Online Discovery and Classification of Operational Regimes From an Ensemble of Time Series Data

Chandrachur Bhattacharya
Departments of Mechanical and Electrical Engineering,
Pennsylvania State University,
University Park, PA 16802
e-mail: chandrachur.bhattacharya@gmail.com

Asok Ray
Fellow ASME
Departments of Mechanical Engineering and Mathematics,
Pennsylvania State University,
University Park, PA 16802
e-mail: axr2@psu.edu

One of the pertinent problems in decision and control of dynamical systems is to identify the current operational regime of the physical process under consideration. To this end, there has been an upsurge in (data-driven) machine learning methods, such as symbolic time series analysis, hidden Markov modeling, and artificial neural networks, which often rely on some form of supervised learning based on preclassified data to construct the classifier. However, this approach may not be adequate for dynamical systems with a variety of operational regimes and possible anomalous/failure conditions. To address this issue, the technical brief proposes a methodology, built upon the concept of symbolic time series analysis, wherein the classifier learns to discover the patterns so that the algorithms can train themselves online while simultaneously functioning as a classifier. The efficacy of the methodology is demonstrated on time series of: (i) synthetic data from an unforced Van der Pol equation and (ii) pressure oscillation data from an experimental Rijke tube apparatus that emulates the thermoacoustics in real-life combustors where the process dynamics undergoes changes from the stable regime to an unstable regime and vice versa via transition to transient regimes. The underlying algorithms are capable of accurately learning and capturing the various regimes online in a (primarily) unsupervised manner. [DOI: 10.1115/1.4047449]

Keywords: pattern discovery and classification, symbolic dynamics, anomaly detection

1 Introduction

Large-scale dynamical systems, depending on their complexity, may exhibit a variety of operational regimes and a number of possible anomalous/failure conditions. Thus, when a control command is issued, it needs to be conditioned on both present and intended operational regimes. Accordingly, traditional classification techniques are often based on the concept of supervised learning, wherein the classifier is trained on labeled data corresponding to various regimes and anomalous characteristics of the underlying process.

Pattern recognition is a mature field of research, which has found its applications in many aspects of machine learning. Several methods [1,2] exist for identification of various classes that the observed data emanate from. Many researchers have made use of several machine learning techniques for time series classification in diverse applications. Bagnall et al. [3] have reported a comprehensive review of time series analysis for online detection, classification, and decision and control. Sugimura and Matsumoto [4] have reported feature extraction from time series to generate a decision tree for pattern classification. The concepts of reduced-order Markov modeling and probabilistic finite state automata (PFSA) have been studied for analysis of combustion instabilities [5], failure prognosis of structural materials [6] and rolling-element bearings [7], as well as usage of sensor networks for detection of moving targets [8]. Several researchers have also used deep neural networks [9] as well as recurrent neural networks for time series classification [10]. However, not all of these techniques are suitable for online detection, which requires real-time classification (e.g., by using as short a time series window as possible). Online regime detection and classification have been demonstrated by Mondal et al. [11], in the setting of hidden Markov models and by Lim and Harrison [12] in the setting of neural networks, for example.

In a vast majority of the methods mentioned previously, the models are trained offline (i.e., no simultaneous learning and classification). Furthermore, all of the above algorithms belong to the category of supervised or semisupervised learning in the sense that they need labeled data, where the labeling is done by (human) domain experts. Even when using an unsupervised technique such as K-means [2] for time series analysis [13], a human expert must decide the number of regimes (i.e., the parameter K). Other algorithms such as Gaussian mixture models are also used for learning subclasses in populations; however, for this approach to work, data from all possible classes need to be considered during offline training. These methods are flexible for learning multiple classes only if a human supervisor pre-informs the algorithm about the actual number of classes the given data needs to be divided into. In such a situation, if the algorithm comes across a hitherto unfamiliar anomaly or signal during the online classification phase, it may misclassify the operation as one of the trained regimes. Often, it is not possible to have a training dataset containing all possible events, especially in systems that may be subjected to occurrence of rare events.

This technical brief proposes a low-computational-cost methodology that is built upon a concept that adaptively learns new patterns and signal forms in an online fashion (i.e., in real-time) without the need to know the total number of classes a priori. The efficacy of the proposed methodology is demonstrated on both synthetic data from an unforced Van der Pol oscillator and experimental data of pressure time series from an electrically heated Rijke tube apparatus [11,14,15] that emulates thermoacoustic instabilities (TAI) in real-life combustors, which can lead to failure of mechanical structures, loud noise, and flow-reversal in the combustion system. From these perspectives, major contributions of the technical brief are delineated below.

- *Novelty of the algorithms*: The classifier is trained offline only on a single ensemble of time series data corresponding to an a priori known regime. Subsequently, other regimes are autonomously identified and learned as they are encountered online, which enable the algorithms to decide whether or not a block of time series belongs to one of the trained regimes.
- *Experimental validation*: The proposed algorithms have been tested and validated with time series of both synthetic data and experimental data to establish their feasibility for potential commercial applications.

2 Mathematical Theory

Before embarking on an explanation of the proposed algorithms, it is necessary to provide the background for construction of PFSA (see Sec. 2.1) [16,17] and D-Markov machines (see Sec. 2.2) [18].

Contributed by the Dynamic Systems Division of ASME for publication in the Journal of Dynamic Systems, Measurement, and Control. Manuscript received August 18, 2019; final manuscript received May 21, 2020; published online July 10, 2020. Assoc. Editor: Umesh Vaidya.

2.1 Probabilistic Finite State Automata.

Time series of the measured signal is quantized and then symbolized as a symbol string. In this process, the signal space is partitioned into a finite number of cells, where the cardinality $|\Sigma|$ of the (symbol) alphabet Σ is identically equal to the number of cells. A symbol from the alphabet Σ is assigned to each (signal) value corresponding to the cell to which it belongs [19,20]; the details are reported in Ref. [18]. Thus, a symbol is associated with a data point at a given instant of time when the value of that data point is located in the particular cell corresponding to that symbol. The following definitions, which are available in standard literature (e.g., Refs. [16] and [18]), are recalled for completeness of the technical brief.

Definition 1. *A finite state automaton (FSA) G, having a deterministic algebraic structure, is a triple (Σ, Q, δ) where:*

- *Σ is a (nonempty) finite alphabet, i.e., its cardinality $|\Sigma|$ is a positive integer.*
- *Q is a (nonempty) finite set of states, i.e., its cardinality $|Q|$ is a positive integer.*
- *$\delta : Q \times \Sigma \rightarrow Q$ is a state transition map.*

Definition 2. *A symbol block, also called a word, is a finite length string of symbols belonging to the alphabet Σ, where the length of a word $w \triangleq s_1 s_2 \cdots s_\ell$ with every $s_i \in \Sigma$ is $|w| = \ell$, and the length of the empty word ϵ is $|\epsilon| = 0$. The parameters of FSA are extended as:*

- *The set of all words, constructed from symbols in Σ and including the empty word ϵ, is denoted as $\Sigma^\star$.*
- *The set of all words, whose suffix (respectively, prefix) is the word w, is denoted as $\Sigma^\star w$ (respectively, $w\Sigma^\star$).*
- *The set of all words of (finite) length ℓ, where ℓ is a positive integer, is denoted as Σ^ℓ.*

Remark 3. A symbol string (or word) is generated from a (finite length) time series by symbolization.

Definition 4. *A PFSA K is a pair (G, π), where:*

- *The deterministic FSA G is called the underlying FSA of the PFSA K.*
- *The probability map $\pi : Q \times \Sigma \rightarrow [0, 1]$ is called the morph function (also known as symbol generation probability function) that satisfies the condition: $\sum_{\sigma \in \Sigma} \pi(q, \sigma) = 1$ for all $q \in Q$.*
- *The $(|Q| \times |\Sigma|)$ morph matrix Π, which is converted into the $(|Q||\Sigma| \times 1)$ morph vector ν to serve as a feature in the sequel, is generated by the morph function π.*

Equivalently, a PFSA is a quadruple $K = (\Sigma, Q, \delta, \pi)$.

2.2 *D*-Markov Machines.

The PFSA representation of a D-Markov machine generates symbol strings $\{s_1 s_2 \cdots s_\ell : \ell \in \mathbb{N}^+ \text{ and } s_j \in \Sigma\}$ on the underlying Markov process. In the construction of a D-Markov machine, it is assumed that the generation of the next symbol depends only on a *finite* history of at most D consecutive symbols, i.e., a symbol block of length not exceeding length D. A D-Markov machine [18] is defined as follows.

Definition 5. *A D-Markov machine [16] is a PFSA in the sense of Definition 4 and it generates symbols that solely depend on the (most recent) history of at most D consecutive symbols, where the positive integer D is called the depth of the machine. Equivalently, a D-Markov machine is a statistically stationary stochastic process $S = \cdots s_{-1} s_0 s_1 \cdots$, where the probability of occurrence of a new symbol depends only on the last consecutive (at most) D symbols,* i.e.,

$$P[s_n | \cdots s_{n-D} \cdots s_{n-1}] = P[s_n | s_{n-D} \cdots s_{n-1}] \quad (1)$$

Consequently, for $w \in \Sigma^D$ (see Definition 2), the equivalence class $\Sigma^\star w$ of all (finite length) words, whose suffix is w, is qualified to be a D-Markov state that is denoted as w.

As the proposed D-Markov algorithms need a uniform dimension of the feature vectors, the concept of state-merging [18] has not been included. There are four primary parameters in the algorithms as enumerated below:

(1) *Alphabet size* ($|\Sigma|$): Larger is the alphabet size, more distinct are the different regimes, but more training data would be needed. There are several algorithms for selection of the optimal alphabet size (e.g., Ref. [21]); but, to demonstrate efficacy of the algorithms, an alphabet size $|\Sigma| = 16$ is chosen in the technical brief for the synthetic dataset and an alphabet size $|\Sigma| = 6$ for the experimental dataset. The choice of alphabet size is dependent primarily on how closely spaced the data are from each regime, with higher alphabet sizes typically yielding better regime separability (if needed).

(2) *Partitioning method*: While there are many data partitioning techniques, maximum entropy partitioning (MEP) [18–20], which is a commonly used partitioning technique, has been chosen in this technical brief.
Depth (D) in the D-Markov machine: Higher values of the positive integer D may lead to better results at the expense of increased computational time due to larger dimension of the space and need for more training. In this technical brief, $D = 1$ has been chosen in order to keep lower word lengths and smaller PFSAs, which lead to faster training and testing.

(3) *Choice of feature*: The feature needs to be one that best captures the nature (e.g., texture) of the signal. The morph vector ν (see Definition 4) has been chosen in this technical brief as the feature, because it is not only easily computed but also captures pertinent dynamics embedded in the signal.

3 The Proposed Algorithms

To address the problem of online discovery and classification of operational regimes, the proposed algorithms are first executed upon a time series at a given time epoch, not in its entirety, but in a windowed fashion with or without an overlap. The algorithms require only a single regime to be labeled, where a (human) expert is required to label a part of the ensemble of training time series data, which corresponds to a specific regime, called the *base regime*. Typically, this regime would represent the nominal operating condition of the physical process under consideration; however, it is not restricted to be so.

The algorithms have two learning phases: the first is the (offline) learning/training phase of the base regime and the second is the (online) discovery-classification phase for learning and classifying the remaining regimes.

3.1 Learning the Base Regime.

A windowed segment of time series from the training data (e.g., corresponding to the base regime) is first analyzed. Each segment is symbolized by MEP [19] with a preset alphabet size, followed by construction of a PFSA[1] as described in Sec. 2. The boundaries of MEP are computed for every window. Training is done for assigned time series, where each time series possibly comprises of multiple windowed segments and the corresponding morph vector serves as the feature of each windowed time series, and is stored. After training, the mean of all morph vectors is taken as the centroid corresponding to the regime under consideration in the higher dimensional space, similar to what is done in K-means [1]. The Euclidean distance $||\bullet||_e$ of each of the training morph vectors from the base centroid is obtained as

[1] The PFSA codes used in this paper are developed in-house by the authors and are available at: https://github.com/Chandrachur92/PFSA

$$d_j^i = ||C_j - v_j^i||_e, \quad i = 1, \ldots, N_j \tag{2}$$

where v_j^i is the morph vector of segment i in regime j ($= 1$ for this phase), and C_j is the centroid of regime j defined as: $C_j \triangleq \frac{1}{N_j}\sum_{i=1}^{N_j} v_j^i$. It is noted that the segment number i runs from 1 to N_j, which is the number of windowed segments used to train regime j. A neighborhood of the base regime (i.e., $j = 1$) is generated about its centroid with radius ρ_1 that is computed by setting $j = 1$ in the following equation:

$$\rho_j = \frac{1}{N_j}\sum_{i=1}^{N_j} d_j^i + \gamma\sqrt{\frac{\sum_{i=1}^{N_j}\left(d_j^i\right)^2}{N_j - 1}} \tag{3}$$

where the first term on the right hand of Eq. (3) is the mean of all distances in regime j, and the second term is the standard deviation multiplied by a user-selected parameter γ. The rationale for choosing the specific structure in Eq. (3) is that the distribution of d_j^i is assumed to be nearly Gaussian, because the cumulative effects of many independent random variables in the construction of a PFSA tend to yield a Gaussian distribution. For a Gaussian distribution, the 4σ band contains almost 100% of the data and also ensures that obvious spurious outliers do not contaminate the estimate of the radius; the parameter $\gamma = 4$ is chosen in this technical brief. However, for a non-Gaussian (e.g., fat-tailed) distribution, γ should be appropriately chosen.

It is noted that, at the end of this primary training, the algorithms are only aware of the feature centroid and neighborhood corresponding only to the base regime and are unaware of any other regimes (e.g., anomalies or other operational states). At this point, the number of trained regimes is $T = 1$.

3.2 Online Discovery and Classification. The algorithms executes on a window of data, hereafter referred to as a data segment, from an unknown time series. The partitioning parameters of MEP are recomputed with the same alphabet size to construct a PFSA from the data segment and to compute the morph vector to serve as an extracted feature. If the morph vector is within the neighborhood of a known regime, the time-window is classified to belong to that regime; otherwise, the algorithms treat the data segment as belonging to a newly *discovered* regime.

The algorithms keep track of the number of segments that they receive corresponding to the new regime and store the feature vector for each data segment; it is noted that the segments need not be observed consecutively. As the algorithms *discover* a new regime, the feature vector (v_{new}^1) is taken to be the initial guess for the centroid of the new regime (C_{new}). However, there may not be enough data to estimate the radius of the neighborhood. To allow for larger initial search regions, the parameters, $\alpha_j > 1$, are defined as follows:

$$\alpha_j \triangleq \frac{\frac{1}{N_j}\sum_{i=1}^{N_j} d_j^i + \beta\sqrt{\frac{\sum_{i=1}^{N_j}\left(d_j^i\right)^2}{N_j - 1}}}{\frac{1}{N_j}\sum_{i=1}^{N_j} d_j^i - \beta\sqrt{\frac{\sum_{i=1}^{N_j}\left(d_j^i\right)^2}{N_j - 1}}}, \quad j = 1, 2, \ldots, T \tag{4}$$

where T is the number of trained regimes and the user-set parameter β determines how much of the symmetric tail of the probability density is removed. The value of the parameter $\beta = 1.5$ has been used for both the cases (i.e., synthetic data and Rijke tube data) in Sec. 4. For each trained regime, α_j is computed, and a single parameter α is obtained as the minimum of α_1 (corresponding to the base regime $j = 1$) and the average of α_j for all T trained regimes

$$\alpha \triangleq \min\left(\alpha_1, \frac{1}{T}\sum_{j=1}^{T}\alpha_j\right) \tag{5}$$

A restriction $\alpha_1 \geq (\alpha_1)_{\text{min}}$ is imposed to ensure numerical stability, where the user-set lower bound is selected in this paper as $(\alpha_1)_{\text{min}} = 1.5$.

Now, the yet unknown radius, (ρ_{new}), of a newly discovered regime is initially assigned a value computed as the mean of ρ_j's, over all T trained regimes, multiplied by the factor α (see Eq. (5)), as below:

$$\rho_{\text{new}} = \frac{\alpha}{T}\sum_{j=1}^{T}\rho_j \text{ as the first (initial) guess} \tag{6}$$

For the subsequent $(M_1 - 1)$ online training segments, (where $M_1 \geq 1$ is a user-defined parameter) the radius ρ_{new} is updated as

$$\rho_{\text{new}} \leftarrow \left(\frac{\rho_{\text{new}}N_{\text{new}} + \alpha\,||C_{\text{new}} - v_{\text{new}}^{N_{\text{new}}}||_e}{N_{\text{new}} + 1}\right) \tag{7}$$

where N_{new} is number of segments in the discovered regime; and $v_{\text{new}}^{N_{\text{new}}}$ is the observed morph vector from the N_{new}th segment that was classified to belong to the "new" discovered regime.

The procedure in Eqs. (6) and (7) allows for larger initial search regions until the algorithms yield a better estimate of the radius of the new regime. This is necessary because each regime may have a different variance and would need a different neighborhood for accurate classification. The updating of α in Eq. (5) ensures that, as the algorithms become more confident, it attempts to make less conservative (i.e., tighter) estimates of the neighborhood radius.

The next M_2 online training segments, where M_2 is also a user-set parameter, the radius ρ_{new} is updated as

$$\rho_{\text{new}} \leftarrow \frac{1}{N_{\text{new}}}\sum_{i=1}^{N_{\text{new}}} d_{\text{new}}^i + \gamma\sqrt{\frac{\sum_{i=1}^{N_{\text{new}}}\left(d_{\text{new}}^i\right)^2}{N_{\text{new}} - 1}} \tag{8}$$

For each of the above $(M_1 + M_2)$ online training segments, the centroid of the regime is shifted as $C_{\text{new}} \triangleq \frac{1}{N_{\text{new}}}\sum_{i=1}^{N_{\text{new}}} v_{\text{new}}^i$ as the mean of all observed feature vectors of that regime changes.

Once the number of training segments in a given regime reaches the set number of $M_1 + M_2$, the training for that regime is deemed completed and no further parameter updating occurs for that regime. The values of M_1 and M_2 largely determine how long the algorithms will consider a given detected regime as *untrained.* Low values of M_1 and M_2 may degrade the quality of training. On the other hand, high values of M_1, and M_2 would enhance learning in general; however, it may cause poor learning in the presence of an unknown regime (e.g., a rare event) occurring for a short duration, because of insufficient data.

In the event that a feature vector of an observed segment (v^{obs}) lies in the intersection of two or more yet untrained regimes, the segment is assigned to the regime, in which it has a higher probability of belonging to, as described below:

$$\text{classified regime} = \arg\max_j\left(\frac{||C_j - v^{\text{obs}}||_e}{\rho_j}\right) \tag{9}$$

After the classifier has seen most of the typical regimes, it can decide on one of these regimes or detect a new regime. These algorithms are capable of learning and augmenting the regime library, and the operation can be executed in real-time for simultaneous learning and classification.

The pseudo-code of Algorithm 1 describes the first phase of base regime training, which is repeatedly followed by the second phase of discovery and classification for each new segment of time series as described in the pseudo-code of Algorithm 2.

Remark 6. The proposed method is also capable of merging regimes when they are too close to each other. This task of merging is achieved by checking if the new centroid of an untrained regime lies within the neighborhoods of any trained regime. In that case, all morph vectors corresponding to the untrained regime are assigned to the trained regime and the new values of centroid and radius of the amalgamated regime is recomputed using Eqs. (4)–(8).

Algorithm 1 Base regime training from ensemble of data

```
Input: Time series dataset corresponding to the base regime
    Initialization: (#segments in base regime dataset) $N_1$; (#untrained regimes) U = 0; (#trained regimes) T = 1
    User-set parameters: $\beta$, $\gamma$, and $(\alpha_1)_{min}$
Output: Base regime information: Centroid $C_1$; Neighborhood radius $\rho_1$; (#Segments) $N_1$; Parameters $\alpha_1$ and $\alpha$
for $i = 1$ to $N_1$ do
    Symbolize time series segment $i$ and generate PFSA
    Store morph vector $v_1^i$ as the extracted feature
end for
$C_1 = \frac{1}{N_1}\sum_{i=1}^{N_1} v_1^i$
for $i = 1$ to $N_1$ do
$d_1^i = ||C_1 - v_1^i||_e$ % Euclidean norm
end for
$\rho_1 = \frac{1}{N_1}\sum_{i=1}^{N_1} d_1^i + \gamma\sqrt{\frac{\sum_{i=1}^{N_1}(d_1^i)^2}{N_1 - 1}}$
$\alpha_1 = \max\left((\alpha_1)_{min}, \frac{\frac{1}{N_1}\sum_{i=1}^{N_1} d_1^i + \beta\sqrt{\frac{\sum_{i=1}^{N_1}(d_1^i)^2}{N_1-1}}}{\frac{1}{N_1}\sum_{i=1}^{N_1} d_1^i - \beta\sqrt{\frac{\sum_{i=1}^{N_1}(d_1^i)^2}{N_1-1}}}\right)$
$\alpha = \alpha_1$
return $C_1, \rho_1, N_1, \alpha_1, \alpha$
```

Algorithm 2 Regime identification from time series window

```
Input: Time series data segment windowed from the process
    Initialization: (# Untrained regimes) $U$; (# Trained regimes) $T$; Parameter $\alpha_1$ generated in Algorithm 1; Parameter $\alpha$ generated in Algorithm 1 and possibly updated in a previous execution of Algorithm 2; (# Segments) $N_j$, Centroid $C_j$, and neighborhood radius $\rho_j$, $j = 1, \ldots, U + T$
User-set parameters: $\beta$, $\gamma$, $M_1$ and $M_2$
Output: Regime that the time series data belong to (i.e., the classified regime); (# Untrained regimes) $U$; (# Trained regimes) $T$; Centroid $C_j$ and neighborhood radius $\rho_j, j = 1, \ldots, U + T$; Parameter $\alpha$
Symbolize time series and generate $D$-Markov Machine
Extract morph vector $v^{new}$ to serve as a feature
for $j = 1$ to $(T + U)$ do
    if $||C_j - v||_e \leq \rho_j$ then
        if $N_j > (M_1 + M_2)$ then
            Classified regime = $j$ % as trained regime $j$
    else
            $C_j = \frac{1}{N_j+1}\left(\sum_{i=1}^{N_j} v_j^i + v^{new}\right)$
            $d_j^{N_j+1} = ||C_j - v^{new}||_e$ and $v_j^{N_j+1} = v^{new}$
            for $i = 1$ to $N_j$ do
                $d_j^i = ||C_j - v_j^i||_e$ % Euclidean norm
            end for
                if $N_j \leq M_1$ then
                $\rho_j \leftarrow \frac{\rho_j N_j + \alpha d_j^{N_j+1}}{N_j+1}$
                $N_j = N_j + 1$
                else if $M_1 < N_j < M_2$ then
                    $N_j = N_j + 1$
                    $\rho_j = \frac{1}{N_j}\sum_{i=1}^{N_j} d_j^i + \gamma\sqrt{\frac{\sum_{i=1}^{N_j}(d_j^i)^2}{N_j - 1}}$
                end if
                if $N_j \geq M_1 + M_2$ then
                $T \leftarrow T + 1$ %#trained regime increased by 1
                $U \leftarrow U - 1$ %#untrained regime decreased by 1
                for $\tilde{j} = 1$ to $T$ do
                $\alpha_{\tilde{j}} = \frac{\frac{1}{N_{\tilde{j}}}\sum_{i=1}^{N_{\tilde{j}}} d_{\tilde{j}}^i + \beta\sqrt{\frac{\sum_{i=1}^{N_{\tilde{j}}}(d_{\tilde{j}}^i)^2}{N_{\tilde{j}} - 1}}}{\frac{1}{N_{\tilde{j}}}\sum_{i=1}^{N_{\tilde{j}}} d_{\tilde{j}}^i - \beta\sqrt{\frac{\sum_{i=1}^{N_{\tilde{j}}}(d_{\tilde{j}}^i)^2}{N_{\tilde{j}} - 1}}}$
                end for
                $\alpha = min\left(\alpha_1, \frac{1}{T}\sum_{\tilde{j}=1}^{T}\alpha_{\tilde{j}}\right)$
            end if
            Classified regime = $j$ % as untrained regime $j$
        end if
  else
        $N_{T+U+1} = 1$ Z$N_{T+U+1} = 1$; $C_{T+U+1} = v$; $\rho_{T+U+1} = \alpha \times \frac{1}{T}\sum_{j=1}^{T}\rho_j$
        Classified regime = $T + U + 1$ % as a new regime
        $U = U + 1$ %#untrained regime increased by 1
        end if
end for
return Classified regime
```

4 Results and Discussion

This section addresses testing and validation of the proposed algorithms, where time series data have been obtained from two sources. The first source is synthetically designed using the (unforced) Van der Pol equation [22], wherein the change-points are very firmly defined. The second source generates an ensemble of experimental data of pressure oscillations, collected from an electrically heated Rijke tube apparatus [11,23], where the operational regimes of the physical process are changed from stable to unstable and vice versa through various stages of transience.

4.1 Synthetic Data.

The synthetic time series consist of data points of length 50,000 corresponding to an observation period of 10,000 s at a sampling frequency of 5 Hz. This ensemble of datasets is constructed from the standard unforced Van der Pol oscillator whose governing equation is given below:

$$\frac{d^2y}{dt^2} + \mu\left(y^2 - 1\right)\frac{dy}{dt} + y = 0 \quad \text{for} \quad \mu > 0 \qquad (10)$$

The generated dataset comprises of four regimes, each corresponding to a distinct value of the damping parameter μ, where μ takes the values of 0.1, 1.0, 1.5, and 2.5. The change-points have been randomly selected for each time series that has at most nine

Table 1 Confusion matrix: regime classification using synthetic data

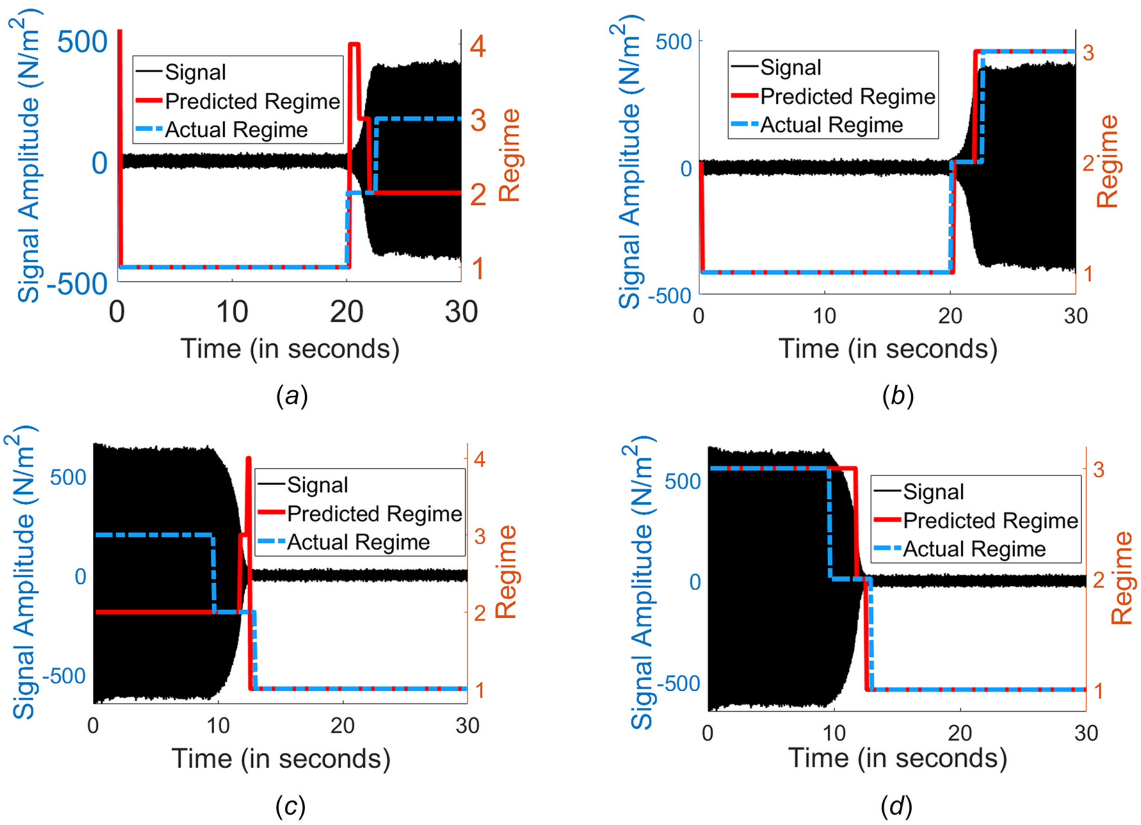
	Classified As				
True regime	Regime 1	Regime 2	Regime 3	Regime 4	Regime 5 (error)
Regime 1	89.14%	0.1%	3.66%	0.1%	7.00%
Regime 2	0.19%	84.27%	7.18%	0.23%	8.13%
Regime 3	0.06%	0.67%	93.11%	0.34%	5.82%
Regime 4	0.11%	0.11%	2.75%	93.95%	3.08%

sections, where each section has any one of the above regimes randomly assigned to it. Sixty such time series have been generated. This ensemble of time series serves as the ground truth for testing the proposed algorithms, because it is generated by a known algorithm with specified regimes and the associated change-points. In order to verify whether an algorithm is working as expected, the ensemble of time series is split into three parts: (i) the first part (30% of total data) for training the base regime, which is randomly chosen as any one of the above-mentioned four regimes, (ii) the second part (60% of total data) to learn the other regimes online, and (iii) the third part (the remaining 10% of total data) to test whether the regimes have been learned correctly by verification against the ground truth.

For conversion of the time series into online data, individual datasets are windowed, with the windowing occurring every 100 data points and each window having a size of 1000 data points. In this analysis, the PFSA has an alphabet size $|\Sigma| = 16$ and depth $D = 1$. Values of the parameters M_1 and M_2 have been chosen as 50 and 250, respectively. The average error has been computed over 20 different runs, where each of these runs is executed on a newly generated dataset, a new randomly selected base regime, and newly trained algorithm centroids and radii.

The results are presented in Table 1 as a confusion matrix, where the failure to correctly identify a classified regime (i.e., the algorithms erroneously deciding that the given data belongs to none of the four actual regimes) is called an error (regime 5), with the algorithms having an overall average error-rate below 10%. This is considered to be a low error because the assigned regimes are very close to each other in the feature space. While the algorithms could have identified more than four regimes in the first few trials, it has been capable of refining the results to the four true regimes by merging of regimes in subsequent runs.

4.2 Experimental Rijke Tube Data. This subsection demonstrates the efficacy of the proposed methodology by showing its capability to perform on an emulated real-life problem. To this end, the underlying algorithms have been tested and validated on experimental data of pressure time series from an electrically heated Rijke tube apparatus [11,23]. The Rijke tube [14] is a commonly used experimental apparatus that emulates the phenomena of TAI [15] encountered in real-life gas-turbine combustors. It is known that occurrence of TAI can be detrimental to the safe operation and health of a combustor. Thus, there is a need to be able to detect and identify the regime that the combustor is presently operating at and identify the transient regime, wherein the combustor is expected to move either from a stable regime to an unstable regime or vice versa. Regime identification is also the first step to perform monitoring and control on the combustor system. Several techniques have been suggested for detection of TAI using different algorithms (e.g., Refs. [5], [11], [24], and [25] to name a few). However, as mentioned in Sec. 1, all these techniques need the assistance of a human expert



(*a*)

(*b*)

(*c*)

(*d*)

Fig. 1 Detection and classification from Rijke tube data: (*a*) time series 1 with separate transient regimes, (*b*) time series 1 with amalgamated transient regimes, (*c*) time series 2 with separate transient regimes, and (*d*) time series 2 with amalgamated transient regimes

Table 2 Confusion matrix: regime classification using rijke tube data

Truly	Classified stable	Classified transient	Classified unstable
Stable	99.73%	0.27%	0%
Transient	11.21%	57.76%	31.03%
Unstable	0%	0%	100%

for classifying the training samples and to create appropriate labels.

The Rijke tube apparatus, described by Mondal et al. [11], has been used to generate transient signals described above. In this work, the detection algorithms are trained for the nominal stable condition as the base regime. Using the methodology described above, other transient and unstable regimes are identified as shown in Fig. 1.

The ensemble of data consists of a total 40 time series that were collected at a sampling rate of 8192 Hz and subsequently filtered with a cut-off frequency of 40 Hz [11]. Then, data windowing has been done at ~10 Hz, i.e., the length of each data window is ~800. The time series is split into three groups, 20% of total data for training the base regime (i.e., regime 1), 70% for online identification, and the remaining 10% for testing.

The *D*-Markov machines (see Sec. 2.2) have been constructed from the ensemble of above windowed data with an alphabet size $|\Sigma| = 6$ and depth $D = 1$. The parameters M_1 and M_2 (see Sec. 3.2) have been chosen as 20 and 180, respectively; the rationale for choosing these low values is that transience occurs over a very short time span and so the algorithms need to train quickly (e.g., within 36 datasets), and the reduced values of M_1 and M_2 achieve the goal. In this process, there is no available ground truth as to when the transient state begins and ends. Various researchers have used different approximations for deciding the change-points for similar events. For example, a cumulative sum approach has been commonly used to approximately determine the change-points ([11]), for which the regimes identified by the algorithm have been cross-checked.

The four plates in Fig. 1 show regime predictions for two sets of time series of experimental data. The two left-hand plates, (*a*) and (*c*), in Fig. 1 show the actual regimes (i.e., the a priori estimated ground truth) for the time series 1 and 2, respectively. The objective here is to identify the change of regime from stable to transient and then to unstable. In this study, the algorithms also identify a transient regime that is not apparent otherwise. In the right-hand images in the plates, (*b*) and (*d*), of Fig. 1, the two predicted transient regimes have been amalgamated into a single transient regime for ease of comparison with the estimated ground truth, where it is seen that the ground truth and the predictions match closely, but not perfectly, because the ground truth itself may be flawed. However, the algorithms are capable of correctly detecting the changes and still have good accuracy, especially in identifying the learned unstable regime. This is seen in the confusion matrix (computed as an average over four test series across 20 repeated runs) in Table 2. Figure 1 also shows that the algorithms identify the most important change (i.e., from stable to transient states nearly perfectly). This capability is very useful for initiating a control action to prevent instability, because the transient regime is where the active controller should quench the pressure oscillations as early as possible.

5 Summary, Conclusions, and Future Work

This technical brief has developed a (partially) unsupervised methodology for online identification and classification of operational regimes in dynamical systems. An objective here is to discover new regimes in an online fashion, based on the a priori knowledge of only a single selected regime. Although the proposed method constructs centroids and regime-radii (in the feature space) representing each regime, it is not iterative like the standard K-means [1,2] and thus can be used for online discovery. As a new time series from an unknown regime is observed, the algorithm is updated online (without the need to iterate) until the regime model converges. The underlying algorithms have been tested and validated with synthetic data from an unforced Van der Pol equation and also with experimental data from an electrically heated Rijke tube apparatus that emulates pressure oscillations in real-life combustors.

While there are many areas of theoretical and experimental research, which must be investigated before the proposed methodology can be implemented in real-life applications, the following topics of future research are suggested.

- Enhancement of the algorithms to accommodate smaller data window lengths, i.e., faster detection and identification of regimes.
- Modification of the algorithms for guaranteed robustness to noise and uncertainties.
- Usage of other machine learning tools (e.g., hidden Markov models, Gaussian process modeling or neural networks [1,2]) as alternatives to symbolic dynamics.
- Verification of the algorithms' efficacy to be able to classify other standard dynamical systems, especially chaotic systems like the forced Van der Pol oscillator and the Rössler dynamical system.
- Autonomous learning of the algorithms' hyper-parameters, namely $|\Sigma|$, γ, β, $(\alpha_1)_{\min}$, $M_{1,}$ and M_2.
- Further investigation within a sensor fusion framework to improve identification of unforeseen regimes that may emerge under rare events.

Acknowledgment

The authors are grateful to Dr. Sudeepta Mondal at Penn State, who kindly provided the experimental data. The reported work has been supported in part by the U.S. Air Force Office of Scientific Research (AFOSR) under Grant Nos. FA9550-15-1-0400 and FA9550-18-1-0135 in the area of dynamic data-driven application systems (DDDAS). The first author is also thankful to Indo-U.S. Science and Technology Forum (IUSSTF) for granting him the Research Internship for Science and Engineering (RISE) scholarship. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

Funding Data

- U.S. Air Force Office of Scientific Research (AFOSR) (Grant Nos. FA9550-15-1-0400 and 558 FA9550-18-1-0135; Funder ID: 10.13039/100000181).

References

[1] Bishop, C., 2006, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag, Berlin.

[2] Murphy, K., 2012, *Machine Learning: A Probabilistic Perspective*, The MIT Press, Cambridge, MA.

[3] Bagnall, A., Lines, J., Bostrom, A., Large, J., and Keogh, E., 2017, "The Great Time Series Classification Bake Off: A Review and Experimental Evaluation of Recent Algorithmic Advances," Data Min. Knowl. Discovery, **31**(3), pp. 606–660.

[4] Sugimura, H., and Matsumoto, K., 2011, "Classification System for Time Series Data Based on Feature Pattern Extraction," IEEE International Conference on Systems, Man, and Cybernetics, Anchorage, AK, Oct. 9–12, pp. 1340–1345.

[5] Sarkar, S., Chakravarthy, S., Ramanan, V., and Ray, A., 2016, "Dynamic Data-Driven Prediction of Instability in a Swirl-Stabilized Combustor," Int. J. Spray Combust. Dyn., **8**(4), pp. 235–253.

[6] Ghalyan, N., and Ray, A., 2020, "Symbolic Time Series Analysis for Anomaly Detection in Measure-Invariant Ergodic Systems," ASME J. Dyn. Syst. Meas. Control, **142**(6), p. 061003.

[7] Jha, D., Virani, N., Reimann, J., Srivastav, A., and Ray, A., 2018, "Symbolic Analysis-Based Reduced Order Markov Modeling of Time Series Data," Signal Process., **149**, pp. 68–81.

[8] Li, Y., Jha, D. K., Ray, A., and Wettergren, T. A., 2018, "Information-Theoretic Performance Analysis of Sensor Networks Via Markov Modeling of Time Series Data," IEEE Trans. Cybern., **48**(6), pp. 1898–1909.

[9] Wang, Z., Yan, W., and Oates, T., 2017, "Time Series Classification From Scratch With Deep Neural Networks: A Strong Baseline," International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, May 14–19, pp. 1578–1585.

[10] Husken, M., and Stagge, P., 2003, "Recurrent Neural Networks for Time Series Classification," Neurocomputing, **50**, pp. 223–235.

[11] Mondal, S., Ghalyan, N., Ray, A., and Mukhopadhyay, A., 2019, "Early Detection of Thermoacoustic Instabilities Using Hidden Markov Models," Combust. Sci. Technol., **191**(8), pp. 1309–1328.

[12] Lim, C., and Harrison, R., 2003, "Online Pattern Classification With Multiple Neural Network Systems: An Experimental Study," IEEE Trans. Syst., Man, Cybern., Part C, **33**(2), pp. 235–247.

[13] Huang, X., Ye, Y., Xiong, L., Lau, R., Jiang, N., and Wang, S., 2016, "Time Series k-Means: A New k-Means Type Smooth Subspace Clustering for Time Series Data," Inf. Sci., **367–368**, pp. 1–13.

[14] Rijke, P. L., 1859, "Notiz Über Eine Neue Art, Die in Einer an Beiden Enden Offenen Röhre Enthaltene Luft in Schwingungen zu Versetzen," Ann. Phys., **183**(6), pp. 339–343.

[15] Matveev, K., 2003, "Thermoacoustic Instabilities in the Rijke Tube: Experiments and Modeling," Ph.D. thesis, California Institute of Technology, Pasadena, CA.

[16] Ray, A., 2004, "Symbolic Dynamic Analysis of Complex Systems for Anomaly Detection," Signal Process., **84**(7), pp. 1115–1130.

[17] Gupta, S., and Ray, A., 2007, "Symbolic Dynamics Filtering for Data-Driven Pattern Recognition," *Pattern Recognition: Theory and Application*, Nova Science Publishers, Hauppage, NY, pp. 17–71.

[18] Mukherjee, K., and Ray, A., 2014, "State Splitting and Merging in Probabilistic Finite State Automata for Signal Representation and Analysis," Signal Process., **104**, pp. 105–119.

[19] Rajagopalan, V., and Ray, A., 2006, "Symbolic Time Series Analysis Via Wavelet-Based Partitioning," Signal Process., **86**(11), pp. 3309–3320.

[20] Subbu, A., and Ray, A., 2008, "Space Partitioning Via Hilbert Transform for Symbolic Time Series Analysis," Appl. Phys. Lett., **92**(8), p. 084107.

[21] Sarkar, S., Chattopdhyay, P., and Ray, A., 2016, "Symbolization of Dynamic Data-Driven Systems for Signal Representation," Signal, Image Video Process., **10**(8), pp. 1535–1542.

[22] Cartwbight, M. L., 1960, "Balthazar Van Der Pol," J. London Math. Soc., **s1-35**(3), pp. 367–376.

[23] Bhattacharya, C., Mondal, S., Ray, A., and Mukhopadhyay, A., 2020, "Reduced-Order Modeling of Thermoacoustic Instabilities in a Two-Heater Rijke Tube," Combust. Theory and Model., **23**(3), pp. 530–548.

[24] Gotoda, H., Amano, M., Miyano, T., Ikawa, T., Maki, K., and Tachibana, S., 2012, "Characterization of Complexities in Combustion Instability in a Lean Premixed Gas-Turbine Model Combustor," Chaos, **22**(4), p. 043128.

[25] Nair, V., and Sujith, R., 2014, "Multifractality in Combustion Noise: Predicting an Impending Combustion Instability," J. Fluid Mech., **747**, pp. 635–655.