Data Mining Methods in Structural Reliability Estimation

I-Tung Yang[1], Willy Husada[2], and Tri Joko Wahyu Adi[3]

1) Professor, Department of Civil and Construction Engineering, National Taiwan University of Science and Technology, Taipei, TAIWAN; President, Taiwan Construction Research Institute, TAIWAN. Email: ityang@mail.ntust.edu.tw
2) Graduate Student, National Taiwan University of Science and Technology, Institut Teknologi Sepuluh Nopember (ITS), Surabaya, INDONESIA. Email: willyhusada_srw@hotmail.com
3) Institut Teknologi Sepuluh Nopember (ITS) Surabaya, INDONESIA. Email: trijoko_w@yahoo.com

Abstract:

The goal of reliability-based design optimization (RBDO) is to find the optimal structural design with minimum cost subject to the required level of reliability, which is defined as one minus the target failure probability. Since the target failure probability is usually small, it takes long computation time to perform Monte Carlo simulation for accurate estimation. In the literature, surrogate models have been created to replace the time-consuming reliability analysis. The process of RBDO would include three parts: perform sampling in the solution space, build a surrogate model based on the concept of data mining, and use the surrogate model to perform the reliability analysis. In this empirical study, we use commonly used data mining methods, namely Classification and Regression Tree (CART), Artificial Neural Network (ANN) and Support Vector Machine (SVM), to create the surrogate models. The aim is to examine the performance of these data mining method in predicting the failure probability when new designs are found during the process of optimization. The present study addresses two cases of RBDO where reliability is treated as the constraint or treated as an additional objective function. In the former case, the data mining methods are used to identify whether a design leads to failure. In the latter case, the data mining methods are applied to estimate the probability of failure. Because the probability of failure may be expressed in different forms (exact value, logarithmic scale, or safety index), we also investigate whether these forms would influence the estimation accuracy. This paper does not report optimization results because our focus is not on optimization but on the prediction of reliability, which is of essence in the RBDO process.

Keywords: Reliability, Structural Design, Data Mining, Soft Computing, Uncertainty Modeling

1. INTRODUCTION

Design quality is an important part in the structural construction project. A structural design should lead to a structure that is reliable enough subjected to uncertain conditions such as variability from construction process, material properties and external loads. Design optimization is used to improve the design quality so that the actual structure can have adequate safety with minimum cost. One of the most popular design optimization methods is reliability-based design optimization (RBDO). RBDO includes two processes, design optimization and reliability analysis which aim to find the optimal design with minimum structure cost or weight subjected to maximum failure probability limit. In practical, RBDO involves highly non-linear limit state functions and non-normally distributes distributed random variables. These issues create significant challenges for accurate reliability analysis (Deb 2001).

There are three integration frameworks of RBDO: double-loop, single-loop and decoupled. The double-loop method requires a full reliability analysis at every step of the design optimization process and too computationally expensive for practical application (Yang & Gu 2004). In single-loop method, a surrogate model is created to replace the time-consuming reliability analysis (Steenackers et al. 2011). Despite the enhanced efficiency, the single-loop method may be inaccurate in estimating the structure failure probability because the surrogate model is associated with certain errors. Decoupled method divides double-loop method into sequential cycles and then improve the reliability by formulating a new optimization constraint in the next cycle for violated reliability constraints (Liao & Ha 2008). Yet, no proof of convergence exists for the decoupled methods. Moreover, the decoupled methods rely heavily on nonlinear constrained optimizers, which are intrinsically limited by the lack of gradient information (Yang & Hsieh 2013).

To improve the accuracy of the single-loop RBDO method, a better surrogate model is in need. Popular data mining methods can be used to improve the surrogate model. This study attempts to implement and compare different data mining methods in predicting the probability of failure of structural designs. The prediction is divided into two parts: binary classification and regression. Binary classification deals with single-objective optimization with focus on minimization of cost whereas regression handles multiple-objective optimization by simultaneously minimizing cost and failure probability. The surrogate models are examined and compared through an empirical benchmark case study, a ten-bar truss, to demonstrate their prediction accuracy and computation time.

2. METHODS

At first, we conduct a preliminary experiment to select the best three data mining methods from popular methods in the market. The advantage of the data mining methods under consideration is that they are inherently non-parametric. In other words, no assumptions are made regarding the underlying distribution of values of the predictor variables. There are 7 (seven) methods for classification and 4 (four) methods for regression as shown in Table 1. Among them, we select the top three because they yield the highest prediction accuracy. Because of space limitation, the present paper only provides a brief introduction of the methods in the following sections.

Table 1. Classification and regression methods

Classification methods	Regression methods
● CART = Classification and Regression Tree	● CART = Classification and Regression Tree
● ANN = Artificial Neural Networl	● ANN = Artificial Neural Networl
● SVM = Support Vector Machine	● SVM = Support Vector Machine
● CHAID = CHi-Squared Automatic Interaction Detection	● LR = Linear Regression
● BAYES = Bayesian Network	
● QUEST = Quick, Unbiased and Efficient Statistical Tree	
● LOG = Logistic Regression	

2.1 CART

The classification and regression tree (CART) is a decision tree method by constructing a classification tree or regression tree according to its output variable data type, which can be either categorical/class or numerical (Breiman et al., 1984). The construction of the surrogate model using CART is extremely fast because it can provide a clear indication of which inputs are more important for making the prediction easier.

The CART method is creating a surrogate model by constructing a decision tree. In the beginning, the decision tree is started with a single parent node and then this parent node will split into two partition nodes. To split a parent node, we need to select a tree-split that gives the smallest impurity among possible splits. Impurity in the CART method is the measurement of how many records in a data set would be incorrectly classified associated with the chosen tree-split.

Depending on the target field, two impurity measurements can be used to select a tree-split for the CART model: Entropy and Gini Index. The entropy at a given node t is defined in Eq. 1:

$$Entropy(t) = -\sum_{j} p(j \mid t)\, log\, p(j|t) \tag{1}$$

where p(*j*|*t*) = relative frequency of class *j* at node *t*

After obtaining the entropy, the CART model will split the decision tree based on information gain in terms of entropy with the following Eq. 2:

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i)\right) \tag{2}$$

where $GAIN_{split}$ = reduction measurement in entropy gained because of the split
p = parent node
n = total records in parent node p
k = partitions of records in parent node p
n_i = number of records in partition node i

The Gini Index g(t) at a node t is calculated using Eq. 3:

$$g(t) = 1 - \sum_{i=1}^{j} {p_i}^2 \tag{3}$$

where $g(t)$ = Gini Index
j = number of different classes occurred after tree-splitting

p = probability of node t belonging to class i

After obtaining the Gini Index, the CART model will split the decision tree based on the GINI Gain in terms of Gini Index with the following Eq. 4:

$$GINI_{GAIN} = g(t) - \sum_{i=1}^{n} \frac{|t_i|}{|t|} GINI(t_i) \quad (4)$$

where t_i = partition of node t induced by the value of class i

Because *GAIN* and *GINI* measures the reduction in entropy achieved after the splitting of the decision tree, the tree-split with maximum *GAIN* (or *GINI*) value should be chosen as it gives minimum impurity to the surrogate model.

2.2 ANN

Artificial Neural Network (ANN) is a computational method that is based on the neuron cell structure of the biological nervous system (Adeli and Hung, 1995). Given a training set of data, the neural network can learn the pattern of the data set with a learning algorithm. Through some propagation, the neural network forms a mapping between inputs and desired outputs from the training set by altering weighted connections within the network.

An ANN has one input layer, one output layer and one or more hidden layers with a certain number of neurons. Each neuron is associated with a weight and bias. Fig. 1 offers an example.

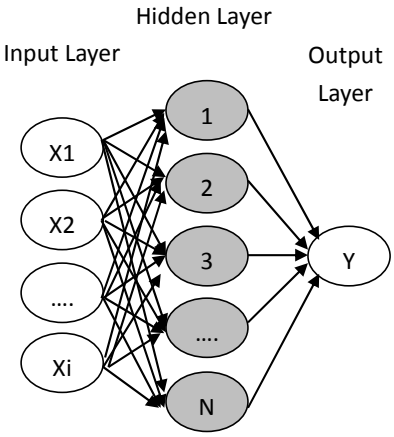

Figure 1. Artificial Neural Network

In this study, we adopt a feedforward backpropagation neural network that has input-to-unit and unit-to-unit connection modified by a weight. In addition, each unit has an extra input that is assumed to have a constant value of one. The weight that modifies this extra input is called the bias. In the feedforward phase, all of the information from the input layer is fed to the network in forward direction from the first hidden layer to the output layer. This phase will activate the activation functions in the output layer. In the backpropagation phase, the activated output layer will propagate backward the error or difference between the predicted output and the actual output through the network. Then, the weight in every connection is adjusted by the error proportions that were propagated backward, thus improving the model (Hagan et al., 2002).

2.3 SVM

Support vector machine (SVM) is a supervised learning algorithm that has an idea to build a hyperplane of classification or a decision function based on the training samples. Once the decision function is available, one can classify future data points or predict outcome (regression) with ease. The training data and the corresponding class (label) are expressed in Eq. 5:

$$D = \{(X_i, y_i) | X_i \in R^p, y_i \in \{-1,1\}\} \; ; \; i = 1, \ldots, N \quad (5)$$

where D = training data
X_i = p-dimensional vector
y_i = label index (either -1 or 1)

In the higher-dimensional plane, we may formulate a linear function as $w \cdot X - b = 0$ where w and b represent the weight bad bias, respectively. In Fig. 2, two boundary functions ($w \cdot X - b = 1$ and $w \cdot X - b = 0 - 1$) are defined to separate the data points while points A and B are located on the boundaries.

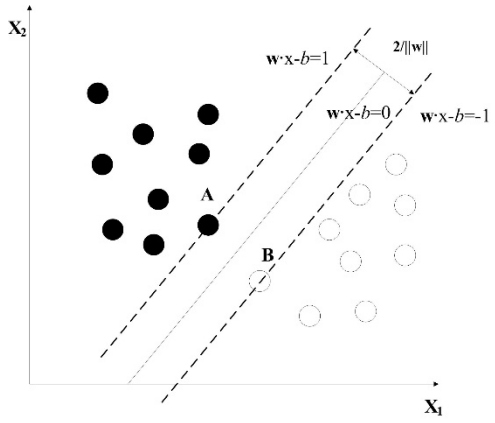

Figure 2. Support Vector Machine

To obtain the best classifier of SVM, we need to maximize the margin between two support vectors (Vapnik, 1995). This is equivalent to the minimization of w because the width of separation between two support vectors is 2/||w||. The selection of parameters w and b can be formulated as a non-linear optimization problem in Eq. 6:

$$\min_{w,b} \frac{1}{2}\|w\|^2 \quad \text{s.t.} \quad y_i(w \,.\, X_i - b) \geq 1 \; \forall_i \tag{6}$$

SVM can be extended to a non-linear problem where the data points are not linearly separable. To perform the linear separation, the original space should be transformed into a higher-dimensional space. Then, the training process for SVM requires selection of kernel functions (i.e., linear, radial basis, polynomial, or sigmoid function) to do such transformation. Default values of kernel parameters are highly dependent on their kernel type and the software implemented. The SVM model requires the user to specify two parameters, which are the regularization parameter and the kernel parameter.

2.4 Monte Carlo Simulation

To generate samples for the experiments, we employ Monte Carlo Simulation (MCS) because it is robust and flexible (being able to handle non-linear and non-smooth limit state function and non-Gaussian distributions) to perform probabilistic reliability analysis of a system (Rubinstein and Kroese, 2011). MCS starts with the Law of Large Number. According to the law, the average of the results obtained from a large number of trials in a simulation should be close to the expected value and will tend to become closer as more trials are performed to the simulation.

The failure probability (P_f) estimated by MCS is the total number of failure samples N_f (when the load exceeds the resistance of the structure) divided by the total number of samples N_{Total}:

$$P_f = \frac{N_f}{N_{Total}} \tag{7}$$

The accuracy of MCS depends on the sample size of the trials. The Coefficient of Variation (CoV) of the estimated failure probability can be used to measure the dispersion of the estimated structure reliability. The CoV of the failure probability can be expressed as follows:

$$COV(P_f) = \sqrt{\frac{1 - P_f}{N \times P_f}} \tag{8}$$

where $COV(P_f)$ = coefficient of variation of the failure probability

P_f = failure probability

N = sample size or number of simulations/trials

3. STEPS

The specific steps of the present study are as follows:

1. Adopt the ten-bar plane truss example, which will be introduced below, to be the experimental case.
2. Use MCS to conduct the reliability analysis for preparing the training data set.
3. Conduct preliminary experiments to select the best methods among popular data mining methods in the estimation of reliability.
4. Develop surrogate models based on the selected methods: CART, ANN and SVM.
5. Fine-tune the control parameters of the surrogate models.
6. Evaluate and compare the performance of the surrogate models using ten-fold cross validation.
7. Draw conclusions based on the experiment results.

4. CASE STUDY

The benchmark case used in this study is a ten-bar truss problem that was introduced in (Haftka & Gurdal 1992). The shape, geometry and loading of the ten-bar truss structure are shown in Fig. 3. As the present study has to perform computationally intensive simulations for three methods, we adopt the ten-bar truss example because of its simplicity. Note that no comparison is made to previous RBDO results because the focus of the present study is on the prediction of reliability, not on the performance of optimization techniques.

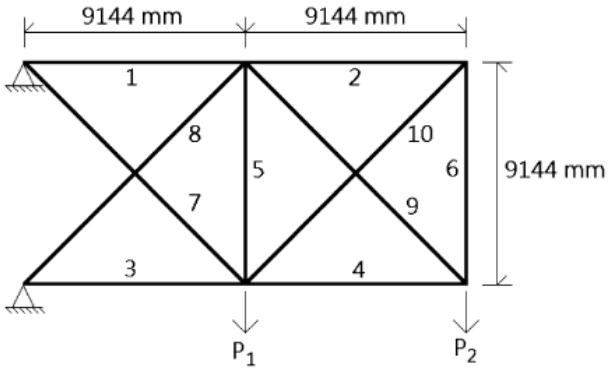

Figure 3. Ten-bar truss

The ten-bar truss is pin-jointed and subjected to two external loads, P_1 and P_2. Every bar is made of hollow carbon steel pipes and may have different sizes. In this study, the carbon steel pipes are selected from an industry standard established by Chinese National Bureau of Standards in 1995 (CNS 4435, G3102, 1995). The bar sizes have 36 options shown in Table 2. The selection of bars represents a discrete set with three features: pipe outside diameter (D), wall thickness (t) and cross-sectional area (A). The steel pipe also has the following mechanical properties: modulus of elasticity (E) of 200,000 N/mm^2 and density (ρ) of 2.768 x 10^{-6} kg/mm^3. In total, there are 36 discrete options that can be selected from the list and these options form a design space of 36^{10} discrete combinations which is more than 3.65 x 10^{15} options. This amount of possible options is considered huge even for a relatively small RBDO problem.

Table 2. Size options of bars

Option	D (mm)	t (mm)	A (mm^2)	Option	D (mm)	t (mm)	A (mm^2)
#1	216.3	7	4602.7	#19	355.6	12	12953.4
#2	216.3	8	5235.2	#20	406.4	7.9	9890.2
#3	216.3	8.2	5360.9	#21	406.4	9	11236.2
#4	267.4	6	4927.3	#22	406.4	9.5	11845.5
#5	267.4	6.6	5407.6	#23	406.4	12	14868.5
#6	267.4	7	5726.5	#24	406.4	12.7	15707.9
#7	267.4	8	6519.4	#25	406.4	16	19623.6
#8	267.4	9	7306.1	#26	457.2	9	12672.6
#9	267.4	9.3	7540.9	#27	457.2	9.5	13361.7

#10	318.5	6	5890.5	#28	457.2	12	16783.6
#11	318.5	6.9	6754.6	#29	457.2	12.7	17734.8
#12	318.5	8	7803.7	#30	457.2	16	22177.1
#13	318.5	9	8750.9	#31	508	7.9	12411.8
#14	318.5	10.31	9982.2	#32	508	9	14108.9
#15	355.6	6.4	7021.1	#33	508	9.5	14877.8
#16	355.6	7.9	8629.4	#34	508	12	18698.8
#17	355.6	9	9799.9	#35	508	12.7	19761.6
#18	355.6	9.5	10329.4	#36	508	14	21727.3

The present study considers variability and error of the random variables due to the nature of uncertainty and practical manufacturing variation as follows:

1. Randomness of external loads: P_1 and P_2
2. Manufacturing variation of cross section area: $A_{1\sim10}$
3. Randomness of steel yield stress: F_y
4. Estimation error of external loads: e_{P1} and e_{P2}
5. Estimation error of cross section areas: $e_{A1\sim10}$
6. Estimation error of steel yield stress: e_F
7. Error in the mathematical modeling of stress: e_σ

Tables 3 and 4 show the probabilistic distributions with mean value and dispersion. The loads are assumed to follow the extreme value distribution, which has widely been used in load modeling (Wen, 1990). The cross-sectional areas are uniformly distributed within a certain range caused by manufacturing imperfections. The yield stress is assumed to follow the lognormal distributions, which exclude negative values (Agrawal and Bhattacharya, 2011).

Table 3. Probability distributions of random variables

Variable	Distribution type	Mean	Dispersion
P_1, P_2	Extreme Value Type I	150 kN	10% c.o.v.
$A_{1\sim10}$	Uniform	Ā	±4%
F_y	Lognormal	517.28 N/mm^2 (9th Bar), 172.43 N/mm^2 (Others)	8% c.o.v.

Table 4. Probability distributions of error

Error	Distribution type	Mean	Dispersion
e_{P1}, e_{P2}	Uniform	0	±10%
$e_{A1\sim10}$	Uniform	0	±3%
e_F	Uniform	0	±20%
e_σ	Uniform	0	±5%

Two structural failure modes are considered in this case: yield stress and buckling stress. The truss is considered failed when any of the members fails either in terms of yield stress or buckling stress. For yield stress, a failure occurs when the axial stress exceeds the yield strength of the bar. Buckling stress occurs in a truss when the bars subjected to high compressive stress, which causes the bar to lose stiffness and strength capacity. The critical buckling stress is computed according to AISC-LRFD rules (Salmon et al., 2008).

5. EXPERIMENT RESULTS

5.1 Performance criteria

To make comparisons, we define the following performance criteria for prediction accuracy. For classification, the prediction accuracy is calculated by using a confusion matrix. A confusion matrix is a specific table layout that allows visualization of the classification model performance. Each column of the matrix represents the predicted output class while each row of the matrix represents the actual output class. The idea of the confusion matrix is to separate and count the total number of actual output class correspond to the predicted output class. The prediction accuracy of the classification model is the number of correct predictions divided by the number of total samples. All correct predictions by the classification model are located in the diagonal cells of the confusion matrix. Therefore, the prediction error is represented by non-diagonal elements.

For regression, the prediction accuracy is measured using four performance indicators: Linear Correlation Coefficient (R), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Root Mean

Squared Error (RMSE). To provide a comprehensive performance measurement, the four performance indicator value should be combined into one unified index. The performance indicators (1-R, MAE, MAPE and RMSE) are combined into the synthesis index (Chou & Pham, 2013):

$$SI = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{P_i - P_{min,i}}{P_{max,i} - P_{min,i}}\right) \tag{9}$$

where m is the number of performance measures, for this study $m = 4$. P_i is the i-th performance measure, $P_{min,i}$ is the minimum value of i-th performance measure while $P_{max,i}$ is the maximum value of i-th performance measure.

The regression models are compared with each other under the same baseline by normalizing the prediction performance. SI ranges between 0 and 1. When the SI value gets close to 0, the associated regression model has the highest accuracy among the other models.

Ten-fold cross validation is used to minimize bias associated with random feature of training data set sampling process. In the ten-fold cross validation method, a fixed number of data samples from a data set are divided into ten folds. Among these ten folds, nine folds will served as training data set to build surrogate model, while the rest one fold will served as testing data set to verify and validate the accuracy of surrogate model. Then, the accuracy of the surrogate model can be expressed as the average accuracy acquired by the ten rounds of validation process. To reduce variability, every round from ten-fold cross validation process is performed using different partitions.

5.2 Setups

In the preliminary study, we perform grid search to find the most suitable setups for ANN and SVM, in terms of the highest accuracy. The selected ANN model has 3 hidden layers, each of which has 5 neurons with the training functions being the Scale Conjugate Gradient method for classification and the Levenberg-Marquardt algorithm for regression. The transfer function is Log-Sigmoid. For SVM, we choose Least Square Support Vector Machine (LSSVM) with the Gaussian Radial Basis Function being the kernel function.

5.3 Comparisons

Table 5 compares the classification accuracy of the three methods: CART, ANN, and SVM. All the methods can achieve more than 90% accuracy. The performance of ANN is the best while the difference between ANN and CART is moderate. Both ANN and CART are significantly better than SVM. However, ANN and SVM require a lot more computational time (7.96 and 14.31 minutes, respectively) than CART (0.33 minute). Most of the computation time is spent in parameter tuning.

Table 5. Comparison in Classification

Methods	Classification Accuracy	
	Mean	Standard Deviation
CART	94.57%	1.44%
ANN	95.83%	1.64%
SVM	92.67%	1.05%

Table 6 compares the performance of the three methods in regression. Overall, there is still room for improvement for all the three methods. ANN surpasses the other two in every criterion. Again, CART spends much shorter time (only 20 seconds), compared with ANN (5.03 minutes) and SVM (13.25 minutes).

Table 6. Comparison in Regression

Methods	Regression Performance					
	R	MAE	MAPE	RMSE	SI	Rank
CART	0.8465	0.0022	36.42%	0.0082	0.723	2
ANN	0.9253	0.0020	32.33%	0.0060	0.000	1
SVM	0.8746	0.0025	41.05%	0.0078	0.867	3

6. CONCLUSIONS

In the present study, we investigate the applications of three popular data mining methods (CART, ANN, and SVM) in building surrogate models in the single-loop RBDO analysis. The three methods serve two purposes: classification and regression. The former is to classify whether a structure design would lead to failure. The latter is to estimate the exact probability of failure.

A benchmark case, ten-bar truss, is used to compare the three methods. Comparison results show that ANN performs better than the other two in both classification and regression. Nevertheless, CART is much more efficient than the other two. To sum up, ANN would be the best choice when selecting the most accurate surrogate model (both classification and regression) for the RBDO analysis. In contrast, CART is advantageous when computation time is limited. The conclusions drawn above are valuable because the suggested surrogate models can predict the failure probabilities with higher accuracy. As a result, the designer can be more confident that the optimal design truly satisfies the reliability constraint.

In current design codes, partial safety factors have been applied to either the loads or the strengths to account for their respective uncertainties. The safety factors, however, do not separate the impacts of various uncertainties (e.g., manufacturing variability, material inconsistence, and measurement error). In the present environment of design codes, the RBDO analysis may serve as a complementary tool as it can address the impacts of different levels of uncertainties on the optimal design.

REFERENCES

Adeli, H., and Hung, S. (1995). *Machine Learning: Neural Networks. Genetic Algorithms And Fuzzy Systems*, John Wiley & Sons, Inc.

Agrawal, G., and Bhattacharya, B. (2011). "Effect of relative failure consequences in reliability based dual performance design." *Canadian Journal of Civil Engineering*, 10.1139/l11-081, 1216-1226.

Breiman, L., Friedman, J., Stone, C.J., and Olshen, R.A. (1984). *Classification and regression trees*, CRC press.

Chou, J.S., and Pham, A.D. (2013). Enhanced artificial intelligence for ensemble approach to predicting high performance concrete compressive strength. *Construction and Building Materials*, 49, pp. 554-563.

CNS 4435, G3102 (1995). Carbon steel pipes for general structures. Chinese National Bureau of Standards, Ministry of Economic Affairs, Taiwan, http://www.cnsonline.com.tw/en/.

Deb, K. (2001). Multi-objective optimization using evolutionary algorithms. New York: Wiley.

Haftka, R. T., and Gürdal, Z. (1992). *Elements of structural optimization*: Springer Science & Business Media.

Hagan, M.T., Demuth, H.B., and Beale, M.H. (2002). *Neural network design*: University of Colorado Bookstore.

Liao, K.W. and Ha, C. (2008). "Application of reliability-based optimization to earth-moving machine: hydraulic cylinder components design process," *Structural and Multidisciplinary Optimization*, vol. 36, pp.523–536.

Rubinstein, R.Y., and Kroese, D.P. (2011). *Simulation and the Monte Carlo method* (Vol. 707): John Wiley and Sons.

Salmon, C.G., Johnson, J. E., & Malhas, F.A. (2008). *Steel structures: design and behavior*, 5th edition: Prentice Hall, New Jersey.

Steenackers, D., Versluys, R., Runacres, M., and Guillaume, P. (2011). "Reliability-based design optimization of computation-intensive models making use of response surface models," *Quality and Reliability Engineering International*, vol. 27, no.4, pp.555–568.

Vapnik, V. (1995). *The nature of statistical learning theory*: Springer Science & Business Media.

Wen, Y.K. (1990). *Structural Modeling and Combination for Performance and Safety Evaluation*, Elsevier, New York, NY.

Yang, I.T., and Hsieh, Y.H. (2013). "Reliability-based design optimization with cooperation between support vector machine and particle swarm optimization." *Engineering with Computers*, 29(2), pp. 151-163.

Yang, R., and Gu, L. (2004). "Experience with approximate reliability-based optimization methods," *Structural and Multidisciplinary Optimization*, vol. 26, pp.152–159.