-linear dynamical system, represented by the forced Duffing equation, with a slowly varying dissipation parameter. The time series data are converted to symbol sequences, which are then represented as a finite-state automaton, called the D-Markov machine [1]. The evolution of a state probability vector yields a measure of the anomaly to be measured. In Bayesian methods, a nonlinear state estimator is designed for the nominal condition of the system. Two such nonlinear estimators, studied in this paper, are Unscented Kalman Filter (*UKF*) [5] and the Particle Filter (*PF*) [4]. This nonlinear estimator is then applied to the system as the anomaly evolves, and probability density of the resulting innovation sequence is the pattern vector. A similar procedure is applied for neural networks and also for statistical methods of principal component analysis [14] and kernel regression analysis [8], which are first trained on the nominal condition, and then are analyzed on the anomalous system. The distance between the state error probability density functions serves as a measure of the evolving anomaly.

The conclusions, derived from the work reported in this paper, are delineated as follows:

- 1) Symbolic Dynamic Filtering provides the best results in terms of early detection capability, speed of ex-

ecution, smoothness of anomaly detection curve and a sharp representation at the bifurcation point that is analogous to the onset of a large failure.

- 2) Statistical and neural network tools of pattern recognition perform at a speed comparable to that of symbolic dynamic filtering. However, they do not capture the gradual evolution of anomalies as early as *SDF* does.
- 3) Bayesian methods have the advantage of computing the estimated states. However, prolonged computation time requirements make them difficult to implement in an online scenario.

A major advantage of working with *SDF* is that pattern vectors are computed in real time and can be effectively transmitted over mobile wireless networks, thus making *SDF* ideally suited for online health monitoring and failure prognosis at a remote location. This is extremely important, for example, in a sensor network scenario, where both memory and processor time of local computers might be severely constrained.

Further theoretical and experimental research is recommended in the following areas:

- 1) Input independence of *SDF* by taking advantage of model-based analysis and system identification
- 2) Robustness assessment under noise contamination of time series data
- 3) Implementation of *SDF* on a sensor network for real time, online fault detection

REFERENCES

- [1] A. Ray, "Symbolic dynamic analysis of complex systems for anomaly detection," *Signal Processing*, vol. 84, no. 7, pp. 1115–1130, 2004.
- [2] S. Gupta, A. Ray, and E. Keller, "Symbolic time series analysis of ultrasonic data for early detection of fatigue damage," *Mechanical Systems and Signal Processing*, vol. 21, no. 2, pp. 866–884, 2007.
- [3] V. Rajagopalan and A. Ray, "Symbolic time series analysis via wavelet-based partitioning," *Signal Processing*, vol. 86, no. 11, pp. 3309–3320, 2006.
- [4] C. Andrieu, A. Doucet, S. Singh, and V. B. Tadic, "Particle methods for change detection, system identification, and control," *Proceedings IEEE*, vol. 92, no. 3, pp. 423–438, 2004.
- [5] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 477–482, 2000.
- [6] P. Li and V. Kadiramanathan, "Particle filtering based likelihood ratio approach to fault diagnosis in nonlinear stochastic systems," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 31, no. 3, pp. 337–343, 2001.
- [7] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. John Wiley, 2001.
- [8] J. Shawe-Taylor, *Kernel Methods for Pattern Analysis*. Cambridge, U.K.: Cambridge University Press, 2004.
- [9] C. Beck and F. Schögel, *Thermodynamics of Chaotic Systems: An Introduction*. Cambridge University Press, 1993.
- [10] M. B. Kennel and M. Buhl, "Estimating good discrete partitions from observed data: Symbolic false nearest neighbors," *Physical Review Letters*, vol. 91, no. 8, p. 084102, 2003.
- [11] S. Mallat, *A Wavelet Tour of Signal Processing*, 2nd ed. Boston, MA: Academic Press, 1998.
- [12] D. Lind and M. Marcus, *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, 1995.
- [13] J. M. T. Thompson and H. B. Stewart, *Nonlinear Dynamics and Chaos*. John Wiley, Chichester, United Kingdom, 1986.
- [14] K. Fukunaga, *Statistical Pattern Recognition*, 2nd ed. Boston, MA: Academic Press, 1990.