Artificial Intelligence for Engineering Design, Analysis and Manufacturing

cambridge.org/aie

One-class support vector machines with a bias constraint and its application in system reliability prediction

Zhengwei Hu[1], Zhangli Hu[1] and Xiaoping Du[2]

[1]Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology, Rolla, MO 65409, USA and [2]Department of Mechanical and Energy Engineering, Indiana University – Purdue University Indianapolis, Indianapolis, IN 46202, USA

Research Article

Cite this article: Hu Z, Hu Z, Du X (2019). One-class support vector machines with a bias constraint and its application in system reliability prediction. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **33**, 346–358. https://doi.org/10.1017/S0890060419000155

Received: 10 July 2018
Revised: 5 March 2019
Accepted: 8 March 2019
First published online: 3 May 2019

Key words:
First-order reliability method; optimization; support vector machines; system reliability; statistical dependence

Author for correspondence: Xiaoping Du, E-mail: duxi@iu.edu

Abstract

Support vector machine (SVM) methods are widely used for classification and regression analysis. In many engineering applications, only one class of data is available, and then one-class SVM methods are employed. In reliability applications, the one-class data may be failure data since the data are recorded during reliability experiments when only failures occur. Different from the problems handled by existing one-class SVM methods, there is a bias constraint in the SVM model in this work and the constraint comes from the probability of failure estimated from the failure data. In this study, a new one-class SVM regression method is proposed to accommodate the bias constraint. The one class of failure data is maximally separated from a hypersphere whose radius is determined by the known probability of failure. The proposed SVM method generates regression models that directly link the states of failure modes with design variables, and this makes it possible to obtain the joint probability density of all the component states of an engineering system, resulting in a more accurate prediction of system reliability during the design stage. Three examples are given to demonstrate the effectiveness of the new one-class SVM method.

Introduction

The support vector machine (SVM) method was originally developed for classifying data from two different classes (Boser *et al.*, 1992; Vapnik and Vapnik, 1998; Vapnik, 2013). Two-class SVM methodologies obtain an optimal decision boundary by maximizing the margin between the training patterns. More specifically, given a data set composed of points from two different classes, an optimal boundary is built in the form of a hyperplane or hypersurface defined by the maximum margin between the points and the boundary, and the points on the maximum margin are the so-called support vectors.

SVM can be analyzed theoretically based on statistical learning theory and optimization methods, thus it outperforms other learning algorithms in many aspects. The advantage of SVM is attributed to its essence based on the principle of the maximal margin (Tian *et al.*, 2012), the dual theory, and the kernel trick, which enable SVM to solve machine learning problems with only limited training points. It overcomes traditional difficulties due to the curse of dimensionality and over-fitting. This makes SVM highly successful and effective in real applications, and thus it has recently received considerable attention in various domains, such as pattern recognition (Frias-Martinez *et al.*, 2006; Peng, 2011; Truong and Kim, 2012), data mining (Cortez, 2010), fault detection (Mahadevan and Shah, 2009; Chen *et al.*, 2011; Gryllias and Antoniadis, 2012), space frame structure optimization (Hanna, 2007), and reliability analysis (Li *et al.*, 2006; das Chagas Moura *et al.*, 2011; Hu and Du, 2018*a*; Wang *et al.*, 2018).

Most traditional SVM methods assume more or less equally balanced data from both classes, and the decision boundary is therefore determined by the data belonging to different classes. However, when encountering with imbalanced data sets where the number of data from one of these two classes far outnumbers that from the other class or even equals to zero, the performance of the general two-class SVM may drop dramatically (Akbani *et al.*, 2004). This situation is very common in real-world applications, especially in certain domains such as reliability analysis and design. For example, to evaluate the reliability of a system or a component, designers may perform reliability testing repeatedly until the system or the component fails. They then record the failure data, such as sizes, loads, and the temperature at the time when the failure occurs. In this case, all the training points belong to only one class (failure). Due to the need of dealing with one-class data, many methods have been developed, and they have been used in applications such as novelty detection (Ma and Perkins, 2003), document classification (Manevitz and Yousef, 2001), and disease diagnosis (Dreiseitl *et al.*, 2010).

© Cambridge University Press 2019

CAMBRIDGE UNIVERSITY PRESS

The existing one-class SVM methods create the optimal hyperplane (decision boundary) with a weight vector (normal vector) and a bias (intercept), which determine the orientation and location of the hyperplane, respectively. Due to the regularization of the optimization model, only the weight vector is actually to be determined, and the bias is treated separately after the weight vector is obtained. In some engineering applications, such as the aforementioned system reliability prediction, the bias is available, leaving only the weight vector unknown and to be determined.

To accommodate the known bias, in this work, we propose a new one-class SVM method. The constraint of the known bias geometrically forms a hypersphere centered at the origin. By maximizing the minimum distance between one-class training points and the hyperplane that is tangent to the known hypersphere, the proposed method produces the optimal weight vector (orientation) of the desired hyperplane. The hyperplane function is thus determined by the obtained weight vector and the known bias. Since the hyperplane function explicitly defines the decision boundary which classifies the training points, it could then be used for further analysis, such as the aforementioned system reliability estimation, where the hyperplane function is actually the reconstructed computational model of the component.

The rest of the paper is organized as follows. In the section "Methodology review," we briefly review the methodology of general one-class SVM. Section "A new algorithm for one-class support vector machines with a bias constraint" introduces the proposed one-class SVM algorithm with a bias constraint. The application of this new method to the system reliability analysis is discussed in the section "Application of the new one-class SVM in system reliability prediction." One mathematical example and two engineering examples are provided in the section "Examples for methodology validation," followed by conclusions and future work in the section "Conclusions."

Methodology review

The one-class SVM (Schölkopf *et al.*, 2001) is a variant of the general SVM and is used for only one class of training points. This method regards the training points available as belonging to the first class and the origin as being the second class. Then the general two-class SVM techniques could be employed. A decision boundary is built by maximizing the distance between training points and the origin, as shown in Figure 1.

For m training points $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)$, $y_i = +1$, $i = 1, 2, \ldots, m, \mathbf{x} \in \mathbf{R}^n$, belonging to the only one class, the optimization model is given by

$$\begin{cases} \min\limits_{\boldsymbol{\omega}, \boldsymbol{\xi}, \rho} \dfrac{1}{2}\boldsymbol{\omega}\boldsymbol{\omega}^{\mathrm{T}} + \dfrac{1}{\mathrm{mv}}\sum\limits_{i=1}^{m}\xi_i - \rho \\ \text{s.t.} \quad \boldsymbol{\omega}\cdot\psi(\mathbf{x}_i) + \xi_i - \rho \geq 0, \xi_i \geq 0, i = 1, 2, \ldots, m \end{cases} \tag{1}$$

in which $\boldsymbol{\omega}$ and ρ are the to-be-determined weight vector and bias, respectively. The regularization variable $v \in (0, 1)$ indicates the maximum value of the fraction of training data set errors, and $\boldsymbol{\xi} = [\xi_1, \xi_2, \ldots, \xi_m]$ is a vector of slack variables that allow point $\mathbf{x}_i$ to locate on the other side of the optimal hyperplane.

With introducing Lagrange multipliers η_i and γ_i, the Lagrangian function of Eq. (1) is given by

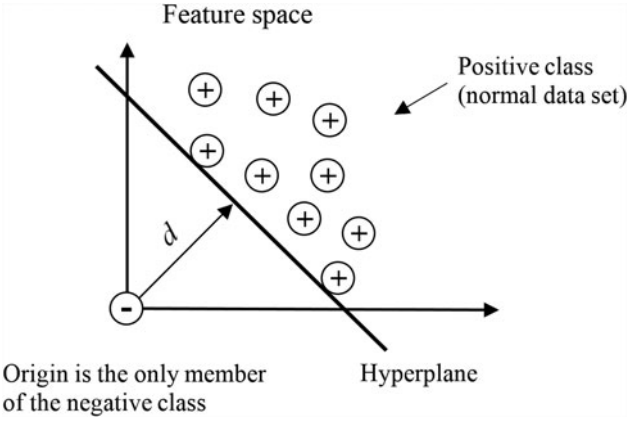


Fig. 1. Basic principle of general one-class SVMs.

$$L(\boldsymbol{\omega}, \boldsymbol{\xi}, \rho) + = \frac{1}{2}\boldsymbol{\omega}\boldsymbol{\omega}^{\mathrm{T}} + \frac{1}{mv}\sum_{i=1}^{m}\xi_i - \rho - \sum_{i=1}^{m}\gamma_i(\boldsymbol{\omega}\cdot\psi(\mathbf{x}_i) + \xi_i - \rho) - \sum_{i=1}^{m}\eta_i\xi_i \tag{2}$$

With the appropriate kernel function $K(\mathbf{x}_i, \mathbf{X}) = K(\psi(\mathbf{x}_i), \psi(\mathbf{X}))$, the optimization model is then written in the dual form

$$\begin{cases} \min\limits_{\gamma} \dfrac{1}{2}\sum\limits_{i,j=1}^{m}\gamma_i\gamma_j K(\mathbf{x}_i, \mathbf{X}) \\ \text{s.t.} \quad 0 \leq \gamma_i \leq \dfrac{1}{mv}, \sum\limits_{i=1}^{m}\gamma_i = 1 \end{cases} \tag{3}$$

Note that when v approaches zero, most of the training points locate inside the estimated support. Then the upper bound of γ_i in Eq. (3) tends to infinity, making the second inequality constraint useless, which is similar to the hard margin algorithm used in two-class SVM. Since there is no constraints for bias ρ, the original optimization model can still be solved by assigning a large negative value to ρ (Schölkopf *et al.*, 2001).

Standard quadratic programing can be used to solve for $\gamma_1, \gamma_2, \ldots,$ and γ_m. The weight vector of the hyperplane is computed by

$$\boldsymbol{\omega} = \sum_{i=1}^{m}\gamma_i\psi(\mathbf{x}_i) \tag{4}$$

And the bias is calculated by

$$\rho = \sum_{i,j=1}^{m}\gamma_i K(\mathbf{x}_i, \mathbf{x}_j) \tag{5}$$

With the determined $\boldsymbol{\omega}$ and b, the decision boundary for one-class SVM is given by

$$f(\mathbf{X}) = \sum_{i=1}^{m}\gamma_i K(\mathbf{x}_i, \mathbf{X}) - \sum_{i,j=1}^{m}\gamma_i K(\mathbf{x}_i, \mathbf{x}_j) \tag{6}$$

https://doi.org/10.1017/S0890060419000155 Published online by Cambridge University Press

A new algorithm for one-class support vector machines with a bias constraint

In this work, we propose a new one-class SVM method with a bias constraint. In the general SVM algorithm, although a bias exists, it is treated separately and does not appear in the optimization model. In the present problem, a bias exists and it is used to formulate a constraint function of the optimization model. The existence of the bias simplifies the optimization model.

The problem arises in the field of system reliability analysis. The proposed method works for the following situation. For the prediction of the reliability associated with a failure mode, repeated reliability testing is performed, and the failure data are recorded until failures occur. Then there is only one class of data. With the failure data, the reliability, which is the probability that the failure mode does not occur, can be estimated. This reliability determines the bias. While more background information about reliability will be provided in the section "Application of the new one-class SVM in system reliability prediction," the new one-class SVM problem we are dealing with is summarized below.

Information available includes the following:

- A data set of m training points and responses is given by $(\mathbf{u}_1, y_1), (\mathbf{u}_2, y_2), \ldots, (\mathbf{u}_m, y_m)$, $\mathbf{u} \in \mathbf{R}^n$, $y_i = -1$, $i = 1, 2, \ldots, m$. Note that different from the general SVM where training points are denoted by $\mathbf{X}$, here we use $\mathbf{u}$ for training points because it is a common notation for reliability analysis where the new method will be used. The data set is from reliability testing, and it may include load, dimensional, temperature, and other parameters that cause a failure. The corresponding response is the state of the component under testing, and $y_i = -1$ represents a failure state. All data points belong to the class of failure, and no data points belong to the class of safety.
- We know the shortest distance β from the origin to the domain to which the data set belongs. This distance comes from the known reliability.

The assumptions we make for the new SVM method are as follows:

- We assume that the boundary of the domain to which the data set belongs is a hyperplane. This assumption is valid for reliability applications where the first order reliability method (FORM) (Cruse, 1997) is applicable.
- The hyperplane is given by

$$Y = \beta + \boldsymbol{\alpha}\mathbf{U}^{\mathrm{T}} \tag{7}$$

where β is a constant, and $\boldsymbol{\alpha}$ is a unit vector. In the reliability application concerned by this study, β is given and is determined by the reliability estimated from the training points, and $\boldsymbol{\alpha}$ happens to be a unit vector. This assumption does not affect the generality of the proposed method.

Our task is to determine the unit vector $\boldsymbol{\alpha}$. In sum, our present problem is to find the optimal normal vector $\boldsymbol{\alpha}$ of a hyperplane given its distance to the origin being β and a data set $(\mathbf{u}_1, y_1), (\mathbf{u}_2, y_2), \ldots, (\mathbf{u}_m, y_m), \mathbf{u} \in \mathbf{R}^n$.

As demonstrated in Figure 2, the problem is to find a hyperplane that is tangent to a hypersphere with a radius of β, and the hyperplane also maximizes the distance from any training points to the hypersphere.

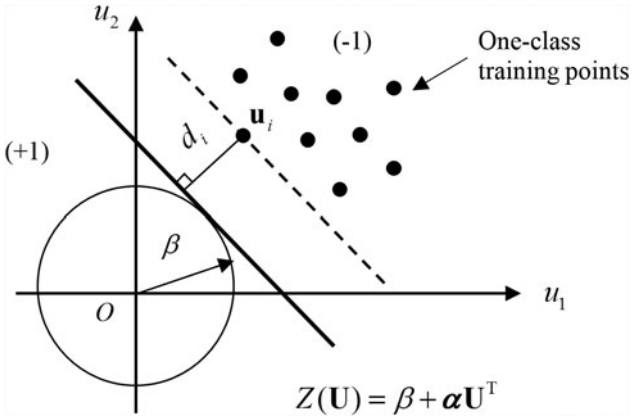


Fig. 2. Basic principle of the proposed one-class SVM.

Denote the distance from $\mathbf{u}_i$ to the hypersphere by d_i. Points located in the negative region enable $Y < 0$; otherwise, $Y > 0$ holds. The minimum distance is given by

$$d = \min\{d_i\} = \min\{-(\beta + \boldsymbol{\alpha}\mathbf{u}_i^{\mathrm{T}})\} \tag{8}$$

in which the negative sign indicates that the training points locate in the negative side of the hyperplane $Z(\mathbf{U})$, thereby making d positive. Since β is a known constant, d is actually determined by minimizing $d_i + \beta$, which is equal to $-\boldsymbol{\alpha}\mathbf{u}_i^{\mathrm{T}}$. Note that $d_i + \beta$ indicates the scalar projection of $\mathbf{u}_i$ onto $\boldsymbol{\alpha}$. Since $\boldsymbol{\alpha}\mathbf{u}_i^{\mathrm{T}}$ is negative, the direction of $\boldsymbol{\alpha}$ is opposite to that of $\mathbf{u}_i$. The geometrical meaning of $d_i + \beta$ is shown in Figure 3.

To construct the optimal hyperplane, our task then becomes to find the maximum d, which can be obtained from the following optimization model

$$\begin{cases} \max\limits_{\boldsymbol{\alpha}, d} d \\ \text{s.t.} \quad -(\beta + \boldsymbol{\alpha}\mathbf{u}_i^{\mathrm{T}}) \geq d \quad (i = 1, 2, \ldots, m), \end{cases} \tag{9}$$

This is the basic model of the proposed one-class SVM with a bias constraint determined by the given constant β. Let $h = d + \beta$, and Eq. (9) is rewritten as

$$\begin{cases} \max\limits_{\boldsymbol{\alpha}, h} h \\ \text{s.t.} \quad -\boldsymbol{\alpha}\mathbf{u}_i^{\mathrm{T}} \geq h \quad (i = 1, 2, \ldots, m). \end{cases} \tag{10}$$

Setting $\boldsymbol{\omega} = \alpha/h$, we have $h = 1/\parallel \boldsymbol{\omega} \parallel$. Then Eq. (10) becomes

$$\begin{cases} \max\limits_{\boldsymbol{\omega}} \dfrac{1}{\parallel \boldsymbol{\omega} \parallel} \\ \text{s.t.} \quad \boldsymbol{\omega}\mathbf{u}_i^{\mathrm{T}} \leq -1 (i = 1, 2, \ldots, m), \end{cases} \tag{11}$$

which is equivalent to the constrained quadratic programing problem as follows:

$$\begin{cases} \min\limits_{\boldsymbol{\omega}} \dfrac{1}{2}\boldsymbol{\omega}\boldsymbol{\omega}^{\mathrm{T}} \\ \text{s.t.} \quad \boldsymbol{\omega}\mathbf{u}_i^{\mathrm{T}} + 1 \leq 0 \quad (i = 1, 2, \ldots, m). \end{cases} \tag{12}$$

https://doi.org/10.1017/S0890060419000155 Published online by Cambridge University Press

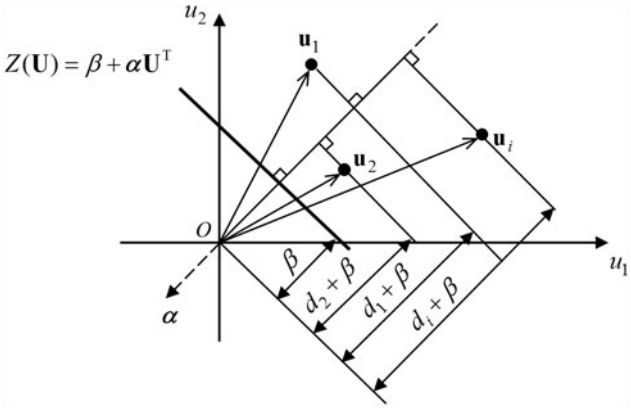


Fig. 3. Geometric meaning of $d_i + \beta$.

With Lagrange multipliers $\lambda_i \geq 0$, the Lagrangian function is given by

$$L(\boldsymbol{\omega}) = \frac{1}{2}\boldsymbol{\omega}\boldsymbol{\omega}^{\mathrm{T}} - \sum_{i=1}^{m} \lambda_i(-\boldsymbol{\omega}\mathbf{u}_i^{\mathrm{T}} - 1) \tag{13}$$

According to the KKT conditions, we have

$$\frac{\partial L}{\partial \boldsymbol{\omega}} = 0 \rightarrow \boldsymbol{\omega} = -\sum_{i=1}^{m} \lambda_i \mathbf{u}_i \tag{14}$$

Substituting Eq. (14) into Eq. (13), the Lagrangian function is rewritten as

$$\begin{aligned} L(\boldsymbol{\omega}) &= \frac{1}{2}\sum_{i,j=1}^{m} \lambda_i\lambda_j\mathbf{u}_i\mathbf{u}_j^{\mathrm{T}} - \sum_{i=1}^{m}\lambda_i\left(\left(\sum_{i=1}^{m}\lambda_i\mathbf{u}_i\right)\mathbf{u}_i^{\mathrm{T}} - 1\right) \\ &= \frac{1}{2}\sum_{i,j=1}^{m} \lambda_i\lambda_j\mathbf{u}_i\mathbf{u}_j^{\mathrm{T}} - \sum_{i,j=1}^{m}\lambda_i\lambda_j\mathbf{u}_i\mathbf{u}_j^{\mathrm{T}} + \sum_{i=1}^{m}\lambda_i \\ &= \sum_{i=1}^{m}\lambda_i - \frac{1}{2}\sum_{i,j=1}^{m}\lambda_i\lambda_j\mathbf{u}_i\mathbf{u}_j^{\mathrm{T}} \end{aligned} \tag{15}$$

Thus, the dual form of the quadratic programing problem in Eq. (12) is given by

$$\begin{cases} \max\limits_{\boldsymbol{\lambda}} \sum\limits_{i=1}^{m}\lambda_i - \frac{1}{2}\sum\limits_{i,j=1}^{m}\lambda_i\lambda_j\mathbf{u}_i\mathbf{u}_j^{\mathrm{T}} \\ \text{s.t.} \quad \lambda_i \geq 0, \forall i = 1, 2, \ldots, m \end{cases} \tag{16}$$

Solving the optimization model in Eq. (16) yields the Lagrange multipliers λ_1, λ_2, …, λ_m. Substituting them into Eq. (14) produces the weight vector $\boldsymbol{\omega}$. The unit vector $\boldsymbol{\alpha}$ is then recovered by $\boldsymbol{\alpha} = \boldsymbol{\omega}/\parallel \boldsymbol{\omega} \parallel$, which thereby constructs the function $Z(\mathbf{U})$ for the hyperplane. Similar to the general one-class SVM algorithm, the training points satisfying $\{\mathbf{x}_i: i \in 1, 2, \ldots, m, \lambda_i > 0\}$ are support vectors by which the optimal hyperplane is finally determined.

With the known bias β and the acquired normal vector $\boldsymbol{\alpha}$, the function of the hyperplane is determined by

$$Y = Z(\mathbf{U}) = \beta + \boldsymbol{\alpha}\mathbf{U}^{\mathrm{T}} \tag{17}$$

which is a decision boundary defining the domain of the one class data set and could then be used to predict the state of a new sample. Substitute a new sample $\mathbf{u}_N$ into Eq. (17). If $Y < 0$, the new sample belongs to the same class as that of the training points, and $y = -1$; otherwise, it is outside the domain of the training points and belongs to the class of $y = +1$.

The proposed one-class SVM algorithm can easily accommodate the bias constraint, which is derived from the given one-class data set. The new algorithm only focuses on this data set without considering the origin as the second class. The optimal hyperplane is constructed based on the hard margin associated with the bias constraint. Specifically, if we regard seeking such an optimal hyperplane as a dynamic process, the general one-class SVM technique attempts to move the hyperplane to the desired position through rotations and translations. While in the proposed method the hyperplane only rotates around the origin while keeping tangent to the hypersphere with a radius of β. In other words, the hyperplane rolls without slipping on the hypersphere. Also, since no slack variables ξ; and regularization parameter ν are introduced, the constraints for the optimization model are relatively simple thereby increasing the computation efficiency.

Application of the new one-class SVM in system reliability prediction

System reliability is the probability of a system working normally without failures. Since the system state (safe or failed) is determined by the states of its components and it may be hard to predict the system reliability directly, the system reliability is usually estimated based on component states. Physics-based methods (Cruse, 1997; Mahadevan, 1997; Hu and Du, 2016, 2019; Hu *et al.*, 2017) and statistics-based methods (Lawless, 1983; Hoyland and Rausand, 2004; Meeker and Escobar, 2014; Hu and Du, 2017*b*) are two possible choices for component reliability analysis. We first briefly review the concepts and basic techniques of the two kinds of reliability methods and then explain how the proposed algorithm works for system reliability analysis.

Physics-based reliability methods

Physics-based reliability methods use computational models to estimate reliability, which predict the component failure state based on physical principles. The computational model is called a limited-state function, denoted by $y = g(\mathbf{X})$, where $\mathbf{X}$ is a vector of basic random variables, which are root variables that affect the state of the failure mode, such as component shape and dimensions, loadings, material properties, and environmental factors; y is the state variable. For each failure mode, a limit-state function is built. If $Y > 0$, the state is safe. Otherwise, a failure occurs. The reliability with respect to the failure mode is given by

$$R = \Pr\{\text{state} = \text{safe}\} = \Pr\{y = g(\mathbf{X}) > 0\} \tag{18}$$

The probability of failure p_f is given by

$$p_f = \Pr\{\text{state} = \text{failed}\} = \Pr\{y = g(\mathbf{X}) < 0\} = 1 - R \tag{19}$$

Since it is difficult to compute Eq. (19) analytically, many approximation methods have been proposed, including FORM (Cruse, 1997; Chiralaksanakul and Mahadevan, 2005), the second order reliability method (Zhao and Ono, 1999), the saddlepoint approximation method (Du and Sudjianto, 2004; Hu and Du,

https://doi.org/10.1017/S0890060419000155 Published online by Cambridge University Press

2018c), the partial safety factor method (Hu and Du, 2018b), and Monte Carlo simulation (MCS) (Green *et al.*, 2013). In this work, we adopt FORM to approximate a linear form of $g(\mathbf{X})$, then the component probability of failure could be easily estimated. The procedure of FORM is briefly summarized in the following three steps.

Step 1: Transform random variables into standard normal variables

Assume that all the random variables in the X-space are independent. The original random variables $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ are transformed into standard normal random variables $\mathbf{U} = (U_1, U_2, \ldots, U_n)$ in the U-space. The transformation is given by (Rosenblatt, 1952)

$$F_i(x_i) = \Phi(u_i)(i = 1, 2, \ldots, n) \tag{20}$$

where $F_i(\cdot)$ and $\Phi(\cdot)$ are the cumulative distribution functions (CDF) of X_i and a standard normal variable, respectively. The transformation could also be given in the form of

$$x_i = T(u_i) = F^{-1}(\Phi(u_i)) \ (i = 1, 2, \ldots, n) \tag{21}$$

in which $T(\cdot)$ denotes the transformation function.

Step 2: Approximate a linear limit-state function

After the transformation, the component probability of failure is computed by

$$p_f = \Pr\{g(T(\mathbf{U})) < 0\} \tag{22}$$

FORM then approximates $g(T(\mathbf{U}))$ at the most probable point (Cruse, 1997), and yields a linear limit-state function

$$Z(\mathbf{U}) = \beta + \boldsymbol{\alpha}\mathbf{U}^{\mathrm{T}} \tag{23}$$

Step 3: Compute p_f

With the new limit-state function $Z(\mathbf{U})$ in Eq. (23), which is a linear combination of standard normal random variables, p_f is calculated by

$$p_f = \Pr\{Z(\mathbf{U}) < 0\} = \Phi(-\beta) \tag{24}$$

Statistics-based reliability methods

A statistics-based method relies on field or testing data related to failures of a component. The component reliability R is estimated by

$$R = \Pr\{\text{state} = \text{safe}\} \approx \frac{N - N_f}{N} \tag{25}$$

where $\Pr\{\cdot\}$ denotes a probability, N_f is the number of failed component, and N is the total number of components.

SVM is widely used with the statistics-based method which creates a reliability model using the provided training data with no need for physical principle of the component. Note that the recorded field or testing data belong to either the safe region or failure region. SVM can therefore identify the safety-failure boundary by solving a binary classification problem (Hu and Du, 2017b). As is mentioned above, the general two-class SVM is only available for cases where two classes of training data are provided.

Application of the new method

We now discuss how to use the proposed one-class SVM approach to achieve a linear decision boundary (limit-state function) if only a one-class training data set is given. The details are as follows.

We still use $y = g(\mathbf{X})$ as the component limit-state function, and the original random variables, denoted by $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ are independent. The counterpart of $\mathbf{X}$ in the U-space, denoted by $\mathbf{U} = (U_1, U_2, \ldots, U_n)$, are standard normal random variables. Given a data set of m training points from reliability testing at failure states as follows:

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m), \mathbf{x} \in R^n \quad y_i = -1, \quad i = 1, 2, \ldots, m$$

The bias β is known, which comes from the component reliability estimated by the supplier using the given training points.

Step 1: Transform $\mathbf{X}$ into $\mathbf{U}$

Similar to FORM, the transformation is given by

$$x_j = T(u_j) = F^{-1}(\Phi(u_j)), \ j = 1, 2, \ldots, n \tag{26}$$

Step 2: Approximate a linear limit-state function based on one-class SVM

According to the proposed one-class SVM discussed in the section "A new algorithm for one-class support vector machines with a bias constraint," the optimal normal orientation of the to-be-determined decision boundary is given by $\boldsymbol{\omega} = -\sum_{i=1}^{m} \lambda_i \mathbf{u}_i$, in which λ_i is available after solving the dual form of the Lagrangian in Eq. (16), and $\mathbf{u}_i$ is obtained in step 1. Since the bias β is also available, the linear form of the component limit-state function is then obtained by

$$Z(\mathbf{U}) = \beta + \boldsymbol{\alpha}\mathbf{U} \tag{27}$$

in which $\boldsymbol{\alpha} = \boldsymbol{\omega}/\|\boldsymbol{\omega}\|$.

Since β is known, there is no need to recalculate component reliability using $Z(\mathbf{U})$. $Z(\mathbf{U})$ is actually used for the system reliability prediction by integrating with other available limit-state functions from FORM. Next, we will discuss how to do so.

System reliability analysis

System reliability could be estimated either by a physics-based approach, a statistics-based approach, or the integration of both. Predicting system reliability is an important task, especially for systems with outsourced components. Outsourcing is a common practice because more and more industrial firms function like system integrators with numerous components outsourced (Click and Duening, 2004), resulting in urgent demand for integrating both statistics- and physics-based approaches. Accurately predicting the system reliability requires complete design information for both in-house and outsourced components, such as the limit-sate functions and distributions of basic random variables. System designers may know everything about the in-house components; however, the design details of outsourced components are usually unavailable since they are proprietary to outside suppliers. This makes it hard to directly use traditional methods for system reliability analysis (Cheng and Du, 2016). To address this issue, the proposed one-class SVM method with a bias constraint is used to reconstruct the limit-state functions for outsourced

https://doi.org/10.1017/S0890060419000155 Published online by Cambridge University Press

components, thereby integrating the new algorithm with physics-based methods for accurate system reliability prediction.

A proof-of-concept method (Hu and Du, 2017*a*) has been recently developed, and it validates the feasibility of this study. This work is an extension of the algorithm proposed in Hu and Du (2017*a*) with a bias constraint derived from the data set provided by the component supplier, such as the reliability data at failure states. We now introduce how to use the proposed method for system reliability prediction. The application scope is summarized as follows:

- The system has m components (failure modes) and $m \geq 2$.
- Component states are dependent.
- There are two types of components: (1) type-I components, whose probabilities of failure are obtained through physics-based methods, have available limit-state functions $g_i^{\mathrm{I}}(\cdot), (i = 1, 2, \ldots, m_1)$, where m_1 is the component number. (2) For the other $m_2 (m_2 = m - m_1)$ type-II components, no limit-sate functions are available, but the data set of training points from fields or testing is provided, and the probabilities of failure are evaluated using a statistics-based method.
- Assume the system is in series.
- Distributions of all basic random variables are known.

For a type-I component, the limit-state functions in the U-space are transformed by

$$g_i^{\mathrm{I}}(\mathbf{X}) \xrightarrow{\mathbf{X} \to T(\mathbf{U})} Z_i^{\mathrm{I}}(\mathbf{U}) = \beta_i^{\mathrm{I}} + \boldsymbol{\alpha}_i^{\mathrm{I}} \mathbf{U}^T (i = 1, 2, \ldots, m_1) \quad (28)$$

For type-II components, the limit-state functions produced by the proposed one-class SVM is given in the form of

$$\begin{aligned} Z_j^{\mathrm{II}}(\mathbf{U}) &= \beta_j^{\mathrm{II}} + \frac{\boldsymbol{\omega}_j}{\| \boldsymbol{\omega}_j \|} \mathbf{U}^T \\ &= \beta_j^{\mathrm{II}} + \boldsymbol{\alpha}_j^{\mathrm{II}} \mathbf{U}^T (j = m_1 + 1,\ m_1 + 2, \ldots, m) \end{aligned} \quad (29)$$

Since the components of $\mathbf{U}$ follow a standard normal distribution, the reconstructed limit-state functions $Z_i^{\mathrm{I}}(\mathbf{U})$ and $Z_j^{\mathrm{II}}(\mathbf{U})$ also follow normal distributions $Z_i^{\mathrm{I}}(\mathbf{U}) \sim N(\mu_i^{\mathrm{I}}, \sigma_i^{\mathrm{I}})$ and $Z_j^{\mathrm{II}}(\mathbf{U}) \sim N(\mu_j^{\mathrm{II}}, \sigma_j^{\mathrm{II}})$, respectively, in which $\mu_i^{\mathrm{I}} = \beta_i^{\mathrm{I}}$ and $\mu_j^{\mathrm{II}} = \beta_j^{\mathrm{II}}$ are their vectors of means, and the covariance of $Z_i^{\mathrm{I}}(\mathbf{U})$ and $Z_j^{\mathrm{II}}(\mathbf{U})$ is ρ_{ij}, which will be given in Eq. (32). Thus, the joint PDF of $Z_i^{\mathrm{I}}(\mathbf{U})$ and $Z_j^{\mathrm{II}}(\mathbf{U})$, denoted by $\phi_{\mathbf{U}}(\mathbf{u})$, is actually the PDF of a multivariate normal distribution with a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$. $\boldsymbol{\mu}$ is given by

$$\boldsymbol{\mu} = (\beta_1^{\mathrm{I}}, \beta_2^{\mathrm{I}}, \ldots, \beta_{m_1}^{\mathrm{I}}, \beta_{m_1+1}^{\mathrm{II}}, \beta_{m_1+2}^{\mathrm{II}}, \ldots, \beta_m^{\mathrm{II}}) \quad (30)$$

in which $\beta_i^{\mathrm{I}} (i = 1, 2, \ldots, m_1)$ is obtained from FORM, and $\beta_j^{\mathrm{II}} (j = m_1+1, m_1+2, \ldots, m)$ is calculated by $\beta_j^{\mathrm{II}} = -\Phi^{-1}(p_{\mathrm{fj}})$. $\boldsymbol{\Sigma}$ is given by

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1 & \rho_{12} & \cdots & \rho_{1m} \\ \rho_{21} & 1 & & \rho_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{m1} & \rho_{m2} & \cdots & 1 \end{bmatrix}_{m \times m} \quad (31)$$

in which ρ_{ij} is the correlation coefficient between the i-th and j-th components and is computed by

$$\rho_{ij} = \rho_{ji} = \begin{cases} \boldsymbol{\alpha}_i^{\mathrm{I}} (\boldsymbol{\alpha}_j^{\mathrm{I}})^{\mathrm{T}}, i < j \leq m_1 \\ \boldsymbol{\alpha}_i^{\mathrm{I}} (\boldsymbol{\alpha}_j^{\mathrm{II}})^{\mathrm{T}}, i \leq m_1 < j \\ \boldsymbol{\alpha}_i^{\mathrm{II}} (\boldsymbol{\alpha}_j^{\mathrm{II}})^{\mathrm{T}}, m_1 < i < j \end{cases} \quad (32)$$

From Eq. (29), we find $\boldsymbol{\alpha}_j^{\mathrm{II}} = (\boldsymbol{\omega}_j / \| \boldsymbol{\omega}_j \|) (j = m_1+1, m_1+2, \ldots, m)$, in which $\boldsymbol{\alpha}_j^{\mathrm{II}}$ has the same direction as $\boldsymbol{\omega}_j$.

With $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ available, the complete joint PDF $\phi_{\mathbf{U}}(\mathbf{u})$ is also available and is given by

$$\phi_{\mathbf{U}}(\mathbf{u}) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2} (\mathbf{u} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{u} - \boldsymbol{\mu}) \right) \quad (33)$$

The probability of system failure is computed by

$$p_{\mathrm{fs}} = \Pr \bigcup_{i=1}^{m_1} Z_i^{\mathrm{I}}(\mathbf{U}) < 0 \bigcup \bigcup_{j=m_1+1}^{m} Z_j^{\mathrm{II}}(\mathbf{U}) < 0 \quad (34)$$

And the system reliability is

$$\begin{aligned} R_{\mathrm{s}} &= \Pr\left(\bigcap_{i=1}^{m_1} -Z_i^{\mathrm{I}}(\mathbf{U}) < 0 \cap \bigcap_{j=m_1+1}^{m} -Z_j^{\mathrm{II}}(\mathbf{U}) < 0 \right) \\ &= \int_{\Omega} \phi_{\mathbf{U}}(\mathbf{u}) d\mathbf{u} \end{aligned} \quad (35)$$

where Ω is the system safe region defined by

$$\begin{aligned} \Omega = \{ \mathbf{U} | &- Z_i^{\mathrm{I}}(\mathbf{U}) < 0, -Z_j^{\mathrm{II}}(\mathbf{U}) \\ &< 0 (i = 1, 2, \ldots, m_1; j = m_1 + 1, m_1 + 2, \ldots, \mathrm{m}) \} \end{aligned} \quad (36)$$

Thus R_{s} can be easily evaluated by solving the integral in Eq. (35), and the probability of system failure is then $p_{\mathrm{f_s}} = 1 - R_{\mathrm{s}}$. A schematic diagram of the proposed method is given in Figure 4.

The proposed method makes the following contributions to reliability analysis: (1) at the component level, it provides a new way to approximate component limit-state functions with only estimated probabilities of failure and limited field or testing failure data. (2) At the system level, since the provided component limit-state functions are linearized using FORM, which produce the same form as the approximated limit-state functions obtained from the proposed one-class SVM, system reliability analysis could be easily conducted. (3) It improves the accuracy of the system reliability prediction because it accounts for the dependency between components automatically. (4) It dramatically reduces the computational cost due to the linear forms of all the limit-state functions.

Examples of methodology validation

Three examples are used to demonstrate the effectiveness and accuracy of the proposed method. Example 1 is a numerical problem showing how to apply the proposed method step by step. Examples 2 and 3 involve engineering problems with multiple failure modes.

https://doi.org/10.1017/S0890060419000155 Published online by Cambridge University Press

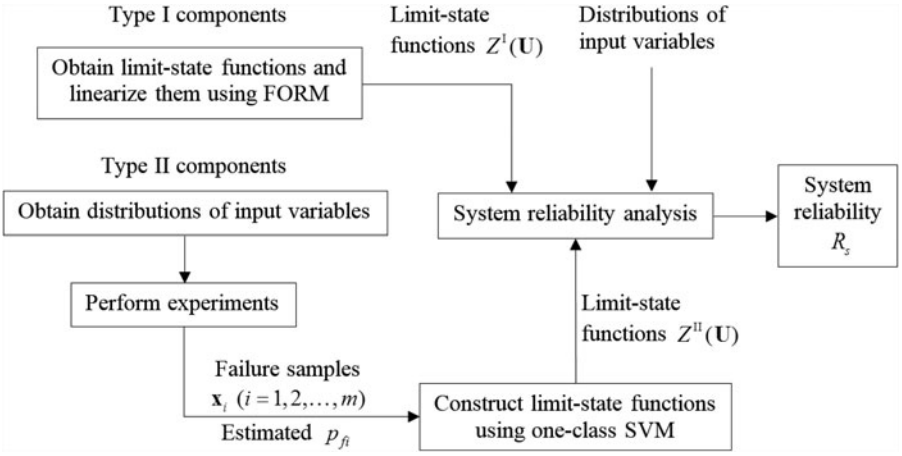


Fig. 4. Schematic diagram of the proposed method.

Example 1: a numerical problem

A system is comprised of two physical components, and each has one failure mode. If either of components fails, the system fails. There are two independent basic random variables $\mathbf{X} = (X_1, X_2)$, where both X_1 and X_2 follow normal distributions of $X_1 \sim N(12, 1^2)$ and $X_2 \sim N(40, 2^2)$, respectively. The limit-state function of the first component is available and is given by

$$g_1^{\mathrm{I}}(\mathbf{X}) = -260 + 8.5X_1 + 5.2X_2 \tag{37}$$

Thus, the component is a type-I component.

FORM produces a linear model, which is given by

$$Z_1^{\mathrm{I}}(\mathbf{U}) = \beta_1^{\mathrm{I}} + \boldsymbol{\alpha}_1^{\mathrm{I}}\mathbf{U}^{\mathrm{T}} = 3.7225 + 0.6328U_1 + 0.7743U_2 \tag{38}$$

in which $\beta_1^{\mathrm{I}} = 3.7225$ and $\boldsymbol{\alpha}_1^{\mathrm{I}} = (0.6328, 0.7743)$.

Component two is a type-II component since no model is available. The probability of failure $p_{\mathrm{f2}} = 2.5517 \times 10^{-5}$ is estimated by a statistics-based reliability method using the recorded testing points, which come from reliability testing.

Although no model is available, to analyze the accuracy, we assume the true model in the X-space is given by

$$g_2^{\mathrm{II(true)}}(\mathbf{X}) = -325 + 5.6X_1 + 8.2X_2 \tag{39}$$

The linear model in the U-space is

$$Z_2^{\mathrm{II(true)}}(\mathbf{U}) = 4.0508 + 0.3231U_1 + 0.9463U_2 \tag{40}$$

We then use computer experiments to mimic the physical reliability testing. With MCS and the model in Eq. (39), we generate a set of training points and transform them into the U-space as shown in Table 1. We also assume that the value of p_{f2} given above is known and is equal to the one estimated using Eq. (39).

Assume that the linear model for component two is given by

$$Z_2^{\mathrm{II}}(\mathbf{U}) = \beta_2^{\mathrm{II}} + \boldsymbol{\alpha}_2^{\mathrm{II}}\mathbf{U}^{\mathrm{T}} \tag{41}$$

in which β_2^{II} is calculated by $\beta_2^{\mathrm{II}} = -\Phi^{-1}(p_{\mathrm{f2}}) = 4.0508$, and $\boldsymbol{\alpha}_2^{\mathrm{II}}$ is the to-be-determined unit vector. Using the proposed one-class

Table 1. Training points

	u	
No.	U_1	U_2
1	−1.4342	−3.8442
2	−2.1810	−4.6721
3	−0.5057	−4.1478
4	−1.3443	−3.9958
5	−1.9988	−3.9879
6	−1.7684	−3.7517
7	−2.5285	−3.6413
8	−1.9114	−3.8673
9	−3.0726	−3.3558
10	−0.8223	−4.1205

SVM method, we solve for $\boldsymbol{\alpha}_2^{\mathrm{II}}$ by

$$\begin{cases} \max\limits_{\boldsymbol{\lambda}} \sum\limits_{i=1}^{10} \lambda_i - \dfrac{1}{2}\sum\limits_{i,j=1}^{10} \lambda_i\lambda_j\mathbf{u}_i\mathbf{u}_j^{\mathrm{T}} \\ \text{s.t.} \quad \lambda_i \geq 0, \forall i = 1, 2, \ldots, 10 \end{cases} \tag{42}$$

where $\mathbf{u}_1 = (-1.4342, -3.8442)$, $\mathbf{u}_2 = (-2.1810, -4.6721)$, …, and $\mathbf{u}_{10} = (-0.8223, -4.1205)$ as shown in Table 1. After solving the above model, we have the Lagrange multipliers $\boldsymbol{\lambda} = (0.0459, 0, 0.0099, 0, 0, 1.734 \times 10^{-4}, 0, 0, 0, 0)$, and the three support vectors $\mathbf{u}_1$, $\mathbf{u}_3$, and $\mathbf{u}_6$ marked by the circles in Figure 5 are determined by the non-zero multipliers. Substituting $\boldsymbol{\lambda}$, $\mathbf{u}_1$, $\mathbf{u}_2$, …, and $\mathbf{u}_{10}$ into $\boldsymbol{\omega}_2 = -\sum_{i=1}^{10} \lambda_i\mathbf{u}_i$, we obtain the weight vector $\boldsymbol{\omega}_2 = (0.0712, 0.2182)$, resulting in a unit vector $\boldsymbol{\alpha}_2^{\mathrm{II}} = (\boldsymbol{\omega}_2 / \| \boldsymbol{\omega}_2 \|) = (0.3101, 0.9507)$. Thus the linear model of component two is reconstructed by

$$Z_2^{\mathrm{II}}(\mathbf{U}) = \beta_2^{\mathrm{II}} + \boldsymbol{\alpha}_2^{\mathrm{II}}\mathbf{U}^{\mathrm{T}} = 4.0508 + 0.3101U_1 + 0.9507U_2 \tag{43}$$

The corresponding optimal hyperplane is also shown in Figure 5, separating the one class training points (data set at failure state) clearly from the circle with a radius of β_2^{II}.

https://doi.org/10.1017/S0890060419000155 Published online by Cambridge University Press

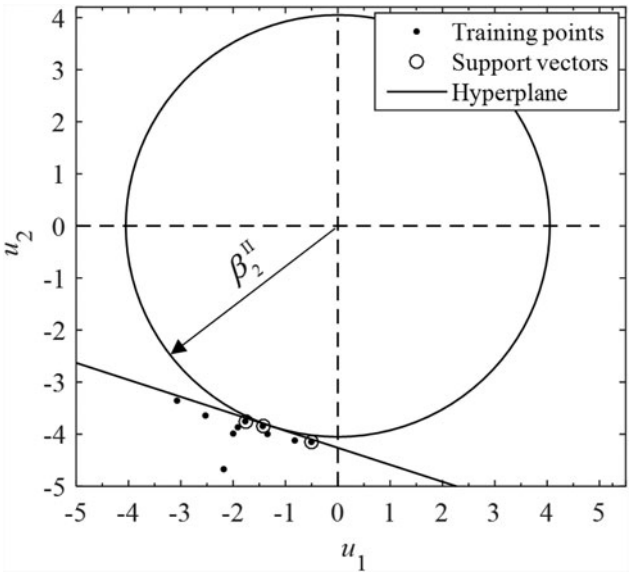


Fig. 5. Support vectors and optimal hyperplane.

The approximated limit-state function in Eq. (43) is very close to the true one given in Eq. (40), thereby leading to high accuracy of system reliability prediction, the details of which are shown below.

Since the components of $\mathbf{u}$ follow standard normal distributions, the two dimensional random vector $\mathbf{Z} = [Z_1^{\mathrm{I}}(\mathbf{U}), Z_2^{\mathrm{II}}(\mathbf{U})]$ follows a multivariate normal distribution with the joint PDF

$$\phi_{\mathbf{U}}(\mathbf{u}) = \frac{1}{\sqrt{(2\pi)^2|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{u}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{u}-\boldsymbol{\mu})\right) \tag{44}$$

where the mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ are given by

$$\boldsymbol{\mu} = (\beta_1^{\mathrm{I}}, \beta_2^{\mathrm{II}}) = (3.7225, 4.0508) \tag{45}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1 & \rho_{12} \\ \rho_{21} & 1 \end{bmatrix} = \begin{bmatrix} 1 & \boldsymbol{\alpha}_1^{\mathrm{I}}(\boldsymbol{\alpha}_2^{\mathrm{II}})^{\mathrm{T}} \\ \boldsymbol{\alpha}_1^{\mathrm{I}}(\boldsymbol{\alpha}_2^{\mathrm{II}})^{\mathrm{T}} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0.9324 \\ 0.9324 & 1 \end{bmatrix} \tag{46}$$

The system reliability is calculated by

$$R_{\mathrm{s}} = \Pr\left(-Z_1^{\mathrm{I}}(\mathbf{U}) < 0 \cap -Z_2^{\mathrm{II}}(\mathbf{U}) < 0\right) = \int_{\Omega} \phi_{\mathbf{U}}(\mathbf{u}) d\mathbf{u} \tag{47}$$

where Ω is the system safe region defined by

$$\Omega = \{\mathbf{U} | -Z_1^{\mathrm{I}}(\mathbf{U}) < 0, -Z_2^{\mathrm{II}}(\mathbf{U}) < 0\} \tag{48}$$

Substituting Eq. (44) into Eq. (47), we have $p_{\mathrm{f_s}} = 1 - R_{\mathrm{s}} = 1.0537 \times 10^{-4}$.

We now discuss the case where the traditional system reliability method is used and then compare the results from both methods. The traditional method (Yong Cang, 1993) assumes that the states of all the components are independent. Then the system reliability is calculated by

$$R_{\mathrm{s}} = \prod_{i=1}^{m} R_i \tag{49}$$

where R_i is the reliability of the i-th component. The result is given in Table 2 in the "Independence assumption method" column. Although this method is easy to use and effective, it may produce large errors when the components are highly dependent.

To verify the accuracy, we also use the true limit-state functions $g_1^{\mathrm{I}}(\mathbf{X})$ and $g_2^{\mathrm{II(true)}}(\mathbf{X})$ in Eqs. (45) and (47) to evaluate the system reliability based on FORM and consider this value as a benchmark. The result obtained is 1.0478×10^{-4}. Table 2 shows all the results from different methods. The independence assumption method has a large error of 18.46%, which is due to the neglected strong correlation indicated by $\rho_{12} = 0.9324$. The proposed method produces an error of only 0.56%, which shows much higher accuracy.

Table 2. Results of system reliability from different methods

	Proposed method	Independent assumption method	True value
p_{fs}	1.0537×10^{-4}	1.2413×10^{-4}	1.0478×10^{-4}
Error (%)	0.56	18.46	–

Example 2: a cantilever beam

A cantilever beam is subject to moments m_1 and M_2, forces Q_1 and Q_2, and distributed loads denoted by (q_{L1}, q_{R1}) and (q_{L2}, q_{R2}) as shown in Figure 6. Assume that m_1, M_2, and Q_1; the dimensions variables a_1, a_2, and b_1; the yield strength S_a; and the allowable shear stress τ_a are basic random variables, which are assumed to be independent and listed in Table 3. Deterministic parameters are listed in Table 4.

The cantilever beam fails due to three failure modes, and each is considered as a component, thus the reliability of the beam is regarded as system reliability. The first failure mode is caused by excessive normal stress, and its limit-state function is known and given by

$$g_1^{\mathrm{I}}(\mathbf{X}) = S_a - \frac{6M}{wh^2} \tag{50}$$

in which M is the bending moment at the root calculated by

$$M = \sum_{i=1}^{2} M_i + \sum_{i=1}^{2} F_i b_i + \sum_{i=1}^{2} q_{Li}(d_i - c_i)(d_i + c_i)/2 + \sum_{i=1}^{2} [(q_{Ri} - q_{Li})(d_i - c_i)/2][c_i + 2(d_i - c_i)/3] \tag{51}$$

Since the limit-state function is provided, this failure mode is treated as a type-I component.

The second failure mode comes from the excessive shear stress with a known limit-state function given by

$$g_2^{\mathrm{I}}(\mathbf{X}) = \tau_a - \tau_{\max} \tag{52}$$

https://doi.org/10.1017/S0890060419000155 Published online by Cambridge University Press

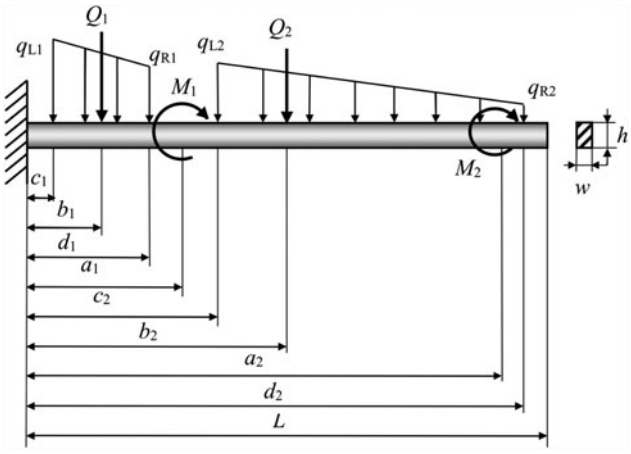


Fig. 6. A cantilever beam system.

in which τ_a is the allowable shear stress, and $\tau_{\max}$ is the maximal shear stress computed by

$$\tau_{\max} = \frac{3}{2wh}\left(\sum_{i=1}^{2} F_i + \sum_{i=1}^{2} q_{Li}(d_i - c_i) + \sum_{i=1}^{2} \frac{(q_{Ri} - q_{Li})(d_i - c_i)}{2}\right) \tag{53}$$

Similarly, this failure mode is also a type-I component.

The third failure mode (FM3) is due to the excessive deflection with an unknown limit-state function. It is therefore a type-II component. The probability of failure p_{f3} due to this failure mode is then evaluated using statistics-based methods with training points. Note that the training points used in this example actually come from computer simulation, since it is hard for us to perform real physical experiments due to lack of measuring devices. Assume the true limit-state function for FM3 is

$$g_3^{\mathrm{II(true)}}(\mathbf{X}) = v_a - v_{\max} \tag{54}$$

in which $v_a = 8.4$mm is the allowable deflection, and $v_{\max}$ is the maximal tip deflection given by

$$\begin{aligned} v_{\max} = {} & \frac{1}{EI}\left[\frac{ML^2}{2} + \frac{BL^3}{2} + \sum_{i=1}^{2}\frac{M_i(L-a_i)^2}{2} - \sum_{i=1}^{2}\frac{F_i(L-b_i)^3}{6}\right] \\ & + \frac{1}{EI}\left[-\sum_{i=1}^{2}\frac{q_{Li}(L-c_i)^4}{24} - \sum_{i=1}^{2}\frac{(q_{Ri}-q_{Li})(L-c_i)^5}{120(d_i-c_i)}\right. \\ & \left. + \sum_{i=1}^{2}\frac{q_{Ri}(L-d_i)^4}{24}\right] + \frac{1}{EI}\sum_{i=1}^{2}\frac{(q_{Ri}-q_{Li})(L-d_i)^5}{120(d_i-c_i)} \end{aligned} \tag{55}$$

where B is the reaction force at the fixed end. The Young's modulus is $E = 200 \times 10^9$ Pa, and the moment of inertia is $I = wh^3/12$. Based on the given limit-state function in Eq. (54), 12 training points at failure states are generated by simulation and are transformed into the U-space as listed in Table 5. Since S_a and τ_a do not affect the third failure mode, their components U_7 and U_8 are absent in the training points. As discussed previously, p_{f3} is estimated by a statistics-based reliability method using the data set from reliability testing and is assumed equal to the probability of failure $p_{f3} = 2.864 \times 10^{-4}$ produced by FORM with the true-limit state function in Eq. (54).

Table 3. Basic random variables

	Random variables	Distribution
X_1	M_1(Nm)	$N(50 \times 10^3, (2 \times 10^3)^2)$
X_2	M_2(Nm)	$N(30 \times 10^3, (2 \times 10^3)^2)$
X_3	a_1(m)	$N(1.5, 0.005^2)$
X_4	a_2(m)	$N(4.5, 0.005^2)$
X_5	Q_1(N)	$N(65 \times 10^3, (13 \times 10^3)^2)$
X_6	b_1(m)	$N(0.7, 0.005^2)$
X_7	S_o(Pa)	$N(62.5 \times 10^6, (1 \times 10^6)^2)$
X_8	τ_o(Pa)	$N(3.6 \times 10^6, (1 \times 10^5)^2)$

Table 4. Deterministic parameters

	Parameters	Values
1	Q_2(N)	30×10^3
2	b_2(m)	2.5
3	q_{L1}(N/m)	30×10^3
4	q_{L2}(N/m)	20×10^3
5	c_1(m)	0.25
6	c_2(m)	1.75
7	q_{R1}(N/m)	20×10^3
8	q_{R2}(N/m)	1×10^3
9	d_1(m)	1.25
10	d_2(m)	4.75
11	L(m)	5.1
12	w(m)	0.204
13	h(m)	0.403

Assume the linear model for FM3 is given by

$$Z_3^{\mathrm{II}}(\mathbf{U}) = \beta_3^{\mathrm{II}} + \boldsymbol{\alpha}_3^{\mathrm{II}}\mathbf{U}^{\mathrm{T}} \tag{56}$$

where $\beta_3^{\mathrm{II}} = -\Phi^{-1}(p_{f3}) = 3.4442$. $\boldsymbol{\alpha}_3^{\mathrm{II}}$ is obtained from the following optimization model

$$\begin{cases} \max\limits_{\boldsymbol{\lambda}} \sum\limits_{i=1}^{12}\lambda_i - \dfrac{1}{2}\sum\limits_{i,j=1}^{12}\lambda_i\lambda_j\mathbf{u}_i\mathbf{u}_j^{\mathrm{T}} \\ \text{s.t. } \lambda_i \geq 0, \forall i = 1, 2, \ldots, 12 \end{cases} \tag{57}$$

in which $\mathbf{u}_i$ represent the training points given in Table 5. Solving the above the model, we obtain the Lagrange multipliers $\boldsymbol{\lambda} = (0, 0, 0.0345, 0, 0.0047, 0.0037, 0.0203, 0, 0, 4.25 \times 10^{-4}, 0, 0.0108)$; therefore, six support vectors $\mathbf{u}_3$, $\mathbf{u}_5$, $\mathbf{u}_6$, $\mathbf{u}_7$, $\mathbf{u}_{10}$, and $\mathbf{u}_{12}$ are determined by the nonzero components λ_3, λ_5, λ_6, λ_7, λ_{10}, and λ_{12} in $\boldsymbol{\lambda}$. Then using $\boldsymbol{\lambda}$, $\mathbf{u}_1$, $\mathbf{u}_2$, …, and $\mathbf{u}_{12}$ in $\boldsymbol{\omega}_3 = -\sum_{i=1}^{12}\lambda_i\mathbf{u}_i$, we have $\boldsymbol{\omega}_3 = (-0.0787, -0.2207, 0.0113, -0.0023, -0.1157, -0.0197)$, which produces the unit vector $\boldsymbol{\alpha}_3^{\mathrm{II}} = (\boldsymbol{\omega}_3 / \| \boldsymbol{\omega}_3 \|)$

https://doi.org/10.1017/S0890060419000155 Published online by Cambridge University Press

Table 5. Training points for FM3

	U					
No.	$U_1(M_1)$	$U_2(M_2)$	$U_3(a_1)$	$U_4(a_2)$	$U_5(F_1)$	$U_6(b_1)$
1	2.8351	2.0504	−1.6190	−1.5710	2.0410	0.8219
2	1.7433	2.2424	−0.1380	0.9161	2.5797	2.1406
3	1.0681	2.6931	−1.2685	−0.5628	2.0397	−0.1730
4	4.0026	2.6572	−1.4512	−0.4765	2.0030	−0.4871
5	2.2738	3.5800	0.4564	0.7455	−0.2342	−0.7254
6	0.2692	2.6336	1.3638	0.2184	2.8383	0.4362
7	1.0906	3.1686	0.4705	−0.3467	0.9793	1.6114
8	1.0306	3.3218	−0.1654	−1.9311	0.9750	1.0332
9	1.4163	3.3888	0.7300	−0.4728	1.1202	0.9889
10	1.0380	3.0489	−0.2377	−0.0370	1.3500	0.6203
11	1.0432	2.6388	−1.6206	0.3320	2.6083	−0.3964
12	0.6974	3.2863	1.4655	2.2559	1.4333	−0.5095

$= (-0.3001, -0.8414, 0.0430, -0.0087, -0.4409, -0.0750)$. Thus, the linear model in Eq. (56) is determined and is given by

$$Z_3^{\mathrm{II}}(\mathbf{U}) = 3.4442 - 0.3001U_1 - 0.8414U_2 + 0.0113U_3 - 0.0023U_4 - 0.1157U_5 - 0.0197U_6 \quad (58)$$

Since the first two failure modes are type-I components, FORM could be directly used with the following linear models:

$$Z_i^{\mathrm{I}}(\mathbf{U}) = \beta_i^{\mathrm{I}} + \boldsymbol{\alpha}_i^{\mathrm{I}}\mathbf{U}^{\mathrm{T}} (i = 1, 2) \quad (59)$$

in which $\beta_1^{\mathrm{I}} = 3.4989$, $\beta_2^{\mathrm{I}} = 3.2470$, $\boldsymbol{\alpha}_1^{\mathrm{I}} = (-0.181, -0.181, 0, 0, -0.826, -0.046)$, and $\boldsymbol{\alpha}_2^{\mathrm{I}} = (0, 0, 0, 0, -0.92, 0)$.

Thus, vectors $[Z_1^{\mathrm{I}}(\mathbf{U}), Z_2^{\mathrm{I}}(\mathbf{U}), Z_3^{\mathrm{II}}(\mathbf{U})]$ follow a multivariate normal distribution with the joint PDF given by

$$\phi_{\mathbf{U}}(\mathbf{u}) = \frac{1}{\sqrt{(2\pi)^3|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{u}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{u}-\boldsymbol{\mu})\right) \quad (60)$$

where the mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ are given by

$$\boldsymbol{\mu} = (\beta_1^{\mathrm{I}}, \beta_2^{\mathrm{I}}, \beta_3^{\mathrm{II}}) = (3.4989, 3.2470, 3.4442)$$

and

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1 & \rho_{12} & \rho_{13} \\ \rho_{12} & 1 & \rho_{23} \\ \rho_{13} & \rho_{23} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0.7608 & 0.5744 \\ 0.7608 & 1 & 0.4062 \\ 0.5744 & 0.4062 & 1 \end{bmatrix},$$

where ρ_{12}, ρ_{13}, and ρ_{23} are the correlation coefficients between $Z_1^{\mathrm{I}}(\mathbf{U})$ and $Z_2^{\mathrm{I}}(\mathbf{U})$, $Z_1^{\mathrm{I}}(\mathbf{U})$, and $Z_3^{\mathrm{II}}(\mathbf{U})$, and $Z_2^{\mathrm{I}}(\mathbf{U})$ and $Z_3^{\mathrm{II}}(\mathbf{U})$, respectively.

The system reliability is then calculated by

$$R_{\mathrm{s}} = \Pr\left(-Z_1^{\mathrm{I}}(\mathbf{U}) < 0 \cap -Z_2^{\mathrm{I}}(\mathbf{U}) < 0 \cap -Z_3^{\mathrm{II}}(\mathbf{U}) < 0\right) = \int_{\Omega} \phi_{\mathbf{U}}(\mathbf{u}) d\mathbf{u} \quad (61)$$

where Ω is the system safe region defined by

$$\Omega = \{\mathbf{U} | -Z_1^{\mathrm{I}}(\mathbf{U}) < 0, -Z_2^{\mathrm{II}}(\mathbf{U}) < 0, -Z_3^{\mathrm{II}}(\mathbf{U}) < 0\} \quad (62)$$

Then Eq. (61) yields $p_{\mathrm{f_s}} = 1 - R_{\mathrm{s}} = 1.022 \times 10^{-3}$.

For validation, we use FORM and all the given limit-state functions $g_1^{\mathrm{I}}(\mathbf{X})$, $g_2^{\mathrm{I}}(\mathbf{X})$, and $g_3^{\mathrm{II}}(\mathbf{X})$ to solve for the true system reliability. Likewise, we also use the independence assumption method. The results are shown in Table 6. The proposed method outperforms the independence assumption method with much higher accuracy.

Example 3: a slider mechanism

As shown in Figure 7, a slider system consists of four major components. An external moment is applied to joint A. The task is to find the system reliability when $\theta_2 = \pi/2$.

Component 1 is beam AB with a length of l_1, and its cross-section is defined by the width b_1 and height h_1. The failure mode (FM1) of AB is the excessive normal stress, and the limit-state function is known and is given by

$$g_1^{\mathrm{I}}(\mathbf{X}) = S_{a1} - S_1 \quad (63)$$

in which S_{a1} is the allowable normal stress, and $S_1 = (M(h_1/2))/(b_1 h_1^3/12)$ is the maximal normal stress developed in the beam.

Component 2 is beam BC with a length of l_2 and a cross-section defined by the width b_2 and height h_2. Beam BC has one failure mode (FM2) due to buckling with a known limit-state function given by

$$g_2^{\mathrm{I}}(\mathbf{X}) = P_{\mathrm{cr}} - F_{\mathrm{BC}} \quad (64)$$

https://doi.org/10.1017/S0890060419000155 Published online by Cambridge University Press

Table 6. Results from different methods

	Proposed method	Independent assumption method	True value
p_{f_s}	1.0198×10^{-3}	1.1028×10^{-3}	1.0155×10^{-3}
Error (%)	0.42	8.59	–

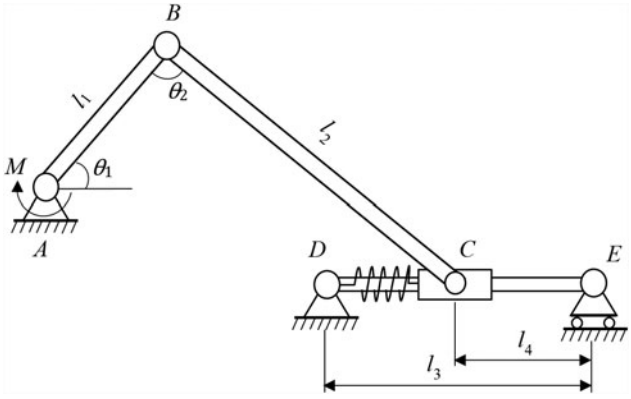


Fig. 7. A crank-slider system.

in which $P_{cr} = \pi^2 E_2 I_2/(Kl_2)^2$ is the critical force for buckling where $I_2 = b_2 h_2^3/12$, and $F_{BC} = M/l_1$ is the internal force in the beam.

Component 3 is shaft DE with a diameter of d_4 and a length of l_3. Two failure modes (FM3 and FM4) exist in the shaft which are caused by excessive deflection and excessive normal stress, respectively. The corresponding limit-state functions are known and given by

$$\begin{cases} g_3^{\mathrm{I}}(\mathbf{X}) = \delta_{a3} - \delta_3 \\ g_4^{\mathrm{I}}(\mathbf{X}) = S_{a4} - S_4 \end{cases} \tag{65}$$

in which δ_{a3} is the allowable deflection, and δ_3 is the maximal deflection given by

$$\delta_3 = \frac{F_{BC} \sin(\pi/2 - \theta_1) l_4 (l_3^2 - l_4^2)^{3/2}}{9\sqrt{3} l_4 E_4 (\pi/4)(d_4/2)^4} \tag{66}$$

where E_4 is the Young's modulus of shaft DE. S_{a4} is the allowable normal stress, and S_4 is the maximal normal stress developed in the shaft and is calculated by

$$S_4 = \frac{F_{BC} \sin(\pi/2 - \theta_1)(d_4/2)}{(\pi/4)(d_4/2)^4} \tag{67}$$

Component 4 is spring CD with one failure mode (FM5) due to excessive shear stress in the spring coils. The limit-state function of FM5 is unknown while the probability of failure is given by $p_{f3} = 1.04 \times 10^{-3}$. Likewise, to simulate the testing, we assume the true limit-state function is given by

$$g_5^{\mathrm{II(true)}}(\mathbf{X}) = \tau_{a5} - \tau_5 \tag{68}$$

in which $\tau_{a5} \sim N(100 \times 10^6, (25 \times 10^6)^2)$Pa is the allowable shear stress of the spring coil, and τ_5 is the developed maximal shear stress and calculated by

$$\tau_5 = \frac{F_{BC} \cos(\pi/2 - \theta_1) D}{\pi d^3} \left(\frac{4D - d}{4D - 4d} + \frac{0.615d}{D} \right) \tag{69}$$

in which $D \sim N(34.7 \times 10^{-3}, 10^{-4})$m is the outer diameter of the spring, and $d = 29.5 \times 10^{-3}$m is the spring inner diameter. We then generate 12 training points of $\mathbf{X}$ with the corresponding failure states determined by $g_5^{\mathrm{II(true)}}(\mathbf{X}) < 0$.

Table 7. Random variables

	Random variables	Distribution
X_1	M_1(Nm)	$N(350, 65^2)$
X_2	l_1(m)	$N(0.3, (1 \times 10^{-4})^2)$
X_3	l_2(m)	$N(0.9, (2 \times 10^{-3})^2)$
X_4	b_1(m)	$N(0.022, (5 \times 10^{-4})^2)$
X_5	h_1(m)	$N(0.019, (5 \times 10^{-4})^2)$
X_6	b_2(m)	$N(0.015, (5 \times 10^{-4})^2)$
X_7	h_2(m)	$N(0.009, (5 \times 10^{-4})^2)$
X_8	d_4(m)	$N(0.0228, (1 \times 10^{-4})^2)$

Table 8. Deterministic parameters

No.	Deterministic parameters	Values
1	E_2(Pa)	200×10^9
2	E_4(Pa)	200×10^9
3	K	1
4	l_3(m)	0.65
5	l_4(m)	0.95
6	S_{a1}(Pa)	400×10^6
7	S_{a4}(Pa)	460×10^6
8	δ_{a4}(m)	0.0032

Table 7 shows all the random variables known by the system designers, and Table 8 lists all the known deterministic parameters. Since D and τ_{a5}, denoted by X_9 and X_{10}, respectively, are only known by the spring supplier, they are not listed in Table 7. There are actually ten basic random variables in the system. For FM5, the training points are provided in the form of (X_1, X_2, X_9, X_{10}).

At the system level, the five FMs in the system are actually treated as five components. The first four FMs with known limit-state functions $g_i^{\mathrm{I}}(\mathbf{X})(i = 1, 2, 3, 4)$ are type-I components, and FM5 is a type-II component since its limit-state function $g_5^{\mathrm{II}}(\mathbf{X})$ is not available.

For type-I components, $g_i^{\mathrm{I}}(\mathbf{X})$ could be approximated by FORM as

$$Z_i^{\mathrm{I}}(\mathbf{U}) = \beta_i^{\mathrm{I}} + \boldsymbol{\alpha}_i^{\mathrm{I}} \mathbf{U}^T \ (i = 1, 2, \ldots, 4) \tag{70}$$

https://doi.org/10.1017/S0890060419000155 Published online by Cambridge University Press

Table 9. Results from different methods

	Proposed method	Independent assumption method	True value
p_{f_s}	0.0142	0.0238	0.0141
Error (%)	0.47	68.15	–

in which

$$\beta_1^{\mathrm{I}} = 2.5099, \boldsymbol{\alpha}_1^{\mathrm{I}} = (-0.91, 0, 0, 0.16, 0.38, 0, 0, 0, 0, 0)$$
$$\beta_2^{\mathrm{I}} = 2.6609, \boldsymbol{\alpha}_2^{\mathrm{I}} = (-0.60, 1.4 \times 10^{-3}, -0.02, 0, 0, 0.14, 0.79, 0, 0, 0)$$
$$\beta_3^{\mathrm{I}} = 2.5653, \boldsymbol{\alpha}_3^{\mathrm{I}} = (-0.99, 2.6 \times 10^{-4}, 1.6 \times 10^{-2}, 0, 0, 0, 0, 0.14, 0, 0)$$
$$\beta_4^{\mathrm{I}} = 2.4145, \boldsymbol{\alpha}_4^{\mathrm{I}} = (-0.99, 2.6 \times 10^{-4}, 1.5 \times 10^{-2}, 0, 0, 0, 0, 0.10, 0, 0) \quad (71)$$

For the type-II component, the limit-state function is reconstructed by the proposed one-class SVM method and is given by

$$Z_5^{\mathrm{II}}(\mathbf{U}) = \beta_5^{\mathrm{II}} + \boldsymbol{\alpha}_5^{\mathrm{II}} \mathbf{U}^T \quad (72)$$

in which $\beta_5^{\mathrm{II}} = -\Phi^{-1}(p_{f5}) = 3.0785$ and $\boldsymbol{\alpha}_5^{\mathrm{II}} = (-0.109, 0.066, 0, 0, 0, 0, 0, 0.068, 0.989)$.

Then $Z_i^{\mathrm{I}}(\mathbf{U})(i = 1, 2, \ldots, 4)$ and $Z_5^{\mathrm{II}}(\mathbf{U})$ follow a multivariate normal distribution with the mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ given by

$$\boldsymbol{\mu} = (-2.5099, -2.6099, -2.5653, -2.4145, -3.0785) \quad (73)$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.546 & 0.903 & 0.907 & 0.099 \\ 0.546 & 1 & 0.593 & 0.595 & 0.065 \\ 0.903 & 0.593 & 1 & 0.999 & 0.107 \\ 0.907 & 0.595 & 0.999 & 1 & 0.108 \\ 0.099 & 0.065 & 0.107 & 0.108 & 1 \end{bmatrix} \quad (74)$$

Thus the probability of system failure is calculate by $p_{f_s} = 1\text{-}R_s = 0.0133$.

Based on the known limit-state functions $g_i^{\mathrm{I}}(\mathbf{X})(i = 1, 2, 3, 4)$ and $g_5^{\mathrm{II(true)}}(\mathbf{X})$, the true system reliability can be directly obtained by FORM. The results from different methods are shown in Table 9, which demonstrate that the proposed method is close to the true value and outperforms the independence assumption method.

Conclusions

Motivated by the need for creating component models from one-class failure data in system reliability prediction, this study develops a new one-class SVM method for data set that is on the one side of a hyperplane, which is tangent to a hypersphere with a known radius. Different from traditional SVM methods, the new method creates a linear model using both the given data set and the radius; in other words, only the orientation of the hyperplane is determined.

The advantages of the proposed method for system reliability prediction are multifold. At first, it reveals the relationship between component states (safe or failed) with factors that affect the state, such as component dimensions, loading, and environment. Second, the method makes it possible to account for the dependence between component states through the created models. Third, the method obtains a complete probability density function of all the component states. Fourth, the method provides a feasible way to integrate physics- and statistics-based reliability methods. As a result, an accurate system reliability prediction can be produced.

There are several assumptions for the application of the proposed method, such as the distributions of basic random variables are known, the reliability resulting from the FORM is accurate, and no stochastic processes are involved. In our future study, we will extend the method to time-dependent problems where the data set varies with respect to time.

Acknowledgments. This material is based upon work supported by the National Science Foundation under Grant No. CMMI 1562593.

References

Akbani R, Kwek S and Japkowicz N (2004) Applying support vector machines to imbalanced datasets. *Machine Learning: ECML* **2004**, 39–50.

Boser BE, Guyon IM and Vapnik VN (1992) A training algorithm for optimal margin classifiers. *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152. ACM.

Chen K-Y, Chen L-S, Chen M-C and Lee C-L (2011) Using SVM based method for equipment fault detection in a thermal power plant. *Computers in Industry* **62**, 42–50.

Cheng Y and Du X (2016) System reliability analysis with dependent component failures during early design stage—a feasibility study. *Journal of Mechanical Design* **138**, 051405.

Chiralaksanakul A and Mahadevan S (2005) First-order approximation methods in reliability-based design optimization. *Journal of Mechanical Design* **127**, 851–857.

Click RL and Duening TN (2004) *Business Process Outsourcing: The Competitive Advantage*. Hoboken, NJ: John Wiley & Sons.

Cortez P (2010) Data mining with neural networks and support vector machines using the R/rminer tool. In *Advances in Data Mining. Applications and Theoretical Aspects*, Part of the Lecture Notes in Computer Science book series (LNCS, volume 6171), pp. 572–583.

Cruse TA (1997) *Reliability-Based Mechanical Design*. Boca Raton, FL: CRC Press.

das Chagas Moura M, Zio E, Lins ID and Droguett E (2011) Failure and reliability prediction by support vector machines regression of time series data. *Reliability Engineering & System Safety* **96**, 1527–1534.

Dreiseitl S, Osl M, Scheibböck C and Binder M (2010) Outlier detection with one-class SVMs: An application to melanoma prognosis. *AMIA Annual Symposium Proceedings*, Vol. 2010, p. 172. American Medical Informatics Association.

Du X and Sudjianto A (2004) First order saddlepoint approximation for reliability analysis. *AIAA journal* **42**, 1199–1207.

Frias-Martinez E, Sanchez A and Velez J (2006) Support vector machines versus multi-layer perceptrons for efficient off-line signature recognition. *Engineering Applications of Artificial Intelligence* **19**, 693–704.

Green RC, Wang L, Alam M and Singh C (2013) Intelligent state space pruning for Monte Carlo simulation with applications in composite power system reliability. *Engineering Applications of Artificial Intelligence* **26**, 1707–1724.

Gryllias KC and Antoniadis IA (2012) A support vector machine approach based on physical model training for rolling element bearing fault detection in industrial environments. *Engineering Applications of Artificial Intelligence* **25**, 326–344.

Hanna S (2007) Inductive machine learning of optimal modular structures: estimating solutions using support vector machines. *AI EDAM* **21**, 351–366.

Hoyland A and Rausand M (2004) *System Reliability Theory: Models, Statistical Methods, and Applications*. Hoboken, NJ: Wiley-Interscience.

Hu Z and Du X (2016) A physics-based reliability method for components adopted in new series systems. *2016 Annual Reliability and Maintainability Symposium (RAMS)*, pp. 1–7. IEEE.

https://doi.org/10.1017/S0890060419000155 Published online by Cambridge University Press

Hu Z and Du X (2017*a*) System reliability analysis with in-house and outsourced components. *System Reliability and Safety (ICSRS), 2017 2nd International Conference on*, pp. 146–150. IEEE.

Hu Z and Du X (2017*b*) System reliability prediction with shared load and unknown component design details. *AI EDAM* **31**, 223–234.

Hu Z and Du X (2018*a*) Integration of statistics- and physics-based methods – A feasibility study on accurate system reliability prediction. *Journal of Mechanical Design* **140**, 074501.

Hu Z and Du X (2018*b*) A partial safety factor method for system reliability prediction with outsourced components. *ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. V02BT03A050–V002BT003A050. American Society of Mechanical Engineers.

Hu Z and Du X (2018*c*) Saddlepoint approximation reliability method for quadratic functions in normal variables. *Structural Safety* **71**, 24–32.

Hu Z and Du X (2019) An exploratory study for predicting component reliability with new load conditions. *Frontiers of Mechanical Engineering* **14**, 76–84.

Hu Z, Nannapaneni S and Mahadevan S (2017) Efficient kriging surrogate modeling approach for system reliability analysis. *AI EDAM* **31**, 143–160.

Lawless J (1983) Statistical methods in reliability. *Technometrics* **25**, 305–316.

Li H-s, Lü Z-z and Yue Z-f (2006) Support vector machine for structural reliability analysis. *Applied Mathematics and Mechanics* **27**, 1295–1303.

Ma J and Perkins S (2003). Time-series novelty detection using one-class support vector machines. *Neural Networks*, 2003. *Proceedings of the International Joint Conference on*, Vol. **3**, pp. 1741–1745. IEEE.

Mahadevan S (1997) Physics-based reliability models. In Cruse TA (ed.), *Reliability-based Mechanical Design*. Boca Raton, FL: CRC Press, pp. 197–232.

Mahadevan S and Shah SL (2009) Fault detection and diagnosis in process data using one-class support vector machines. *Journal of Process Control* **19**, 1627–1639.

Manevitz LM and Yousef M (2001) One-class SVMs for document classification. *Journal of Machine Learning Research* **2**, 139–154.

Meeker WQ and Escobar LA (2014). *Statistical Methods for Reliability Data*. Hoboken, NJ: John Wiley & Sons.

Peng X (2011) TPMSVM: a novel twin parametric-margin support vector machine for pattern recognition. *Pattern Recognition* **44**, 2678–2692.

Rosenblatt M (1952) Remarks on a multivariate transformation. *The Annals of Mathematical Statistics* **23**, 470–472.

Schölkopf B, Platt JC, Shawe-Taylor J, Smola AJ and Williamson RC (2001) Estimating the support of a high-dimensional distribution. *Neural Computation* **13**, 1443–1471.

Tian Y, Shi Y and Liu X (2012) Recent advances on support vector machines research. *Technological and Economic Development of Economy* **18**, 5–33.

Truong TX and Kim J-M (2012) Fire flame detection in video sequences using multi-stage pattern recognition techniques. *Engineering Applications of Artificial Intelligence* **25**, 1365–1372.

Vapnik V (2013) *The Nature of Statistical Learning Theory*. New York: Springer Science & Business Media.

Vapnik VN and Vapnik V (1998). *Statistical Learning Theory*. New York: Wiley.

Wang H-Q, Cai Y-N, Fu G-Y, Wu M and Wei Z-H (2018) Data-driven fault prediction and anomaly measurement for complex systems using support vector probability density estimation. *Engineering Applications of Artificial Intelligence* **67**, 1–13.

Yong Cang Z (1993) High-order reliability bounds for series systems and application to structural systems. *Computers and Structures* **46**, 381–386.

Zhao YG and Ono T (1999) A general procedure for first/second-order reliability method (FORM/SORM). *Structural Safety* **21**, 95–112.

Zhengwei Hu received his PhD from the Department of Mechanical Engineering at the Missouri University of Science and Technology. His research interests include system reliability analysis, statistical modeling, design for reliability, machine learning, and optimization.

Zhangli Hu is currently pursuing his PhD in the Department of Mechanical Engineering at the Missouri University of Science and Technology. His research interests include time-dependent reliability analysis, reliability-based optimization, robust design, and uncertainty quantification.

Dr Xiaoping Du is Professor in the Department of Mechanical and Energy Engineering at Indiana University – Purdue University Indianapolis (IUPUI). Before joining IUPUI, he was Curator's Distinguished Teaching Professor at the Missouri University of Science and Technology. In addition to his academic jobs, he assumed mechanical engineer positions at two companies. He has served as PI for seven grants from the National Science Foundation. He has authored over 100 journal papers. He is Fellow of ASME and currently Associate Editor of *Journal of Mechanical Design* and Editor of *Structural and Multidisciplinary Optimization*.

https://doi.org/10.1017/S0890060419000155 Published online by Cambridge University Press