# Attack Detection for Securing Cyber Physical Systems

Weizhong Yan, *Senior Member, IEEE*, Lalit K. Mestha, *Fellow, IEEE*, and Masoud Abbaszadeh, *Senior Member, IEEE*

***Abstract*—Cyber-physical systems (CPSs) security has become a critical research topic as more and more CPS applications are making increasing impacts in diverse industrial sectors. Due to the tight interaction between cyber and physical components, CPS security requires a different strategy from the traditional information technology (IT) security. In this paper, we propose a machine learning-based attack detection (AD) scheme, as part of our overall CPS security strategies. The proposed scheme performs AD at the physical layer by modeling and monitoring physics or physical behavior of the physical asset or process. In developing the proposed AD scheme, we devote our efforts on intelligently deriving salient signatures or features out of the large number of noisy physical measurements by leveraging physical knowledge and using advanced machine learning techniques. Such derived features not only capture the physical relationships among the measurements but also have more discriminant power in distinguishing normal and attack activities. In our experimental study for demonstrating the effectiveness of the proposed AD scheme, we consider heavy-duty gas turbines of combined cycle power plants as the CPS application. Using the data from both the high-fidelity simulation and several real plants, we demonstrate that our proposed AD scheme is effective in early detection of attacks or malicious activities.**

***Index Terms*—Attack detection (AD), cyber security, cyber-physical system (CPS), feature engineering, gas turbines, machine learning.**

## I. Introduction

CYBER-PHYSICAL systems (CPSs) are an integral system featuring strong interactions between its cyber (e.g., networks and computation) and physical components [1]. CPS applications have been making great business impacts in various industrial sectors, such as energy, transportation, healthcare, and manufacturing. With the advent of Internet of Things (IoT), more and more devices with security vulnerabilities are linked to CPS, which makes CPS vulnerable to adversary attacks. In recent years, we have been experiencing an increasing number of CPS attack incidents, especially since the Stuxnet attack in 2010 [2]. The CPS attacks, if not detected quickly and mitigated properly, can have enormous consequences, e.g., equipment damage, financial losses, and public safety. Thus, CPS security is of critical importance. Securing CPS is also challenging because of: 1) the heterogeneity of components; 2) the complexity of cyber-physical interactions; and 3) the complexity of attack surfaces (attackers can target cyber, physical or both) [3].

Attack detection (AD) is one of the important strategies for securing CPS from malicious attacks. By detecting malicious behaviors and attacks early, proper counter-attack measures and mitigation actions can be taken to minimize or prevent their impacts. Traditional intrusion detection systems (IDSs), primarily designed for conventional information technology (IT) systems, are not enough for CPS since they do not take into account the physical side of CPS. A new strategy for CPS AD is to perform AD at the physical/application layer to provide the last line of defense in case that attacks penetrate through the IT protection layer and reach the physical space to "hijack" the physical system. By modeling and monitoring the "physics" or physical properties of the physical asset or process, the so-called physical-domain AD (pdAD) techniques have proven to be more effective in securing CPS from malicious attacks [4].

Properly modeling the physical behavior, or specifically physical relationships among physical measurements, is the key to success of the pdAD techniques. State space models, such as Kalman filter, have been the predominant techniques used by the control community as a means of modeling dynamic systems. State space methods have also been used for CPS AD, for example, [5]. One issue with the state space model-based AD is that, for complex CPS, the state space models may have difficulty in achieving the desired model accuracy. Another issue with the state space model-based AD is that they may not be effective in detecting stealthy attacks [6].

More recently, machine learning has been adopted for pdAD for CPS applications [7]. Machine learning-based AD methods can be either semisupervised or supervised. In semi-supervised setting, the "normal behavior," i.e., physical relationship among the physical measurements and control commands, under attack-free condition, is modeled first using machine learning modeling techniques and attack is declared if the model behavior deviates from the normal behavior. Semi-supervised AD methods are also called "anomaly-based detection." In supervised setting, the AD problem is formulated

Manuscript received April 15, 2019; accepted May 15, 2019. Date of publication June 3, 2019; date of current version October 8, 2019. This work was supported in part by the United States Department of Energy Cyber-Security for Energy Delivery Systems Research and Development Program awarded in 2016 under Contract DEOE0000833. *(Corresponding author: Weizhong Yan.)*

W. Yan is with AI and Machine Learning, GE Global Research Center, Niskayuna, NY 12309 USA (e-mail: yan@ge.com).

L. K. Mestha was with GE Global Research Center, Niskayuna, NY 12309 USA. He is now with KinetiCor, Inc., San Diego, CA 92131 USA (e-mail: lalit.mestha@gmail.com).

M. Abbaszadeh is with Controls and Optimization, GE Global Research Center, Niskayuna, NY 12309 USA (e-mail: abbaszadeh@ge.com).

Digital Object Identifier 10.1109/JIOT.2019.2919635

as a binary classification problem that is to classify normal versus attack directly based on the available physical measurements. Several studies have shown that machine learning-based methods often outperform state estimation-based methods in many CPS security applications [8], [9]. More recently, Mestha *et al.* [10] documented a novel cyber-attack accommodation algorithm by estimating the true operational states of the system with new boundary and performance constrained resilient estimators while the system is continuously operating and is under attack. The approach is based on combining data driven machine learning and physics-based domain knowledge with traditional resilient estimation.

In this paper, we propose a new machine learning-based AD scheme. The proposed AD scheme belongs to the pdAD techniques but has a key differentiation. Unlike other AD methods that directly work on the physical measurements, the proposed AD scheme uses features as the input to its detection model, where the features or fingerprints are intelligently derived from the noisy measurements by leveraging physical/domain knowledge and using advanced machine learning techniques, including deep learning. Also, the proposed AD scheme differs from other existing machine learning-based detection methods in that it adopts the extreme learning machines (ELMs), an advanced machine learning technique [11], as the supervised machine learning attack detector. For demonstration purpose, in this paper the heavy-duty gas turbines of combined cycle power plants are considered as the CPS application and the proposed AD scheme is applied specifically for gas turbine AD. The proposed AD scheme itself, however, is general and can be applied to other CPS security applications.

The main contributions of this paper include the following.

1) Proposing a comprehensive feature generation scheme that utilizes multidisciplinary techniques, e.g., statistical, physical/domain knowledge, and deep learning, for identifying salient features that better capture complex spatio-temporal relationships of physical assets.
2) Adoption of ELMs, an advanced machine learning technique, to CPS security applications, which has not been done before, to the best of our knowledge.
3) Application of AD to a new CPS security application—heavy-duty gas turbines of power plants, which has never done before, to the best of our knowledge. Power plants are critical infrastructures and hence, an important CPS application for nations energy security.
4) Validation of the proposed AD scheme by using data from the industrial-scale, high-fidelity simulation, and real-world power plants, and by comparing it against other machine learning-based detection methods.

The reminder of this paper is organized as follows. Section II reviews relative work on CPSs' AD. The proposed methodology is described in detail in Section III. Section IV presents our experimental study and its results, while Section V concludes this paper.

## II. Related Work

With the growing interest in CPS security, there have been an increasing number of publications on CPS security in recent years. While some of these publications, for example, Humayed and Luo [3] and He and Yan [12], focused on general discussions of challenges and research directions of CPS security, others, e.g., Mitchell and Chen [13] and Han *et al.* [14], were on the specific topics of intrusion or AD of CPS.

Out of the different intrusion detection methods discussed in the literature, a majority of them have focused on cyber domain attacks, where communication network packages were the main source of information for AD. For the pdAD methods concerned in this paper, Urbina *et al.* [4] surveyed different physics-based AD methods for control systems. Most of these physics-based AD methods involve modeling the normal behavior of the physical system and declaring attack if the model behavior deviates from the normal behavior. This group of methods is often called "anomaly-based AD." State space modeling techniques have been popularly used for modeling the CPS, especially the industrial control systems (ICSs), (e.g., [5], [15], and [16]).

Recently, we have seen both semisupervised and supervised machine learning techniques being adopted for CPS AD. For semi-supervised AD, for example, Paridari *et al.* [8] used one-class support vector machines (SVMs) for AD of industrial control systems. Other efforts on semi-supervised AD methods include [17]–[19], where different deep learning models, e.g., CNN, RNN, and DNN were explored for CPS security applications.

Supervised machine learning AD methods have been more popularly used for CPS security. Ozay *et al.* [7] developed a physical layer AD framework for detecting false data injection attacks in smart grids. In their framework, diverse machine learning techniques, including neural networks, SVM, ensemble learning, and online learning strategies, were investigated. Using various IEEE test systems, they demonstrated that machine learning algorithms can do better than the state vector estimation methods in detecting the attack. Wallace *et al.* [20] proposed a cyber event detection system for power grid following a context specific approach. It derived cyber conclusions by utilizing the physical knowledge of the power system. Specifically, it extracted information contained within historical or contingency power system states using principal component analysis and used a Naïve-Bayes classifier on the calculated Hotelling's t-squared metric of the principal components. More recently, in the work by Wang *et al.* [21], the margin setting algorithm (MSA) was proposed for detecting false data injection attacks in smart grids. Mestha *et al.* [10] used a classifier based on SVMs both for supervised and semi-supervised learning along with a Gaussian kernel to cover the nonlinearity on the feature space. Cyberattack on physical systems is signal corruption induced by an adversarial agent. Anubi *et al.* [22] formulated the problem of signal reconstruction in the presence of cyber-attacks as a constrained optimization problem by merging developments in the field of machine learning and estimation theories. Abbaszadeh *et al.* [23] investigated using feature-based dynamic ensemble forecasting method for anomaly forecasting and early warning generation in industrial control systems.

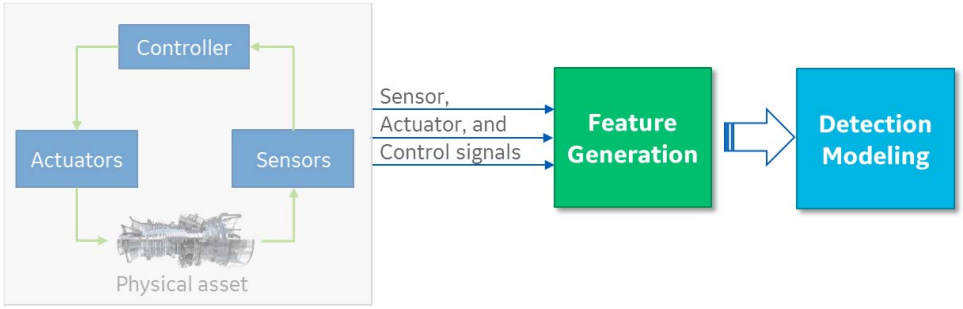


Fig. 1. High-level block diagram of the proposed AD scheme.

In terms of CPS security applications where AD has been applied, smart grids and industrial control systems probably are the dominant applications widely discussed in various publications [24]–[27]. For smart grids applications, as more intelligent sensors are exposed to cyber-attacks [28], cyber incidents may lead to large-scale blackout of power systems [29]. Other CPS applications with less amount of research effort include medical devices [30], smart manufacturing [31], [32], smart homes [33], and smart cars [3]. CPS security of power plant systems, e.g., gas turbines, generators, etc., has rarely been done. In this paper, we address the CPS security of heavy-duty gas turbines of power plants by introducing a new machine learning-based AD scheme that uses advanced machine learning for modeling and monitoring the physical behavior of assets. To the best of our knowledge, machine learning-based AD using a large number of sophisticated features extracted and learned from the physical measurements has not been done before.

Feature engineering—feature extraction and feature dimensionality reduction—is an important subfield in machine learning. Traditionally feature engineering is a labor-intensive, manual effort; and linear transformation, such as PCA, is the dominant approach for feature dimensionality reduction. In recent years, using deep learning to learn features, i.e., representation learning [34], has attracted a lot of research attention. For CPS security applications concerned in this paper, however, feature engineering has not been actively studied.

## III. Proposed Machine Learning-Based Attack Detection

Our AD scheme proposed in this paper is a physical-domain approach, that is, it performs AD at the physical layer by modeling and monitoring the physics or physical behavior of the physical asset or process, where modeling and monitoring are performed using advanced machine learning techniques. Since the number of physical measurements available for modeling can be overwhelmingly large as CPS become increasingly complex, modeling, and monitoring the system based on the physical measurements become challenging. Our effort in developing the proposed AD scheme focuses on deriving salient signatures or features from the noisy measurements and ensuring that the generated features not only capture the complex relationships among the measurements extremely well, but also more importantly, have more discriminant power in distinguishing normal operation and attack events, thus enabling us to achieve more accurate and robust detection performance. The overall structure of the proposed AD scheme is shown in Fig. 1. We will provide detailed descriptions of the two critical components, *feature generation* and *AD modeling*, of the proposed AD scheme in the following two sections, respectively.

### A. Feature Generation

Feature generation or feature engineering in general is an important process in developing predictive analytical solutions. In literature, there are numerous feature extraction methods available, ranging from traditional statistics-based to modern deep representation learning [35]. In this paper, for CPS AD, we focus on generating features that can better capture the physics of the CPS physical asset and are better in discriminating attacks from normal activities. While our features can be any signatures calculated from any number of physical measurements, for the proposed AD scheme we propose using three categories of features, namely, statistics-based, physics-based, and learning-based, to capture both spatial and temporal effects of the physical system. Spatially we calculate features on individual (univariate) and multiple (multivariate) measurements, respectively. To capture the temporal effects or dynamics of the underlying system, we perform our feature calculations over the sliding window (sliding over time). Let us assume we have $n$ physical measurements, $s^{(1)}, s^{(2)}, \ldots, s^{(n)}$, covering sensor measurements, actuator measurements, and potentially time-varying control parameters, and the window width for the sliding window is $w$.

*1) Univariate-Based Features:* For each individual measurement, $s^{(i)}$, its windowed segment of measurements at time $t$ is $m_t^{(i)} = s_{t-w}^{(i)}, s_{t-w+1}^{(i)}, \ldots, s_t^{(i)}$. Several statistical descriptors can be calculated for this segment of measurements, $m_t^{(i)}$, for example

$$f_1^{(i)} = \text{median}\left(m_t^{(i)}\right) \tag{1}$$

$$f_2^{(i)} = \text{std}\left(m_t^{(i)}\right) \tag{2}$$

$$f_3^{(i)} = \max\left(m_t^{(i)}\right) \tag{3}$$

$$f_4^{(i)} = \max\left(m_t^{(i)}\right) - \min\left(m_t^{(i)}\right) \tag{4}$$

$$f_5^{(i)} = s_t^{(i)}. \tag{5}$$

In addition, for each measurement we also calculate the maximum rate-of-change of the segment of measurements, $m_t^{(i)}$. That is,

$$f_6^{(i)} = \max\Big(\text{abs}\Big(\text{diff}\Big(m_t^{(i)}\Big)\Big)\Big). \tag{6}$$

*2) Multivariate-Based Features:* To capture the relationships among many variables, we calculate three groups of features using multiple variables. The first group of the multivariate-based features are to capture the relations between pairs of measurements, which are either defined by domain experts or learned from the data. The relations can be, for example, the difference or the ratio of two measurements, and can also be covariance of the two measurements. The number of features resulted in this group is equal to the number of relations defined.

The second group of features are the residuals of the physical models, that is, the differences between the true measurements and the models' predictions. The number of physical models, as well as the input and output of each of the physical models, are system-dependent and predefined by domain experts. Thus, we refer this group of features as "physical model-based." The physical models can be built by using the first principal or data-driven methods. Assume $i$th model has $n$ inputs, $x^{(i)} \in \Re^n$, and $m$ outputs, $y^{(i)} \in \Re^m$. At each time stamp (sample), this model will give us $m$ residuals, $R_j^{(i)} = |y_j^{(i)} - \overline{y}_j^{(i)}|,\ j = 1, 2, \ldots, m$, where $\overline{y}^{(i)} \in \Re^m$ is the model predictions. We can directly use these $m$ residuals as the features. Alternatively, for each of the residuals, we can calculate statistics (e.g., mean and standard deviation) of the residuals over the sliding window, $w$, and use the calculated statistics as the features.

The third group of multivariate-based features are those directly learned from the multiple measurements. We thus call this group of features as the "learned features" and they are completely data-driven. In recent years, feature learning, also referred to "representation learning," becomes a hot research topic as deep learning advances and becomes increasingly popular [34]. Deep representation learning employs deep learning architecture for learning features. By stacking up multiple layers of shallow learning blocks, higher layer features learned from lower layer features represent more abstract aspects of the data, and thus can be more informative and more robust to variations [36]. For learning features from physical measurements of CPS, in this paper, we propose using stacked denoising autoencoder (SDAE) as the deep learning architecture. DAE is a variant form of the classic autoencoders (AEs) [37]. Our previous studies showed that DAE is robust and effective for applications with noise sensor measurement data [38].

An AE, in its basic form, has two parts: an encoder and a decoder. The encoder is a function that maps an input $x \in \Re^{d_x}$ to hidden representation $h(x) \in \Re^{d_h}$, that is, $h(x) = s_f(Wx + b_h)$, where $s_f$ is a nonlinear activation function, typically a logistic sigmoid function. The decoder function maps hidden representation $h$ back to a reconstruction $y = s_g(W'h + b_y)$, where $s_g$ is the decoder's activation function, typically either the identity function (yielding linear reconstruction) or a sigmoid function.

AE training involves finding parameters $\theta = \{W, b_h, b_y\}$ that minimize the reconstruction error on a training set of examples, $D$, that is: $\text{argmin}_\theta(\sum_{x\in D} \|x - y\|^2)$.

To avoid trivial hidden representations, certain regularization mechanisms are needed. Denoising AE (DAE) is one form of such regularization, which involves corrupting input $x$ during training the AE, more specifically, corrupting the input $x$ in the encoding step, but still reconstructing the clean version of $x$ in the decoding step. By advocating denoising as part of the training criteria, it ensures the extracted features to have better representation capabilities.

With DAE being defined as above, we can stack up individual DAEs to form a deep network called SDAE, where the outputs of the hidden neurons at the lower layer of DAE will be the input to the upper layer of DAE (see [37] for details). The hidden representations at the last layer of the SDAE can be used as the learned features.

Concatenating all of the calculated features, that is, the univariate features, the physical model residual features, and the learned features, gives us the final feature set, which is used as the input to our AD model.

### B. Detection Modeling

Detection modeling is the process of creating a mapping function for mapping features to decisions (normal or attack). Putting it in the machine learning terminology, detection is a classification task. In literature, there are many machine learning-based detection methods available and those methods are broadly categorized in two groups, supervised and semisupervised. While supervised methods require both normal and attack samples, semi-supervised methods work on normal data only. It has also shown that supervised methods generally perform better than semi-supervised ones [39]. In this paper, we adopt supervised classification methods as our detection model. Specifically, we use the ELM as the detection model, due to several unique characteristics associated with it. ELM is a special type of feed-forward neural networks introduced by Huang *et al.* [40]. Unlike in traditional feed-forward neural networks where training the network involves finding all connection weights and bias, in ELM, connections between input and hidden neurons are randomly generated and fixed, that is, they do not need to be trained. Thus, training an ELM becomes finding connections between hidden and output neurons only, which is simply a linear least squares problem whose solution can be analytically solved by the generalized inverse of the hidden layer output matrix [41]. Because of such special design of the network, ELM training becomes extremely efficient. Furthermore, empirical studies and recently some analytical studies as well have shown that ELM has better generalization performance than other machine learning algorithms including SVMs and is more efficient and effective for both classification and regression tasks [41].

Consider a dataset, $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, $\boldsymbol{x_i} \in \mathbb{R}^{\boldsymbol{d}}, y_i \in \mathbb{R}^k$ with $k$ classes, and a network with $L$ hidden neurons. Then the output of the network for an input $\boldsymbol{x}$ is expressed as

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{i=1}^{L} \boldsymbol{\beta_i} h_i(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x})\boldsymbol{\beta} \tag{7}$$

where $h_i(\boldsymbol{x}) = G(w_i, b_i, \boldsymbol{x})$, $w_i \in \mathbb{R}^d$ and $b_i \in \mathbb{R}^1$, is the output of the $i$th hidden neuron with respect to the input $\boldsymbol{x}$; $G(w, b, x)$ is a nonlinear piecewise continuous function satisfying ELM universal approximation capability theorems [40]; $\boldsymbol{\beta}_i$ is the output weight vector between the $i$th hidden neuron to the $k \geq 1$ output nodes. $h(\boldsymbol{x}) = [h_1(\boldsymbol{x}), \ldots, h_L(\boldsymbol{x})]$ is a random feature map mapping the data from the $d$-dimensional input space to the $L$-dimensional random feature space (or the ELM feature space).

For the equality optimization constraints-based ELM, the unknown parameter, $\boldsymbol{\beta}$, is found through the following optimization:

$$\begin{aligned} \text{Minimize: } & \boldsymbol{L}_p = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{1}{2}C\sum_{i=1}^{N}\|\boldsymbol{\xi}_i\|^2 \\ \text{Subject to: } & \boldsymbol{h}(x_i)\boldsymbol{\beta} = \boldsymbol{y}_i^T - \boldsymbol{\xi}_i^T, \ i = 1, \ldots, N \end{aligned} \tag{8}$$

where $\boldsymbol{\xi}_i = [\xi_{i,1}, \ldots, \xi_{i,k}]^T$ is the training error vector of the $k$ output nodes with respect to the training sample $\boldsymbol{x}_i$ and the constant $C$ controls the tradeoff between the output weights and the training error.

The equivalent dual optimization objective function of (8) is

$$\boldsymbol{L}_d = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{1}{2}C\sum_{i=1}^{N}\|\boldsymbol{\xi}_i\|^2 - \sum_{i=1}^{N}\sum_{j=1}^{k}\alpha_{i,j}\big(\boldsymbol{h}(\boldsymbol{x_i})\boldsymbol{\beta_j} - y_{i,j} + \xi_{i,j}\big). \tag{9}$$

Based on the Karush–Kuhn–Tucker (KKT) condition, we can have the solution for the ELM output function $\boldsymbol{f}(\boldsymbol{x})$ as follows (refer to [40] for details)

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x})\boldsymbol{\beta} = \boldsymbol{h}(x)\boldsymbol{H^T}\left(\frac{\boldsymbol{I}}{\boldsymbol{C}} + \boldsymbol{H}\boldsymbol{H^T}\right)^{-1}\boldsymbol{Y} \tag{10}$$

where $\boldsymbol{H}$ is the hidden layer output matrix

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{h}(\boldsymbol{x_1}) \\ \vdots \\ \boldsymbol{h}(\boldsymbol{x_N}) \end{bmatrix} = \begin{bmatrix} h_1(x_1) & \ldots & h_L(x_1) \\ \vdots & \vdots & \vdots \\ h_1(x_N) & \ldots & h_L(x_N) \end{bmatrix}. \tag{11}$$

For two- or more-class classification, the predicted class label for the input $\boldsymbol{x}$ is

$$\text{label}(x) = \underset{i \in 1,2,\ldots,k}{\text{argmax}} \ \big[f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})\big]. \tag{12}$$

For one-class ELM, one can see from (10) that when the target class, $\boldsymbol{Y}$, is one class only (e.g., all 1 s), $\boldsymbol{\beta}$ becomes a linear approximation mapping from $\boldsymbol{h(x)}$ to $\boldsymbol{Y}$, which geometrically is a hyper plane approximation [42]. Then the distance of a test sample, $\bar{x}$, to the hyper plane constructed by the ELM is defined as

$$d(\bar{x}) = \big|\boldsymbol{h}(\bar{x})^T\boldsymbol{\beta} - \boldsymbol{y}\big| = \left|\boldsymbol{h}(\bar{x})^T\boldsymbol{H}^T\left(\frac{\boldsymbol{I}}{C} + \boldsymbol{H}\boldsymbol{H}^T\right)^{-1} - 1\right|. \tag{13}$$

The distance can conveniently serve as the anomaly score, that is, the larger the distance is, the more likely the sample is an anomaly. To perform anomaly detection, we simply apply a threshold to the anomaly scores. That is, $\bar{x}$ is abnormal if $d(\bar{x}) \geq T_h$, otherwise it is normal. The threshold can be determined by cross validation, while considering the distribution of distances for all normal samples.

In summary, the overall structure of the proposed AD scheme is shown in Fig. 2. Overall it consists of two phases: 1) offline training and 2) real-time detection. During the offline training phase, both normal and attack data samples are made available and a portion of the normal data are used for training the physical models and the SDAE model (see Section III-A2). The features described in Section III-A are then calculated for the rest of the normal data and the attack data as well. The detection model (ELM classifier) is finally trained using the obtained feature set, and the decision threshold of the classifier is determined using cross validation. All of the feature calculations (including the trained physical models and the SDAE model) are encapsulated into a single, measurements-to-features mapping function, while the trained ELM classification model and the decision threshold are encapsulated into the features-to-decisions mapping function. During the real-time detection phase, at any given time, $t$, a window of physical measurements is first sent to the measurements-to-features mapping function to obtain the feature vector; the features-to-decisions mapping function then takes the feature vector as the input and outputs the status (attack or normal) decision for the given time, $t$.

## IV. Experiments—Case Study

To validate our proposed AD scheme, in this section we consider the heavy-duty gas turbines of combined cycle power plants as a CPS application and develop a pdAD system for this CPS application.

### A. Data

For the gas turbine CPS considered, we use the simulated data for training and testing its AD model. We also use the data from several real planta for validating the model. Followings are the details of both the simulated the real plant data.

*1) Simulated Data:* For simulation, we use our in-house developed, high-fidelity, industrial-scale, hardware-in-the-loop (HWIL) threat simulator (Fig. 3). For developing our gas turbine AD algorithms, we use the simulator to generate multiple data sets, including normal, attack, and evaluation data sets.

Normal operational data are simulated to cover a wide range of turbine operation conditions. Specifically, ambient temperature, pressure, and humidity are varied to capture environmental variations. Fuel composition, compressor flow variation and turbine efficiency are also varied to capture the expected operating conditions for the gas turbine. A design of experiments (DOEs) was performed to blend these factors into a reduced set of simulation runs. It was determined that steady state operating points were required as well as dynamic load conditions. Load level and load rate of change were variations that factored into the DOE runs. The DOE runs contained three different types of variations: 1) Plackett–Burman (PB) full factorial [43]; 2) PB 11 factorial; and 3) Pseudorandom

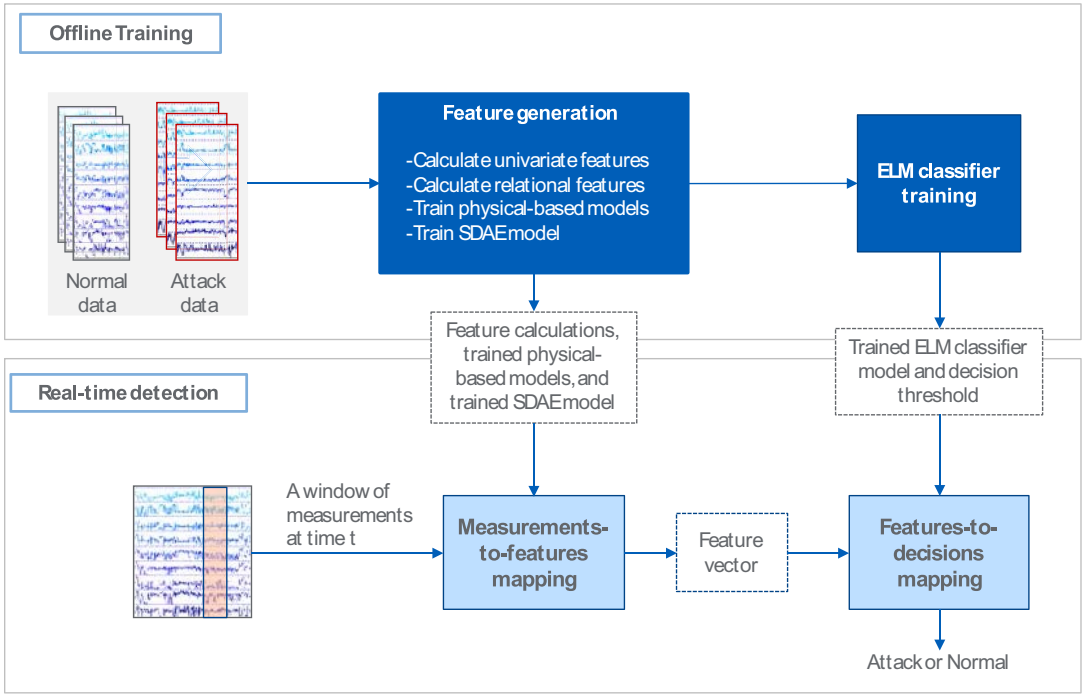


Fig. 2. Overall flow diagram of the proposed AD scheme.

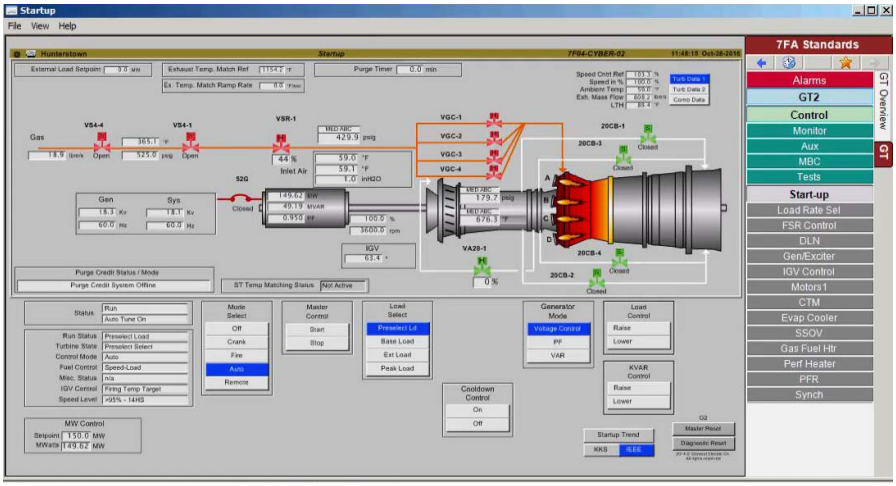


Fig. 3. Plant HMI of the simulator used for data generation.

binary signal (PRBS) [44]. Table I summarizes the DOE runs under normal operations.

For attack data, we consider four different types of attacks: 1) scaling attack—multiply the true signal with a scaling factor; 2) ramp attack—gradually modify the true signal; 3) step attack—add a positive or negative value; and 4) random attack—add a random value to the true signal. We simulate the different attacks on 11 attack nodes, each with three attack levels. Table II shows the summary of the Plackett–Burman DOE runs.

Note that those simulation runs have different lengths in time. For DOE runs, the lengths vary from 200 to 300 s; for PRBS runs, the lengths vary from 2000 to 25 000 s.

*2) Real Power Plant Data:* To further validate the performance of the proposed AD scheme, we retrieve the historical data from 11 gas turbines of five different power plants across the Continental USA (East Coast, Midland, and West Coast). Those gas turbines have power capacities ranging from 20 to 236 MW and are operated at ambient temperatures from 21 °F to 91 °F. For each of the gas turbines, we retrieved once per second data for the period between January 1, 2017 and May 22, 2017, which give us a total of 7 804 600 data samples.

### *B. AD Modeling and Performance Evaluation*

*1) AD Model Details:*

*a) Feature generation:* Both the simulator and real-world gas turbines have a large number of physical measurements available. For AD modeling in this paper, we down-select

TABLE I
SUMMARY OF SIMULATION RUNS FOR NORMAL OPERATIONS

| Descriptions | # of simulation runs |
|---|---|
| *Full factorial DOE* | |
| 4 factors (ambient temp., ambient pressure, relative humidity, and compressor flow) at 3 levels; and load at 8 levels (from 16.8MW to 200MW) | 648 |
| *Placket-Burman DOE* | |
| 11 factors at 2 levels; and load at 8 levels (from 16.8MW to 200 MW) | 96 |
| *Pseudo-random binary signal (PRBS) runs* | |
| Fixed ambient conditions, while varying load from 1MW to 200MW with ramp rate varying from 10 MW/min to 18MW/min | 7 |
| **Total =** | **751** |

TABLE II
SUMMARY OF SIMULATION RUNS FOR ATTACKS

| Descriptions | # of simulation runs |
|---|---|
| *Placket-Burman DOE* | |
| 11 factors Placket Burman DOE at 3 attack levels, 3 ambient temperatures, and 8 levels of load (from 16.8MW to 200 MW) | 936 |
| **Total =** | **936** |

Note that those simulation runs have different lengths in time. For DOE runs, the lengths vary from 200 seconds to 300 seconds; for PRBS runs, the lengths vary from 2000 seconds to 25000 seconds.

24 most commonly used measurements as the monitor nodes, out of which 15 are sensor measurements, 5 are actuator measurements, and 4 are control signals. Out of the 24 selected physical measurements, we take 20 measurements for univariate feature calculation and other four for calculating two pair-wise relation features.

For each of the 20 measurements, we calculate six statistical features, that is, median, variance, kurtosis, range, current value, and maximum rate of change, of the signals within a sliding window with a window width of 50 sample points (see Section III-A1 for details). For the four measurements, we form two pairs and for each pair, we calculate mean difference between the two measurements. Based on physical knowledge, we decide to build three physics models. Model 1 has seven inputs and four outputs, model 2 has three inputs and one output, and model 3 has two inputs and one output. The three models are all ELM regression models and are trained with normal data only. We calculate five residual statistics for each of the six model outputs, which gives us 30 physics-based features. To learn features that capture the nonlinear relationship among the 20 physical measurements, we use the 2-layer SDAE network, (see Section III-A2) and the network structures for the first-layer DAE and second-layer DAE are 20-50-20 and 50-10-50, respectively, where the 50 hidden neuron outputs of the first-layer DAE are used as the inputs to the second-layer DAE. The outputs of the ten hidden neurons of the second-layer DAE are taken as the features, which gives us ten learned features.

So overall, we have 162 $(120 + 2 + 30 + 10)$ features calculated for the sliding window at each time stamp.

*b) ELM detection modeling:* For ELM model design, we set the number of hidden neurons to be the default value of 1000, as suggested in [11]. The activation function for the hidden neurons is the sigmoid function, $G(w, b, x) = 1/(1 + \exp(-(W^T x + b))$. The model parameter, $C$, is empirically determined via grid search of 20 different values, i.e., $C = [2^{-9}, 2^{-8}, \ldots, 2^{10}]$.

*2) Performance Metrics and Evaluation Method:* To assess the performance of the proposed AD scheme, we use the commonly used receiver operating characteristic (ROC) curve as the classification performance measures. The ROC curve captures the tradeoff between the two quantities, true positive rate (TPR) and false positive rate (FPR) [45]. We employ tenfold cross validation for model training and validation. To obtain more robust comparison we run the tenfold cross validation ten times, each time with different randomly splitting of tenfold of the data. For quantitative comparison, we also calculate the average TPRs, at the given FPR of 1.0%, over the ten independent runs. All experiments conducted in this paper are performed in MATLAB environment.

### C. Attack Detection Results

To help understand how good the 162 calculated features are in terms of distinguishing normal and attack samples, we project these features from the original 162-D space to a 2-D space so that we can visualize them. For the projection, we adopt the $t$-distributed stochastic neighbor embedding ($t$-SNE) [46], a projection technique that can maximally preserve the sample distribution while projecting to lower-dimension space. One can see from Fig. 4 that: 1) both normal and attack samples have multimodal distributions, respectively, in the feature space and 2) the normal and attack samples are locally separable in the feature space, which indicates that our ELM detection model trained on the features should perform well in separating normal and attack, albeit the decision boundary of the model may have to be highly nonlinear.

Fig. 5 shows the ROC curves of the proposed AD scheme for ten different tenfold cross validation runs on the simulated dataset. The calculated average TPRs, at the given FPR of 1.0%, over the ten independent runs is 99.46%.

For the real plant data, since no attacks occurred during the time period, we cannot assess the TPR. Instead we can only validate how many false alarms our algorithm

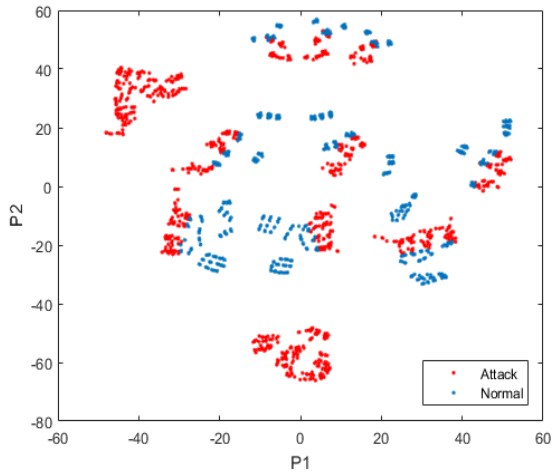


Fig. 4. *t*-SNE projection of the 162 features to 2-D space.

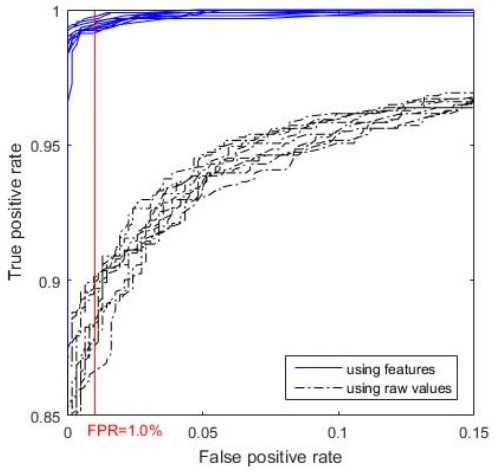


Fig. 6. ROC comparison between two models: using features and using raw values.

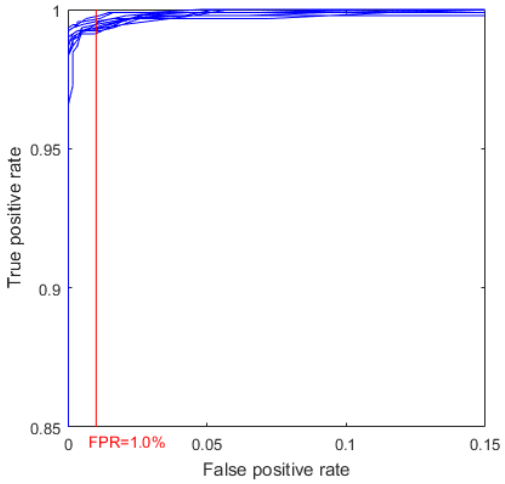


Fig. 5. ROC curves of the proposed AD scheme.

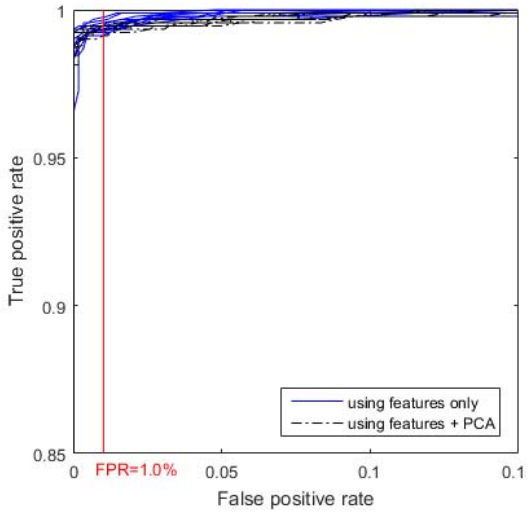


Fig. 7. ROC comparison between two models: using features only and using features + PCA.

generates. By applying the proposed AD scheme to the 7 804 600 data samples collected from 11 different turbines (see Section IV-A2), we obtain an FPR of 0.0006%, which is well below the required FPR of 1%.

### *D. Comparisons and Discussions*

In order to further validate the effectiveness and the superiority of the proposed AD scheme, we perform several comparison studies, comparing our scheme against other modeling designs, as follows.

*1) Comparing Against the Detection Model Using Raw Values as the Inputs:* To appreciate the significance of the feature generation of our proposed AD scheme, we also build a detection model that directly uses the raw physical measurements of the 24 monitoring nodes as the inputs, i.e., without using the generated features. For fair comparison, we keep the detection model, the ELM classifier, to be identical. The ten runs of ROCs for this model with raw value as inputs are compared against the ones of our proposed AD scheme in Fig. 6. Clearly, the proposed AD scheme that uses the generated features as inputs significantly improve the detection performance over the detection model using the raw measurements, which validates the importance of the feature generation step of the proposed AD scheme. Quantitatively, the calculated average TPR, at the given FPR of 1.0%, over the ten independent runs, is 88.81% for this design, as opposed to 99.46% for the proposed feature-based AD scheme, which once again shows a significant difference.

*2) Comparing Against the Detection Model With Feature Dimensionality Reduction:* Realizing that the number of features used for our proposed AD scheme is relatively high with respect to the number of samples we have, we want to know whether feature dimensionality reduction would improve the performance of the proposed AD scheme. For that we perform the PCA, a popular feature transformation method, on the feature set. Once again, we keep the classifier models the same for both designs. Fig. 6 compares ROCs of our proposed AD scheme against the "feature + PCA" design, where, in the feature + PCA design, the first 25 principal components,

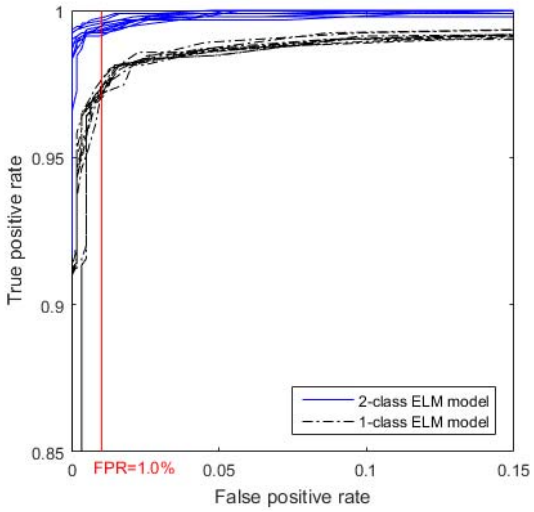


Fig. 8. ROC comparison between two-class ELM and one-class ELM models.

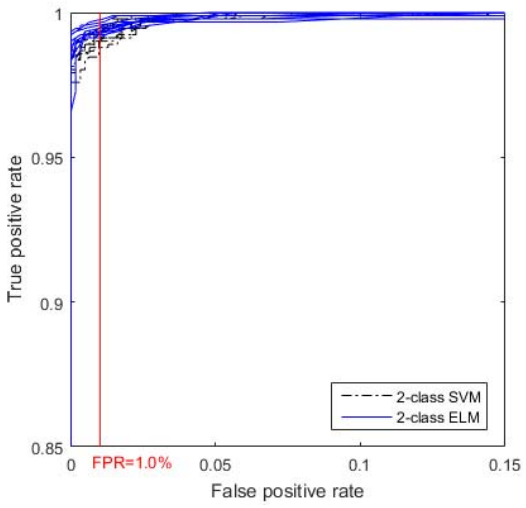


Fig. 9. ROC comparison between two-class ELM and two-class SVM models.

which explain about 99.98% of overall variance of the data, are used as the input to the ELM classifier. From the ROC curves (Fig. 7), one can see that using PCA to reduce the feature dimensionality from 162 to 25 has minimal change to the ROC curves. With PCA, the calculated average TPR at the FPR of 1% is 99.35%, compared to 99.46% of the proposed AD scheme, which indicates that feature dimensionality reduction, such as PCA, has insignificant effect to our ELM detection model. The results support the claim that the ELM has certain sparse regularization capability that can handle feature redundancy [11].

*3) Comparing Against the Detection Model With One-Class ELM:* We also investigate how well the semisupervised, one-class ELM AD model performs, compared with the supervised, two-class ELM AD model used in the proposed AD scheme. Fig. 8 shows the ROCs for both two-class ELM and one-class ELM models, respectively, noting that both models use the same generated features, i.e., the 162 features, as the input. Quantitatively, the average TPR at 1% FPR for one-class ELM model is 97.20%, as opposed to 99.46% for the two-class ELM model used in our proposed AD scheme. By comparing the ROCs and the average TPRs, one can conclude that two-class ELM performs better than one-class ELM, when the same features are used. Thus, using supervised, two-class ELM as the detection model in the proposed AD scheme is a valid choice.

*4) Comparing Against the Detection Model With Two-Class SVM:* We further compare our ELM detection model against SVM detection model in terms of detection performance. It is worth pointing out that for both ELM and SVM models the same 162 features are used as the inputs to the detection models. Fig. 9 shows the ROC comparison between our proposed two-class ELM and two-class SVM. From Fig. 9 one can see that the e-class ELM performs slightly better than the two-class SVM does and the difference between the two models is statistically insignificant in terms of ROC. The calculated average TPR at the FPR of 1% is 99.15% for the SVM model, as opposed to 99.46% for the proposed ELM model. Even though the detection performance between the two models is comparable, ELM model has a clear advantage of being much more efficient in model training.

## V. Conclusion

Geared toward improving CPS security, this paper presents a machine learning-based AD scheme and its application to power plants' gas turbine control systems. Our proposed AD scheme is a physical domain approach, that is, it performs AD by modeling and monitoring the physics of the physical asset or process. The key ingredient of the proposed scheme is its feature generation that enables to derive salient features from noisy physical measurements by leveraging the physical/domain knowledge and using advanced machine learning techniques. With the generated features we are able to better capture the complex, nonlinear relationships of the physical systems; and when combined the features with the supervised machine learning detection model, ELM, we can achieve a high accuracy in early detection of attacks and malicious behaviors. By applying the proposed AD scheme to a CPS application—the gas turbine control system of power plants, we demonstrate, using both simulated and real plant data, that our proposed scheme is effective in detecting malicious attacks of CPSs.

Currently, we are expanding our AD capabilities to cover other power plant components, e.g., steam turbines and generators. We are also actively working on attack localization and attack mitigation—two other important components of CPS security strategies. The results of these ongoing efforts will be part of our future publications.

## Acknowledgment

The authors would like to thank the DOE/CEDS Research and Development Staff, and J. John and D. Holzhauer from GE Global Research, and M. McKinley from GE Power for useful discussions and providing simulation infrastructure with high fidelity models.

## REFERENCES


[1] S. K. Khaitan and J. D. McCalley, "Design techniques and applications of cyber physical systems: A survey," *IEEE Syst. J.*, vol. 9, no. 2, pp. 350–365, Jun. 2015.

[2] N. Falliere, L. O. Murchu, and E. Chien, "W32.Stuxnet dossier," Mountain View, CA, USA, Symantec Corp., White Paper, 2011.

[3] A. Humayed and B. Luo, "Cyber-physical security for smart cars: Taxonomy of vulnerabilities, threats, and attack," in *Proc. ACM/IEEE 6th Int. Conf. Cyber Phys. Syst.*, Seattle, WA, USA, Apr. 2015, pp. 252–253.

[4] J. Urbina *et al.*, *Survey and New Directions for Physics-Based Attack Detection in Control Systems*, document GCR 16-010, Nat. Inst. Stand. Technol., Gaithersburg, MD, USA, Nov. 2016.

[5] J. Kim, L. Tong, and R. J. Thomas, "Subspace methods for data attack on state estimation: A data-driven approach," *IEEE Trans. Signal Process.*, vol. 63, no. 5, pp. 1102–1114, Mar. 2015.

[6] F. Miao, Q. Zhu, M. Pajic, and G. J. Pappas, "Coding schemes for securing cyber-physical systems against stealthy data injection attacks," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 1, pp. 106–117, Mar. 2017.

[7] M. Ozay, I. Esnaola, F. T. Y. Vural, S. R. Kulkarni, and H. V. Poor, "Machine learning methods for attack detection in the smart grid," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 8, pp. 1773–1786, Aug. 2016.

[8] K. Paridari, N. O'Mahony, A. E.-D. Mady, R. Chabukswar, M. Boubekeur, and H. Sandberg, "A framework for attack-resilient industrial control systems: Attack detection and controller reconfiguration," *Proc. IEEE*, vol. 106, no. 1, pp. 113–128, Jan. 2018.

[9] M. Ozay, I. Esnaola, F. T. Y. Vural, S. R. Kulkarni, and H. V. Poor, "Sparse attack construction and state estimation in the smart grid: Centralized and distributed models," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 7, pp. 1306–1318, Jul. 2013.

[10] L. Mestha, O. M. Anubi, and M. Abbaszadeh, "Cyber-attack detection and accommodation algorithm for energy delivery systems," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, 2017, pp. 1326–1331.

[11] G.-B. Huang, H. M. Zhou, X. J. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012.

[12] H. He and J. Yan, "Cyber-physical attacks and defenses in the smart grid: A survey," *IET Cyber Phys. Syst. Theory Appl.*, vol. 1, no. 1, pp. 13–27, 2016.

[13] R. Mitchell and I.-R. Chen, "A survey of intrusion detection techniques for cyber-physical systems," *ACM Comput. Surveys*, vol. 46, no. 4, p. 55, 2014.

[14] S. Han, M. Xie, H.-M. Chen, and Y. Ling, "Intrusion detection in cyber-physical systems: Techniques and challenges," *IEEE Syst. J.*, vol. 8, no. 4, pp. 1049–1059, Dec. 2014.

[15] Y. Liu, Y. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," in *Proc. ACM/IEEE 6th Int. Conf. Cyber Phys. Syst.*, Seattle, WA, USA, Apr. 2009, pp. 252–253.

[16] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on the smart grid," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 645–658, Dec. 2011.

[17] M. Kravchik and A. Shabtai, "Detecting cyber attacks in industrial control systems using convolutional neural networks," in *Proc. ACM Workshop Cyber Phys. Syst. Security PrivaCy (CPS-SPC)*, New York, NY, USA, 2018, pp. 72–83.

[18] J. Goh, S. Adepu, M. Tan, and Z. Lee, "Anomaly detection in cyber physical systems using recurrent neural networks," in *Proc. IEEE 18th Int. Symp. High Assurance Syst. Eng. (HASE)*, 2017, pp. 140–145.

[19] J. Inoue, Y. Yamagata, Y. Chen, C. M. Poskitt, and J. Sun, "Anomaly detection for a water treatment system using unsupervised machine learning," in *Proc. IEEE Int. Conf. Data Min. Workshops (ICDMW)*, New Orleans, LA, USA, 2017, pp. 1058–1065.

[20] N. Wallace, S. Ponomarev, and T. Atkison, "A dimensional transformation scheme for power grid cyber event detection," in *Proc. Cyber Inf. Security Res. Conf.*, 2014, pp. 1–12.

[21] Y. Wang, N. M. Amin, J. Fu, and H. B. Moussa, "A novel data analytical approach for false data injection cyber-physical attack mitigation in smart grids," *IEEE Access*, vol. 5, pp. 26022–26033, 2017.

[22] O. M. Anubi, L. Mestha, and H. Achanta, "Robust resilient signal reconstruction under adversarial attacks," *arXiv:1807.08004*, Jul. 20, 2018. [Online]. Available: https://arxiv.org/abs/1807.08004v1

[23] M. Abbaszadeh, L. Mestha, and W. Yan, "Forecasting and early warning for adversarial targeting in industrial control systems," in *Proc. 57th IEEE Conf. Decis. Control*, 2018, pp. 7200–7205.

[24] X. Li, X. Liang, R. Lu, X. Shen, X. Lin, and H. Zhu, "Securing smart grid: Cyber attacks, countermeasures, and challenges," *IEEE Commun. Mag.*, vol. 50, no. 8, pp. 38–45, Aug. 2012.

[25] K. Manandhar, X. Cao, F. Hu, and Y. Liu, "Detection of faults and attacks including false data injection attack in smart grid using Kalman filter," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 4, pp. 370–379, Dec. 2014.

[26] S. Ponomarev and T. Atkison, "Industrial control system network intrusion detection by telemetry analysis," *IEEE Trans. Depend. Secure Comput.*, vol. 13, no. 2, pp. 252–260, Mar./Apr. 2016.

[27] Y. Zhang, L. Wang, W. Sun, R. C. Green, and M. Alam, "Distributed intrusion detection system in a multi-layer network architecture of smart grids," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 796–808, Dec. 2011.

[28] Y. Jiang, C.-C. Liu, M. Diedesch, E. Lee, and A. K. Srivastava, "Outage management of distribution systems incorporating information from smart meters," *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 4144–4154, Sep. 2016.

[29] Y. Jiang *et al.*, "Blackstart capability planning for power system restoration," *Int. J. Elect. Power Energy Syst.*, vol. 86, pp. 127–137, Mar. 2017.

[30] O. Kocabas, T. Soyata, and M. K. Aktas, "Emerging security mechanisms for medical cyber physical systems," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 13, no. 3, pp. 401–416, May 2016.

[31] D. Z. Wu *et al.*, "Cybersecurity for digital manufacturing," *J. Manuf. Syst.*, vol. 48, pp. 3–12, Jul. 2018.

[32] Y. Pan *et al.*, "Taxonomies for reasoning about cyber-physical attacks in IoT-based manufacturing systems," *Int. J. Interact. Multimedia Artif. Intell.*, vol. 4, no. 3, pp. 45–54, 2017.

[33] E. Fernandes, J. Jung, and A. Prakash, "Security analysis of emerging smart home applications," in *Proc. 37th IEEE Symp. Security Privacy*, San Jose, CA, USA, May 2016, pp. 636–654.

[34] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[35] W. Yan, "Feature engineering for PHM applications," in *Tutorial Annu. Conf. Prognostics Health Manag.*, San Diego, CA, USA, 2015. [Online]. Available: https://www.phmsociety.org/events/conference/phm/15/tutorials

[36] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2016.

[37] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Mar. 2010.

[38] W. Yan and L. J. Yu, "On accurate and reliable anomaly detection for gas turbine combustors: A deep learning approach," in *Proc. Prognostics Health Manag. Conf.*, San Diego, CA, USA, Oct. 2015, pp. 1–8.

[39] N. Görnitz, M. Kloft, K. Rieck, and U. Brefeld, "Toward supervised anomaly detection," *J. Artif. Intell. Res.*, vol. 46, no. 1, pp. 235–262, 2013.

[40] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006.

[41] G. B. Huang, "An insight into extreme learning machines: Random neurons, random features and kernels," *Cogn. Comput.*, vol. 6, no. 3, pp. 376–390, 2014.

[42] Q. Leng, H. Qi, J. Miao, W. Zhu, and G. Su, "One-class classification with extreme learning machine," *Math. Problems Eng.*, vol. 2015, Nov. 2014, Art. no. 412957. [Online]. Available: http://dx.doi.org/10.1155/2015/412957

[43] R. L. Plackett and J. P. Burman, "The design of optimum multifactorial experiments," *Biometrika*, vol. 33, no. 4, pp. 305–325, Jun. 1946.

[44] A. H. Tan and K. R. Godfrey, "The generation of binary and near-binary pseudorandom signals: An overview," *IEEE Trans. Instrum. Meas.*, vol. 51, no. 4, pp. 583–588, Aug. 2002.

[45] M. H. Zweig and G. Campbell, "Receiver-operating characteristic (ROC) plots: A fundamental evaluation tool in clinical medicine," *Clin. Chem.*, vol. 39, no. 8, pp. 561–577, 1993.

[46] L. J. P. van der Maaten and G. E. Hinton, "Visualizing high-dimensional data using *t*-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

**Weizhong Yan** (M'04–SM'08) received the Ph.D. degree from Rensselaer Polytechnic Institute, Troy, NY, USA, in 2002.

He has been with General Electric Company, Boston, MA, USA, since 1998. He is currently a Principal Scientist with the AI and Machine Learning Discipline, GE Global Research Center, Niskayuna, NY, USA. He was an Adjunct Professor of mechanical engineering with Rensselaer Polytechnic Institute from 2004 to 2008. He has authored over 90 publications in refereed journals and conference proceedings and has filed over 60 U.S. patents. His current research interests include neural networks, ensemble learning, feature engineering and representation learning, and time-series forecasting.

Dr. Yan has been a recipient of numerous professional awards. He was the winner of the GRC Dushman Technology Excellent Award in 2018, the winner in the IJCNN-2007 Time-series Forecasting competition, the first-place winner of the GRC APAF Growth Competition Award in 2006, the winner of GRC Technical Excellent Award in 2007, and the winner of the Lockheed Martin One-Company-One-Team Award in 2004. He is an Editor of the *International Journal of Artificial Intelligence* and an Associate Editor of the *International Journal of Prognostics and Health Management*.

**Lalit K. Mestha** (F'11) received the B.E. degree from the University of Mysore, Mysuru, India, in 1981 and the Ph.D. degree from the University of Bath, Bath, U.K., in 1985.

He is currently the Director of biometric research with KinetiCor, San Diego, CA, USA, developing innovative solutions for biomedical systems. He was a Principal Engineer with GE Global Research Center, Niskayuna, NY, USA, and a Research Fellow with Xerox PARC, Palo Alto, CA, USA. He is an Adjunct Professor with the University of Texas at Arlington, Arlington, TX, USA. He currently holds 250 U.S. patents and numerous pending applications. He has published over 80 papers, coauthored one reference book, and written chapters for two books. His current research interests include particle accelerators, digital production printers, biomedical systems, and critical infrastructures.

Dr. Mestha is a fellow of the National Academy of Inventors.

**Masoud Abbaszadeh** (S'06–M'08–SM'16) received the B.Sc. degree in electrical and computer engineering from the Amirkabir University of Technology, Tehran, Iran, in 2000, the M.Sc. degree in electrical and computer engineering from the Sharif University of Technology, Tehran, in 2002, and the Ph.D. degree in electrical and computer engineering from the University of Alberta, Edmonton, AB, Canada, in 2008.

He is currently with GE Global Research Center, Niskayuna, NY, USA. From 2011 to 2013, he was a Senior Research Engineer with the United Technologies Research Center, East Hartford, CT, USA. From 2008 to 2011, he was with Maplesoft, Waterloo, ON, Canada. He was the Principal Developer of MapleSim Control Design Toolbox and a member of a research team working on the Maplesoft–Toyota joint projects. His current research interests include estimation and detection theory, robust and nonlinear filtering, and statistical machine learning with applications such as cyber-physical security and autonomous systems.

Dr. Abbaszadeh is an Associate Editor of the *Intelligent Industrial Systems* (Springer) and serves as a technical program committee member in various conferences.