Applied Energy 240 (2019) 561–582

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy

An anomaly detection framework for dynamic systems using a Bayesian hierarchical framework

Ramin Moghaddass[a,*], Shuangwen Sheng[b]

[a] University of Miami, Coral Gables, FL, USA
[b] National Renewable Energy Laboratory (NREL), Golden, CO, USA

HIGHLIGHTS

- A cost-sensitive anomaly detection framework for sensor-intensive systems.
- A generative structure to model output-inputs using a subset of features and samples.
- An optimal cost-sensitive model for both supervised and unsupervised settings.
- The Bayesian setting that avoid overfitting and conduct regularization.
- Numerical experiments using simulation and a data set for wind turbine monitoring.

ARTICLE INFO

Keywords:
Anomaly detection
Wind turbine
Dynamic systems
Sensor-intensive energy systems
Bayesian Modeling

ABSTRACT

Complex systems are susceptible to many types of anomalies, faults, and abnormal behavior caused by a variety of off-nominal conditions that may ultimately result in major failures or catastrophic events. Early and accurate detection of these anomalies using system inputs and outputs collected from sensors and smart devices has become a challenging problem and an active area of research in many application domains. In this article, we present a new Bayesian hierarchical framework that is able to model the relationship between system inputs (sensor measurements) and outputs (response variables) without imposing strong distributional/parametric assumptions while using only a subset of training samples and sensor attributes. Then, an optimal cost-sensitive anomaly detection framework is proposed to determine whether a sample is an anomalous one taking into consideration the trade-off between misclassification errors and detection rates. The model can be used for both supervised and unsupervised settings depending on the availability of data regarding the behavior of the system under anomaly conditions. The unsupervised model is particularly useful when it is prohibitive to identify in advance the anomalies that a system may present and where no data are available regarding the behavior of the system under anomaly conditions. A Bayesian hierarchical setting is used to structure the proposed framework and help with accommodating uncertainty, imposing interpretability, and controlling the sparsity and complexity of the proposed anomaly detection framework. A Markov chain Monte Carlo algorithm is also developed for model training using past data. The numerical experiments conducted using a simulated data set and a wind turbine data set demonstrate the successful application of the proposed work for system response modeling and anomaly detection.

1. Introduction

Anomalies are often defined as data points, patterns, or behaviors that do not conform to expected behavior and that have different characteristics from normal instances. Most complex systems are susceptible to many anomalies caused by a variety of off-nominal conditions that may ultimately result in major failures, catastrophic events, or costly repair. The importance of anomaly detection is because of the fact that it can translate to significant, and often critical, actionable information in a wide variety of application domains [1]. In energy-intensive industries, accurate and timely detection of anomalies and abnormal system behaviors can potentially result in significant energy efficiency benefits and cost savings. In the context of reliability and maintenance, early detection of anomalies is of the utmost importance

* Corresponding author.
E-mail address: Ramin@miami.edu (R. Moghaddass).

https://doi.org/10.1016/j.apenergy.2019.02.025
Received 12 November 2018; Received in revised form 23 January 2019; Accepted 5 February 2019
Available online 20 February 2019
0306-2619/

and can provide significant health monitoring information and actionable insights. Such information and insights can prevent costly events, such as failure and downtime, particularly in capital-intensive industries, such as wind turbine, that have a considerable amount of operations and maintenance costs. The detection of anomalous events through continuous monitoring of working systems can be a part of a preventive maintenance policy to prevent further damages and avoid costly maintenance operations as well as a part of a predictive maintenance policy to assist maintenance operators to better plan for any future maintenance interventions. Anomaly detection can provide situational awareness to inform/alert maintenance operators regarding the original source of abnormal behavior in order to assess the seriousness and severity of the anomaly. Many research articles have utilized anomaly detection for the purpose of maintenance planning and decision-making (see for instance [2]).

Because of the complexity of today's systems and their stochastic nature, anomalies are often not directly observable from system inputs and outputs and need to be diagnosed indirectly using available data, particularly through monitoring sensors. Thus, it is desirable to develop new data mining approaches that can employ observable sensor data to identify anomalies in a timely manner. The availability of large-scale data produced by a variety of sensors over the past decade has opened up many new real-time or near real-time analytics opportunities to improve detection and prediction of anomalies and to help prevent problems before they lead to costly or catastrophic events. However, because of the complex structure of systems, and the high volume, velocity, and dimensionality of the data collected from these systems, current data-driven methods are often inefficient at generating trustable insights with respect to anomalies. The problem with the majority of these data-driven methods is that they are often a byproduct of an algorithm originally designed for a purpose other than anomaly detection. As a result, they are not optimized to detect anomalies and may lead to too many false alarms or too few detected anomalies [3].

Among available anomaly detection methods, nonparametric statistical modeling has gained popularity because of its appealing properties, such as less distributional and parametric assumptions regarding the relationship between predictors and response variables and the ability to handle some of the current challenges of anomaly detection methods. For example, nonparametric techniques do not generally assume knowledge of the underlying distribution of data and instead gain insights from past data. Also, there is no need to estimate additional unknown model parameters and perform goodness-of-fit tests. To define the relationship between multivariate predictors and response variables, we develop a new nonparametric approach that benefits from some of the advantages of a well-known nonparametric regression method called kernel regression, such as simplicity and mathematical convenience, and partially addresses its two main shortcomings of curse of dimensionality and computational complexity. We then utilize a Bayesian hierarchical setting that has a generative structure (i.e., a top-down structure where response variables are generated from measurements and other variables) to better describe the structure of the model, accommodate uncertainty, and assist in the interpretation of model parameters and the control of model complexity.

Regardless of what type of model is used to define the relationship between the system's multivariate measurements/predictors and the response variable, anomaly detection is a task that should be treated as a decision-making process. In data-driven frameworks, the system's operators will have to analyze system inputs and outputs and then, based on some predefined decision-making criteria, decide whether to label a sample as an anomalous sample. One major factor that defines these decision-making criteria in the context of anomaly detection is the trade-off between misclassification errors and detection rates. Accounting for this trade-off is particularly important when the assumption that all misclassification errors have the same weight/cost is not valid. This trade-off can be represented (and quantified) by a mathematical function referred to in this article as a cost function. The word *cost* (sometimes referred to as risk) is used in this article in the general sense and may include any criterion in addition to the monetary cost. A decision-making process that minimizes such a cost function is often referred to as a cost-sensitive or risk-sensitive decision-making process. This article develops a cost-sensitive anomaly detection framework and its optimal structure based on the trade-off between false alarms and missed anomalies and provides both analytical and empirical results.

The article is mainly motivated by the need to develop sensor-based mathematical models that can efficiently generate actionable insights and decision-making intelligence for systems working under dynamic operating and environmental conditions, which cover the vast majority of condition-monitored systems, particularly in the wind and power industries. Generating alarms/warnings to operators and decision makers who determine when to start preparing for expensive and time consuming maintenance activities ahead of failure is very common and critical for the wind turbine industry. By triggering warnings using knowledge of the degradation state, decision makers may be able to detect early indications of possible problems and make more accurate and timely maintenance decisions. Although the results of this article can potentially be used for any type of system under condition monitoring, the article is motivated by the wind industry, and as a result, the application focus of the numerical experiments section is mainly on wind turbine condition monitoring.

The main contributions made in this article are summarized below. A new generative structure without strong distributional/parametric assumptions is first developed to model system outputs in terms of system inputs in the sense that only a subset of features and training samples is kept in the training pool. Then, an optimal cost-sensitive anomaly detection model for both supervised and unsupervised settings is proposed that can be used to determine whether a data sample is an anomalous sample. Two optimal strategies (one for the supervised setting and one for the unsupervised setting) are proposed to detect anomalies based on the system's policy represented by the trade-off between false alarms and missed anomalies. The unsupervised case is particularly useful in many practical cases where it is prohibitive to recognize in advance all the possible anomalies that a system may present. Also, it can be used when either no labeled data are available from past data or anomalies and their underlying system behavior are not fully known. The Bayesian setting in the model can help with accommodating uncertainty and generating a more realistic picture by providing the stochastic distribution of the parameter posteriors and anomaly status, avoiding overfitting, and conducting regularization. Unlike many of the black box approaches, our model has the benefit of interpretability, both in model structure and model parameters. The novelty of the article is mainly in its power to transform multi-dimensional sensor data into insights regarding whether a system is under a previously known or an unknown anomaly condition in an optimal manner using only a subset of attributes and training samples. We should point out that although the Bayesian setting and its hierarchical forms have been used in the literature for reliability analysis, utilizing it in an interpretable manner for system's response modeling while regularizing both training samples and feature set is one of the article's contributions. One important advantage of our model in real applications is that the potential users do not need to know/define the physical characteristics of anomalies and the relationship between system inputs and outputs.

The rest of this article is organized into the following sections. Section 2 provides an overview of relevant anomaly detection models in the literature and their shortcomings. In Section 3, we describe the model and the Bayesian inference procedure. The structures of the hierarchical model for modeling the system output and the anomaly detection framework are illustrated in Section 4. We then use a series of numerical experiments in Sections 5 and 6 to evaluate the proposed framework and demonstrate its application for wind turbine anomaly detection. Finally, we conclude in Section 7 and introduce a few

important directions for future research.

2. Related work

In this section, we provide an overview of the literature on various techniques used for anomaly detection. Then, we summarize the limitations of available work and discuss how the proposed framework can address these limitations.

2.1. A review of anomaly detection methods

Many different outlier and anomaly detection methods have been developed in the literature and applied in practice. Available anomaly detection methods can be categorized into three main classes: density-based, distance-based, and model-based (or physics-based) approaches [3]. For a comprehensive review of anomaly detection methods in each of these three major categories, interested readers may refer to [1,3]. The majority of available models are model-based, which are based on detecting off-nominal or outlier stochastic behaviors according to a known (often parametric) model. The key success factor of these methods is their ability to construct a model that accurately reflects the nominal behavior of systems [4]. Although most of these methods have been successfully applied in a number of domains, many can only detect hazardous behaviors on a predefined list and thus miss important risks that are unlisted or unknown. Thus, they can only be used in a supervised manner, which requires an accurate set of labeled training data. For example, in modern aircraft systems, current analytical methods, such as Multiple Kernel Anomaly Detection and Exceedance Detection, can only detect anomalies based on a predefined list [5]. It is known that unsupervised anomaly detection methods are more promising for practical applications, as they do not require anomaly labels and high-quality training data from past anomalies [6].

Many anomaly detection methods construct a profile of normal instances and then identify anomalies as those that do not conform to the normal profile [3]. For example, the residuals between the sensed engine outputs and the model predicted outputs were used in [4] for anomaly detection. In [7], a normal behavior model of computers and networks was first built based on the Markov chain and then large deviations of the observed data calculated based on the chi-square distance monitoring method were employed as indicators of cyber attacks. In [8], a weighted version of the least squares support vector regression that can model wind turbine responses as a function of a few variables was proposed in which a large absolute value of residuals was used to indicate possible abnormal situations. The most commonly used models are neural network, support vector machine regression, and K-nearest neighbors (KNN). The main limitations of the neural network-based models for anomaly detection are their black box structure, lack of interpretability, and computational complexity. Common limitations of the above models are their deterministic nature, use of the whole training set for model learning, and finding a proper kernel and key parameters in SVM and K in KNN. In addition, none of these models can automatically consider the trade-off between prediction performance measures during the task of anomaly detection. Because many of these approaches are byproducts of an algorithm originally designed for a purpose other than anomaly detection, they are not optimized to detect anomalies and may lead to too many false alarms or too few detected anomalies [3].

From a methodological point of view, available anomaly detection models can be categorized to classification-based, nearest neighbor-based, clustering-based, statistical (parametric, nonparametric), information theoretic, and spectral anomaly detection methods, each with certain assumptions, advantages, disadvantages, and unique strengths and weaknesses [1]. Among anomaly detection models, nonparametric models have gained popularity, particularly in data-intensive and high-dimensional settings where finding a parametric relationship between system inputs and outputs is not possible. Kernel density estimation is one of the most common nonparametric density estimators for multivariate data [9] and was first introduced into statistical learning by [10]. Many extensions of conventional kernel models have been developed in the literature 11, see for instance. Kernel methods are particularly well known to be consistent density estimators under suitable conditions on the bandwidth [9]. Many kernel-based methods, such as SVM [12], kernel principal component analysis (PCA) [13], and kernel Reed-Xiaoli (RX) methods [14], have been employed for anomaly detection in the literature. Although kernel-based models are relatively simple, easy to use, and easy to understand, available kernel-based anomaly detection methods have important shortcomings that have limited their applications in practical problems. Some of these shortcomings, such as the curse of dimensionality, are very common in conventional kernel methods [11]. Typical kernel-based methods have two main limitations. First, they are not efficient at handling too many features, particularly in high-dimensional settings. Second, kernel-based models are computationally expensive because all training samples are used for the prediction of a new sample's response variable. Also, most kernel methods assume that the training set is clean and has no anomalous points, which is not true particularly in sensor-based data sets that are contaminated with noise and may also contain samples that are not generated under the system's normal conditions. Also, using conventional techniques that detect anomalies in a full-dimensional space is problematic, as anomalies often appear in a subset of all dimensions [15]. In order to extract meaningful information from sensors, it is important to select data, sensors, and recommend features that are strong predictors of a fault, are highly relevant, and minimally redundant [16].

This article focuses on addressing some of the above-mentioned shortcomings of available models. The proposed model selects a limited set of samples for model training and can potentially identify and remove anomalous samples during the training process (by removing them from the training pool). Also, only a subset of features is selected for predicting the response variable and detecting anomalies. Because certain variables may be more informative than others when used to identify certain anomalies, regularization can impose a sparse model that removes uncorrelated variables falsely found to be correlated with system dynamics. The proposed framework can be applied in both supervised and unsupervised manners depending on the availability of sufficient labeled training data.

2.2. Cost-sensitive anomaly detection

Most anomaly detection methods are cost insensitive. That is, they assume that all misclassification errors (false alarms or missed anomalies) have the same cost (or risk), and there is no difference in the representation and penalization of normal vs. anomalous points [17]. In those models, either there is no major performance criterion for detecting anomalies or the error rate (i.e., the expected fraction of misclassification) is used as the main tool to build a binary classifier for anomaly detection. Examples of cost insensitive anomaly detection models can be found in [18,19]. Most anomaly detection or classification problems include class imbalance and are naturally cost-sensitive [20] and thus should be treated as a cost-sensitive decision-making problem. The majority of published cost-sensitive classification algorithms assume the availability of supervised training data in which all instances are labeled [17]. There are also reported cost-sensitive anomaly detection frameworks that are modeled as a binary classification problem in which classifier parameters are trained to improve the detection of important anomalies and reduce the total losses [21]. In this article, we develop a cost-sensitive decision-making process that can predict anomalies while optimizing an important performance criterion referred to as the total cost (or alternatively called risk). To our knowledge, no similar cost-sensitive anomaly detection framework exists that can optimally detect anomalies in both supervised and unsupervised settings. In addition, our article can include operational

usability factors by imposing limits on performance measures, such as the rates of false and true detection.

2.3. Anomaly detection in wind turbines

Anomaly detection has been extensively applied in a variety of applications, including fraud detection [22], fault detection [23], flight safety prediction [24], network intrusion detection [25], image denoising [26], social networks [27], nuclear components and systems [28], engine health management in gas turbines [29], building energy data and operational patterns [6], and health care [30]. Condition monitoring and fault diagnosis of wind turbines have received a high priority over the past years because of the continuous growth of wind energy generating sources and increasing demand for more careful planning and control of operation and maintenance costs [31]. The wind power industry all over the globe is constantly seeking more cost-sensitive operations and maintenance actions. As a result, various strategies have been proposed for wind turbines [32]. For a recent review of wind turbine reliability studies, interested readers may refer to [33].

Many anomaly detection models have been developed for wind turbines [34]. Most of the reported references have used supervisory control and data acquisition (SCADA) data and the residual of one or multiple output variables for anomaly detection (see for instance [35]). In [36], the residual error of the system output together with a critical threshold were used for anomaly detection in which a simple forward feature selection approach was employed to identify important features. The well-known method of back-propagation neural network was utilized to estimate the target variables. In [37], an index called the abnormal level index was proposed based on SCADA data to quantify the abnormal level of wind turbine condition parameters. The back-propagation neural network was used for prediction of wind turbine condition parameters, and a fuzzy synthetic evaluation was used to integrate the selected condition parameters. In [38], a self-organizing map was first used to model normal behavior by projecting multi-dimensional SCADA data into a two-dimensional map. Then, the deviation between the current status and normal behavior was used as the indicator of a suspicious anomaly, and a threshold was defined based on the quantile function to screen the anomaly. In [39], an adaptive neuro-fuzzy interference system was used to develop a normal behavior model to monitor and detect anomalies by considering the prediction error in SCADA outputs. A series of work in [40,41] focused on developing models for wind turbine condition monitoring and anomaly detection using SCADA data. In all of these references, a neural network was used to generate the normal behavior model. Also, all three references considered only a subset of features, either through expert judgment [41] or using well-known statistical methods [40]. It is also known that most of the state-of-the-art approaches on wind turbine condition monitoring are based on the labeled data and thus, unsupervised anomaly detection has been rarely discussed [42].

Our article is different from the above reference because (i) the proposed model uses only a subset of important/effective training samples and features in which this subset is fully trained with past data and (ii) an optimal cost-sensitive framework is developed to detect anomalies based on the trade-off between misclassification errors as well as operational usability factors (e.g., controlling the false alarm rate). Our proposed framework can be used in both supervised and unsupervised manners. Identifying unknown anomalies in wind turbines is particularly important as they operate under dynamic operating and environmental conditions where many types of unknown anomalies occur over time.

2.4. Summary of the related work and the knowledge gap

The main limitations of related studies are summarized here. First, most available models either require training data with actual labels of normal vs. anomaly or simply assume that training data are clean and have no anomalous points. This is unrealistic for many situations in which past data are contaminated with anomalies. Also, almost all available studies utilize the whole data for model training. Second, much of the available work assumes that all variables contribute to the detection of anomalous events. The ones with feature selection often use off-the-shelf feature selection models, such as backWard or forward feature selection algorithms. Feature selection allows us to build a separate model with a unique set of attributes for each type of known anomaly. This is a benefit over models that employ the same variables and samples for detecting different types of anomalies. Third, most anomaly detection frameworks are cost insensitive and cannot consider the system's policy and operational usability factors (e.g., controlling false alarms). Forth, available anomaly detection frameworks are often only applicable in a supervised manner.

To address the above limitations, accommodate uncertainty, and regulate the degree of sparsity, we propose a Bayesian hierarchical structure in which the response variable is formulated as a generative process according to a subset of important features and important samples. The importance of features and samples is represented by binary variables that are trained from past data. This method is particularly useful when there is a large number of irrelevant attributes and redundant and anomalous samples. Our model also does not require any functional relationship between predictors and response variables, which is a benefit over some traditional anomaly detection algorithms. The proposed model is simple, interpretable (which is a benefit for industrial applications), and relatively fast (because it uses only partial samples and features in the training data). The Bayesian setting can help with accommodating uncertainty and obtaining more reasonable estimates of the parameter posteriors, avoiding overfitting, and conducting regularization. With the hierarchical setting and providing closed-form conditional probabilities, we benefit from developing analytical solutions for parameter estimation. Using prior information, we can control model parameters and model complexity. For instance, we can change the average fraction of samples being used in the training pool and as a result, the sparseness and the complexity of our model are under control. The main disadvantage of using Bayesian settings compared to non-Bayesian settings is the intensive computation during the model training step, which we try to partially overcome with the selection of features and samples.

To the best of our knowledge, the closest research to this article is the work by [43]. Although there are similarities in terms of the kernel between that work and ours, our model is different as it uses only a subset of features and samples and utilizes a Bayesian hierarchical framework to build the model structure. In addition, we provide a cost-sensitive anomaly detection framework. We should point out that although the Bayesian setting and its hierarchical forms have been used extensively in the literature for reliability analysis (see for instance [44] for prognostics in Li-ion batteries and [45] for predicting railway track geometry degradation), utilizing it in an interpretable manner (as will be described later in the article) for system's response modeling while regularizing both training samples and feature set is one of the contributions made in the article. Our method is different from distance-based outlier detection methods reported in [46,47] that define outliers by their distance to neighboring samples from two main aspects. First, our model optimally detects anomalies according to a cost-sensitive decision-making process that considers the trade-off between false alarms and missed anomalies. Second, our model works only on a representative subset of training samples and feature space.

3. The main framework

3.1. System response modeling – a generative approach

In this section, the structure of the hierarchical framework for the generation of the system's response is presented. The main assumptions

made with regard to the system under study are as follows:

- The single-unit system is under continuous monitoring at discrete time points. At each monitoring point, sensor measurements (system inputs) and the system response/output are directly observable.
- No parametric distribution between the system's input and the output is known.
- The operating condition of the system (anomaly vs. normal) is not directly observable.
- The training (historical) data may or may not contain anomaly-labeled data.
- A data point is considered to be an anomaly sample only if it is extreme with respect to the response value (y), not the input (x) values. For the above reason, data with noise in the inputs and high-leverage observations are not considered in the proposed framework.
- The trade-off between misclassification errors and detection rates can be quantified.

Let $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\} \in \mathbb{R}^P$ be a set of N independent and identically distributed (iid) P-vector random samples associated with P features (measurement types) from a joint distribution with density $f_X(\boldsymbol{x})$ and $y_1, ..., y_N$ be the corresponding one-dimensional (continuous) response variables. We assume that $\boldsymbol{x}$ and y are from a data generating process that can be functionally described for sample n by the following equation:

$$y_n = \phi(\boldsymbol{x}_n) + \epsilon_n, \quad n \in \{1, ..., N\}, \tag{1}$$

where $\epsilon_i, ..., \epsilon_N$ are iid random variables with mean 0 and variance σ^2 (and $cov(\epsilon_i, \epsilon_j) = 0$). It should be pointed out that this model can be simply modified for the case in which the variance of y is not a constant function (e.g., $Var(y|\boldsymbol{x}) = \sigma^2(\boldsymbol{x})$). To define function $\phi(\boldsymbol{x}_n)$ non-parametrically with minimal distributional assumptions, we assume that it is in the class of linear smoothers that transforms the feature vector to the response variable ($\mathbb{R}^P \to \mathbb{R}$) using a number of observations around the same neighborhood in the training pool. Based on this function, the estimator is linear in the sequence y_n, $n \in \{1, ..., N\}$. Let us define $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N]$, $\mathbf{Y} = [y_1, y_2, ..., y_N]$ as the sets of N independent samples of sensor measurements and system response values. We present a data-driven approach to estimate $\phi(\boldsymbol{x})$ using past data, denoted by $\mathscr{D} = \{\mathbf{X}, \mathbf{Y}\}$, where $\mathbf{X}$ and $\mathbf{Y}$ are from a random experiment as described below:

$$\phi\left(\boldsymbol{x} \middle| \mathscr{D}\right) = \sum_{n=1}^{N} w\left(\boldsymbol{x}, \boldsymbol{x}_n\right) y_n. \tag{2}$$

Here $w(\boldsymbol{x}, \boldsymbol{x}_n)$ is the weight of the observation n with respect to $\boldsymbol{x}$ and $\sum_{n=1}^{N} w(\boldsymbol{x}, \boldsymbol{x}_n) = 1$. We define a kernel-based weight as

$$w\left(\boldsymbol{x}, \boldsymbol{x}_n\right) = \frac{r_n K_{\mathbf{H}}^{\boldsymbol{q}}(\boldsymbol{x}, \boldsymbol{x}_n)}{\sum_{i=1}^{N} r_i K_{\mathbf{H}}^{\boldsymbol{q}}\left(\boldsymbol{x}, \boldsymbol{x}_i\right)}, \quad n \in \left\{1, ..., N\right\}, \tag{3}$$

where r_i is a binary indicator denoting whether sample i in the training pool $\mathscr{D}$ is an effective/important sample (or prototype) and q_j is a binary indicator denoting whether feature j is important. Here, $K_{\mathbf{H}}$ is a kernel function with bandwidth $\mathbf{H}$ and feature importance vector $\boldsymbol{q} = \{q_1, ..., q_P\}$. Based on this formula, the response variable of a new sample y_n is the feature-dependent weighted average of selected y values around the effective neighborhood of $\boldsymbol{x}_n$ plus some noise. An important feature of the model is that the kernel is calculated over the set of important features and important samples only. It is clear that the model becomes the Nadaraya-Watson regression developed by [48,49] if $r_n = 1$ and $q_p = 1$ for all $n \in \{1, ..., N\}$ and $p \in \{1, ..., P\}$. Although our model is not limited to any type of kernel, the focus in the article is on the Gaussian kernel because of its common usage, convenient mathematical structure, computational efficiency, and ability to control the degree of smoothing. Now the modified Gaussian kernel based on feature importance vector $\boldsymbol{q}$ can be written as

$$K_{\mathbf{H}}^{\boldsymbol{q}}\left(\boldsymbol{x}, \boldsymbol{x}_i\right) = (2\pi)^{-d/2} \left|\mathbf{H}\right|^{-1/2} e^{-\frac{1}{2}(|\boldsymbol{x}-\boldsymbol{x}_i|\cdot\boldsymbol{q})^T \mathbf{H}^{-1}\left(\left|\boldsymbol{x}-\boldsymbol{x}_i\right|\cdot\boldsymbol{q}\right)},$$

where $\mathbf{H}$ is a symmetric positive definite $P \times P$ matrix known as the bandwidth matrix that controls the amount and orientation of smoothing induced ($\mathbf{H} = \text{diag}(h_1, ..., h_P)$) and thus plays the role of a smoothing parameter. We can also use a bandwidth matrix proportional to Σ^{-1}, where Σ is the covariance matrix of the data. The bandwidth matrix can be estimated by cross-validation. Based on this kernel function, only important dimensions in $\boldsymbol{q}$ are taken into account. Similar to other kernel methods, our model should provide consistent estimators under suitable conditions when the bandwidth vector tends to zero for large N. By selecting only important prototypes in the training data and feature space (partial subset of training data), our model provides a sparse framework leading to large computational savings and performance improvement.

With the form defined for $\phi(\boldsymbol{x}_n)$, which is constructed according to the information derived from past data, we do not need to assume that prior knowledge or a fully parametric formulation is available for the relationship between predictors $\boldsymbol{x}$ and response y. Besides, there is no need to estimate additional unknown model parameters and perform goodness-of-fit tests. More importantly, the form that function $\phi(\boldsymbol{x}_n)$ is defined allows us to directly impose regularization for both samples (by defining r_i) and features (by defining q_j) and provides flexibility for possible complex interactions between predictors in producing the response variable. We should also point out that a potential issue in the proposed model compared to fully parametric models is that a larger number of samples may be required in the original data set so that final important samples and features are properly estimated.

3.2. Model training

The objective of model training is to find reasonable values for the main model parameters and characterize the model structure using data. The set of unknown parameters is denoted by $\Theta = [\boldsymbol{r}, \boldsymbol{q}, \sigma]$, where $\boldsymbol{r} = [r_1, ..., r_N]$, and $\boldsymbol{q} = [q_1, ..., q_P]$. Thus, the number of model parameters is $N + P + 1$. A standard parameter estimation approach is to use the ordinary least squares (OLS) or the maximum likelihood model as follows:

$$\mathscr{L}\left(\Theta \middle| \mathscr{D}\right) = \sum_{i=1}^{N} \log p\left(y_i \middle| \boldsymbol{x}_i, \Theta\right) = \sum_{i=1}^{N} \log \mathscr{N}\left(y_i; \phi\left(\boldsymbol{x}_i \middle| \mathscr{D}\right), \sigma\right), \tag{4}$$

where $\mathscr{D} = [\mathbf{X}, \mathbf{Y}]$ includes the set of past data for N samples. Maximizing the likelihood function in the above form is numerically not tractable because of the complex relationship between $\boldsymbol{x}$ and y and the existence of many binary variables (i.e., $\boldsymbol{r}$ and $\boldsymbol{q}$). The model described so far is a basic model that does not define the relationship between model variables and does not account for uncertainty. The hierarchical form described below will explain the structure of the model and will assist in the interpretation of model parameters and the control of model complexity. All parameters and hyperparameters in our model can be interpreted; thus, our model is not a pure black box model. Based on our hierarchical framework, the model parameters and their stochastic characteristics can be defined in a generative form as follows:

Variable r_i: This variable shows the relative importance of sample i in the training pool and is assumed to follow a Bernoulli distribution as $r_i \sim \text{Bern}(\alpha) i \in \{1, ..., N\}$, where $0 \le \alpha \le 1$. In addition to mathematical convenience, choosing the Bernoulli distribution helps with interpretability. That is, α can be simply interpreted as the proportion of

important samples. Having this variable, the samples that are either noisy or add no information to the prediction of response variables are expected to leave the training sample pool. This will not only help with computational savings but also help with robustness. Here, hyperparameter α gives the ability to the users to control the sparseness of the training pool and the complexity of the model. This parameter (α) can either be predefined by the users or treated as an unknown variable with a noninformative or informative prior depending on how much information is available. We should point out that importance and effectiveness in our article refer to the relative importance/effectiveness with respect to the response/target variables. This does not mean that samples with $r_i = 0$ are not important, it just means that the selected samples are sufficient to model system outputs in terms of sensor inputs.

Variable q_j: This variable shows the importance of feature j and is assumed to follow a Bernoulli distribution as $q_p \sim \text{Bern}(\beta)$, $p \in \{1, \ldots, P\}$, where $0 \leqslant \beta \leqslant 1$. In addition to mathematical convenience, choosing a Bernoulli distribution helps with interpretability; that is, β can be interpreted as the proportion of important features. Certain parameters may be more informative than others when used to identify certain anomalies; thus, regularization for variable selection to avoid overfitting is critical. With regularization, we have a sparse model that removes uncorrelated variables falsely found to be correlated with system health status and the response variable. With defining this variable and the hyperparameter β, we can fully control the sparseness and complexity of the model.

Model Hyperparameters: In this work, we do not assume any strong prior for α, β, and σ and use the following noninformative priors:

$$\alpha \sim \mathscr{U}(0, 1),\ \beta \sim \mathscr{U}(0, 1),\ \sigma^2 \sim \text{Inv} - \text{Gamma}(\gamma_1, \gamma_2),$$

where γ_1 and γ_2 are the hyperparameters associated with the shape and scale parameters of the inverse gamma distribution. The directed acyclic graph associated with the full generative structure is shown in Fig. 1, which represents the dependencies and causal relationships among model parameters.

Assuming that γ_1 and γ_2 are known, the full posterior based on the hierarchical model, which is proportional to the product of the likelihood and priors' probability, can be written as the following:

$$p(\boldsymbol{r}, \boldsymbol{q}, \sigma, \alpha, \beta|\mathscr{D}) \propto p(\mathbf{Y}|\mathbf{X}, \boldsymbol{r}, \boldsymbol{q}, \sigma)p(\boldsymbol{r}|\alpha)p(\boldsymbol{q}|\beta)p(\alpha)p(\beta)p(\sigma^2|\gamma_1, \gamma_2). \tag{5}$$

The first term on the right-hand side is the likelihood probability that includes the probability of all response observations given data. Theoretically, Eq. (5) can be used to find the Maximum-a-Posteriori Estimation (MAP) to obtain point estimates of model parameters. The posterior distribution given in Eq. (5) does not have a closed-form and thus optimizing it analytically (i.e., by taking a gradient over each unknown parameter) is not tractable, given the large number of variables and potentially large $\mathscr{D}$. Below we provide a Markov chain Monte Carlo (MCMC) simulation framework based on the hierarchical structure of the model for model training. This method will effectively use the hierarchical structure to recursively estimate model parameters while accounting for uncertainty in the estimation process. Although the joint distribution is not known explicitly and is impossible to sample from directly, the conditional distribution of each variable is known and is easy to sample from. Thus, we develop a Gibbs sampling algorithm that generates an instance from the distribution of each variable in turn, conditional on the current values of other variables. In other words, at iteration t of the model training algorithm, we sample

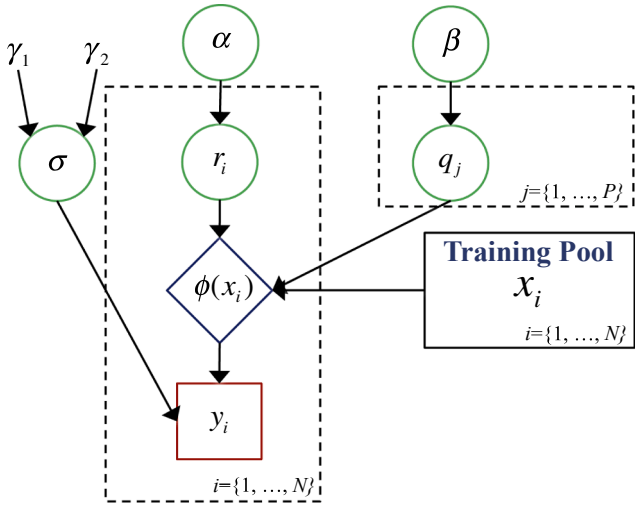

Fig. 1. Directed acyclic graph for the hierarchical model. Circles indicate stochastic nodes; rectangles indicate observable factors; and the rhombus indicates the smoother. The box on the right side represents the training pool. The hyperparameters γ_1, γ_2, and variables α and β regulate the degree of sparsity.

(a) $r_i^{(k)}$ from $p(r_i|r_i^{-(k)})$ using direct Gibbs sampling as

$$p(r_i = v|r_i^{-(k)}) \propto \begin{cases} \alpha\, p(\mathbf{Y}|\mathbf{x}, \boldsymbol{r}^{(k)}, \boldsymbol{q}^{(k)}, \sigma^{(k)}; r_i = 1) & \text{if } v = 1; \\ (1-\alpha) p(\mathbf{Y}|\mathbf{x}, \boldsymbol{r}^{(k)}, \boldsymbol{q}^{(k)}, \sigma^{(k)}; r_i = 0) & \text{if } v = 0. \end{cases} \tag{6}$$

(b) $\boldsymbol{q}^{(k)}$ from $p\left(q_j \middle| q_j^{-(k)}\right)$ using direct Gibbs sampling as

$$p(q_j = v \middle| q_j^{-(k)}) \propto \begin{cases} \beta\, p(\mathbf{Y}|\mathbf{x}, \boldsymbol{r}^{(k)}, \boldsymbol{q}^{(k)}, \sigma^{(k)}; q_j = 1) & \text{if } v = 1; \\ (1-\beta) p(\mathbf{Y}|\mathbf{x}, \boldsymbol{r}^{(k)}, \boldsymbol{q}^{(k)}, \sigma^{(k)}; q_j = 0) & \text{if } v = 0, \end{cases} \tag{7}$$

(c) σ from $p(\sigma|\sigma^{-(k)})$ using direct Gibbs sampling as

$$p\left(\sigma \middle| \sigma^{-(k)}\right) = \text{Inv} - \text{Gamma}\left(\gamma_1 + n/2,\ \gamma_2 + \frac{1}{2}(\mathbf{Y} - \hat{\mathbf{Y}})^T\left(\mathbf{Y} - \hat{\mathbf{Y}}\right)\right), \tag{8}$$

where θ^- refers to all variables except for θ and $\hat{\mathbf{Y}}$ is a N-vector estimate of response $\mathbf{Y}$ obtained from Eq. (2). If α and β are not known a priori, we can use the following equations for Gibbs sampling. In such a situation, the number of model parameters increases to $N + P + 3$.

(d) α from $p(\alpha|\alpha^{-(k)})$ using direct Gibbs sampling as

$$p\left(\alpha \middle| \alpha^{-(k)}\right) \propto \text{Beta}\left(\sum_{i=1}^{N} r_i + 1,\ N + 1 - \sum_{i=1}^{N} r_i\right), \tag{9}$$

(e) β from $p(\beta|\beta^{-(k)})$ using direct Gibbs sampling as

$$p\left(\beta \middle| \beta^{-(k)}\right) \propto \text{Beta}\left(\sum_{p=1}^{J} q_p + 1,\ P + 1 - \sum_{p=1}^{J} q_p\right). \tag{10}$$

Note that (d) and (e), which are derived according to the conjugate property of the uniform distribution on the Bernoulli distribution, are needed only if α and β are assumed to be unknown. The primary benefit of the Gibbs sampler is that it helps avoid computing unfriendly marginal distributions and gradients. The output from the above iterative process is a set of samples

$$\Theta^{(k)} = (\boldsymbol{r}^{(k)}, \boldsymbol{q}^{(k)}, \sigma^{(k)}, \alpha^{(k)}, \beta^{(k)}),$$

for $k = \{1, \ldots, K\}$ iterations after a period of burn-in iterations. Now, we can run a sparse Gibbs sampler for a number of iterations until convergence. For our numerical experiments, we disregard some samples at the beginning (burn-in period). It is clear that the Gibbs sampler is useful when sampling from the conditional distributions for each parameter is possible. In the case that the structure of the hierarchical model is fully changed with a new set of priors so that conditional

distributions cannot be found in closed-forms, the Metropolis-Hastings algorithm can be used as an alternative to Gibbs sampling (it may be more time consuming and less accurate). The summary of the steps for the parameter estimation task is provided in Algorithm 1. It should be pointed out that the users of the model must carefully diagnose the MCMC convergence before using the trained model for anomaly detection. In this article, we evaluate the result of MCMC directly (convergence) and indirectly (using the results obtained from response prediction and anomaly detection).

Algorithm 1. Gibbs Sampler for Parameter Estimation – Summary of the Steps

1. Set $k = 0$ and initialize the unknown variables denoted by $\Theta^{(0)} = [\boldsymbol{r}^{(0)}, \boldsymbol{q}^{(0)}, \sigma^{(0)}, \alpha^{(0)}, \beta^{(0)}]$,
2. **Repeat**
   $k = k + 1$,
   for $i = 1: N$,
   - Sample $r_i^{(k)}$ from Eq. (6),
   end for
   for $j = 1: P$,
   - Sample $q_j^{(k)}$ from Eq. (7),
   end for
   - Sample $\sigma^{(k)}$ from Eq. (8),
   - Sample $\alpha^{(k)}$ from Eq. (9),
   - Sample $\beta^{(k)}$ from Eq. (10),
   - Set $\Theta^{(k)} = [\boldsymbol{r}^{(k)}, \boldsymbol{q}^{(k)}, \sigma^{(k)}, \alpha^{(k)}, \beta^{(k)}]$, **Until** convergence.
3. Output $\{\Theta^{(1)}, ..., \Theta^{(k)}\}$.

3.3. Bootstrapping for prediction intervals and empirical distributions

Bootstrapping is a widely used approach that employs data from one sample to generate a sampling distribution by repeatedly taking random samples from the known sample set with replacement. In this article, we use bootstrap sampling to provide an estimate for the distribution of estimated y_r (denoted by $\hat{y}_r$) for sample r, which can be used to construct prediction intervals. Prediction intervals provide an estimate of an interval in which the response value will fall with a given probability. For a review of some methods that use bootstrapping for prediction intervals, interested readers may refer to [50,51]. Also, the details for bootstrapping's asymptotic behavior are explained in [52]. The outcome of the following steps is an empirical distribution for the prediction y_r from which we can also find the corresponding prediction intervals:

Step 1. For each sample $n = 1, ..., N$ in the finalized training pool, calculate $\hat{y}_n$ from Eq. (2) and from that calculate the residual error as $e_n = y_n - \hat{y}_n$.

Step 2. Resample the training pool for M (to be set by the users) times, each with exactly N samples and with replacement (denote it by $\mathscr{D}_m$). For each bootstrap sample r, calculate the corresponding estimate $\hat{y}$ and denote it by $\hat{y}_{r,m}$, where $m \in \{1, ..., M\}$. Calculate $\mu_{r,m}$ as follows:

$$\mu_{r,m} = \hat{y}_{r,m} - \frac{\sum_{m=1}^{M} \hat{y}_{r,m}}{M}.$$

The set of samples for $\hat{y}_r$ can be recorded in $\mathscr{S}_r$ as follows:

$$\mathscr{S}_r = \{\hat{\phi}(\boldsymbol{x}_r|\mathscr{D}_m) + \mu_{r,m} + e_n,\ m = \{1, ..., M\},\ n = \{1, ..., N\}\},$$

which builds the empirical distribution of $\hat{y}_r$. Now that the empirical distribution of response variables is found, it is possible to find the α-level prediction interval for input vector $\boldsymbol{x}$ as $I_\alpha(\boldsymbol{x}) = \left[\mathscr{S}_{r,\frac{\alpha}{2}}, \mathscr{S}_{r,1-\frac{\alpha}{2}}\right]$, where $\mathscr{S}_{r,\frac{\alpha}{2}}$ and $\mathscr{S}_{r,1-\frac{\alpha}{2}}$ are the $\frac{\alpha}{2}$ and $1 - \frac{\alpha}{2}$ quantiles of the set $\mathscr{S}_r$, respectively. Also, the regions that the prediction cannot perform well can be investigated using the above interval. We will see through numerical experiments that prediction intervals based on our models perform very well.

4. Anomaly detection framework

The ultimate value of statistical techniques for hazard monitoring and reliability analysis lies in their power in generating actionable insights that can help with decision-making. In this section, the structure of the anomaly detection framework and optional operational usability constraints are discussed.

4.1. Structure of the anomaly detection framework

In any data-driven anomaly detection framework, a decision needs to be made as to whether an instance is an anomaly based on the set of available measurements/features. Most available anomaly detection models do not provide an optimal structure for a decision-making framework and often lead to the same prediction regardless of the trade-off between accuracy, false alarms (misclassification of a normal instance as an anomaly), and missed anomalies (misclassification of an anomalous sample as normal). In this section, we develop the structure of an optimal policy for anomaly detection based on two possible scenarios referred to as Case (i) and Case (ii): In Case (i), the training samples are fully labeled, and we know all anomalous instances. That is, some information may be captured regarding the stochastic distribution of response variables under anomaly conditions. In Case (ii), the training samples are not labeled, and no direct information is available regarding the stochastic behavior of response variables under anomaly conditions. The model proposed for the above cases is a cost-sensitive model that aims to minimize the total cost associated with false alarms and missed anomalies.

Let us assume that we are interested in finding whether sample r with measurement vector $\boldsymbol{x}_r$ and response y_r represents an anomaly condition. We define $R_a(\boldsymbol{x}_r, y_r)$ as the total expected risk of classifying sample r as an anomalous sample and $R_{\dot{a}}(\boldsymbol{x}_r, y_r)$ as the total expected risk of classifying sample r as a normal sample. Similar to the majority of cost-sensitive classification models (e.g., in [53]), the words *cost*, *risk*, *penalty*, *benefit*, and *reward* are used in this article in the general sense and may include any criterion in addition to the monetary cost. Thus, the cost function is domain specific and is quantified in arbitrary units. It is assumed in our article that the costs of misclassification and the rewards of correct classification are measured in the same units where the reward is a negative cost. To properly define the cost function so that the trade-off between misclassification errors and detection rates is taken into account, the following cost parameters are defined:

$c_{a|\dot{a}}$: the cost of labeling a normal sample as an anomalous sample (i.e., false positive cost),
$c_{\dot{a}|\dot{a}}$: the reward of labeling a normal sample as a normal sample (true negative reward),
$c_{a|a}$: the reward of labeling an anomalous sample as an anomalous sample (true positive cost), and
$c_{\dot{a}|a}$: the cost of labeling an anomalous sample as a normal sample (false negative cost).

It should be pointed out that the above cost parameters depend on the relative importance of misclassification and true detection and vary from application to application. Decision makers need to define these cost parameters as the inputs for the anomaly detection process. Thus, all cost parameters are assumed to be known a priori in this article. Because the objective of the anomaly detection framework is to minimize the total cost, the numerical values associated with the reward of true detection ($c_{\dot{a}|\dot{a}}$ and $c_{a|a}$) should be nonpositive and the numerical values associated with the cost of false detection ($c_{\dot{a}|a}$ and $c_{a|\dot{a}}$) should be nonnegative. The cost parameters determine the sensitivity of the users for the trade-off between minimizing false alarms and maximizing true detection rates. So, depending on the application and the system

under study, these parameters may vary. For the cases where minimizing false alarms and maximizing true detection rates have the same importance and the user have no preference between these two items, the respective cost parameters can be set equal (which yields $c_0 = 1$). It will also be shown that there is no need to determine the absolute value of the cost parameters and only the unitless measure $\frac{c_{\dot{a}|\dot{a}} - c_{a|\dot{a}}}{c_{a|a} - c_{\dot{a}|a}}$ needs to be quantified. Let us also define $p(a|\boldsymbol{x}_r, y_r)$and $p(\dot{a}|\boldsymbol{x}_r, y_r)$ as the probability of being in an anomaly condition and a normal condition given $\boldsymbol{x}_r$ and y_r, respectively. Thus, we have the following:

$$R_a(x_r, y_r) = c_{a|a}\, p(a|\boldsymbol{x}_r, y_r) + c_{a|\dot{a}}\, p(\dot{a}|\boldsymbol{x}_r, y_r),$$

$$R_{\dot{a}}(x_r, y_r) = c_{\dot{a}|a}\, p(a|\boldsymbol{x}_r, y_r) + c_{\dot{a}|\dot{a}}\, p(\dot{a}|\boldsymbol{x}_r, y_r).$$

The idea is to define a dynamic decision rule (λ) that minimizes the expected total cost $\sum_{i=1}^{N} R_{w_i}(x_i, y_i)$, where $w_i \in \{a, \dot{a}\}$ is the chosen label for sample i and $R_{w_i}(x_i, y_i)$ is the effective cost based on the chosen label w_i. It should be pointed out that depending on the application, decision makers may choose to have other forms of the objective function, such as the average cost per unit of operation. It is clear that a decision that can minimize the expected cost of each sample can minimize the overall expected cost as well. Thus, if we can classify sample r as an anomaly if $R_a(\boldsymbol{x}_r, y_r) < R_{\dot{a}}(\boldsymbol{x}_r, y_r)$ and classify it as normal otherwise, then we have a policy that minimizes the expected total cost/risk. We will show below that there is an optimal structure in a control-limit form that performs better than any other policy, such as confidence interval and fixed-threshold policies. We first simplify the case where anomaly should be chosen for sample r as follows:

$$R_a(\boldsymbol{x}_r, y_r) < R_{\dot{a}}(\boldsymbol{x}_r, y_r) \rightarrow \tag{11}$$

$$p(a|\boldsymbol{x}_r, y_r)[c_{a|a} - c_{\dot{a}|a}] < p(\dot{a}|\boldsymbol{x}_r, y_r)[c_{\dot{a}|\dot{a}} - c_{a|\dot{a}}] \rightarrow$$

$$\frac{p(a\mid \boldsymbol{x}_r, y_r)}{p(\dot{a}\mid \boldsymbol{x}_r, y_r)} > \underbrace{\frac{c_{\dot{a}|\dot{a}} - c_{a|\dot{a}}}{c_{a|a} - c_{\dot{a}|a}}}_{c_0} \rightarrow$$

$$\frac{p(y\mid a, \boldsymbol{x})p(a, \boldsymbol{x})}{p(y\mid \dot{a}, \boldsymbol{x})p(\dot{a}, \boldsymbol{x})} > c_0,$$

where $p(a, \boldsymbol{x})$ is the joint prior distribution of $\boldsymbol{x}$ and an anomaly event. With the independence assumption of the sample's anomaly status and feature vector $\boldsymbol{x}$, we have the following simplified policy:

$$\mathscr{I}_1\left(\boldsymbol{x}_r, y_r\right) = \log\frac{p(y_r|a, \boldsymbol{x}_r)}{p(y_r|\dot{a}, \boldsymbol{x}_r)} > \underbrace{[\log c_0 + \log p(\dot{a}) - \log p(a)]}_{\lambda^*},$$

where $p(\dot{a})$ and $p(a)$ are the prior distribution of normal and anomaly conditions that can be empirically found from data. Now for Case (i), where $p(y_r|a, \boldsymbol{x}_r)$ can be computed from past data, we can classify a sample as an abnormal sample using the following rule:

$$\text{if} \quad \mathscr{I}_1(\boldsymbol{x}_r, y_r) > \lambda^* \rightarrow w_r = a. \tag{12}$$

For Case (ii), where there is no information available with regard to the behavior of the system under the anomaly condition of interest (that is when $p(y_r|a, \boldsymbol{x}_r)$ is not known), we assume a noninformative unconditional probability distribution $p(y_r)$ for $p(y_r|a, \boldsymbol{x}_r)$. Thus, we have

$$p\left(y_r \middle| \dot{a}, \boldsymbol{x}_r\right) < \frac{p(y_r)\, p(a)}{c_0\, p(\dot{a})} \rightarrow$$

$$-\log p\left(y_r \middle| \dot{a}, \boldsymbol{x}_r\right) > \underbrace{-[\log p(y_r) - \log c_0 - \log p(\dot{a}) + \log p(a)]}_{\lambda^{**}} \tag{13}$$

The anomaly detection policy now changes to

$$\mathscr{I}_2(\boldsymbol{x}_r, y_r) = -\log p(y_r|\dot{a}, \boldsymbol{x}_r) > \lambda^{**} \rightarrow w_r = a. \tag{14}$$

Note that λ^{**} depends on the prior distribution of y_r. It can be shown that when $p(y|\dot{a}_r, \boldsymbol{x}_r)$ is uni-model, then the anomaly detection policy can be rewritten as an interval policy with lower bound g_1 and upper bound g_2 where

$$\text{if } y_r \notin [g_1, g_2] \rightarrow \text{sample} r \text{is an anomalous point},$$

where g_1 and g_2 are the solution of $-\log p(y_r = g|\dot{a}, x_r) = \lambda^{**}$. Now, to apply the above policy for a new sample, it is sufficient to have λ^* and λ^{**} as the anomaly indices. It can be seen that both $\mathscr{I}_1(x_r, y_r)$ and $\mathscr{I}_2(x_r, y_r)$ are anomaly indices. The larger these anomaly indices, the more likely the sample is an anomalous sample. It is clear from Eqs. (12) and (13) that as long as the cost ratio of c_o or $\frac{1}{c_0}$ is known, the proposed anomaly detection framework can be applied regardless of whether the absolute values of the cost parameters are known.

Because in practice, we may not be able to accurately calculate λ^* and λ^{**} using their closed-forms, we need to be able to tune them with data. Below, we provide a general optimization problem that can be used to find the optimal thresholds λ^* and λ^{**} using an empirical risk model. Given N_1 samples with known x, y, and true labels for anomaly (that is $o_1, \ldots o_{N_1}$), we have.

$$\text{Min} \quad J\left(\lambda^*, \lambda^{**}\right) = \frac{1}{N_1}\sum_{i=1}^{N_1} c_{w_i|o_i} \tag{15}$$

$$\text{s. t.} \qquad w_i = \begin{cases} a & \text{if } \mathscr{I}_1(x_r, y_r) > \lambda^* \text{ in Case(i)}, \\ a & \text{if } \mathscr{I}_2(x_r, y_r) > \lambda^{**} \text{ in Case(ii)}, \\ \dot{a} & \text{Otherwise.} \end{cases} \quad i = 1, \ldots, N_1.$$

$$-\infty \leqslant \lambda^* \leqslant +\infty, \quad 0 \leqslant \lambda^{**} \leqslant +\infty,$$

The result presented in this section shows that using confidence or prediction intervals or other similar methods for anomaly detection is not necessarily optimal when there are some risks associated with false alarms and missed anomalies. It should be pointed out that although the anomaly indices are not cost- or risk sensitive, the anomaly detection thresholds λ^* and λ^{**} change with the cost parameters. Thus, the anomaly detection task/framework is in fact cost- or risk sensitive. An overview of all necessary steps in the proposed anomaly detection framework is shown as a flowchart in Fig. 2.

4.2. Additional constraints – operational usability factors

For many systems, it may be desirable to build an anomaly detection framework with a lower bound for the true detection rate (or an upper bound for the missed anomalies rate) and an upper bound for the false alarm rate in addition to minimizing the total risk. To achieve such a limitation, the following two constraints may be added to the model given in Eq. (15):

$$\frac{\sum_{i=1}^{N} \mathbf{1}\left\{o_i = w_i = a\right\}}{\sum_{i=1}^{N} \mathbf{1}\left\{o_i = a\right\}} \geqslant P_0, \tag{16}$$

$$\frac{\sum_{i=1}^{N} \mathbf{1}\left\{o_i = \dot{a}, w_i = a\right\}}{\sum_{i=1}^{N} \mathbf{1}\left\{w_i = a\right\}} \leqslant P_1, \tag{17}$$

where P_0 and P_1 are respectively the desired lower bound of the true detection rate and the desired upper bound of the false alarm rate. In this model, as long as the feasible region is not empty, the upper bound of the false alarm rate and the lower bound for the true detection are guaranteed. In case there is no feasible solution, we can add techniques, such as goal programming, to transform constraints (16) and (17) to goal constraints with target values P_0 and P_1, in the sense that unwanted deviations from this set of target values are minimized. The above

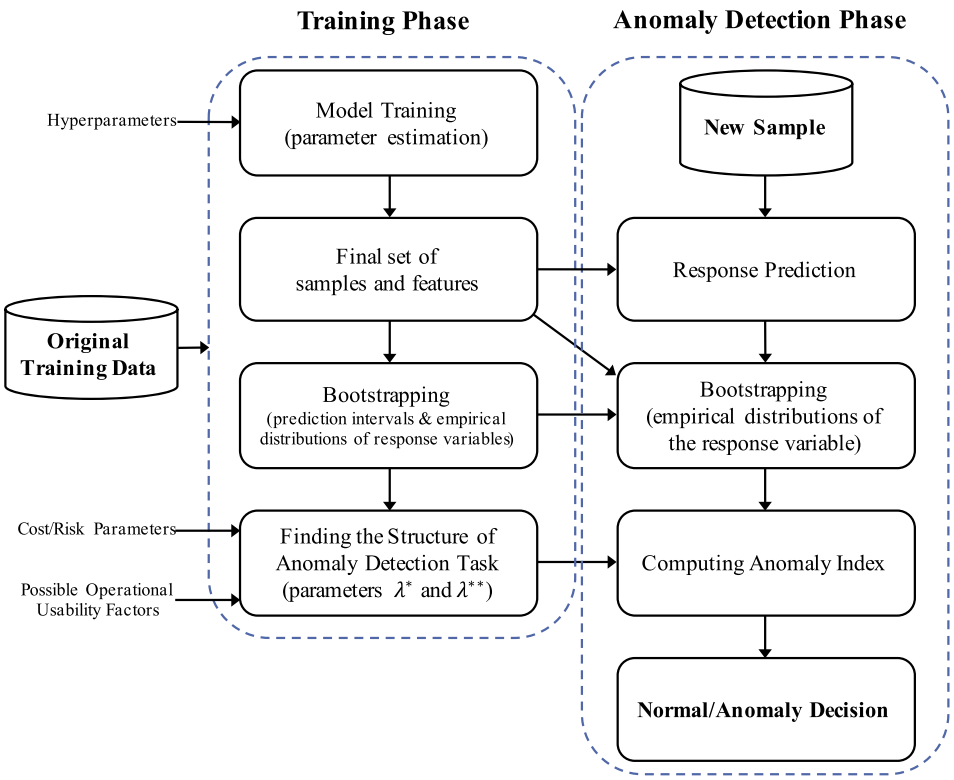

Fig. 2. Summary of all steps for model training and anomaly detection. The inputs of the framework are the set of historical data (training data), hyperparameters, cost/risk parameters, and possible operational usability factors. Note that once the effective set of training samples and features are determined in the training phase, they are removed from the model and are not used during the anomaly detection phase. The thresholds λ^* and λ^{**} can be found theoretically, empirically, or from experience.

problem is a one-dimensional optimization problem with one unknown (λ^* or λ^{**}) and can be easily solved with any single-variable optimization technique. To account for uncertainty, we can use the posterior mean of λ^* or λ^{**} using the generated samples from MCMC.

5. Numerical experiments – simulation

In this section, we demonstrate how our model can be used for anomaly detection using simulation data. We also discuss the performance of the parameter estimation method, prediction intervals, and anomaly detection framework with a set of numerical experiments.

5.1. Simulation setup

The primary objective of the simulation experiments in this article is to evaluate the accuracy and performance of the model with regard to model training, ability to find important samples and remove anomalies from data, predicting system's response, and detecting anomalies in both supervised and unsupervised settings. We also compare our model with similar approaches. We consider a multivariate nonlinear model, which is adapted from [52]. The details of this model are as follows:

- There are 5 features drawn from a multivariate normal distribution with mean vector [0.1,0.2,0,0.05,1.2] and covariance matrix

$$\begin{bmatrix} 1 & 0.43 & 0.45 & -0.29 & -0.69 \\ 0.43 & 1 & 0.25 & -0.36 & -0.36 \\ 0.45 & 0.25 & 1 & -0.91 & -0.36 \\ -0.29 & -0.36 & -0.91 & 1 & 0.49 \\ -0.69 & -0.36 & -0.36 & 0.49 & 1 \end{bmatrix}.$$

- The system's response under normal conditions is generated based on the following nonlinear system:

$$y_i = \exp(x_{i,1}) + x_{i,2}x_{i,3}^2 + \log(|x_{i,4} + x_{i,5}|) + \epsilon_i, \quad (18)$$

where ϵ_i is drawn from $\mathcal{N}(0, 0.01)$. We consider multiple settings by changing the anomaly ratio (proportion of anomalous samples) in the simulated data sets. For example, for a case of 10% anomaly ratio, a total of 10% of the samples are randomly selected to be anomalous samples and do not follow the formula given in Eq. (18). Based on the anomaly ratio (denoted by g) defined for each setting, we randomly generate a binary indicator for each sample to denote whether it is normal or an anomalous sample. In other words,

$$o_i \sim \textbf{Bern}(g), \quad \forall\, i.$$

- The system's response under anomaly conditions is generated based on Eq. (18) except that ϵ_i is drawn from $\mathcal{N}(3.3, 0.01)$.
- To evaluate the power of the model to identify important features, we considered two additional features with no effect on the response y. These two features are randomly generated for each sample using a uniform distribution

$$x_{i,5}, x_{i,6} \sim \textbf{U}(-4, 4), \quad \forall\, i.$$

We selected -4 and 4 as the bounds because they are the closest integers to the minimum and maximum values of features 1–5, respectively. The outcomes of the simulated procedure for a predetermined number of samples N and number of features P are (a) binary variables o_i reflecting whether sample i is anomalous, (b) feature values $x_{i,p}$, and (c) the response variables y_i, for $i = \{1, ...,N\}$ and $p = \{1, ...,P\}$. For each setting, we generate 2,500 samples (2000 for training and 500 for testing).

5.2. Markov Chain Monte Carlo performance and its ability in model training

We first evaluate the performance of the parameter estimation (model training) process. A total of 2000 samples are used for training in the MCMC algorithm. After running the MCMC for a number of iterations, we chose the mean of the samples as the estimate for the parameter of interest. We also repeated the experiments by considering the most likely value of the desired parameter (the mode) as the estimate. We chose the sample value that occurred most commonly, and

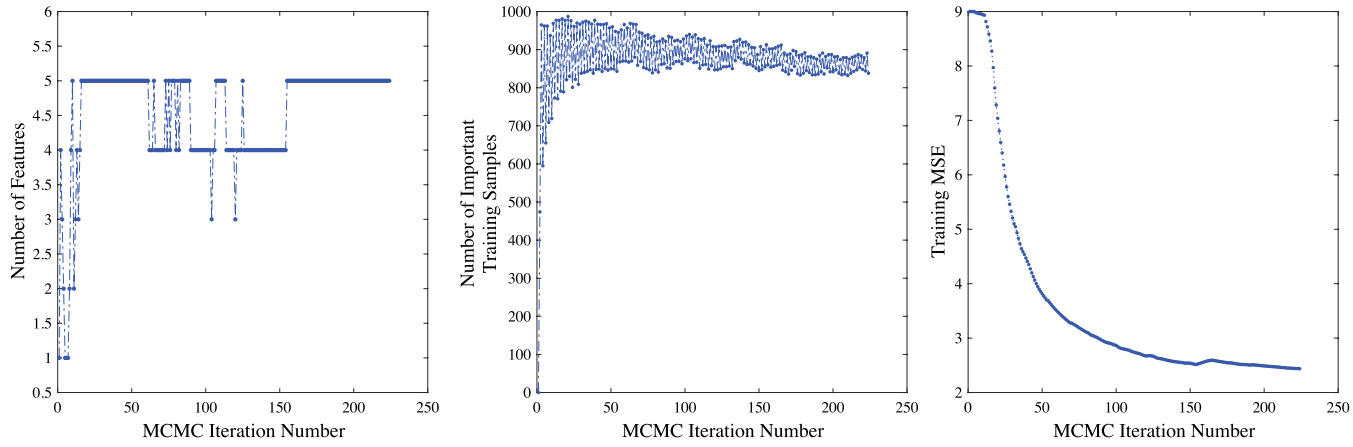

Fig. 3. The results from MCMC iterations. The results shown for each iteration are based on the mean posterior of all variables up to that iteration. These plots verify that the MCMC had a reasonable convergence and performance.

the results were almost the same. To observe how the MCMC iterative process improves the model training error over iterations and find the best number of samples and features, we plotted the results of MCMC for 230 iterations in Fig. 3. It can be seen from the results shown in Fig. 3 that.

(a) the number of important features converges to 5 (which is the true value for P),
(b) the number of important samples converges after some iterations (around 870), and
(c) the mean squared error of response variables in the training data decreases as the number of iterations increases.

All of the above observations verify that the developed MCMC process is able to estimate a reasonable model after a few iterations. It should be noted that to have better mixing and convergence, we can run the model for more iterations. Each iteration of MCMC using Matlab 2017 on a computer with 32 GB, 1600 MHz DDR3, and 4 GHz Intel Core i7 takes approximately 30 s to sample for all parameters (around 2500 variables). Running the experiment for 10,000 iterations gave us a better mixing; however, the mean posteriors were almost unchanged. We believe one of the reasons that the MCMC converges relatively fast was the existence of closed-forms for conditional distributions and the hierarchical structure of the model making Gibbs sampler an efficient parameter estimation process. To make sure that the outcome of the model after removing some features is sufficient, we can check the error in the estimated values of the response variable for the samples in the testing set. As long as the model can accurately predict the response variable using the selected set of features, we assume that the removed features are not important or not necessary to be included in the model. In real applications where the true values of model parameters and numbers of important features and samples are not known, it is possible to use MCMC convergence diagnostics along with performance measures, such as the MSE of predicted response values in the training and/or testing set to evaluate whether trained model is trustable to be further used for anomaly detection. We should point out that the use of binary indicators for each sample (r_i) may impact the computational complexity of the training phase, particularly for large data sets. Also, estimating model parameters when sufficient data are not available is a challenging task.

5.3. Ability to find important samples and remove anomalies from training data

As discussed before, one of the main features of our model is the ability to select a subset of the training pool as the effective set of samples. This feature can help with computational complexity and removing anomalous points from the training data so that they don't disturb the development of the normal behavior model as in Eq. (5.5). To numerically evaluate this feature, we considered 11 different settings based on the anomaly ratios of $g \in \{0{:}\ 0.1{:}\ 10\}$. For each setting, we simulated 2500 samples (2000 for training), ran MCMC, and then checked the final results to find out what % of anomalous points is selected as unimportant (i.e., when $r_i = 0$). We also checked the total number of samples selected for each setting (i.e., $\sum_{i=1}^{2000} r_i$). In Fig. 4, we plotted the number of effective samples with $r_i = 1$ obtained from parameter estimation (left y axis) versus the anomaly ratio (%) in the original training data (x axis) for the 11 settings defined earlier (the dashed blue line with ×). Also, we plotted the % of anomalous samples removed after parameter estimation (right y axis) versus the anomaly ratio (%) in the original training data (x axis) for the same 11 settings (the dashed red line with circles). For instance, it can be seen that when the anomaly ratio in the training data are 10% (that is 200 out of 2000 samples are noisy), then the results of parameter estimation show that around 1150 samples (read from the left y axis) are selected as important/effective samples (i.e., $\sum_{i=1}^{2000} r_i \approx 1,150$) and around 94% (read from the right y axis) of all anomalous samples (i.e., around 188 out of 200 anomalous samples) are removed after parameter estimation. Results in Fig. 4 are summarized below:

(a) The number of effective training samples is much lower than the 2000 samples in the original training set. This is an important computational benefit because fewer samples can be employed later for the anomaly detection task.
(b) In almost all 11 settings defined based on the anomaly ratio in the training data, the majority of the anomalous samples are removed after parameter estimation (i.e., their corresponding r_i is zero). This is important because noisy data in the training set can negatively impact the anomaly detection task.
(c) As the number of anomalous samples in the training data (anomaly ratio) increases, more samples are needed in the effective training pool. This is in complete agreement with our intuition.

We should point out that for item (a) above, finding all anomalous samples in the training data is not always guaranteed with our model. When possible, users of the model should make all possible efforts to remove such points so that the normal condition model is better trained. In real applications where the set of important features and samples are not known apriori, users can run similar experiments to shrink the training set for the task of anomaly detection.

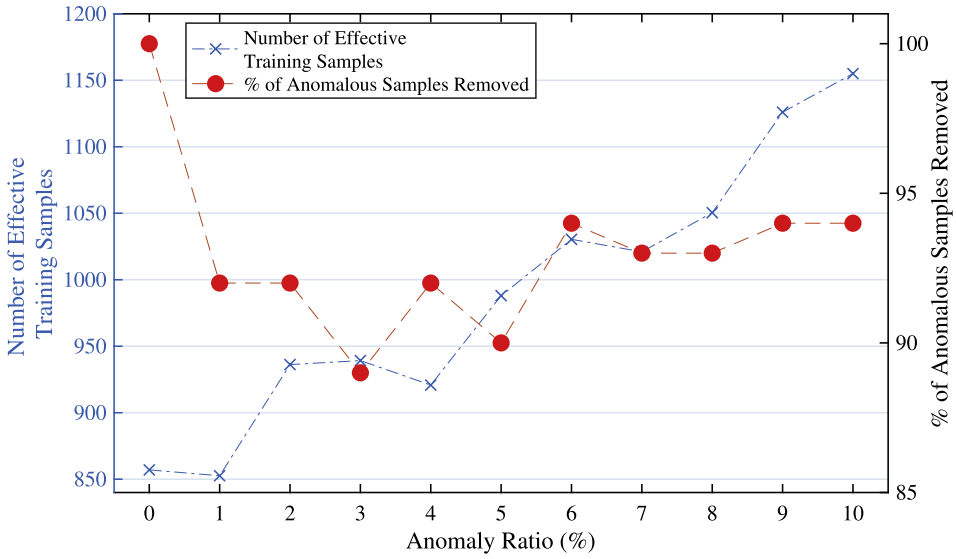

Fig. 4. The number of effective samples with $\pi_i = 1$ obtained from parameter estimation (left y axis) and the % of the anomalous samples removed after model training (right y axis) versus the anomaly ratio (%) in the original training data (x axis).

5.4. System's response prediction

To show how the trained model can predict the response variable accurately, we plotted the true and estimated response values for the case of 10% anomaly ratio using our model and a regular kernel regression (that is, when $q_{1:J} = 1$ and $r_{1:N} = 1$). It can be seen from the results shown in Fig. 5 that our model generates a much better prediction for the response variable while using fewer training samples and features. Also, the residuals of the estimates (right plot) are very close to zero in our model and are smaller than the regular kernel. As expected, both models behave relatively poorly at the boundaries. In our future work, we will investigate methods such as boundary kernel and local likelihood density estimation [54] to address this important shortcoming. In the next subsection, we use a performance measure (root mean squared error [RMSE]) to numerically evaluate the ability of the model in response prediction.

5.5. Comparison with similar models for modeling the response variable

We compared the application of our model for response variable prediction with four similar and widely used models: neural network regression, SVM regression, KNN regression, and kernel regression. We chose these models because of their popularity and the fact that they are technically similar to our model. Also, we used the basic model of each type (as a representative), because we did not aim to show that our model outperforms all these models and their extensions. We briefly explained each of these benchmark models below. In all of these models, the relationship between $\boldsymbol{x}$ and y for sample n is formulated by the following equation:

$$y_n = \phi(x_n, \text{ training data}) + \epsilon_n,$$

where $\epsilon_i, \ldots, \epsilon_N$ are iid random variables with mean 0 and variance σ^2. The difference between these models is on how they define ϕ.

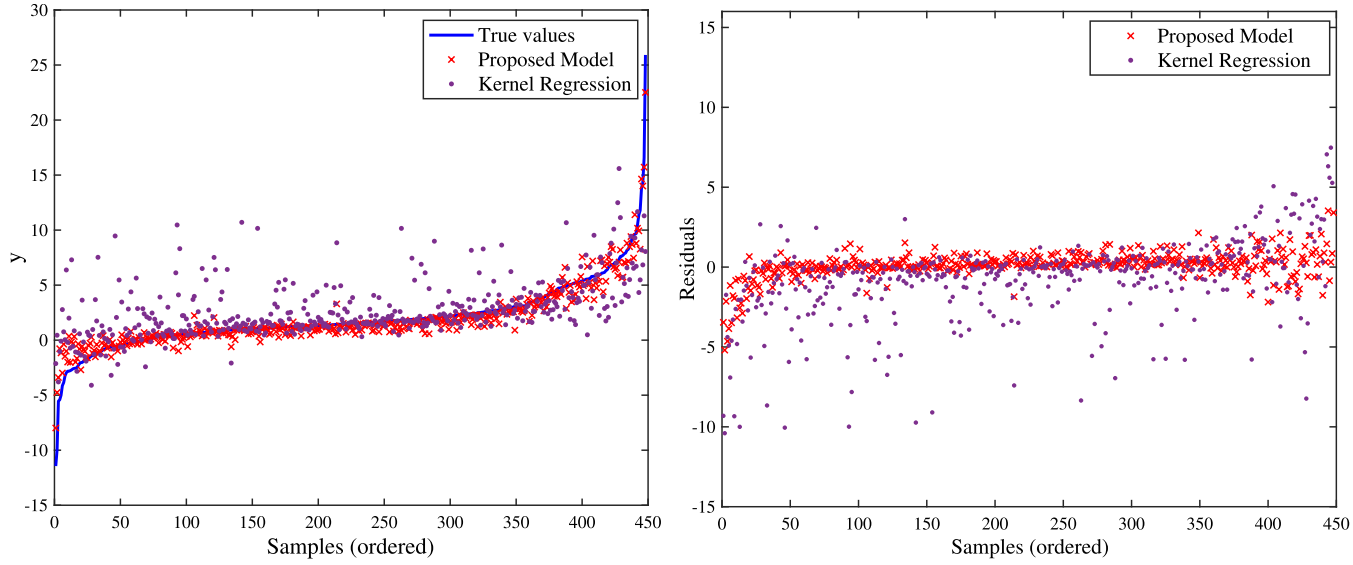

Fig. 5. Prediction of the response variable (in the test set) using our model and the kernel regression for the case of 10% anomaly ratio. The right plot is the residual (true-estimate) plot. The closer the points are to zero, the better prediction we have.

- **Neural Network Regression**: This is an extension of a linear regression, where neural network is used for representing nonlinear mappings between the multi-dimensional input vector and the response variable using some number of layers of hidden units. For each set of experiments using neural network, the number of input neurons is the same as the number of attributes, there is one output layer, and the number of hidden layers is a number between 1 and 10 that was found with a 5-fold cross-validation procedure.
- **SVM Regression**: The main idea of SVM regression is to map the multivariate input data into higher dimensional space using a nonlinear mapping function (kernel) and then perform linear regression in higher dimensional space to minimize a loss function. The SVM parameter, which is the σ in the Gaussian or radial basis function, was optimized using the Hyperparameter Optimization tool in MATLAB.
- **KNN Regression**: In KNN regression, which is in the class of linear smoothers, the average of the response of the K nearest neighbors is used to estimate the response value of a new sample. The main step in KNN regression is to select the K value. This parameters determines the number of neighbors when a value to any new observation is assigned. In this article, we found the parameter K by cross-validation as well (chosen from $K \in \{1{:}\ 20\}$).
- **Kernel Regression**: The kernel regression model is also in the class of linear smoothers and can be considered a special case of our model in which $r_i = 1$ for all samples and $q_j = 1$ for all features. For the kernel regression, we considered the same Gaussian kernel used in our model. All other parameters of the above models were set as the Matlab default.

For each model, we trained the model based on training data and then calculated the mean squared error as the performance measure on the test sets. We also considered 10 levels of anomaly ratio in the training data ($g \in \{1\%, \ldots, 10\%\}$).Results shown in Table 1 verify that our model outperforms SVM regression, KNN regression, and kernel regression and has comparable results with the neural network. We looked more deeply at the estimated values of the response variable using our model and the neural network to find out why neural network works better in some cases. We found that our model provides worse estimates at the boundaries, which is not surprising because of the fact that it is based on a kernel function, which is known to behave poorly close to boundaries. Given that neural network models are highly black box and are not probabilistic (i.e., they normally do not account for uncertainty), our model is a strong competitor because of its interpretable structure and ability to accommodate uncertainty. The proposed model also uses only a subset of training samples and features for prediction. In addition, unlike a typical neural network, our model can be used for cost-sensitive anomaly detection and can consider operational usability factors, as defined in Section 4.2. Because the values of the systems inputs and response variables are assumed to be known over time, the user of our model can simply run similar experiments to evaluate the power of the trained model in detecting known anomalies.

Table 1
MSE of the estimates for the proposed model and benchmark models.

Anomaly ratio (%)	Proposed Model	Neural Network	SVM Regression	KNN Regression	Kernel Regression
0	1.39	1.02	7.54	9.43	4.47
1	1.41	1.16	7.63	9.5	4.86
2	1.45	1.11	7.53	9.46	4.87
3	1.62	1.84	7.58	9.48	5.09
4	1.43	1.5	7.68	9.43	5.90
5	1.12	1.60	7.56	9.46	6.37
6	1.14	1.39	7.44	9.24	6.75
7	1.26	2.07	7.36	9.52	7.30
8	1.37	2.08	7.46	9.32	8.06
9	1.18	2.04	7.47	9.37	8.75
10	1.23	2.76	5.19	7.99	7.55

5.6. The prediction interval for the response variable

Here, we show that the model trained from our framework can be used to provide reasonable prediction intervals for the response variable. In Fig. 6, we plotted the true response from the model in Eq. (2), the estimated values, and the 95% prediction intervals using our model and the regular kernel regression. The samples are sorted based on the true response values. It can be seen from Fig. 6 that the confidence intervals capture the variation in the data and cover the true values reasonably well. Also, it can be seen that our model provides better (narrower) prediction intervals than the regular kernel regression. To better compare the results, we plotted the residual values and their corresponding histograms for the proposed model and the kernel regression model in Fig. 7. Results shown in this figure verify that the residuals in our model are more centered around zero. In Table 2, we reported the desired and the observed coverage samples for the proposed prediction interval for confidence levels of 1%, 2%, 5%, 10%, 15%, and 20% and three anomaly ratios of 0%, 5%, and 10%. As expected, the desired and observed coverage values are reasonably close to each other. The users of the proposed framework can run similar experiments to evaluate the power and the level of uncertainty in representing the response variable in terms of the system inputs.

5.7. Anomaly detection

Our model can be used in both supervised and unsupervised settings (where the stochastic behavior of the system under anomaly conditions may or may not be known in the samples of the training set). For the unsupervised setting, our model can be applied for examining unknown anomalies that were not observed before with the assumption that the response variable behaves differently under an anomalous condition compared to a normal condition. We present the performance of our model for both unsupervised and supervised settings in the following two subsections.

5.7.1. Unsupervised settings with unknown anomalies

In many real applications, the training data do not have sufficient information with respect to the behavior of the system under anomaly conditions. This can happen for various reasons, such as, data collection issues under anomaly conditions and the generation of new anomalies over time that have not been seen before. Here, we assume that the behavior of the system under anomaly conditions is not known. To check how sensitive the results are with respect to the risk parameters, we considered five different cases for the risk ratio as $\frac{c_{\hat{a}|a}}{c_{a|\hat{a}}} \in \left\{0.25, 0.5, 1, 2, 4\right\}$.With changing this ratio, we evaluate the sensitivity of the model with respect to the relative importance of true detection rate versus the false alarm rate. We considered three main models and their extensions as benchmark models. First, we used kernel regression for response modeling (but still our model for anomaly detection) as it is the closest model to ours. Then we considered the model given in [52], which is based on prediction intervals and testing for anomalies with respect to the conditional distribution $f(y|x)$. Based on this cost insensitive model, a prediction interval at level of α, denoted by $I_\alpha(x)$, is first calculated for any new point $\boldsymbol{x}$ based on the results shown in Section 3.3. If the observed response value is not within the range of this interval, then the point is considered an anomalous point.

We considered five values of $\alpha \in \{1\%, 5\%, 10\%, 15\%, 20\%\}$. We also considered another alternative model, which is the prediction interval with optimal quantiles α_1^* and α_2^*, denoted by $I_{\alpha_1^*,\alpha_2^*}(x)$. This model is the one from a set of prediction intervals $I_{\alpha_1,\alpha_2}(x)$ (where α_1 and $\alpha_2 \in \{0{:}\ 0.01{:}\ 0.2\}$) that provides the best detection results in the training set. A sample is considered to be an anomalous sample if its predicted

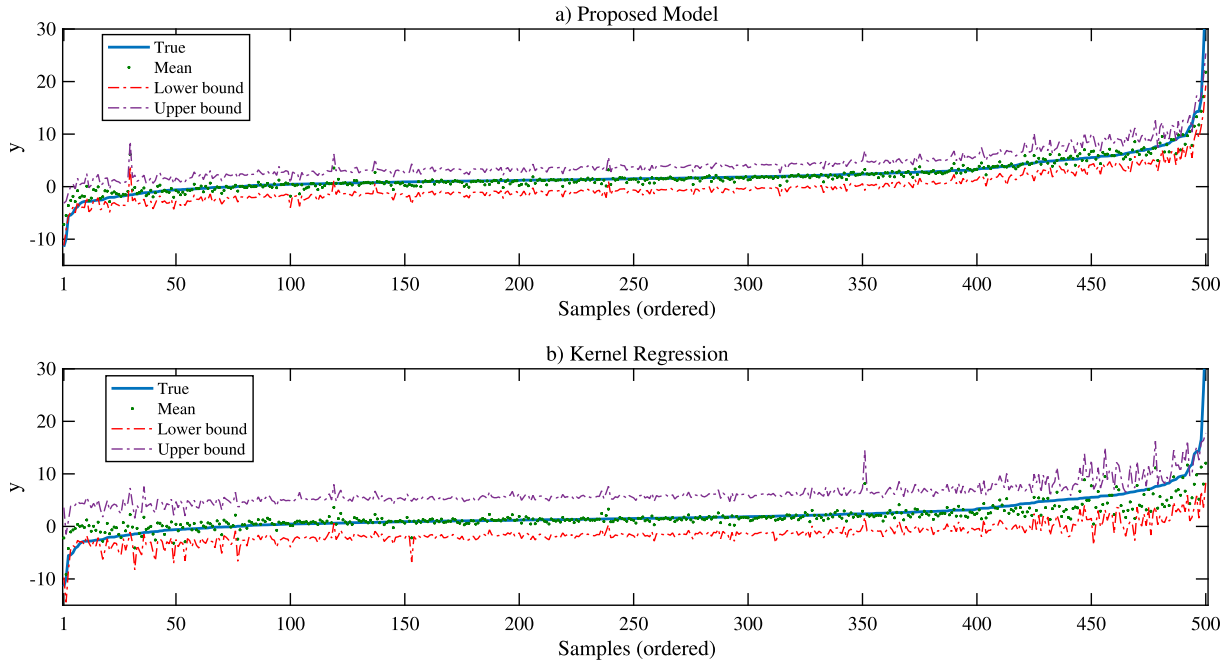

Fig. 6. Comparison of the 95% prediction intervals based on our model and the regular kernel regression. The Mean reported is the estimated values. It can be observed that our model provides very good coverage of true values, particularly for nonboundary points.

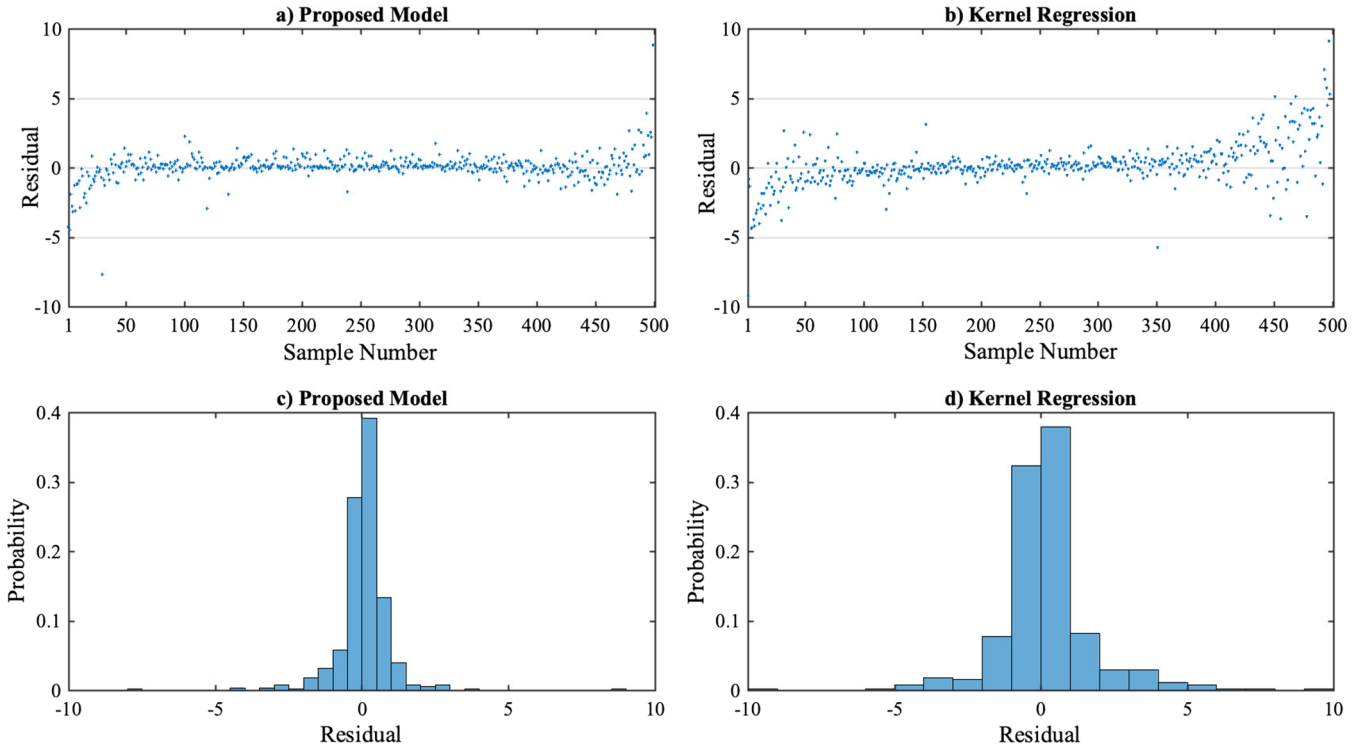

Fig. 7. The residual values (true values - estimated values) and their corresponding histograms for the proposed model and the kernel regression model.

Table 2
Desired and observed coverage of the proposed model - reported as $500 \times (1 - \alpha)$ for $\alpha \in \{1\%, 2\%, 5\%, 10\%, 15\%, 20\%\}$.

Anomaly ratio (%)	Desired ($500 \times (1 - \alpha)$ for)	495	490	475	450	425	400
0	Observed	496	493	481	459	443	419
5		496	490	477	459	441	427
10		497	491	478	463	438	418

response is outside the range defined by $I_{\alpha_1^*,\alpha_2^*}(x)$. The last competitor was the commonly used method of studentized residuals in which all observations whose studentized residuals are greater than 3 in absolute value are considered possible anomalous samples. We refer to this model as the S-R model in the remainder of the article. As mentioned earlier for Eqs. (12) and (13), as long as the ratio of c_o or $\frac{1}{c_0}$ is known, we do not need to know the absolute values of the cost/risk parameters. For each combination of $\frac{c_{\hat{a}|a}}{c_{a|\hat{a}}}$ and each model, we report the average total risk (reported as the risk score), the true detection rate (%), the

Table 3
Comparison between our model and similar models for anomaly detection; in Settings 1 and 2, the anomalous samples follow $\mathcal{N}(0, 5)$ and $\mathcal{N}(0, 30)$, respectively. The term Optimal refers to our proposed model.

Measure>	Total risk					True detection rate (%)					False alarm rate (%)					Accuracy (%)				
$\frac{c_{\hat{a}\|a}}{c_{a\|\hat{a}}}$ Ratio >	0.1	0.5	1	2	10	0.1	0.5	1	2	10	0.1	0.5	1	2	10	0.1	0.5	1	2	10
Setting 1																				
Optimal	26.20	82	140	240	440	50	64.2	64.8	77.2	98.1	5.8	10.3	11	26.9	54.4	82.8	86	86	83.4	61.4
$I_{1\%}(x)$	28.80	112	216	424	2088	35.8	35.8	35.8	35.8	35.8	6.5	6.5	6.5	6.5	6.5	78.4	78.4	78.4	78.4	78.4
$I_{5\%}(x)$	29.40	99	186	360	1752	46.3	46.3	46.3	46.3	46.3	7.4	7.4	7.4	7.4	7.4	81.4	81.4	81.4	81.4	81.4
$I_{10\%}(x)$	31.20	84	150	282	1338	59.3	59.3	59.3	59.3	59.3	8.6	8.6	8.6	8.6	8.6	85	85	85	85	85
$I_{15\%}(x)$	43.60	90	148	264	1192	64.2	64.2	64.2	64.2	64.2	13.3	13.3	13.3	13.3	13.3	85.2	85.2	85.2	85.2	85.2
$I_{20\%}(x)$	55	99	154	264	1144	66	66	66	66	66	17.1	17.1	17.1	17.1	17.1	84.6	84.6	84.6	84.6	84.6
$I_{\alpha_1^*,\alpha_2^*}(x)$	24.60	83	148	262	1144	42.6	56.2	61.1	65.4	66	4.2	6.2	10	15.2	17.1	80.8	84.6	85.2	85	84.6
S-R	32.40	130	252	496	2448	24.7	24.7	24.7	24.7	24.7	9.1	9.1	9.1	9.1	9.1	74.8	74.8	74.8	74.8	74.8
Setting 2																				
Optimal	7.80	21	34	58	184	87.4	91.4	91.4	94	97.4	1.5	2.8	2.8	7.2	26.1	95.8	96.6	96.6	96	88.8
$I_{1\%}(x)$	9.60	24	42	78	366	88.1	88.1	88.1	88.1	88.1	2.2	2.2	2.2	2.2	2.2	95.8	95.8	95.8	95.8	95.8
$I_{5\%}(x)$	16.60	27	40	66	274	91.4	91.4	91.4	91.4	91.4	4.8	4.8	4.8	4.8	96	96	96	96	96	
$I_{10\%}(x)$	18.20	27	38	60	236	92.7	92.7	92.7	92.7	92.7	5.4	5.4	5.4	5.4	96.2	96.2	96.2	96.2	96.2	
$I_{15\%}(x)$	25.80	33	42	60	204	94	94	94	94	94	7.8	7.8	7.8	7.8	7.8	95.8	95.8	95.8	95.8	
$I_{20\%}(x)$	33.80	41	50	68	212	94	94	94	94	94	10.1	10.1	10.1	10.1	10.1	95	95	95	95	95
$I_{\alpha_1^*,\alpha_2^*}(x)$	6.80	22	34	58	204	84.1	92.1	92.1	92.1	94	0.8	3.5	3.5	3.5	7.8	95	96.6	96.6	96.6	95.8
S-R	9.40	23	40	74	346	88.7	88.7	88.7	88.7	88.7	2.2	2.2	2.2	2.2	2.2	96	96	96	96	96

false alarm rate (%), and the accuracy (%) of the model on the test set. We considered two separate noise levels to generate anomalous points. To do this, we generated the response variables using a normal distribution with the same mean of zero but different variances of 5 and 30, respectively. We refer to these two settings as Setting 1 and Setting 2 in Table 3, respectively. It is clear that the higher the variance, the farther the response variables associated with anomalous points are from normal points and thus the more likely they should be detected. Results shown in Table 3 are summarized below:

(a) All models (except for our model and the model based on $I_{\alpha_1^*,\alpha_2^*}(x)$) are cost insensitive; thus, the performance measures of true detection rate, false alarm rate, and accuracy remain constant with changing the cost parameters.
(b) As expected, as the ratio of the risk of missing anomalies to false alarms increases, all models provide higher true detection rates and false alarm rates.
(c) Our model provides lower risk scores compared to all other models.
(d) When the cost ratio c_0 is 1 (that is when classification reward and misclassification errors have the same reward/cost), our model still provides the highest accuracy and the lowest risk score.

When the response variable of anomalous points is very far from a normal behavior model (that is when it is generated based on Setting 2), the performances of all models become very close. This is because anomalous points are easier to detect. To show how the performance of the model changes with different values of c_0, we plotted the total risk (as a performance measure) versus c_0 for Settings 1 and 2 in Fig. 8. Results shown in this figure verify that the optimal policy proposed in this article performs better in terms of reducing the total cost. Also, the performance of models gets closer to each other for Setting 2. It should be reminded that the experiments described in this subsection are only applicable to cases where labeled training data is available. Also, the users need to define the cost elements used in c_0.

5.7.2. Supervised settings with known anomalies

In this case, we assumed that some past samples (or information) are available for the anomalies of interest in the training data. We then used the formula given in Eq. (12) to evaluate the power of our model for separating normal from anomalous samples in the test set. In Fig. 9, we plotted the empirical threshold derived from optimization model (Eq. (15)) and the theoretical threshold derived from Eq. (12) for anomaly detection. We repeated the experiments using 1000, 2000, 5000, and 10,000 for various values of $\frac{c_{\hat{a}|a}}{c_{a|\hat{a}}}$. It can be seen that the empirical and theoretical λ^*s get closer as we have more data. We then used the threshold found from 2000 samples to perform the task of anomaly detection. The left plot in Fig. 10 shows the relationship between true detection rate, false alarms, and the ratio $\frac{c_{\hat{a}|a}}{c_{a|\hat{a}}}$. It can be seen that, as expected, increasing the cost of missing anomaly increases both the true detection and the false alarm rates. The right plot in Fig. 10 shows how the average risk score changes with changing the ratio $\frac{c_{\hat{a}|a}}{c_{a|\hat{a}}}$. To visually see how different thresholds can change the performance of the anomaly detection framework, we plotted the anomaly index for all samples in Fig. 11 together with the λ^*s associated with the ratio $\frac{c_{\hat{a}|a}}{c_{a|\hat{a}}} \in \left\{0.25, 0.5, 1, 2, 4\right\}$. It can be seen from the results in this figure that as the anomaly threshold increases, fewer samples are going to be labeled as anomalous samples. Our results have useful applications for cost-sensitive decision-making situations in which decision makers need to easily analyze how the main performance measures change under different scenarios.

With regard to the anomaly detection task in real applications, finding reasonable estimates for anomaly indices and the optimal values of the thresholds from past data can be problematic, particularly when enough data are not available.

6. Application on wind turbine condition monitoring

The main application domain selected in this article is wind energy, which is presently the fastest growing renewable energy source in the world [55]. The wind industry is challenged by underperforming turbines and premature component failures, which result in increased cost of energy for wind power generation. Both of these drivers can potentially be addressed to some extent by the anomaly detection methodology introduced in this research. In this section, we demonstrate the application of our model for power curve prediction and wind turbine anomaly detection. The data set used in this article is obtained from Riso-DTU (located at Technical University of Denmark) [56]. The original data set has measurements recorded every 10 min for a few different wind farms, where every data set has its own set of turbine

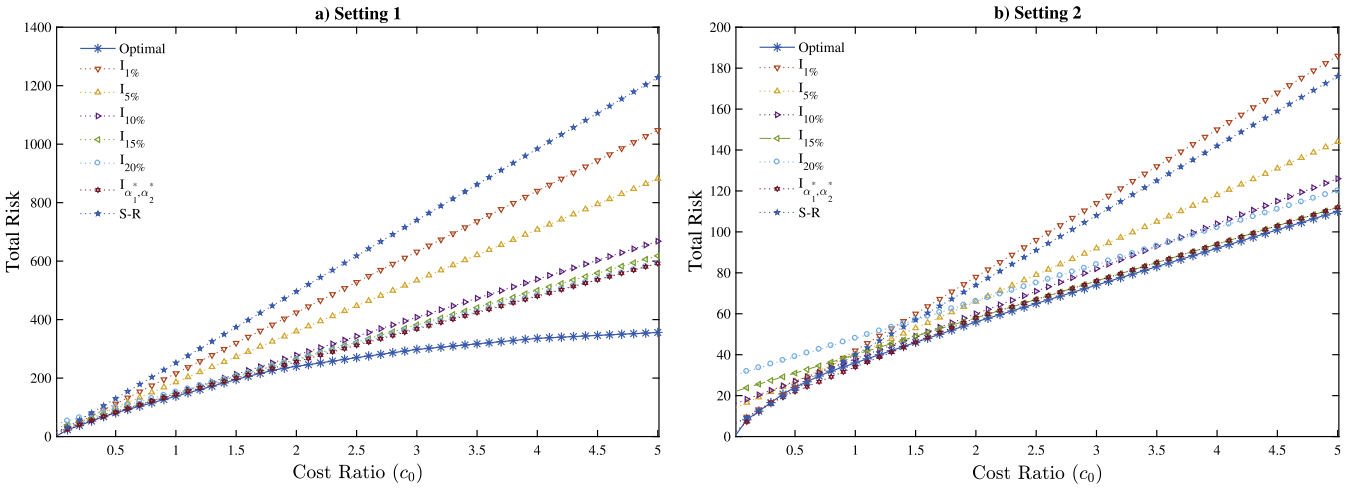

Fig. 8. The total risk associated with the anomaly detection versus the cost ratio (c_0) for the proposed model and the benchmark models for Setting 1 and Setting 2.

characteristics, sensors, and operational and environmental conditions. We considered the Nordtank data set with a stall controlled wind turbine of 500 kW with a diameter of 41 m, 3 blades, and a data acquisition system for recording meteorological properties, wind turbine operational properties, and structural loads in terms of strain gauge measurements. In this article, the sensor measurements are wind speed (2 sensors), wind direction (2 sensors), temperature, yaw angle, and shaft torque (2 sensors), all of which have relatively low missing points in the data set. There were also other measures for motor rpm and bending moments that we did not use because of the confounding effects of other predictors on these two. We considered power generated as the response variable (this is a very common assumption in the literature, see for instance [57]) and wind speed (2 sensors), wind direction (2 sensors), temperature, yaw angle, and shaft torque (2 sensors) as the predictors. We randomly selected 2,500 points from the data set and divided our data set into 80% training set (2000 samples) and 20% testing set (500 samples). The out-of-sample results of our evaluation are based on the testing set. We should point out that because the data set does not have any labeling for anomalous points and we did not have access to pure field data with confirmed anomalies against which we could evaluate our methods, based on the result given in [8] and our discussion with field experts, we considered any point that satisfied the below condition as an anomalous point:

{If wind speed $\geqslant$ 6 m/s& the power generated $\leqslant$ 10 kW}
$\rightarrow$ the sample is an anomalous sample.

The data set was screened based on the above rule, and as a result, 100 anomalous samples were selected for the test set. Thus, only 400 samples (out of 500) in the test set are normal samples. It should be pointed out that although the ratio of 20% for anomalous samples is extremely high and is often not the case in real systems, we set it that high in order to have enough positive samples to better evaluate the results.

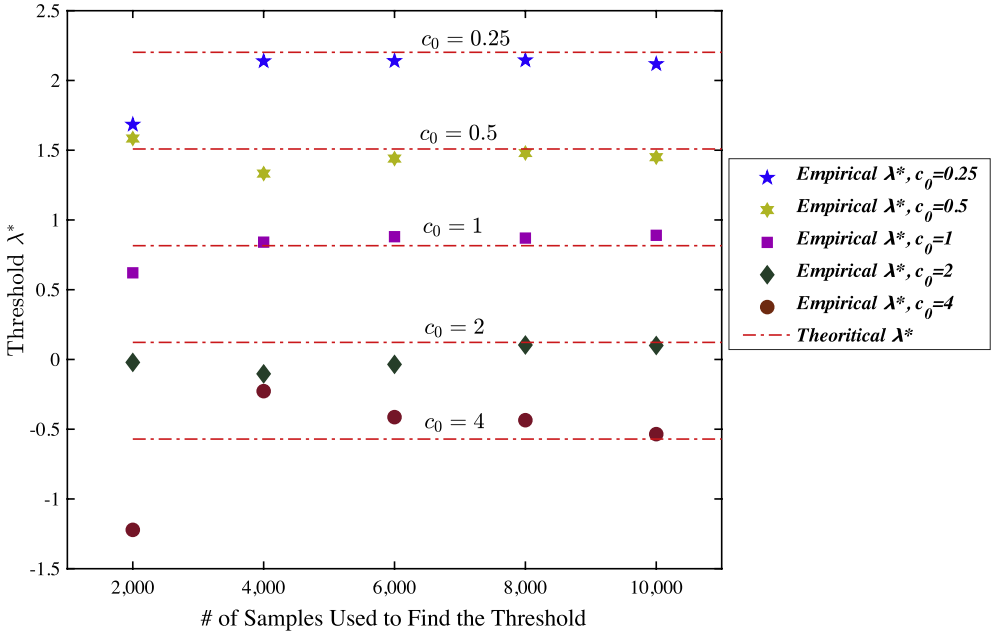

Fig. 9. The comparison between the empirical and theoretical λ^*s based on different values for the number of training samples ($N \in \{2000, 4000, 6000, 8000, 1000\}$) and cost ratio c_0 ($\frac{c_{\hat{a}|a}}{c_{a|\hat{a}}} \in \left\{0.25, 0.5, 1, 2, 4\right\}$). The theoretical values of the thresholds are computed directly from Eq. (12). It can be seen that the empirical values approach the theoretical values as the number of traing samples increases.

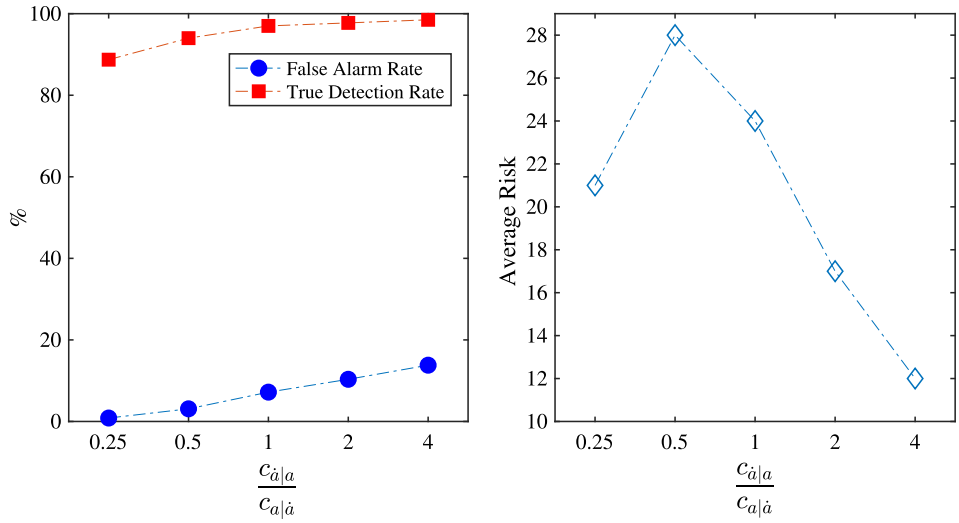

Fig. 10. The performance of the model under the supervised setting versus the cost ratio $\frac{c_{\dot{a}|a}}{c_{a|\dot{a}}} \in \{0.25, 0.5, 1, 2, 4\}$. The left plot shows the false alarm and the true detection rates, and the right plot shows the average risk score.

6.1. Power curve prediction

Power curves are used for many important tasks, including predicting wind power production, assessing turbine energy production efficiency, and monitoring turbine health [43]. We first evaluate the ability of the method developed in this article to predict the generated power based on a set of given measurements. We compare with SVM regression, KNN, neural network, and regular kernel regression. All of these methods are first trained using the training set and then are evaluated based on the testing set. For KNN and neural network, we used cross-validation to tune in the number of clusters and the number of hidden neurons, respectively. The results shown throughout this section are based on the test set. The main performance criterion is the RMSE computed as

$$\text{RMSE} = \left(\frac{\sum_{i=1}^{N_{test}} (\hat{y}_i - y_i)^2}{N_{test}} \right)^{1/2}.$$

Fig. 12 shows the true and actual predicted power and the associated RMSE for all selected methods for the 400 normal samples in the testing set. It can be seen that the estimates from our model outperform the estimates from regular kernel, SVM regression, and KNN regression and are very comparable to the neural network. We have also found that the estimates are reasonable even around the boundaries. The main reason is that training data have multiple points at the boundaries. In summary, we can conclude that our model can satisfactorily predict the power generated by wind turbines working under normal conditions.

In addition to providing a reasonable prediction for response variables, our work has other benefits that contribute to sparsity and model

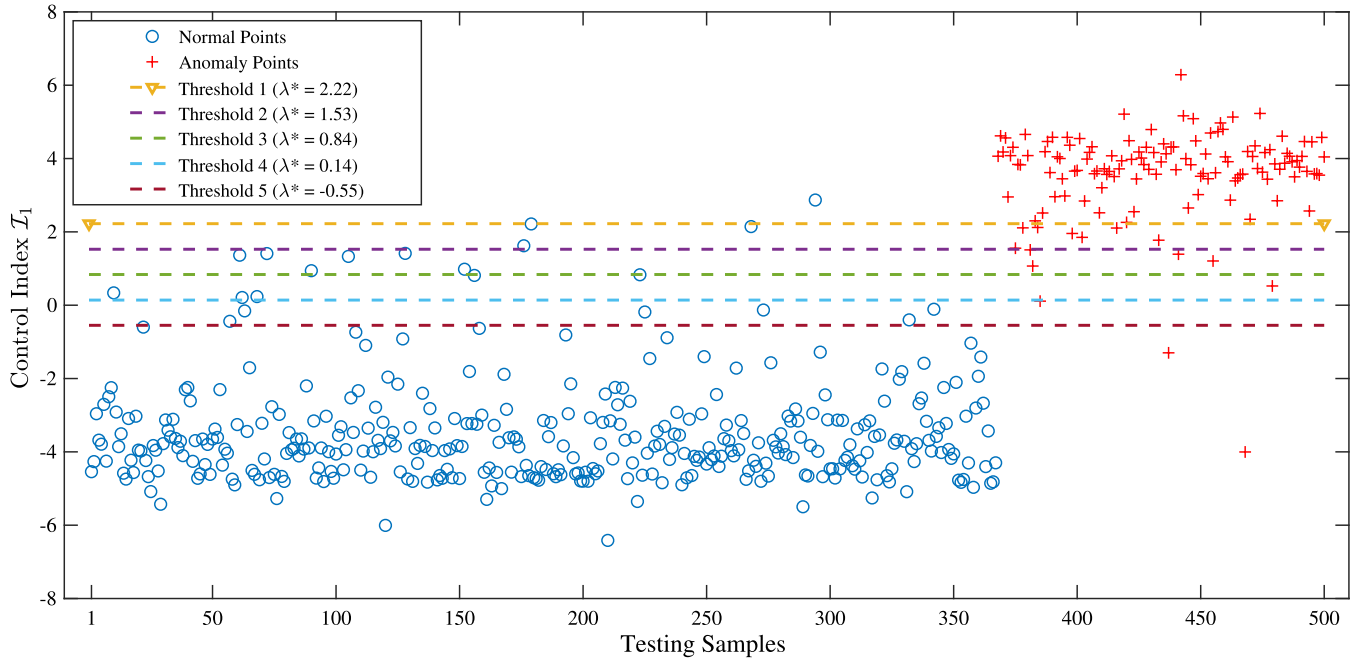

Fig. 11. Detecting anomalies with anomaly index under 5 cases of $\frac{c_{\dot{a}|a}}{c_{a|\dot{a}}} \in \{0.25, 0.5, 1, 2, 4\}$. The five horizontal dash lines are the optimal thresholds for anomalies (one for each $\frac{c_{\dot{a}|a}}{c_{a|\dot{a}}}$). These five different thresholds verify their dynamic nature in the sense that they change with the cost parameters.

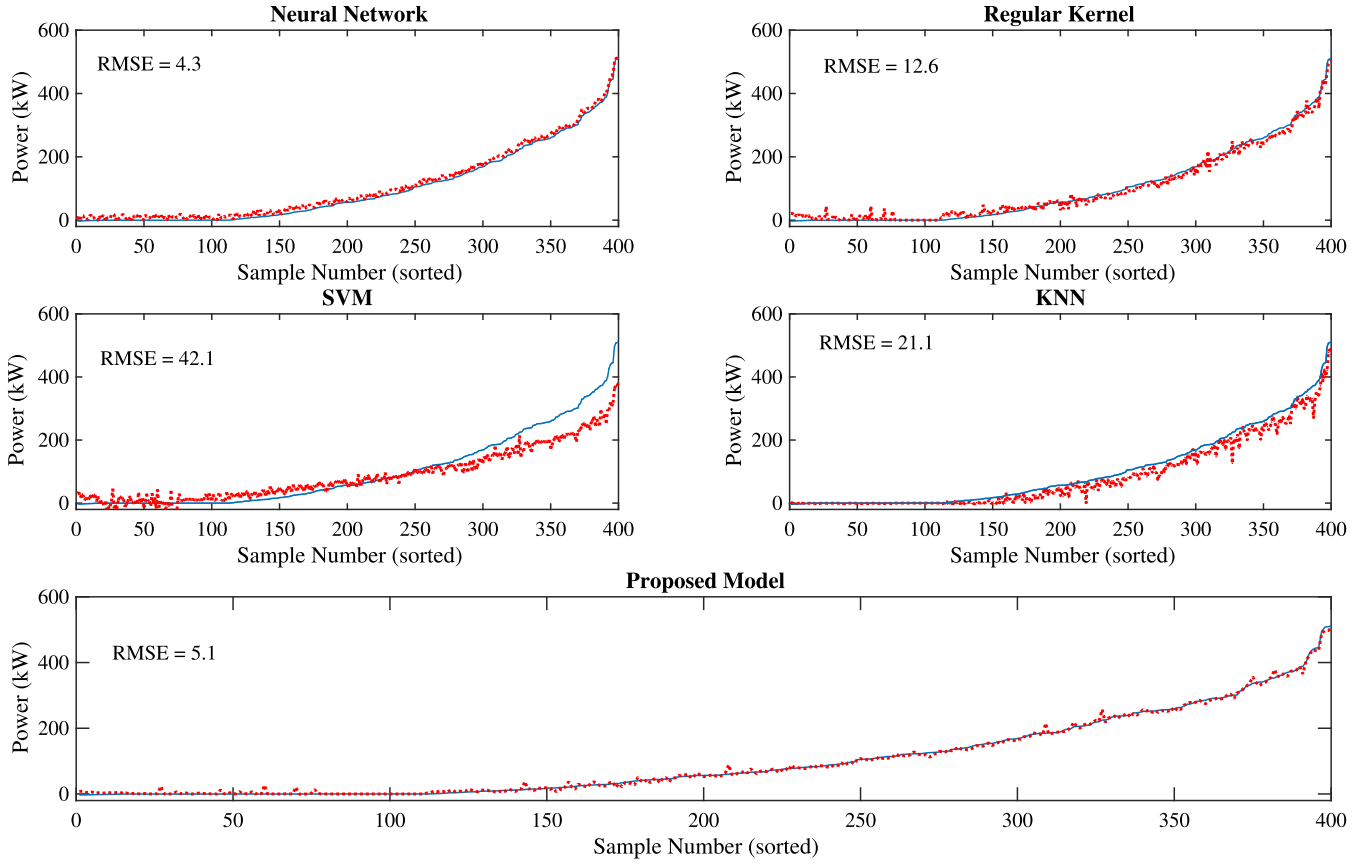

Fig. 12. Power prediction using our work and the benchmark models.

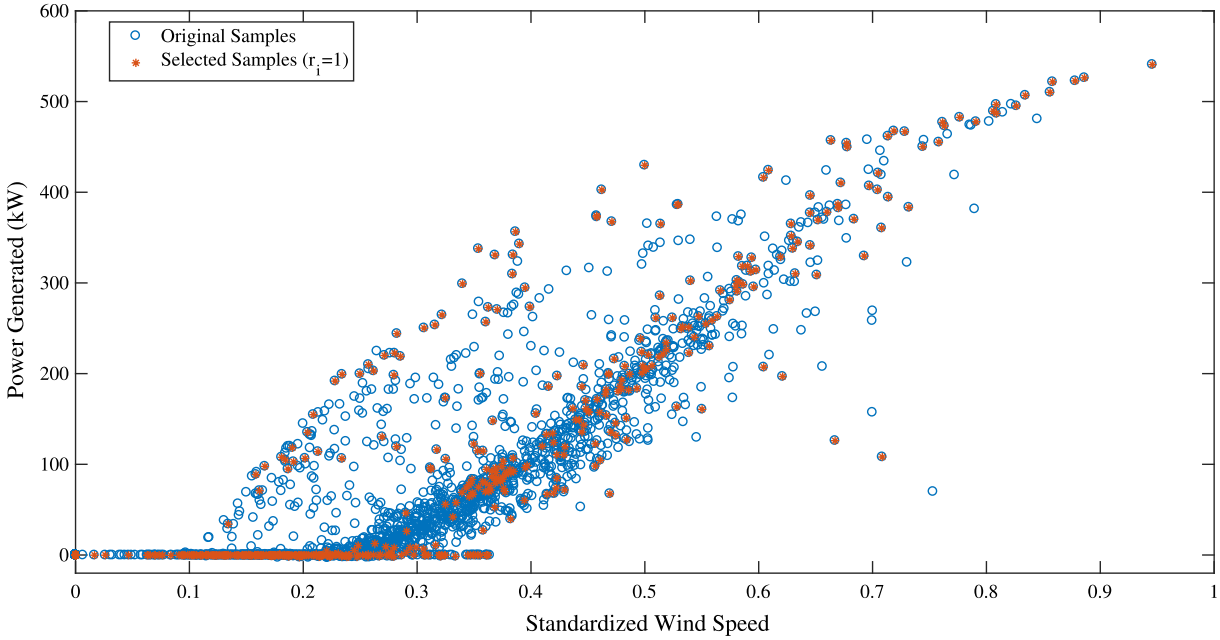

Fig. 13. Effective samples (red points) in the prediction of power curve selected by our model after model training. The effective samples are those with estimated r_i equal to 1.

complexity by selecting only a subset of samples and a subset of features. To show the effective samples chosen in the model training phase (those with $r_i = 1$), we plotted in Fig. 13 the so-called power curve for all samples and then highlighted the ones that are selected in the training samples as important/effective. It can be seen from Fig. 13 that the final effective samples are selected from various regions. It can also be seen that many points with small power generated are selected as important samples. We believe these interesting results are in complete agreement with the intuition behind the model. Those points at the boundaries (points with close to zero power and small wind speed) are not anomalous points and refer to the points where the turbine could not generate enough power. The model simply selects more points in

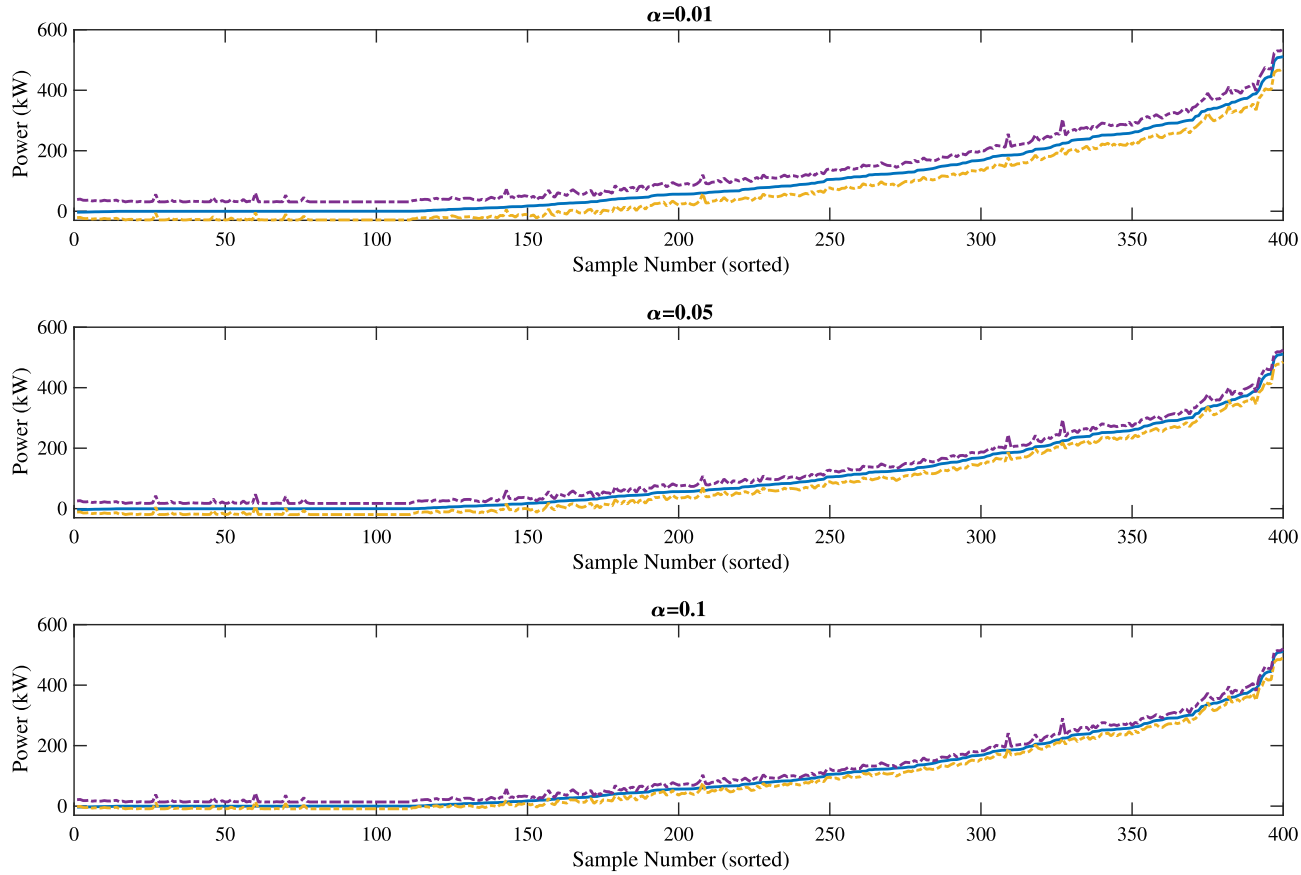

Fig. 14. Prediction interval for the samples in the testing set (sorted based on the power generated).

that region so that normal points are not mistakenly confused for anomalous points. Such results provide the benefit of interpretability to the users of our model. In terms of feature selection, our model quickly converges to select only wind speed, wind direction, and temperature as important features and selects yaw angle and torque as unimportant features for the power generated. This is also in agreement with the physics behind wind power generation. In Fig. 14, we plotted the estimated power (using the mean of the posterior) and the 95% prediction intervals for the normal points in the testing set. Results in Fig. 14 show a reasonable coverage for the true value of power generated. We should point out that we do not claim that our model outperforms many of the existing successful methods for power curve prediction. Instead, we tried to show that it can incorporate multiple environmental and operating features to predict power curve while using only a subset of features and training samples.

6.2. Anomaly detection – supervised

We applied our anomaly detection model to test the ability of our model to detect anomalies. In Fig. 15, we plotted the anomaly index from Eq. (13) for both normal (400 points shown by circles, on the left side of the left plot) and anomalous points (100 points, shown by circles, on the right side of the left plot). Also, the receiver operating characteristic (ROC) curve (right plot) was generated by placing thresholds at each point in the anomaly index and calculating the true positive rate and false positive rate with respect to anomaly. Results show that the model does well in distinguishing normal samples from anomalous samples. In Fig. 16, we plotted the average risk, the false alarm rate, the true detection rate, and the accuracy using our model, the benchmark model among $I_\alpha(x)$ for $\alpha \in \{1\%, 5\%, 10\%, 15\%, 20\%\}$, $I_{\alpha_1^*,\alpha_2^*}(x)$, and the S-R model described before that gives us the best results. We refer to this model as the alternative model. Results verify that our model provides lower risk scores, lower false alarm rates, higher detection rates, and higher accuracy. Also, both models perform equally on the extreme cases where the cost of missing anomalies is very low or very high. Our results provide wind operators a decision-making tool for detecting anomalies based on their preferences, system's policies, and the trade-off between missed anomalies and false alarms.

6.3. Anomaly detection - unsupervised

We applied our anomaly detection model to test the ability of the proposed framework in detecting anomalies in an unsupervised manner, that is, when no information is available with respect to the behavior of the system under anomaly conditions. In order to do such an experiment, we used a training set of 2000 random samples without using the labels assigned in Section 6.1. For the 500 samples in the test set, we calculated the anomaly index $\mathscr{I}_2$ from Eq. (14) and then use various thresholds λ^{**} to observe which points are selected as anomalous points. Here, we used the same test set as Section 6.1 to better evaluate the outcome of the experiment. In Fig. 17, we plotted the standardized wind speed versus the generated power (kW) and highlighted the points that exceed the threshold and are selected as anomalous samples. It can be seen that by increasing the threshold λ^{**}, fewer samples are selected as anomaly. In other words, the model becomes more sensitive to minimizing false alarms and thus tends to select less point as anomaly. This is in complete agreement with the structure of the threshold λ^{**} given in Eq. (13), where larger values of $c_{a|\dot{a}}$ and $p(a)$ result in larger values for λ^{**} and less number of anomalous points. Also, it can be observed from this figure that most selected anomalous samples are located either in the upper boundaries of wind speed and power or in the region that there is sufficient wind but no power is generated.

To compare the performance of the supervised and unsupervised

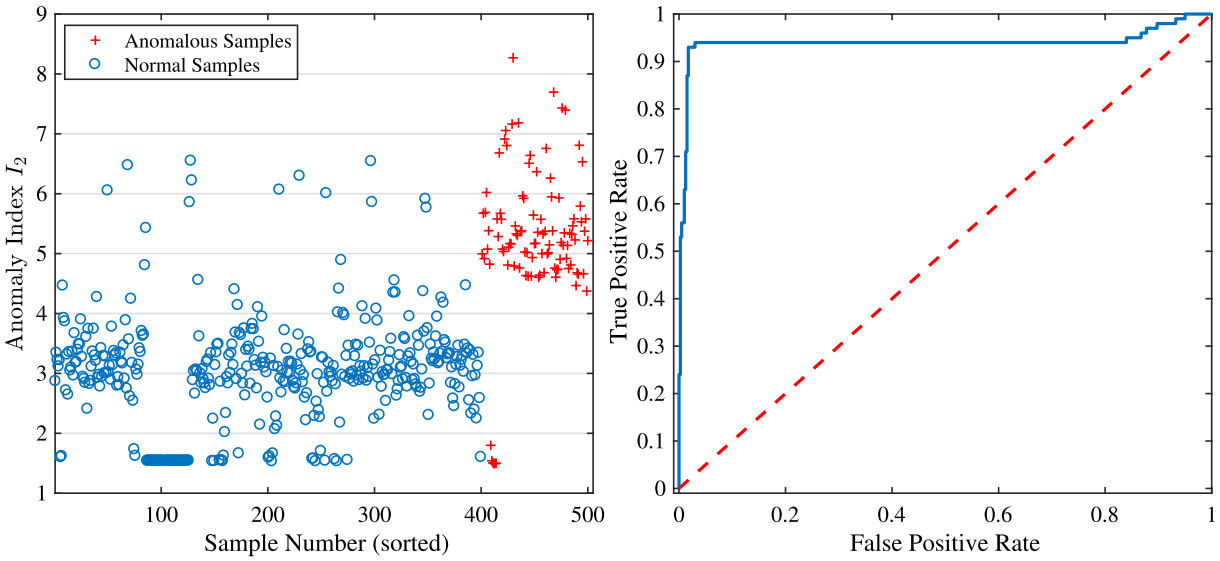

Fig. 15. The anomaly index for normal and anomalous samples (left) and the ROC curve (right) calculated based on $\mathscr{I}_2$ in Eq. (14).

models, we plotted the respective ROC curves that represent the discriminating power of each model with respect to anomalies. It can be seen from Fig. 18 that although, as expected, the unsupervised version of the framework performs worse than the supervised version (i.e., the unsupervised ROC curve is below the supervised ROC curve), its performance is still in a reasonable range. Results in this subsection verify that our model provides reasonable insights with respect to anomalies regardless of whether they are known before. In real applications, it is possible to empirically find different values for threshold λ^{**} for various types of unknown anomalies depending on the tradeoff between missed anomalies and true detection rates defined by c_0.

7. Conclusions and future work

In this article, we developed an anomaly detection framework utilizing system inputs and outputs/response variables collected from sensors and smart devices for which both feature set and training samples are regularized. We utilized a Bayesian hierarchical structure that is able to model the relationship between the system inputs and outputs without imposing strong distributional/parametric assumptions. The Bayesian hierarchical setting helps better describe the structure of the model, accommodate uncertainty, assist in the interpretation of model parameters, and control model complexity and sparsity. Our model can be used for cases in which it is desirable to have fewer points and dimensions, and when there is no known parametric relationship between system inputs and the response variable. The proposed model can also be used for anomaly detection in both supervised and unsupervised settings (when no past information is available with regard to the behavior of the system under some anomaly conditions). The effectiveness and application of the results of this article were shown using both simulation experiments and a case study for wind turbine condition monitoring.

7.1. Limitations of the proposed framework and future work

The results of this article can be used to monitor a single-unit system in which real-time sensor measurements are available and multiple predictors and a single response variable can be clearly defined. The proposed framework has some limitations (to be addressed in our future work) that are summarized below.

– ***Temporal Behavior of Anomalies:*** The static (time-independence)

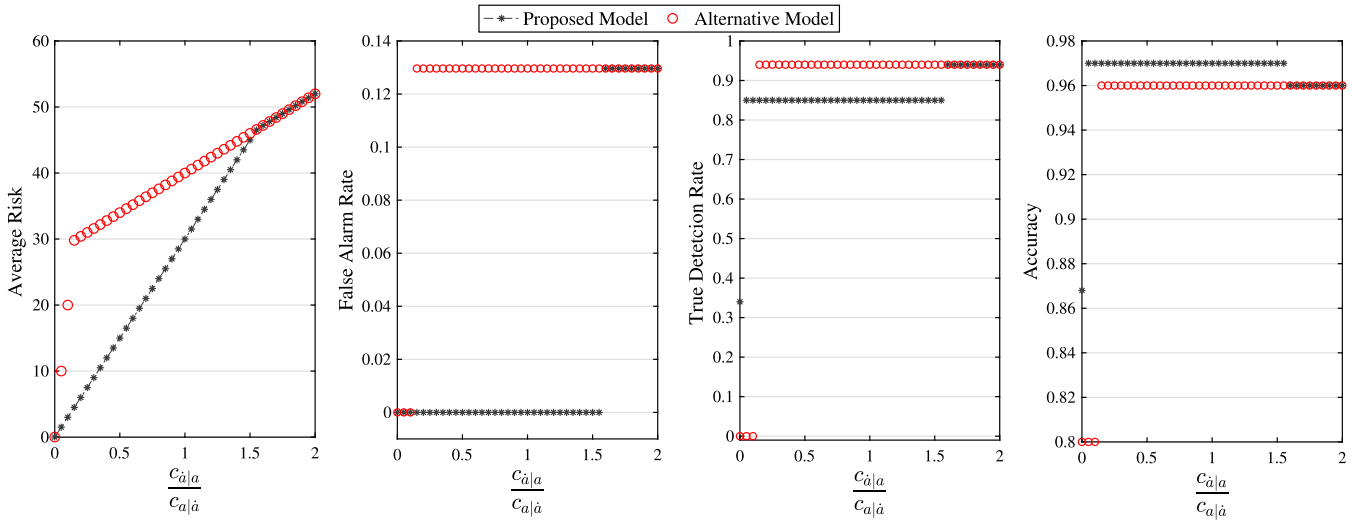

Fig. 16. Performance measures for anomaly detection using the proposed model and the alternative model (i.e., the alternative model is the benchmark model among $I_\alpha(x)$ where $\alpha \in \{1\%, 5\%, 10\%, 15\%, 20\%\}$, $I_{\alpha_1^*,\alpha_2^*}(x)$, and the S-R model that gives the best risk score).

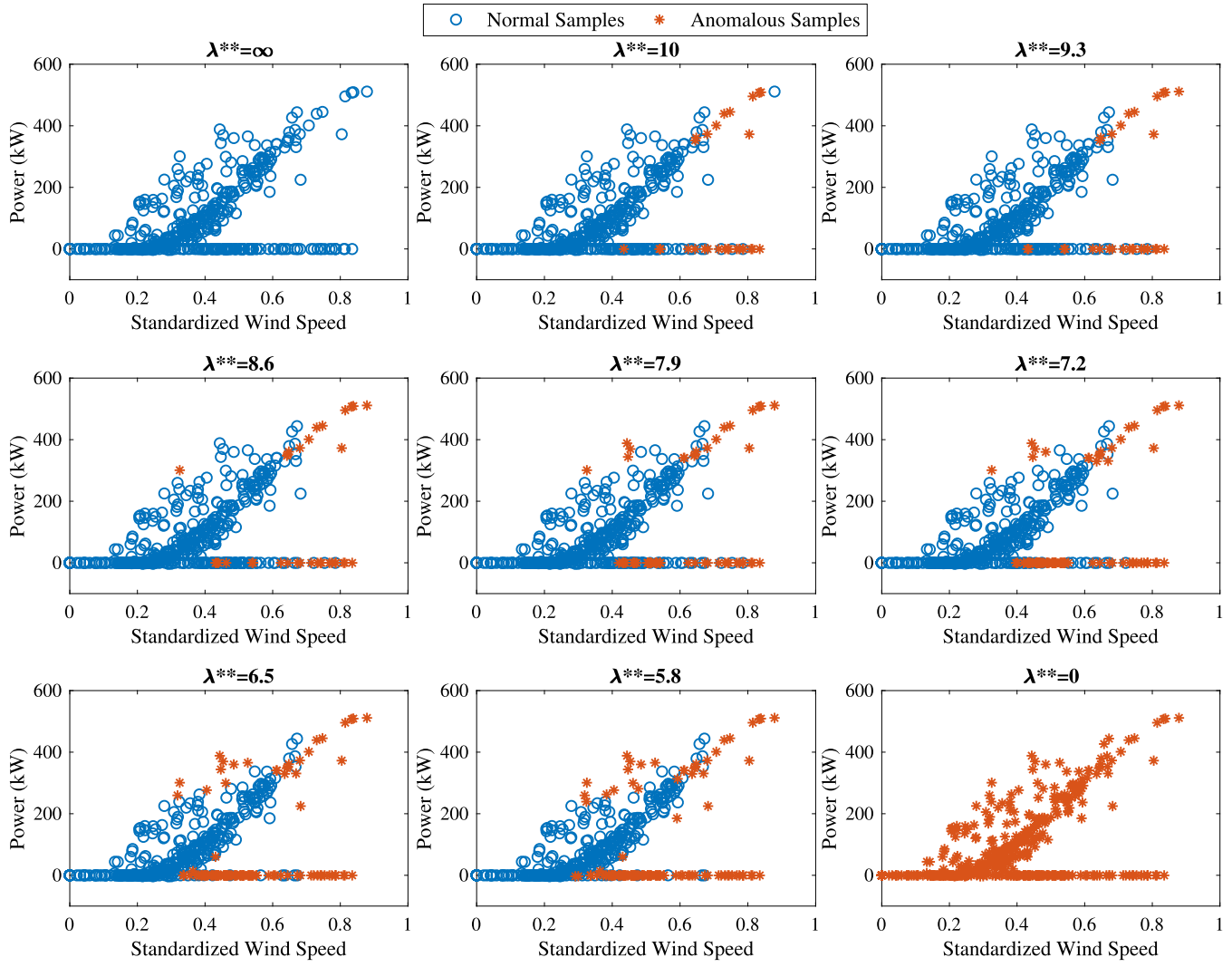

Fig. 17. The samples selected as anomaly using different values for threshold λ^{**} (test set). According to Eq. (13), if the cost ration c_o doubles, then the threshold λ^{**} increases by log(2). That is why we chose the difference between two consecutive thresholds to be log(2) (Excluding the two extreme thresholds of 0 and $-\infty$).

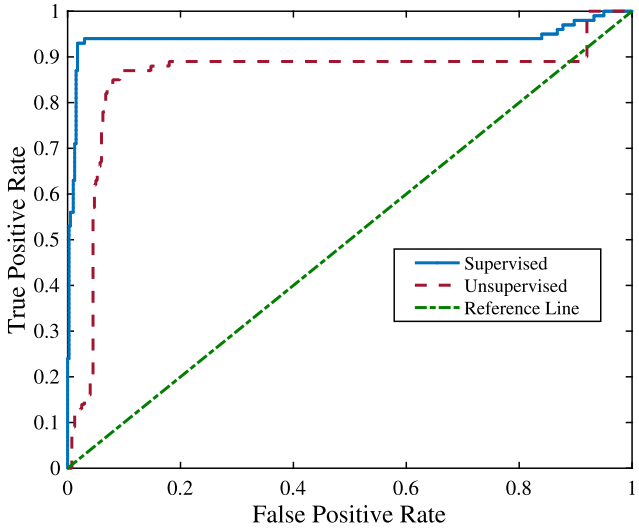

Fig. 18. The ROC curves for the supervised and unsupervised experiments versus the reference line.

nature of the developed anomaly indices, their inability to model the temporal behavior of anomalies, and their independence to maintenance actions are important shortcomings of the proposed anomaly detection framework. In future work, we will use the approaches in articles such as [35] for monitoring the evolution of the anomaly index over time.

- ***Detecting High-Leverage Data Points:*** High-leverage data points are those observations made at extreme or outlying values of the independent variables. It is very important to have a distinction between anomalous points and high-leverage observations because they are often treated differently. There is a distinction between anomalies considered in this article and high-leverage observation. We considered a data point to be an anomaly only if it is extreme with respect to the response values (y), not the input (x) values. For the above reason, high-leverage observations are not considered in the proposed framework and will be considered in our future work.
- ***Dealing with Categorical Features:*** The proposed framework only works with continuous variables. It is possible that we have some features that are discrete or categorical and have a limited number of discrete levels. In such cases, the user should carefully define a more suitable kernel (e.g., Dirac kernel) and also use a suitable noise model to handle categorical variables.
- ***Missing Points:*** Another limitation of the proposed framework is the

ignorance of missing points, which may lead to biased and incorrect inferences if handled inappropriately. A few different statistical approaches may be used to handle missing points. For instance, any method of single or multiple imputation may be used (e.g., [58]). Also, different Bayesian methods for dealing with missing data may be employed [59].

- ***Dealing with Multiple Independent Types of Anomalies:*** The framework proposed in this article cannot be used if multiple independent types of anomalies with known characteristics are to be studied simultaneously. To do so, a separate anomaly behavior model needs to first be developed for each type of anomaly where it is possible that different unique sets of variables are found to detect each type of anomaly. It is clear that such an analysis of multiple independent anomalies with known behavior is subject to two important limitations. First, it requires a large set of parameters that need to be estimated from past data. Second, in order to estimate all these parameters, a large number of accurate training samples is needed for each type of anomaly.
- ***Computational Complexity:*** There are two main phases in the proposed anomaly detection framework: model training and anomaly detection based on the trained model. The former task is completed offline while the latter takes place in real time. Compared to the models that do not have the unknown binary variables r_i for each sample, our model is computationally more expensive in the training phase (because r_is need to be estimated) but more efficient during the inference phase (because fewer samples are used for anomaly detection). For extremely large data sets, it is possible to benefit from MCMC parallelization techniques and performing the computations on a graphics processing unit (e.g., [60]). The trade-off between computation and the benefit of selecting a subset of samples depends on the application.

References

[1] Chandola V, Banerjee A, Kumar V. Anomaly detection: a survey. ACM Comput Surv 2009;41(3):15:1–15:58.
[2] Von Birgelen A, Buratti D, Mager J, Niggemann O. Self-organizing maps for anomaly localization and predictive maintenance in cyber-physical production systems. Proc CIRP 2018;72:480–5.
[3] Liu FT, Ting KM, Zhou ZH. Isolation-based anomaly detection. ACM Trans Knowl Disc Data 2012;6(1).
[4] Simon DL, Rinehart AW. A model-based anomaly detection approach for analyzing streaming aircraft engine measurement data. In: Proceedings of the ASME Turbo Expo, vol. 6; 2014.
[5] Li L, Hansman RJ, Palacios R, Welsch R. Anomaly detection via a Gaussian mixture model for flight operation and safety monitoring. Transport Res Part C, Emerg Technol 2016;64:45–57.
[6] Fan C, Xiao F, Zhao Y, Wang J. Analytical investigation of autoencoder-based methods for unsupervised anomaly detection in building energy data. Appl Energy 2018;211:1123–35.
[7] Ye N, Chen Q. Attack-norm separation for detecting attack-induced quality problems on computers and networks. Qual Reliab Eng Int 2007;23(5):545–53.
[8] Yampikulsakul N, Byon E, Huang S, Sheng S, You M. Condition monitoring of wind power system with nonparametric regression analysis. IEEE Trans Energy Convers 2014;29(2):288–99.
[9] Kim J, Scott CD. Robust kernel density estimation. J Mach Learn Res 2012;13:2529–65.
[10] Aizerman A, Braverman M, Rozoner LI. Theoretical foundations of the potential function method in pattern recognition learning. Autom Rem Control 1964;25:821–37.
[11] Lee G, Ding Y, Xie L, Genton MG. A kernel plus method for quantifying wind turbine performance upgrades. Wind Energy 2015;18(7):1207–19.
[12] Santiago-Paz J, Torres-Roman D, Figueroa-Ypína A, Argaez-Xool J. Using generalized entropies and oc-svm with Mahalanobis kernel for detection and classification of anomalies in network traffic. Entropy 2015;17(9):6239–57.
[13] Chapel L, Friguet C. Anomaly detection with score functions based on the reconstruction error of the kernel pca. Lect Notes Comput Sci (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 2014;8724 LNAI(PART 1):227–41.
[14] Zhou J, Kwan C, Ayhan B, Eismann MT. A novel cluster kernel RX algorithm for anomaly and change detection using hyperspectral images. IEEE Trans Geosci Rem Sens 2016;54(11):6497–504.
[15] Zhang L, Lin J, Karim R. An angle-based subspace anomaly detection approach to high-dimensional data: with an application to industrial fault detection. Reliab Eng Syst Safety 2015;142:482–97.
[16] Hu RL, Granderson J, Auslander DM, Agogino A. Design of machine learning models with domain experts for automated sensor selection for energy fault detection. Appl Energy 2019:117–28.
[17] Kontorovich A, Hendler D, Menahem E. Metric anomaly detection via asymmetric risk minimization. Lect Notes Comput Sci (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 2011;7005 LNCS:17–30.
[18] Pang J, Liu D, Peng Y, Peng X. Anomaly detection based on uncertainty fusion for univariate monitoring series. Meas: J Int Meas Confeder 2017;95:280–92.
[19] Ehsani-Besheli F, Zarandi HR. Context-aware anomaly detection in embedded systems. Adv Intell Syst Comput 2018;582:151–65.
[20] Wang Y, Li X, Ding X. Probabilistic framework of visual anomaly detection for unbalanced data. Neurocomputing 2016;201:12–8.
[21] Lu HY, Chen FY, Xu M, Wang CJ, Xie JY. Never ignore the significance of different anomalies: a cost-sensitive algorithm based on loss function for anomaly detection. In: Proceedings - international conference on tools with artificial intelligence, ICTAI, vol. 2016-January; 2016. p. 1099–1105.
[22] Lee SC, Faloutsos C, Chae DK, Kim SW. Fraud detection in comparison-shopping services: patterns and anomalies in user click behaviors. IEICE Trans Inform Syst 2017;E100D(10):2659–63.
[23] Jin X, Sun Y, Que Z, Wang Y, Chow TWS. Anomaly detection and fault prognosis for bearings. IEEE Trans Instrum Meas 2016;65(9):2046–54.
[24] Balducelli C, Bologna S, Lavalle L, Vicoli G. Safeguarding information intensive critical infrastructures against novel types of emerging failures. Reliab Eng Syst Safety 2007;92(9):1218–29.
[25] Usha M, Kavitha P. Anomaly based intrusion detection for 802.11 networks with optimal features using SVM classifier. Wirel Netw 2017;23(8):2431–46.
[26] Yan H, Paynabar K, Shi J. Anomaly detection in images with smooth background via smooth-sparse decomposition. Technometrics 2017;59(1):102–14.
[27] Noorossana R, Hosseini SS, Heydarzade A. An overview of dynamic anomaly detection in social networks via control charts. Qual Reliab Eng Int 2018;34(4):641–8.
[28] Rocco S CM, Zio E. A support vector machine integrated system for the classification of operation anomalies in nuclear components and systems. Reliab Eng Syst Saf 2007;92(5):593–600.
[29] Li F, Wang H, Zhou G, Yu D, Li J, Gao H. Anomaly detection in gas turbine fuel systems using a sequential symbolic method. Energies 2017;10(5).
[30] Dezman ZDW, Gao C, Yang S, Hu P, Yao L, Li HC, et al. Anomaly detection outperforms logistic regression in predicting outcomes in trauma patients. Prehos Emerg Care 2017;21(2):174–9.
[31] Herp J, Ramezani MH, Bach-Andersen M, Pedersen NL, Nadimi ES. Bayesian state prediction of wind turbine bearing failure. Renew Energy 2018;116:164–72.
[32] Byon E. Wind turbine operations and maintenance: a tractable approximation of dynamic decision making. IIE Trans (Inst Ind Eng) 2013;45(11):1188–201.
[33] Artigao E, Martín-Martínez S, Honrubia-Escribano A, Gómez-Lázaro E. Wind turbine reliability: a comprehensive review towards effective condition monitoring development. Appl Energy 2018;228:1569–83.
[34] Gil A, Sanz-Bobi MA, Rodríguez-López MA. Behavior anomaly indicators based on reference patterns - application to the gearbox and electrical generator of a wind turbine. Energies 2018;11(1).
[35] de Andrade Vieira RJ, Sanz-Bobi MA. Failure risk indicators for a maintenance model based on observable life of industrial components with an application to wind turbines. IEEE Trans Reliab 2013;62(3):569–82.
[36] Yan Y, Li J, Gao DW. Condition parameter modeling for anomaly detection in wind turbines. Energies 2014;7(5):3104–20.
[37] Sun P, Li J, Wang C, Lei X. A generalized model for wind turbine anomaly identification based on SCADA data. Appl Energy 2016;168:550–67.
[38] Du M, Tjernberg LB, Ma S, He Q, Cheng L, Guo J. A SOM based anomaly detection method for wind turbines health management through SCADA data. Int J Prognost Health Manage 2016;7:1–13.
[39] Schlechtingen M, Santos IF, Achiche S. Wind turbine condition monitoring based on SCADA data using normal behavior models. Part 1: System description. Appl Soft Comput J 2013;13(1):259–70.
[40] Mazidi P, Bertling L, Bobi M. Wind turbine prognostics and maintenance management based on a hybrid approach of neural networks and a proportional hazards model. Proc Inst Mech Eng, Part O: J Risk Reliab 2017;231(2):121–9.
[41] Mazidi P, Du M, Tjernberg L, Bobi M. A health condition model for wind turbine monitoring through neural networks and proportional hazard models. Proc Inst Mech Eng, Part O: J Risk Reliab 2017;231(5):481–94.
[42] Yang W, Liu C, Jiang D. An unsupervised spatiotemporal graphical modeling approach for wind turbine condition monitoring. Renew Energy 2018;127:230–41.
[43] Lee G, Ding Y, Genton MG, Xie L. Power curve estimation with multivariate environmental factors for inland and offshore wind farms. J Am Stat Assoc 2015;110(509).
[44] Mishra M, Martinsson J, Rantatalo M, Goebel K. Bayesian hierarchical model-based prognostics for lithium-ion batteries. Reliab Eng Syst Safety 2018;172:25–35.
[45] Andrade AR, Teixeira PF. Statistical modeling of railway track geometry degradation using hierarchical Bayesian models. Reliab Eng Syst Safety 2015;142:169–83.
[46] Knorr EM, Ng RT, Tucakov V. Distance-based outliers: algorithms and applications. VLDB J 2000;8(3–4):237–53.
[47] Bay SD, Schwabacher M. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining; 2003. p. 29–38.
[48] Chandola V, Banerjee A, Kumar V. On estimating regression. Theory Probab Appl 1964;9:141–2.
[49] Watson GS. Smooth regression analysis, Sankhyā. Indian J Stat Ser A 1964;26:359–72.
[50] Schmoyer RL. Asymptotically valid prediction intervals for linear models.

Technometrics 1992;34(4):399–408.
[51] Stine RA. Bootstrap prediction intervals for regression. J Am Stat Assoc 1985;80(392):1026–31.
[52] Kumar S, Srivastava A. Bootstrap prediction intervals in non-parametric regression with applications to anomaly detection. In: The 18th ACM SIGKDD conference on knowledge discovery and data mining. Beijing; China; 2012.
[53] Turney PD. Cost-sensitive classification: empirical evaluation of a hybrid genetic decision tree induction algorithm. J Artif Intell Res 1995;2:369–409.
[54] Sheather SJ. Density estimation. Stat Sci 2004;19(4):588–97.
[55] Sheng S, Veers P. Wind turbine drivetrain condition monitoring - an overview. In: Technical program for MFPT: the applied systems health management conference 2011: Enabling Sustainable Systems; 2011.
[56] Technical University of Denmark & Risø National Laboratory. Database of wind characteristics (Accessed Jul. 2018). < http://www.winddata.com > .
[57] Wang S, Huang Y, Li L, Liu C. Wind turbines abnormality detection through analysis of wind farm power curves. Meas: J Int Meas Confeder 2016;93:178–88.
[58] Little RJA, Rubin DB. Statistical analysis with missing data, second edition. Statistical analysis with missing data. 2nd ed.John Wiley & Sons, Inc; 2014.
[59] Ma Z, Chen G. Bayesian methods for dealing with missing data problems. J Korean Stat Soc 2018;47(3):297–313.
[60] Terenin A, Dong S, Draper D. GPU-accelerated Gibbs sampling: a case study of the horseshoe probit model. Stat Comput 2018:1–10.