12th International Conference on Applications of Statistics and Probability in Civil Engineering, ICASP12
Vancouver, Canada, July 12-15, 2015

Network Reliability Analysis for Cluster Connectivity Using AdaBoost

Raphael E. Stern
Graduate Student, Dept. of Civil and Environmental Engineering, University of Illinois at Urbana Champaign, Urbana, IL, USA

Junho Song
Associate Professor, Dept. of Civil Engineering, Seoul National University, Seoul, Korea

Daniel B. Work
Assistant Professor, Dept. of Civil and Environmental Engineering and Coordinated Science Laboratory, University of Illinois at Urbana Champaign, Urbana, IL, USA

ABSTRACT: In the aftermath of a natural disaster, knowledge of the connectivity of different regions of infrastructure networks is crucial to aid decision makers. For large-scale networks it can be extremely time-consuming to obtain a converged estimate by performing a large number of Monte Carlo simulations to compute the network failure probability. To reduce computational requirements, this work develops a surrogate model using an AdaBoost classifier for predicting probabilities of disconnections between node clusters in lifeline infrastructure networks. The proposed approach uses spectral clustering to partition the network, and it estimates the connectivity of these clusters using an AdaBoost classifier. Numerical experiments on a California gas distribution network demonstrate that using the surrogate model to determine cluster connectivity introduces less than five percent error and is two orders of magnitude faster than methods using an exact network model to estimate the probability of network failure through Monte Carlo simulations.

1. INTRODUCTION

Following a natural disaster, fast response is critical to minimize the loss of life and damage to infrastructure. In lifeline networks such as gas distribution networks, power grids, and water pipelines, the ability to rapidly assess the connectivity of components within the network is vital for prompt disaster relief (Bruneau et al., 2003; Boin and McConnell, 2007). This assessment requires methods to quickly and accurately estimate the probability of network failure given the individual component failure probabilities conditioned on the event.

A variety of *system reliability analysis* (SRA) methods have been introduced to identify the probability that a network will remain functional in the aftermath of an event. One set of approaches, for example those discussed by Rausand and Høyland (2004), aim to exactly compute the network failure probability from individual component failure probabilities. Lim and Song (2012) introduced a method that intelligently enumerates all possible failure combinations by preferentially identifying disjoint cut sets and link sets to calculate the probability of network disconnection, while Reed et al. (2009) and Vugrin et al. (2010) proposed to analyze infrastructure resilience to natural disasters more generally.

While these methods are useful for small networks when precise failure probabilities are known in advance, they have several limitations for application to large, infrastructure-sized networks. As the number of nodes in the network increases, exploring all possible node failure combinations that lead to network failure can quickly become computationally intractable. Furthermore, before the earthquake event, one can compute the failure prob-

abilities of components for each possible earthquake scenario, or compute the aggregate failure probability for all possible earthquakes represented by a probabilistic seismic hazard model. Both of these approaches have inherent uncertainty since they require conditioning on a specific earthquake event. Knowledge of the precise component failure probability for a specific event are not known until after the event occurs, which means that for results with the least uncertainty the analysis may need to be performed in the immediate aftermath of a disaster, when time is critical.

Because of the above limitations, there is recent interest to develop approximate methods to compute network failure probabilities, or bounds on such probabilities (Der Kiureghian and Song, 2008; Song and Der Kiureghian, 2003). For example, *Monte Carlo simulations* (MCS) are frequently used to estimate the probability of network failure in SRA (Papadrakakis et al., 1996; Ditlevsen and Madsen, 1996). However, the Monte Carlo techniques still require one to determine the connectivity of nodes in a network for each realization of the network, which can be very time consuming if reliable estimates are desired.

To further speed up the calculation time, Stern et al. (2014) proposed the idea of using a *surrogate model* to determine the connectivity of two node pairs, instead of testing the connectivity on the actual infrastructure network. The surrogate model is a simplified model that approximates the actual network model of interest and on which calculations can be performed much more quickly. Surrogate models are commonly used to simplify computations in fields as diverse as structural optimization, waste water modeling, and supply chain management (Jansson et al., 2003; Meirlaen et al., 2001; Wan et al., 2005).

Stern et al. (2014) used MCS to approximate the connectivity of a single source and terminal node, using an AdaBoost (Freund and Schapire, 1995) classifier as the surrogate model. This model is trained on a large labeled dataset of network states in an offline pre-processing stage, and the samples are chosen independently of any specific node failure probabilities. As a result, the trained model can be used in the immediate aftermath of an event when the node failure probabilities are realized, by performing an MCS on the surrogate model. The results in Stern et al. (2014) indicate the approximate model can estimate the probability of disconnection of a source and terminal node to within 3% under worst-case conditions, and is six times faster than using a shortest path method to determine network failure in the MCS.

Due to the hierarchical structure and regional divisions of emergency management authorities, decisions regarding disaster response often are made on a regional basis. Therefore, for large-scale risk assessment on infrastructure networks it is important to determine the probability of disconnection of these large regions of the network, or node clusters. In this work, we extend the AdaBoost surrogate model SRA framework to the problem of determining the probability of clusters of nodes becoming disconnected after an event.

The main contribution of this work is as follows. By decomposing the network into densely connected clusters and training a surrogate model to determine inter-cluster connectivity, we show that it is possible to quickly estimate the probability of cluster disconnection using MCS.

The remainder of the article is organized as follows. In Section 2, the AdaBoost algorithm and an efficient sampling technique to generate balanced training data are reviewed. In Section 3, the cluster connectivity surrogate model is presented. Section 4 demonstrates the application of the developed methods in a numerical example of the California gas distribution network. In Section 5 conclusions and future work are discussed.

2. BACKGROUND ON ADABOOST SURROGATE MODELS FOR SRA

In this section, the main techniques used in the source–terminal network connectivity problem (Rai and Aggarwal, 1978) are reviewed. The general methodology of constructing a surrogate model using AdaBoost and training the model on a labeled dataset generated via guided random walk sampling are discussed. These tools will also be leveraged in the cluster connectivity problem used in this work.

2.1. Estimating source–terminal disconnections

The infrastructure network is modeled as a graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, where the number of nodes $n = |\mathscr{V}|$ is the cardinality of the vertex (node) set $\mathscr{V}$, and $\mathscr{E} \subseteq \{(i,j) : i,j \in \mathscr{V}\}$ is the set of edges. A network state is defined as $x \in \{0,1\}^n$, where each element x_i encodes the state of the corresponding network component as

$$x_i = \begin{cases} 0 & \text{if node } i \text{ has failed,} \\ 1 & \text{if node } i \text{ remains intact.} \end{cases} \quad (1)$$

Similarly, the network status $y_{s,t}$ with respect to source node s and terminal node t is defined as follows:

$$y_{s,t} = \begin{cases} 1 & \text{if } s \text{ and } t \text{ are connected,} \\ -1 & \text{otherwise.} \end{cases} \quad (2)$$

Suppose, following an event E, the failure probability $\Pr(x_i = 0|E)$ of each component is known. In order to estimate network failure probability $\Pr(y_{s,t} = -1|E)$, we generate network realizations $x(k)$, where k denotes the sample index, and compute $y_{s,t}(k) = f(x(k))$ for each sample k. The operator f that determines the connectivity of the source and terminal node is a shortest path algorithm. If a finite cost path exists between s and t, then the nodes are connected.

Given K realizations of the network state, the network failure probability is approximated as:

$$\Pr(y_{s,t} = -1|E) \approx \frac{1}{K} \sum_{k=1}^{K} I(f(x(k)) = -1) \quad (3)$$

where $I(\cdot)$ is the indicator function that takes the value 1 if the event occurs, and 0 otherwise. Evaluating f for large K can be computationally expensive. To improve the computational performance, Stern et al. (2014) proposed the use of a fast but approximate surrogate model $\tilde{f}$, which approximates the value of $y_{s,t}$ as $\tilde{y}_{s,t} = \tilde{f}(x)$.

2.2. Generating data to train the surrogate model

When considering the binary status of components in the network, there exist a factorial number of possible network states (combination of failed and non-failed nodes). Therefore, it is necessary to sample a subset of these data points to train and test the surrogate model $\tilde{f}$. Furthermore, due to the inherent robustness of infrastructure networks, there is a bias toward connected networks when sampling likely network states. For best classification performance, it is important to have a dataset that is balanced between examples of failed and not failed networks (Japkowicz and Stephen, 2002).

In order to generate a balanced training dataset, data points are sampled according to a *guided random walk sampling* (GRWS) method (Stern et al., 2014). The sampling method, based on the Metropolis-Hastings algorithm (Metropolis et al., 1953), seeks to identify the boundary between network failure and non-failure cases on the basis of the number of nodes that have failed. A complete description of the guided random walk sampling technique can be found in Stern et al. (2014).

The network status $y_{s,t}$ must be determined for each data point in the training data using the exact model f, which is still computationally expensive. The key benefit is this training data needs to be generated once offline, and future queries can use the surrogate model after it has been trained.

2.3. AdaBoost classification

The surrogate model for source–terminal connectivity is constructed using the AdaBoost machine learning classifier. AdaBoost is chosen because it has been shown to learn complex structure with limited training data (Schapire et al., 1998). AdaBoost is a binary classification algorithm that combines multiple weighted *weak* classifiers that, in aggregate, produce a more sophisticated classifier. This general approach is called *boosting* in the machine learning community (Freund and Schapire, 1999). The final classifier is given as:

$$\tilde{f}(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right), \quad (4)$$

which is a linear combination of T weak classifiers h_t with weights α_t.

Each weak classifier h_t is a one dimensional classification rule. AdaBoost sequentially computes each weak classifier, indexed by t, and determines its contribution α_t to the overall classifier according

to:

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) > 0, \quad (5)$$

where ε_t is the classification error of the t^{th} weak classifier, defined subsequently.

The weak classifier h_t is determined by minimizing the classification error of the m training data points $\{(x(1),y(1)),\ldots,(x(m),y(m))\}$ as:

$$\varepsilon_t = \frac{1}{m}\sum_{k=1}^{m}[W_t(k)\cdot(h_t(x(k)) \neq y(k))]. \quad (6)$$

To compute the classification error, the data points are weighted according to the weights $\{W_t(1),\cdots,W_t(m)\}$.

For the first classifier, these weights are initialized as $W_1(k) = 1/m$. For subsequent classifiers, the new weight for data point $x(k)$ is computed from the previous data weight $W_{t-1}(k)$, the previous classifier h_{t-1} operating on $x(k)$, and the classifier weight α_{t-1} as follows:

$$W_t(k) = \frac{W_{t-1}(k)}{Z_{t-1}}\exp\left(-\alpha_{t-1}y(k)h_{t-1}\left(x(k)\right)\right), \quad (7)$$

where Z_{t-1} is a normalization constant.

The AdaBoost classifier in Stern et al. (2014) is trained on data collected using the guided random walk sampling method, and applied to classify a node pair in the network as connected or disconnected. Cross validation results (not shown) produced similar error on both the training and test data, indicate the model is not being over-fitted.

3. SURROGATE MODELS FOR CLUSTER CONNECTIVITY

In this work, we extend the AdaBoost surrogate model for use in an approximate method to estimate the probability of disconnection of two clusters of nodes. To construct the method, the infrastructure network is first partitioned into a number of densely connected clusters of nodes. Once the clusters have been determined, the connectivity of a cluster–cluster pair can be determined by generating a surrogate model for inter-cluster connectivity. Owing to the good performance on the source–terminal connectivity problem, we again use an AdaBoost classifier, where one classifier is trained for each cluster–cluster pair. The details of this procedure are described next.

3.1. Graph clustering

We assume a common structure for large infrastructure networks. Specifically, we assume there are a number of densely connected components (e.g, in an urban infrastructure grid), which are loosely connected to other densely connected components (e.g., in another urban area) via long-range links. These network structures lend themselves well to cluster connectivity analysis. Clustering of infrastructure networks has also been proposed to understand network hierarchy (Gómez et al., 2013) and to facilitate analysis of large networks on multiple scales (Lim et al., 2015).

Spectral graph clustering (SGC) (von Luxburg, 2007) is used to partition graphs into densely connected clusters. To split a graph into two clusters Laplacian L is computed as

$$L = D - A, \quad (8)$$

where D is a diagonal matrix containing the degree of each node on the diagonal, and A is the adjacency matrix of the graph. Clusters are determined by finding the second smallest eigenvalue of L and assigning node labels, q_i according to

$$q_i = \begin{cases} 0 & \text{if } v_i < 0, \\ 1 & \text{otherwise,} \end{cases} \quad (9)$$

where v_i is the i^{th} element of the eigenvector associated with the second smallest eigenvalue of L. A cluster is the set of all nodes that have the same node label q_i.

For an arbitrary number of clusters, k, the first k eigenvectors of L are clustered using *k-means* clustering and used to determine the cluster assignment (von Luxburg, 2007).

3.2. Training the surrogate model

Similar to the methodology in Stern et al. (2014), we generate training data using the guide random walk sampling method, with the distinction that connectivity is defined with respect to cluster i and j as opposed to the source and terminal nodes in (2). Since each surrogate model only describes

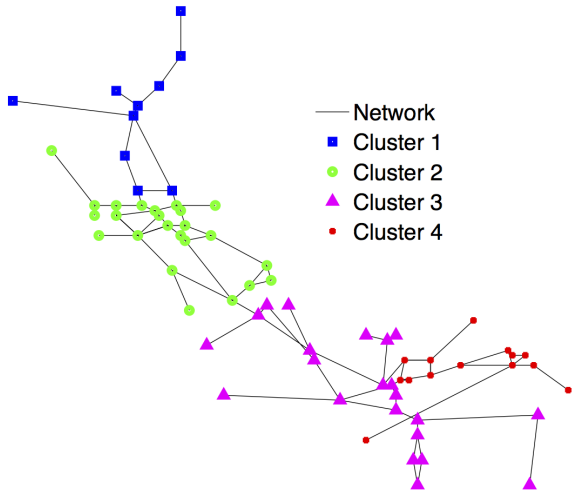


Figure 1: California gas distribution network with four clusters.

the connectivity of two specific clusters within the network, we construct $\binom{n_c}{2}$ models describing each possible cluster-cluster pair, where n_c is the total number of clusters.

Once the surrogate models are built, the model enters the application phase. When an event occurs, fragility models are used to estimate the failure probability of each network component, and network state samples are drawn from these component failure distributions. Each sample is then run through the AdaBoost classifier to estimate if the clusters in question are disconnected for the given network realization. Finally, the samples are used collectively to generate an estimate on the probability of cluster disconnection given the event.

4. NUMERICAL EXAMPLE

4.1. California gas distribution network

We use the California gas distribution network as a numerical example to test the proposed method. The network topology, obtained from Lim et al. (2015), is shown in Figure 1. The network consists of 244 components (70 nodes and 87 bi-directional edges). Each node represents a substation in the gas distribution network, and the edges represent gas pipelines.

We consider a case where all nodes in the network have the same uncorrelated failure probability. Without loss of generality, only nodes are allowed to fail in this network. To relax this assumption, one can place a node at the midpoint of each edge and assign to it the corresponding edge failure probability. We divide the network into four node clusters and consider all possible cluster combinations. A surrogate model is trained for each pair of node clusters in the network. Each model is trained using AdaBoost, and built using $m = 5,000$ training points and a maximum of $T = 200$ weak classifiers.

We evaluate the trained models by estimating the probability that the two clusters become disconnected using the corresponding AdaBoost classifier. All nodes are assumed to have a uniform failure probability $p_f = 0.15$, since it was found to yield the results with the highest error in Stern et al. (2014). A uniform component failure probability is assumed for simplicity, but this can be generalized to non-uniform component failure without modifying the methodology, though performance may vary. The inter-cluster disconnection probability is estimated via MCS with $K = 5,000$ samples. We compare this to the estimate computed using Dijkstra's shortest path algorithm to check connectivity between clusters (Dijkstra, 1956).

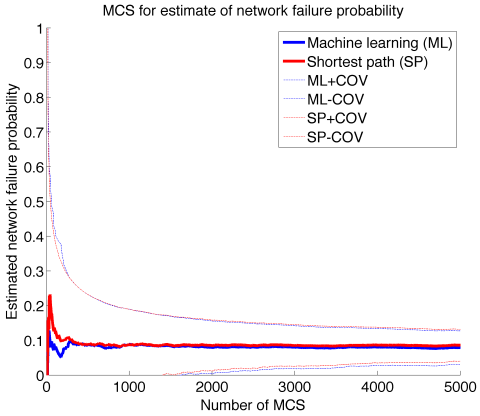
In addition to the failure probability estimate, we also compute the *coefficient of variation* (COV) on the true probability of network failure based on the estimate, $\delta_{\hat{p}_f}$, computed by:

$$\delta_{\hat{p}_f} = \frac{s}{\sqrt{K}\hat{p}_f} \tag{10}$$

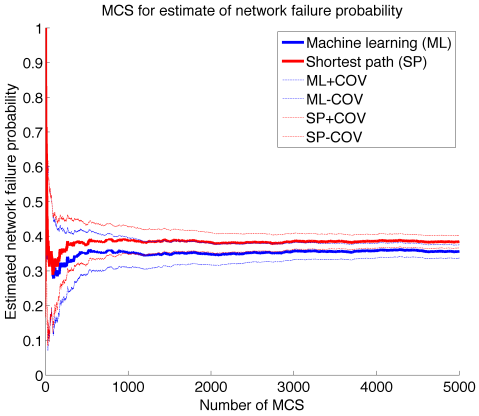
where K is the total number of MCS, s is the sample standard deviation of network failure probability estimates, and $\hat{p}_f$ is the estimated network failure probability. The time required to estimate the network failure probability using the AdaBoost classifier is recorded and compared with the time to estimate the same probability using Dijkstra's shortest path method. The algorithms are implemented in Matlab on a quad core 2.6 GHz MacBook Pro, and are available online at Stern (2015).

4.2. Results

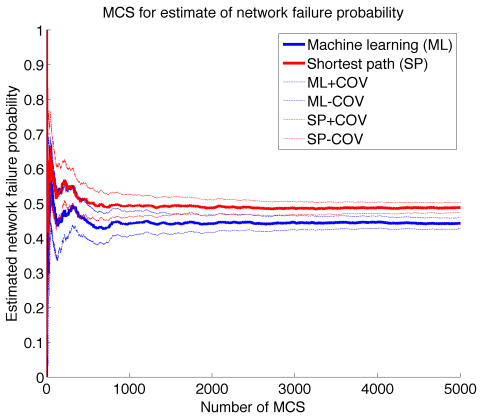
The node clusters for the California gas distribution network are shown in Figure 1. This choice of clusters divides the network into four densely connected regions of approximately equal size.



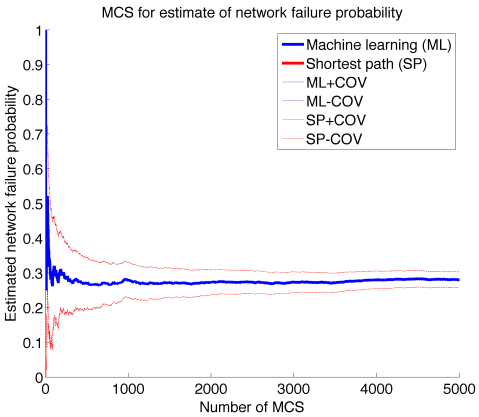
(a) Disconnection probability estimate of clusters 1 and 2 convergence, and COV.



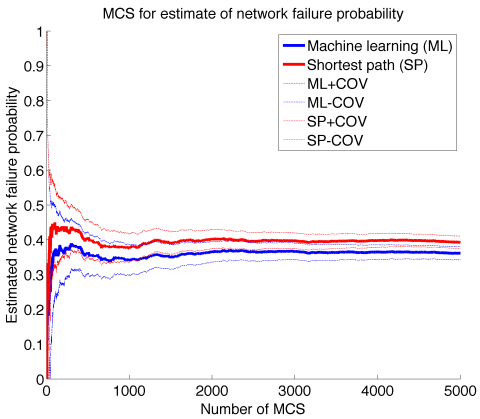
(b) Disconnection probability estimate of clusters 1 and 3 convergence, and COV.



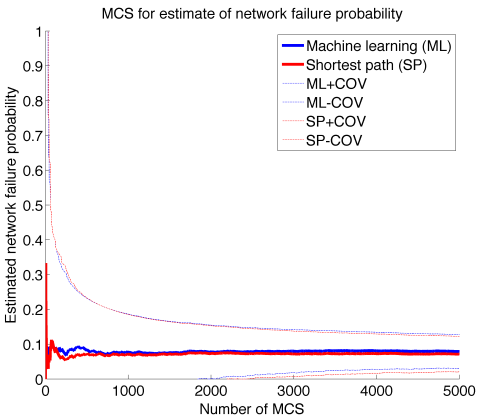
(c) Disconnection probability estimate of clusters 1 and 4 convergence, and COV.



(d) Disconnection probability estimate of clusters 2 and 3 convergence, and COV.



(e) Disconnection probability estimate of clusters 2 and 4 convergence, and COV.



(f) Disconnection probability estimate of clusters 3 and 4 convergence, and COV.

Figure 2: Convergence of disconnection probability estimate of cluster pairs.

The results for each cluster pair are summarized in Table 1, which gives the estimation error, as well as the computational time for the MCS estimate using the AdaBoost surrogate model (T_{ML}), and using the shortest path method (T_{SP}). The error is computed as the difference between the MCS estimates using the exact shortest path method and the surrogate model. Overall, the error introduced due to the surrogate model is less than five percent across the various clusters, while the computation time is approximately two orders of magnitude faster. It is noted that the surrogate model is observed to underpredict failure in almost all cases. However, more testing is needed to identify if this is a general result or specific to the experiments presented in this article.

An interesting result is the perfect performance for predicting connectivity between clusters 2 and 3. This is due in part to the network structure, since there is only one edge between the two clusters. As a result, a model is learned that only requires one weak classifier for accurate classification. Thus, T_{ML} is reduced compared to the other surrogate models. Figure 2d shows the convergence of the the estimated probability of disconnection for both MCS approaches, and is an example of the best-case performance of the surrogate model.

Figures 2a through 2f show the general trend that clusters that are further apart have a higher probability of disconnection than those that are adjacent. Furthermore, the clusters that are further apart exhibit a greater difference between the surrogate model estimate and shortest path estimate. For example, Figure 2c shows that the two estimates have converged on slightly different values. The greatest convergence gains occur in the first 1,000 simulations.

Cluster	Error (%)	T_{ML} (s)	T_{SP} (s)
1 - 2	0.72	110.6	1.480×10^4
1 - 3	2.82	109.3	2.060×10^4
1 - 4	4.52	112.2	1.504×10^4
2 - 3	0.00	1.691	3.727×10^4
2 - 4	0.82	109.4	2.067×10^4
3 - 4	0.74	112.9	2.074×10^4

Table 1: Summary of results for all six cluster-to-cluster models with component failure probability $p_f = 0.15$. Computation time for machine learning method T_{ML} and shortest path method T_{SP} provided.

5. CONCLUSIONS AND FUTURE WORK

This work demonstrates how to use machine learning methods to construct a surrogate model for estimating cluster connectivity via MCS. The approach improves the efficiency significantly and introduces relatively small (less than five percent) errors to the failure probability estimate.

Several extensions to this work are currently under exploration. First, the influence of correlated component failures on the surrogate model accuracy are being investigated, since this may occur in many practical applications. Second, we are also interested in developing a machine-learning based regression approach to predict the probability of network failure in terms of network flow quantities. Furthermore, we are exploring the possibility of combining all cluster-cluster surrogate models into one *super-model* that would incorporate all the individual surrogate models.

6. ACKNOWLEDGMENTS

The authors would like to thank the US National Science Foundation for funding under grant number CMMI 1031318. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The second author, Junho Song would like to acknowledge support by a grant entitled Development of cutting edge technologies for the multi-faceted representation of design earthquake ground motions based on analyses of acceleration records [NEMA-NH-2013-71] provided by the Natural Hazard Mitigation Research Group, National Emergency Management Agency of Korea.

7. REFERENCES

Boin, A. and McConnell, A. (2007). "Preparing for critical infrastructure breakdowns: The limits of crisis management and the need for resilience." *Journal of Contingencies and Crisis Management*, 15(1), 50–59.

Bruneau, M., Chang, S. E., Eguchi, R. T., Lee, G. C., O'Rourke, T. D., Reinhorn, A., Shinozuka, M., Tier-

ney, K., Wallace, W. A., and von Winterfeldt, D. (2003). "A framework to quantitatively assess and enhance the seismic resilience of communities." *Earthquake Spectra*, 19(4), 733–752.

Der Kiureghian, A. and Song, J. (2008). "Multi-scale reliability analysis and updating of complex systems by use of linear programming." *Reliability Engineering & System Safety*, 93(2), 288–297.

Dijkstra, E. (1956). "A note on two problems in connexion with graphs." *Numerische Mathematik*, 1(1), 269–271.

Ditlevsen, O. and Madsen, H. (1996). *Structural Reliability Methods*. John Wiley & Sons, Chichester, UK.

Freund, Y. and Schapire, R. (1995). "A desicion-theoretic generalization of on-line learning and an application to boosting." *Computational Learning Theory*, P. Vitányi, ed., Vol. 904 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 23–37.

Freund, Y. and Schapire, R. E. (1999). "A short introduction to boosting." *Journal of Japanese Society for Artificial Intelligence*, 14(5), 771–780.

Gómez, C., Sanchez-Silva, M., Dueñas-Osorio, L., and Rosowsky, D. (2013). "Hierarchical infrastructure network representation methods for risk-based decision-making." *Structure and Infrastructure Engineering*, 9(3), 260–274.

Jansson, T., Nilsson, L., and Redhe, M. (2003). "Using surrogate models and response surfaces in structural optimization – with application to crashworthiness design and sheet metal forming." *Structural and Multidisciplinary Optimization*, 25(2), 129–140.

Japkowicz, N. and Stephen, S. (2002). "The class imbalance problem: A systematic study." *Intelligent Data Analysis*, 6(5), 429–450.

Lim, H.-W. and Song, J. (2012). "Efficient risk assessment of lifeline networks under spatially correlated ground motions using selective recursive decomposition algorithm." *Earthquake Engineering & Structural Dynamics*, 41(13), 1861–1882.

Lim, H.-W., Song, J., and Nolan, K. (2015). "Seismic reliability assessment of lifeline networks using cluster-based multi-scale approach." *Earthquake Engineering Structural Dynamics*, 44(3), 355–369.

Meirlaen, J., Huyghebaert, B., Sforzi, F., Benedetti, L., and Vanrolleghem, P. (2001). "Fast, simultaneous simulation of the integrated urban wastewater system using mechanistic surrogate models." *Water Science and Technology*, 43(7), 301–307.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). "Equation of state calculations by fast computing machines." *The Journal of Chemical Physics*, 21(6), 1087–1092.

Papadrakakis, M., Papadopoulos, V., and Lagaros, N. D. (1996). "Structural reliability analysis of elastic-plastic structures using neural networks and monte carlo simulations." *Computer methods in applied mechanics and engineering*, 136(1-2), 145–163.

Rai, S. and Aggarwal, K. K. (1978). "An efficient method for reliability evaluation of a general network." *IEEE Transactions on Reliability*, 27(3), 206 – 211.

Rausand, M. and Høyland, A. (2004). *System Reliability Theory: Models, Statistical Methods, and Applications*. John Wiley & Sons, Chichester, UK.

Reed, D., Kapur, K., and Christie, R. (2009). "Methodology for assessing the resilience of networked infrastructure." *Systems Journal, IEEE*, 3(2), 174–180.

Schapire, R. E., Freund, Y., Bartlett, P., and Lee, W. S. (1998). "Boosting the margin: a new explanation for the effectiveness of voting methods." *The Annals of Statistics*, 26(5), 1651–1686.

Song, J. and Der Kiureghian, A. (2003). "Bounds on system reliability by linear programming." *Journal of Engineering Mechanics*, 129(6), 627–637.

Stern, R. (2015). "California gas distribution network source code, <https://github.com/raphaelestern/systemreliability>.

Stern, R. E., Song, J., and Work, D. B. (2014). "Machine learning-based surrogate models for network reliability analysis." 17^{th} *IFIP WG 7.5 July 3-7*.

von Luxburg, U. (2007). "A tutorial on spectral clustering." *Statistics and Computing*, 17(4).

Vugrin, E., Warren, D., Ehlen, M., and Camphouse, R. (2010). "A framework for assessing the resilience of infrastructure and economic systems." *Sustainable and Resilient Critical Infrastructure Systems*, K. Gopalakrishnan and S. Peeta, eds., Springer Berlin Heidelberg, 77–116.

Wan, X., Pekny, J. F., and Reklaitis, G. V. (2005). "Simulation-based optimization with surrogate models—application to supply chain management." *Computers & Chemical Engineering*, 29(6), 1317–1328.