Expediting life cycle cost analysis of infrastructure assets under multiple uncertainties by deep neural networks

Vahid Asghari[1], Shu-Chien Hsu[2,*], and Hsi-Hsien Wei[3]

Abstract: Deteriorating and at-risk infrastructure assets should be maintained at acceptable conditions by asset management systems (AMSs) to ensure the safety and welfare of communities. Project-level AMSs have been proposed to optimize maintenance interventions in the life cycle of assets by incorporating probabilistic and complex models but at the expense of relatively high computation time. To make complex project-level AMSs computationally applicable to all assets in a network, this paper presents a methodology to replace the time-consuming simulation modules of optimization algorithms with a trained machine learning model estimating life cycle cost analysis (LCCA) results. Deep neural network (DNN) models were trained on LCCA results of more than 1.4 million semi-synthesized bridges based on the US National Bridge Inventory considering different intervention actions and uncertainties about condition ratings, hazards, and costs. Our findings show that the trained DNN models can accurately estimate the complex LCCA results 5 order of magnitudes faster than simulation techniques. The proposed methodology helps practitioners reduce the optimization and LCCA computation times of complex AMSs to a feasible level for practical utilization.

Keywords: Project-level asset management, maintenance optimization, deep neural networks, life cycle cost analysis

[1] Ph.D. Candidate, Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, 181 Chatham Road South, Kowloon, Hong Kong. Email: vahid.asghari@connect.polyu.hk
[2,*] Associate Professor, Department of Civil and Environmental Engineering The Hong Kong Polytechnic University, Hong Kong, 181 Chatham Road South, Kowloon, Hong Kong (Corresponding Author). Email: mark.hsu@polyu.edu.hk
[3] Assistant Professor, Department of Building and Real Estate, The Hong Kong Polytechnic University, 181 Chatham Road South, Kowloon, Hong Kong. Email: hhwei@polyu.edu.hk

Introduction

Infrastructure assets are pillars of modern societies all over the world. Significant monetary resources are invested to maintain infrastructure assets at a safe and operational level. During the past two decades, hundreds of studies have focused on enhancing asset management systems at both the project and network levels (Chen and Bai 2019). Project-level asset management is the process of optimizing alternative actions such as maintenance, rehabilitation, and reconstruction (MRR) in the life cycle of an asset. Network-level project management refers to the process of selecting the best subset of projects from a pool of projects given a limited budget to maximize stakeholders' utilities.

Studies proposing project-level asset management systems have usually employed complex models to consider various phenomena affecting the asset and use various optimization techniques to find the best set of actions in the life cycle of an asset (Arif et al. 2015; Chen et al. 2015; Cheng et al. 2020; France-Mensah and O'Brien 2018; Frangopol 2011; Ghodoosi et al. 2018; Kim and Frangopol 2018; Lagaros et al. 2013; Li et al. 2020; Miyamoto et al. 2000; Montazeri and Touran 2019; Saydam and Frangopol 2015; Yang et al. 2019). This approach typically has two computational loops: 1) the simulation loop to model the uncertainties and evaluate the expected life cycle cost analysis (LCCA) results (Salimi et al. 2014; Yang et al. 2012), and 2) heuristic optimization algorithms to find the best set of actions under different constraints to optimize one or several objectives (Soliman et al. 2017; Yang et al. 2019).

Depending on the type of asset management, this flexible framework (i.e., simulating life cycle events and optimizing actions) has its upsides and downsides. On the one hand, the framework allows researchers to assess and investigate the efficiency of various types of modeling techniques such as deterioration models, hazard response models, and optimization algorithms in project-level management. On the other hand, it is currently far from practical for application to each asset at the network level due to the framework's relatively high computational time (Frangopol 2011). The computation time of project-level asset management systems that carry out Monte Carlo simulations (i.e., LCA) and optimization using heuristic algorithms is usually in the order of minutes (Kim and Frangopol 2018; Yang et al. 2012). However, this seemingly short amount of time becomes problematic if it is applied to a large network of assets. For example, if the MRR optimization of a bridge in its life cycle takes 5 minutes, it will take approximately 6 months to perform a similar process for the Texas network of bridges with more than 50,000 highway bridges. Needless to say, the more complex the models and uncertain the phenomena are as implemented in the optimization problem, the greater number of samplings and computational time is required.

Both project- and network-level asset management must be conducted within a workable computation time so that the relevant agencies and decision-makers can analyze and evaluate a multitude of MRR alternatives (Patidar et al. 2011). The importance of low computation time has been raised in several studies (Frangopol 2011; Kandil et al. 2010; Patidar et al. 2011; Yang and Frangopol 2020; Yang et al. 2012). To bypass this obstacle, network-level asset management systems that are conducted as part of real-life decision-making tools have typically used simplified and deterministic models (e.g., IBMS (Sinha et al. 2009)). These models have been used to analyze complex phenomena governing the asset (e.g., non-probabilistic deterioration models (Yang and Frangopol 2020)), provide pre-defined MRR projects (e.g., DTREE in the Indiana bridge management system (Sinha et al. 2009)), and select the best subset of projects by heuristic project selection methods (e.g., incremental utility costs ration heuristics (Patidar et al. 2011)). While simplified models developed in previous studies have played a vital role in practical asset management to date, they are by design not able to fully capture complex natural and environmental phenomena (Sánchez-Silva et al. 2016).

Given the importance of computation time in asset management, studies have employed different techniques such as parallel processing to reduce the computation time in different types of asset management systems (Kim and Frangopol 2018; Yang et al. 2012). Despite the improvements made by these studies, the development of a network-level asset management system based on an LCCA and optimization of assets with complex models has remained a challenge heretofore. Therefore, there is a need for a methodology that can reduce the LCCA computation time (the first computational loop of project-level asset management) to make much-needed headway in network-level asset management without compromising complex models (i.e., non-linear or probabilistic models). Machine learning models, such as deep neural networks, are characterized by their capabilities in learning complex correlations in a variety of systems. They have been widely used in different domains of knowledge for developing managerial and decision support frameworks, such as disaster assessment in transportation engineering (Yudi et al. 2020), hazard assessment of dam infrastructure (Rayan and H. 2020), construction costs of infrastructure assets (Ilker et al. 2020), project management and planning (Mohamad et al. 2021), and implementing renewable energies in communities (Jackson et al. 2020), in the past few years.

To address the above-described limitations, this study puts forth a methodology to estimate LCCA results (e.g., costs and utilities) rather than regular sampling at the time of planning. The proposed methodology enables decision-makers and asset managers to reduce the computation time of complex LCCA of assets by abiding an acceptable overhead computation time for training a deep neural network model. Project-level asset management studies have used machine learning techniques in the components of LCCA rather than attempting to estimate the LCCA results. Therefore, the primary contribution of this study to the body of knowledge is proposing a methodology to reduce the computation time required for the LCCA by providing a clear set of procedures for synthesizing data and training a deep neural network to estimate the LCCA results. As a result, the MRR optimization of assets could be reached in a far shorter time frame, enabling MRR optimization of each asset in a network without compromising on the complex models and inherent uncertainties in the problem. Although bridges were used as an example asset, this methodology could eventually be applied to other types of assets. The source code of the LCCA framework of this study is available online in the GIAMS GitHub repository (https://github.com/vd1371/GIAMS) (Asghari and Hsu 2020). GIAMS is an open-source general infrastructure asset management system that has been previously developed by authors and is freely accessible online. The machine learning modeling was conducted in the Python programming environment by Tensorflow (Abadi et al. 2016), Keras (Chollet 2015), and Sci-kit learn (Pedregosa et al. 2011). The codes for training the neural networks are also fully available online (https://github.com/vd1371/XProject).

The rest of this paper is structured as follows. First, the main parts of the proposed framework are explained in the methodology section. Then, the key aspects of project-level bridge management, its components, and the LCCA module in this study are discussed, followed by a case study drawing upon data from the US National Bridge Inventory (NBI) to illustrate the capabilities of the proposed methodology. Finally, the results and further discussion of them are provided, followed by a summary of key conclusions. The result of this study reveals significant improvement in the computation time of complex LCCA with negligible prediction errors.

Methodology

An overview of the proposed methodology is presented in this section. This methodology primarily consists of three parts: 1) the LCCA module, 2) data synthesizing, and 3) machine learning model training. A high-level flowchart of the proposed methodology is shown in Fig. 1, followed by a brief discussion of each of the procedures and sub-procedures.

Life cycle cost analysis

The LCCA of an asset refers to the process of evaluating various costs such as construction and maintenance incurred by the asset during its life cycle (investment horizon). When multiple sources of uncertainties and stochastic phenomena exist, Monte Carlo (MC) simulation could be used to simulate all incidents, their consequences, and their corresponding costs. To briefly explain this approach, the first step is that the condition rating of assets, which is affected by deterioration, MRR activities, hazards, and post-hazard recovery actions, is simulated in a life cycle. In the next step, agency costs and stakeholders' utilities due to MRR/recovery actions are calculated. Finally, user costs due to transportation delays, excessive fuel consumption, loss of lives, injuries, etc. are evaluated. Using statistical and probabilistic methods, quantitative representations of the simulation results are generated for further analysis and evaluation. These representations could be in the form of a simple average of user costs, agency costs, and utilities (Chen et al. 2015; Frangopol et al. 2017):

$$C_U = \frac{1}{N}\sum_{n=1}^{N}\left(\sum_{t=0}^{T}\left(\sum_{i}\left(X_{it}C_{T_{it}}\right) + C_{L_t|IM}\right)/(1+r)^t\right) \quad (1)$$

$$C_A = \frac{1}{N}\sum_{n=1}^{N}\left(\sum_{t=0}^{T}\left(\sum_{i}\left(X_{it}C_{M_{it}}\right) + C_{R_t|IM}\right)/(1+r)^t\right) \quad (2)$$

$$U = \frac{1}{N}\sum_{n=1}^{N}\left(\sum_{t=0}^{T}\left(\sum_{i}\left(X_{it}U_{M_{it}}\right)\right)/(1+r)^t\right) \quad (3)$$

where C_U, C_A and U are expected user costs, agency costs, and utilities, X_{it} is a binary parameter for action i at t, C_T is MRR user costs, C_M is maintenance costs, $C_{L|IM}$ and $C_{R|IM}$ are loss costs and recovery costs given a certain hazard with intensity measure of IM, U is the utility of actions, r is the discount rate, T is investment horizon, and N is the number of simulations.

Parametrizing LCCA

Asset parameters, MRR actions, MC simulation parameters, and LCCA results can be converted into a vectorized form. The initial condition of elements, length, width, and degradation rates are some of the assets' parameters. MRR actions are usually a vector of binaries in the management horizon. Simulation parameters could comprise parameters such as inflation rate, hazard occurrence probability and magnitude, and management horizon. These parameters, in their general form, that are incorporated in different models inside an LCCA computational core can be fed to a machine learning model for estimating LCCA results (Fig. 2)

Sampling LCCA parameters and results

A large number of samples are required to properly train an ML model. In this context, a large number of $\boldsymbol{A}$, $\boldsymbol{M}$, $\boldsymbol{S}$, and $\boldsymbol{R}$ vectors are required to ensure ML models can cover and predict all points of feature space. The $\boldsymbol{M}$ (MRR actions) and $\boldsymbol{S}$ (MC simulation parameters) could be sampled by selecting different actions for MRR plans and different approaches toward simulation. However, the variety of $\boldsymbol{A'}$ (asset parameters) is not enough due to the limited number of real assets. Synthesized assets with imaginary parameters based on real assets could be fabricated for training the ML model. This process mostly resembles the data augmentation technique in computer science problems (Redmon et al. 2016) where collecting more data is expensive or impossible. Since the number of infrastructure assets is limited, synthesizing fabricated assets is an appropriate method to generate a sufficient number of data samples for training an ML model. The number of required samples for having reliable predictions is affected by the feature space size and complexity of the problem. In the present case, millions of assets with different MRR actions and MC settings would be required to train an ML model with acceptable prediction errors.

Estimating LCCA results

Results of the LCCA computational core can be estimated by a machine learning model if enough LCCA samples for different bridges and MRR actions are available. The abstract idea of this methodology is depicted in Fig. 2. Within this process, all or a subset of parameters could be used for machine learning training purposes. The subsets of parameters include some variables given the experts' and practitioners' requirements. For example, the inflation rate could be considered as a constant in one study and a variable in another. The constants should be omitted to avoid increasing the dimension of the problem without adding information in the dataset for training ML models. Finally, estimation performance and the accuracy of results of a trained machine learning model can be validated by statistical measures such as correlation coefficient and common prediction metrics such as mean absolute percentage error. Depending on problem complexity, sample size, and feature space size, different machine learning models are subject to strengths and limitations and provide different levels of performance and accuracy.

Applicability of different machine learning models

The deep neural networks (DNN) model is an appropriate choice for estimating the results of LCCA because of three main reasons. First, DNN models have been characterized to be universal approximators that can capture any degree of non-linearity. LCCA of assets with stochastic and non-linear models as well as their results are inherently complex and highly non-linear. A candidate ML model must be able to be trained accurately on this type of dataset. Therefore, linear-based models such as simple linear regression, Lasso, and ridge would not yield satisfactory predictions. Second, DNN models are updatable. This means that DNN models can be updated with continuing training with the addition of new observations. Since sampling and training on a large dataset might take numerous steps, it would be time-consuming to start training from scratch after receiving new observations. Therefore, decision tree-based models (e.g., random forest, boosting algorithms), k-nearest neighborhood, and support vector machine regression algorithms would be inefficient. Third, DNN training time on big datasets is relatively shorter than other algorithms given recent advances in data science programming libraries/packages. Using graphical processor units (GPU) computational power, for example, Tensorflow (Abadi et al. 2016) can train complex DNN models on regular computers in a relatively short amount of time. The need for a feasible computation time during training sessions renders support vector machines model unsuitable for this methodology.

Deep neural networks

The deep neural networks model is an algorithm widely used in both academic literature and industrial problems. Layers of nodes and neurons interconnected with non-linear activation functions establish a non-linear relationship between the input parameters (independent variables) and target parameters (dependent variables):

$$\boldsymbol{l^i} = \sigma(\boldsymbol{W^i l^{i-1}} + \boldsymbol{b^i}) \tag{4}$$

where σ is the activation function of each layer, $\boldsymbol{l}^i$ and $\boldsymbol{b}^i$ are the vectorized results and bias vector of layer i, and $\boldsymbol{W}$ is the vectorized nodes' weight. Notably, $\boldsymbol{l}^0$ and $\boldsymbol{l}^n$ refer to the input vector and target value in a DNN structure with n layers.

A variant of gradient descent algorithms (e.g., RMSProp, Adam) can be used to optimize the weights and biases to maximize the similarity between the predicted and actual target values. Table 1 summarizes some of the most common cost functions such as the mean of squared error (MSE), mean of absolute errors (MAE), or mean of absolute percentage error (MAPE). DNN models have several other hyperparameters (e.g., number of hidden nodes and layers, activations functions, and optimizer) that must be tuned before training. Although hyperparameters tuning is a craft of experience, guidelines have been proposed to optimize this process (Ng 2016).

Case study: bridge management systems

In this section, an illustrative example of the proposed methodology using LCCA in bridge management systems is provided. Bridges are one of the most important infrastructure assets of a community and have been the focus of many studies by the end of the 2020s (Chen and Bai 2019). GIAMS, an open-source and freely accessible general infrastructure asset management platform (Asghari and Hsu 2020), is used to evaluate the results of the life cycle analysis of bridges in this example. In this section, first, a brief overview of project-level bridge management systems is provided. Then, details of parametrizing and sampling LCCA results in this example are provided followed by further details of DNN training.

LCCA of bridges in project-level management

Project-level bridge management systems aim to find the optimal set of actions in the life cycle of a bridge given a limited budget and other constraints (FHWA 2012). Depending on the type of study and problem, deterministic optimization methods such as linear programming (Thompson et al. 1998) or heuristic optimization methods such as genetic algorithm (Kim and Frangopol 2018) could be used to minimize the costs and maximize the utilities.

Condition rating and monitoring

Bridge elements such as deck, superstructure, and substructure deteriorate over time due to various reasons such as traffic loads and environmental stresses. The condition of these elements should be inspected periodically for further analysis. For example, the bridge data in the US is collected every 24 months and stored in the NBI (FHWA 2012). The condition rating system of bridge elements varies in different BMSs. For example, the NBI uses a discrete condition rating from 0 to 9, which is summarized in Table 2. In addition, HAZUS damage states are mapped to the NBI condition rating and shown in this table.

Markovian deterioration

Deterioration is the first and main source of uncertainty that affects the condition of bridges and outcomes of LCCA. The first-order Markovian process is a common method for modeling the probabilistic phenomenon of deterioration in infrastructure management when the condition ratings are discrete (Sinha et al. 2009; Thompson et al. 1998). The first-order Markov chain is used based on the assumption that the state of a system at $t+1$ (S_{t+1}) is solely a function of the state at t (Ross 2010). Although time-independent transition probabilities between states (i.e., $\Pr(S_{t+1} = j \mid S_t = i)$) are usually used (Ross 2010; Thompson et al. 1998), time-dependent transition probabilities as a function of elements' age have also been proposed (Sinha et al. 2009) to model deterioration of elements. Deterioration rates of bridge elements in this case study are based on the proposed rates in IBMS (Sinha et al. 2009).

Probabilistic hazards and responses

Hazards and the hazard responses of assets are within the second category of uncertainties in this study. Although hazards are rare incidents, they usually lead to enormous subsequent losses. Hazard occurrence and sampling could be modeled with the Poisson process (Li et al. 2020):

$$p(n) = \frac{(\lambda t)^n e^{-\lambda t}}{n!} \tag{5}$$

where $p(n)$ is the probability of n occurrences with an occurrence rate of λ in t units of time. The response of a bridge to an earthquake occurrence could be evaluated by the fragility curves proposed in the HAZUS (FEMA-NIBS 2003). Fragility curves, Eq. (6), yield the probability ($P_{S\geq S_i|IM}$) of exceeding a damage state S_i given an earthquake intensity (IM):

$$P_{S\geq S_i|IM} = \Phi\left\{\frac{1}{\beta_{S_i}}\ln\left(\frac{IM}{m_{S_i}}\right)\right\} \tag{6}$$

where m_{S_i} and β_{S_i} are the median and standard deviation of ground motion intensity, and Φ is the standard normal cumulative distribution function. However, the parameters of the fragility curves proposed in the HAZUS govern intact assets and not deteriorated ones. In other words, this approach provides a similar probability of exceedance from damage states for both intact and degraded assets. Other studies (Dong et al. 2014; Ghosh and Padgett 2009) have suggested time-variant fragility curves to incorporate deterioration due to corrosion into seismic performance evaluation and finding the conditional probability of damage states in response to earthquakes. Inspired by the HAZUS methodology and without loss of generality, state-dependent fragility curves are used in this study to overcome this limitation. In this approach, the probabilities of exceedance from the deteriorated state to the collapsed state are normalized to keep the sum of probabilities equal to 1. As a result, the probability of exceeding a damage state from state S_j could be quantified as:

$$P_{S\geq S_i|IM,S_j} = \Phi\left\{\frac{1}{\beta_{S_i}}\ln\left(\frac{IM}{m_{S_i}}\right)\right\}/\Phi\left\{\frac{1}{\beta_{S_i}}\ln\left(\frac{IM}{m_{S_j}}\right)\right\}\ ,\ S_i \geq S_j \tag{7}$$

The condition rating corresponding to a damage state can be found in Table 2 which maps the two systems based on their descriptions. Notably, although HAZUS – MH2.1 does not provide information regarding casualty data and losses for bridges, the methodology holds for other assets and their response that would yield different losses.

Costs volatility

Costs volatility is the third source of uncertainty in this study. The uncertainty in costs stems from factors such as fuel price and average daily traffic. The Wiener process has been extensively applied for short/long-term modeling of uncertain prices and values in finance and economics (Brennan and Schwartz 1976; Capasso et al. 2020; George and George 2018; Hirsa and Neftci 2013; Kim et al. 2017; Kim and Lee 2018; Pindyck 1993; Ross 2010). It has also been employed in the construction domain similarly (Ashuri et al. 2012; Ilbeigi and Ashuri, Baabak Hui 2014). The Wiener process is a category of stochastic processes for modeling continuously volatile market prices and indicators (Hirsa and Neftci 2013). Consistent with these studies, it is assumed that user costs volatility follows the Wiener process with drift, Eq. (8), in this study:

$$v(t) = v_0 + \eta t + \sigma W_t \tag{8}$$

where W_t is the Wiener process, η is the drift ratio (the trend of costs) and σ is the standard deviation (volatility of costs), and v_0 is the initial value. The drift ratio and standard deviation of Eq. (8) can be fine-tuned and calibrated with historical data.

MRR plans and recovery actions

Maintenance, rehabilitation, reconstruction, and do nothing are four typical actions that are planned for assets in a time horizon (Hawk and Small 1998; Sinha et al. 2009; Thompson et al. 1998). Fig. 3 also shows a possible MRR plan of a bridge in this case study consisting of these possible actions (i.e., 0: do nothing, 1: maintenance, 2: rehabilitation, 3: reconstruction), represented as a 2-D vector. Recovery actions refer to a set of actions that should be undertaken after the occurrence of a hazard to restore the asset to an acceptable service level. The effectiveness of MRR activities and recovery actions were inspired by previous BMSs, such as BRIDGIT (Hawk and Small 1998), or rationally assumed (i.e., the condition rating of the asset after recovery actions will be similar to that of NBI rating 8).

User costs, agency costs, and utilities

User costs MRR actions for bridges are mainly incurred because of delays in the transportation times of users and commuters. These costs can be modeled as a function of fuel price and workers' hourly wage. The user costs functions that are implemented in this study are based on the estimates provided by the Texas Department of Transportation (2020):

$$C_U = C_d + C_f \tag{9}$$

$$C_d = T \times ADT \times \left[\left(\frac{L_1}{V_a} - \frac{L_1}{V_b}\right)(1 - p_a) + \left(\frac{L_2}{V_c} - \frac{L_1}{V_b}\right)p_a\right] \times \left[p_T C_{d_T} + (1 - p_T C_{d_P})\right] \tag{10}$$

$$C_f = T \times ADT \times [L_1(1 - p_a) + L_2 p_a] \times \left[p_T C_{f_T} + (1 - p_T C_{f_P})\right] \tag{11}$$

where C_U is the total user costs, C_d is costs due to travel delay, C_f is costs due to excessive fuel consumption, T is project duration, ADT is average daily traffic, L_1 is the length of the bridge or MRR projects, L_2 is the length of detour (alternate road), V_a is average speed prior to construction, V_b is the average speed during construction, V_c is the average speed in the detour, p_T is the truck percentage, p_a is the percentage of drivers that would use detour, C_{d_T} and C_{d_P} are values of travel time for trucks and personal vehicles, C_{f_T} and C_{f_P} are marginal costs of trucks and personal vehicles fuel burn. Further details regarding costs and other parts of user costs formulas can be found in (TexasDOT 2020).

Agency costs refer to the direct monetary resources that must be invested in the maintenance, rehabilitation, or reconstruction of bridges (or assets in general). These agency costs could be formulated as a function of the design type of the bridges, element type, material, and area or volume of the project. Sinha et al. (2009) proposed using the Cobb-Douglas production function (Nicholson and Christopher 2011) for estimating the agency costs:

$$c = A \times L^{\alpha} \times W^{\beta} \tag{12}$$

where c is estimated project costs, A, α, β are regression coefficients, L and W are lengths and width of bridges. Given the type of project, elements type, and materials, regression coefficients in

Eq. (12) could differ from one another. These regression coefficients and further details could be found in (Sinha et al. 2009).

Utility theory has been widely used to measure how appealing an MRR plan is to the agencies and decision-makers. In this study, the utility of MRR actions regarding deck, substructure, and superstructure of bridges are (Bai et al. 2013):

$$u_{DC} = 122.75 \times (1 - e^{-0.19x}) \quad (13)$$

$$u_{SP} = 119.13 \times (1 - e^{-0.203x}) \quad (14)$$

$$u_{SB} = 119.49 \times (1 - e^{-0.202x}) \quad (15)$$

where u_{DC}, u_{SP}, u_{SB} are utility of deck, super structure, and substructure with a condition rating of x. Consequently, the utility of an action can be quantified as:

$$U = u_2 - u_1 \quad (16)$$

where u_2 and u_1 are the utility of the element after and before conducting an MRR action. Multi-attribute utility theory is usually used to combine several utilities into one to simplify the optimization process (Bai et al. 2013; Frangopol et al. 2017). Accordingly, the weighted sum of bridge elements' utilities with equal weights is used as the total utility in this case study.

The LCCA module

Monte Carlo simulation is usually used to consider the uncertainties and calculate the expected values of outcomes (i.e., user costs, agency costs, and utilities). Although other factors such as reliability, sustainability, and risk could also be quantified and analyzed for each MRR plan in a life cycle (Frangopol et al. 2017), this study focuses on the average of the user costs, agency costs, and utilities without loss of generality. The current implemented LCCA module in GIAMS can yield agency costs, user costs, and utility of implementing a proposed MRR plan in the investment horizon of a bridge/network. Details of the computational steps in the LCCA module and relations among the implemented models in the case study are provided in Table A1 in Appendix A.

LCCA parameters of bridges

LCCA parameters can be divided into three main groups: 1) constants which are the underlying assumptions in this case study, 2) variables which are bridge specific parameters, and 3) MRR plans which are possible timings for conducting maintenance, rehabilitation, or reconstruction of a bridge. These parameters, including their value or range of values, are summarized in Table 3.

Sampling bridge LCCA parameters and results

Bridges' characteristics, MRR plans, and environmental factors are randomly synthesized to generate sample bridges based on the Indiana bridge network available in NBI. After conducting LCCA for each sample bridge, the life cycle analysis results, as well as other related parameters, are stored in a dataset for training machine learning models. Each synthesized bridge with its random MRR plan is a point in the feature space for the machine learning model. Accordingly, more than 1.4 million synthesized bridges (samples) with random MRR plans were sampled and analyzed. Considering deterioration, earthquakes, and user costs as main sources of uncertainties, approximately 1000 simulations were required to reach a 95% confidence interval for the LCCA results. This estimation was derived based on the central limit theorem which states the average of simulations results follows the normal distribution with an average of $\mu_{\bar{x}}$ and standard deviation of $\frac{\sigma}{\sqrt{n}}$ for n iterations. Accordingly, the confident interval, Eq. (17), and a minimum number of iterations, Eq. (18), could be derived by (Law and Kelton 2000):

$$\pm Z_{\alpha/2}\frac{\sigma_0}{\sqrt{n}} \qquad (17)$$

$$n \geq \left(\frac{Z_{\frac{\alpha}{2}} \times \sigma_0}{\epsilon}\right) \qquad (18)$$

where σ_0 is initial estimate of standard deviation and ϵ maximum allowable error. In this study, 1% of the initial estimate of mean was set as the maximum allowable error (ϵ). Table 4 provides a statistical summary of the synthesized LCCA target value results (i.e., covering user costs, agency costs, and utility).

Estimating LCCA of bridges with DNN

Data preprocessing

The LCCA parameters must be normalized, encoded, and pruned to be able to be fed to machine learning models because of redundant parameters, nominal parameters, ordinal parameters, and differences in the ranges of continuous variables. First and foremost, redundant variables (constants for all samples) should be eliminated to reduce the dimensionality while maintaining useful information from datasets. Constant parameters such as the number of elements are removed since they are shared among all samples and will have an adverse effect on the ML model training. More importantly, not all variable parameters equally affect the three main outputs (i.e., user costs, agency costs, and utility) of the LCCA. For example, detour length affects the user cost while it does not affect agency costs and utilities. To reduce dimensionality and consequently prevent the learning models from overfitting, three different subsets of the dataset were created for user costs, agency costs, and utilities with redundant features removed. These three datasets contain several parameters such as condition ratings in common and some parameters exclusively. Table 5 summarizes the parameters that are excluded from each dataset. Second, one-hot encoding is used to convert nominal parameters (e.g., material, road type, HAZUS classification) into a string of binaries. Each nominal parameter with k categories is converted to $k - 1$ binary parameters by one-hot encoding. Also, MRR actions for each year were converted from categorical to binary variables in a different manner. Do nothing, maintenance, rehabilitation, and reconstruction were first converted to integers, 0, 1, 2, 3, respectively. Then these integer values were converted to binary values (e.g., 3 was converted to 1, 1). As a result of encoding MRR actions and one-hot encoding of other categorical variables, 32 parameters of simulation and bridges' characteristics and 30 parameters of a 20-year horizon MRR plan for three elements were converted to a total of 122 normalized and binary parameters. Finally, since the range of continuous variables varies, they should be normalized to a range between 0 and 1 for ML training:

$$N(\mathbf{X}_i) = \frac{\mathbf{X}_i - \min(\mathbf{X}_i)}{\max(\mathbf{X}_\mathrm{i}) - \min(\mathbf{X}_i)} \tag{19}$$

where $N(\mathbf{X}_i)$, $\min(\mathbf{X}_i)$, $\max(\mathbf{X}_i)$ are the normalization, minimum, and maximum of the parameters $\mathbf{X}_i$. Similarly, normalization should be applied to ordinal parameters (e.g., condition ratings).

Hyperparameter tuning

Following the work of (Asghari et al. 2020), DNN hyperparameters including, but not limited to, optimization algorithm, activation functions, cost function, type, and the number of layers were determined in this study and are summarized in Table 6.

A number of these hyperparameters are set given the nature of the problem. For example, linear function is suggested as the final layer activation function for regression tasks (Ng 2016). Some of the hyperparameters, including Adam optimizer as the optimization function (Kingma and Ba 2015), *ReLu* as the hidden layer activation functions (Xu et al. 2015), *tanh* as the input layer activation function (Ng 2016), are reportedly recommended in the literature based on their superior performance in comparison to their counterparts. Considering the convergence speed and prediction accuracy, batch sizes are suggested to be relatively small and a power of 2 (Keskar et al. 2017). The slicing proportions are arbitrary values that are set based on dataset sizes. To further illustrate, smaller test size portions can be used for big datasets (Ng 2016). Early stopping as a regularization technic can be used to terminate optimization when there is no improvement in the accuracy of prediction results on the cross-validation set (Ng 2016). To this end, a large number of epochs (iterations of optimization) is used not to terminate the optimization before early stopping technic does. Starting from smaller neural networks with few hidden nodes and one hidden layer, different structures should be trained and tested to minimize improve prediction accuracy (reduce the variance problem). After finding a structure for the neural networks that can yield acceptable prediction results (with low variance problem), L1 or L2 regularization technics can be used to mitigate possible overfitting problem (minimize difference between the prediction results on test set and train set) (Ng 2016). Depending on the type of training goals, cost function considerably impacts prediction accuracy and training time. In this study, for example, the range of user cost values is relatively large. If MSE or MAE are chosen as the cost function of DNN to model user costs, the model would try to fit on larger values to attain the lowest possible MSE at the cost of neglecting smaller values. Therefore, by normalizing errors, MAPE would be a better choice for the cost function of the DNN model in this study. To compare the effectiveness of cost functions in this study, Fig. 4 depicts the MAPE of predictions for 3 different cost functions after 100 epochs of training.

Results and Discussion

Calculation of each LCCA using the GIAMS platform and considering the three sources of uncertainties takes 5.3 seconds. The LCCA of the synthesized bridges were analyzed and evaluated using an Intel(R) Xeon(R) E5-2697 CPU, 128 GB RAM, with 72 logical processors. Through leveraging parallel processing and using all 72 processors, the whole bridge sampling session took nearly 31.5 hours. Then the dataset was normalized, encoded, and pruned to form three datasets for training three models for user costs, agency costs, and utility. The neural networks training process was conducted by GPU NVIDIA Quadro P620 and with specialized libraries required for GPU training including CUDA 10.1, and cuDNN 7.4. The training session took approximately 57 minutes.

Other machine learning models (i.e., decision trees, random forest, shallow neural network, and linear regression) are trained to compare their results with that of the trained DNN model. Following previous studies (Wang et al. 2020), the hyperparameters of these models were set as follows: A) Decision tree: maximum branching depth = 5, minimum samples in each leaf for splitting = 2, minimum samples to be in each leaf = 1, B) Random forest: number of trees = 500, number of features to look for when splitting = all features, maximum branching depth = 5, minimum samples in each leaf for splitting = 2, minimum samples to be in each leaf = 1, C) Shallow neural network: similar to the proposed DNN but with only one layer, D) Linear regression:

ordinary least squares (OLS) regression. Notably, the high computation time of support vector machine and k-nearest neighbors on large datasets made them infeasible for evaluation in this study. Since the ranges of the user and agency costs are large, the MSE values are misleading and vague for assessing the prediction performance of the regression models in this study. Therefore, R-squared and MAPE among other prediction accuracy metrics are provided. The results of the regression analyses for all the models trained on the test sets are summarized in Table 7, and the corresponding graphs for visual validation of the regression analysis results are also provided in Fig. 5.

With an R2 of more than 0.98 and MAPE of less than 2% in all models, the numerical results of the regression analysis are satisfactory. These results could also be visually validated in Fig. 5, where the predicted values are drawn against actual values for user cost, agency cost, and utility. In addition, based on the results shown in Table 7, DNN outperformed other models by a large margin, demonstrating it to be a viable approach.

The computation time of the LCCA module in GIAMS with three sources of uncertainties, as well as that of the estimators, is provided in Table 8. Although the data synthesis and training of the models are relatively time-consuming during the training phase, the LCCA estimation is far less time-consuming during the analysis phase. After completing the overhead computation time for sampling and training, the trained DNN model can estimate the LCCA results and yield similar outcomes with an acceptable range of errors 5 order of magnitudes faster than the regular MC simulation method. This trade-off is especially beneficial in terms of computational time if LCCA is to be conducted millions of times in the optimization procedure. Drawing upon the previous discussion on the Texas highway bridges, computation for finding the optimal MRR of each bridge could theoretically be reduced to approximately 105.5 hours from 6 months (assuming optimization by genetic algorithm with 200 generations, 200 individuals in each generation, 50,000 assets, and 12 computational processors).

The proposed methodology, i.e., estimation of LCCA results using machine learning models, could be used across different domains of asset management to reduce the computational time of life cycle optimization. Complex models, different sources of uncertainties, and their consequently large computation time for LCCA are the main barriers to upscaling advanced LCCA frameworks for application to large networks. This methodology could tackle this limitation and be applied to various types of assets such as pavement, railways, and buildings. Notably, the number and range of features affect the asset synthesizing (data augmentation) step. If the management horizon increases from 20 years to 40 years and 4 elements are considered instead of three, 100 new features will be added to the dataset. The issue of dimensionality affects both the LCCA and life cycle optimization, making them even further unfeasible for use in network-level management. However, the proposed methodology would only require more synthesized samples so that the machine learning model could accurately be trained.

Conclusions

This paper puts forward a new methodology to reduce the LCCA computation time by estimating the LCCA results of an asset using deep neural networks. Complex project-level asset management systems that search for the optimal MRR plan of an asset could not be applied for each asset of a network because of their high computational costs. Due to this issue, asset management systems applied in real life usually use simplified models to assign an MRR plan to each asset in a network. To overcome this challenge, DNN models were trained on datasets consisting of numerous synthesized bridges based on the US NBI with randomly generated MRR plans and corresponding LCCA results. Since three sources of stochastic uncertainties were present in the LCCA mode, the LCCA results of each bridge were derived from the Monte Carlo simulation. The three DNN models for the user costs, agency costs, and utility had satisfactory prediction results (i.e., MAPE less than 2%, R-squared more than 0.98). DNN is an appealing option because it can: 1) be updated after observing new samples, 2) capture any degree of non-linearity in complex datasets, and 3) be trained on large datasets with reasonable computation time. Although this methodology has a relatively large overhead computational cost, the trained DNN models can yield similar results but hundreds of times faster than the Monte Carlo simulation that is used in the MRR plan optimization of an asset by heuristic optimization algorithms.

The main limitation of this study is the modeling approach toward uncertain phenomena such as earthquake occurrence, user costs volatility, and deterioration. However, without the loss of generality of the proposed methodology, more complex and advanced models could be used to imitate the underlying phenomena in the LCCA of assets in the analysis. Bridge management is a discipline with great importance in infrastructure asset management, though other assets such as pavement and railway could be studied with the same methodology. Theoretically, more data samples could improve DNN prediction results. Therefore, synthesizing more assets could lead to constructing near-perfect machine learning models.

The proposed flexible methodology for estimating the LCCA results by training a DNN model provides the opportunity to use more complex models in the MRR optimization of each asset in a network. Filling the gap between academic and applied project-level AMSs, this methodology enables practitioners and decision-makers to possibly identify more advantageous MRR strategies by incorporating probabilistic, non-linear, and other advanced techniques into their long-term planning. Future research could compare the MRR optimization results of Monte Carlo simulation and trained DNN models. Further research could focus on utilizing the proposed methodology in other infrastructure asset management systems such as pavement management systems and investigate the efficacy of the results.

Appendix A

Table A1. Algorithm of the LCCA module in the case study

1:	**Input:** Range of bridge characteristics and simulation parameters	// Table 3
2:	C_{u_j}, C_{a_j}, U_j = 0, 0, 0	// C_u, C_a, U_j: holders for user costs, agency costs, and utilities for each element j
3:	MRR = A synthesized MRR plan	// Synthesizing an MRR plan

BRG = A synthesized bridge	// Using values of Table 3
SIM = Synthesized simulation parameters	// Using values of Table 3
for $n \in \{0, 1, \dots, N\}$ **do**:	// N: Number of simulations
$\quad C'_{u_j}, C'_{a_j}, U'_j$ = 0, 0, 0	// C'_u, C'_a, U'_j: holders for user costs, agency costs, and utilities for each element j in each round of simulation
$\quad H$ = A generated hazard sample	// Based on hazard distribution characteristics of SIM parameters and Eq. (5)
$\quad$ **for** t $\in \{0, 2, \dots, T\}$ **do**:	// T: Management horizon in SIM parameters
$\quad\quad$ **for** $j \in \{elements\}$:	// for each element in the bridge
$\quad\quad\quad$ **if** t **in** H:	
$\quad\quad\quad\quad$ **find** S'_j	// S': State of the bridge in response to hazards given BRG parameters and Eq. (7)
$\quad\quad\quad\quad$ RC = **find** recovery plans	// Finding proper recovery actions based on the recovery model
$\quad\quad\quad$ **if** t **in** RC:	
$\quad\quad\quad\quad$ **find** C'_{a_j}, C'_{u_j}	// Find costs based on Eq. (8)– Eq. (12) and corresponding models[1, 2] given SIM and BRG parameters
$\quad\quad\quad\quad$ **find** S'_j	// S': State of the bridge element after recovery based on the recovery model and guidelines in literature[3]
$\quad\quad\quad$ **elif** t **in** MRR and action != DONOT:	
$\quad\quad\quad\quad$ **find** C'_{a_j}, C'_{u_j}, U'_j	// Find costs based on Eq. (8)– Eq. (12) and corresponding models[1, 2] and find utilities based on Eq. (13) to Eq. (16) and corresponding models[4] given SIM and BRG parameters
$\quad\quad\quad\quad$ **find** S'_j	// S': State of the bridge element after MRR actions based on the recovery model and guidelines in literature[3]
$\quad\quad\quad$ **else:**	
$\quad\quad\quad\quad$ **find** S'_j	// S': State after degradation based on degradation model[1] and BRG
$\quad\quad\quad$ $C'_{u_j} \mathrel{+}= C'_{u_j}/(1+r)^t$ $\quad\quad\quad$ $C'_{a_j} \mathrel{+}= C'_{a_j}/(1+r)^t$ $\quad\quad\quad$ $U'_j \mathrel{+}= U'_j/(1+r)^t$	// Adding discounted user costs, agency costs, and utility of actions for each element to its corresponding holder in each round simulation
$\quad\quad\quad$ $S_j = S'_j$	// Update state of bridge element
$\quad\quad\quad$ age_j = **Update** age of element j	// Update age of each element given the type of actions
$\quad\quad$ **end for**	
$\quad$ $C_{uj} = (C_{uj} \times (n-1) + C'_{uj})/n$ $\quad$ $C_{aj} = (C_{aj} \times (n-1) + C'_{aj})/n$ $\quad$ $C_{uj} = (C_{uj} \times (n-1) + C'_{uj})/n$	// Updating user costs, agency costs, and utilities for finding the average results of Monte Carlo simulation
end for	
return C_{u_j}, C_{a_j}, U_j	

Note: 1- (Sinha et al. 2009), 2- (TexasDOT 2020), 3-(Hawk and Small 1998)**,** 4- (Bai et al. 2013)

Data Availability Statement

Some or all data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Craig, C., et al. (2016). "TensorFlow: Large-scale machine learning on heterogeneous systems." *arXiv preprint arXiv:1603.04467*.

Arif, F., Bayraktar, M. E., and Chowdhury, A. G. (2015). "Decision support framework for infrastructure maintenance investment decision making." *Journal of Management in Engineering*, 32(1), 04015030.

Asghari, V., and Hsu, S.-C. (2020). "GIAMS." <https://github.com/vd1371/GIAMS> (Jan. 6, 2021).

Asghari, V., Leung, Y. F., and Hsu, S. C. (2020). "Deep neural network based framework for complex correlations in engineering metrics." *Advanced Engineering Informatics*, Elsevier, 44(February), 101058.

Ashuri, B., Kashani, H., Molenaar, K. R., Lee, S., and Lu, J. (2012). "Risk-neutral pricing approach for evaluating BOT highway projects with government minimum revenue guarantee options." *Journal of Construction Engineering and Management*, 138(4), 545–557.

Bai, Q., Labi, S., Sinha, K. C., and Thompson, P. D. (2013). "Multiobjective optimization for project selection in network-level bridge management incorporating decision-maker's preference using the concept of holism." *Journal of Bridge Engineering*, 18(9), 879–889.

Brennan, M. J., and Schwartz, E. S. (1976). "The pricing of equity-linked life insurance policies with an asset value guarantee." *Journal of Financial Economics*, 3(3), 195–213.

Capasso, G., Gianfrate, G., and Spinelli, M. (2020). "Climate change and credit risk." *Journal of Cleaner Production*, 266, 121634.

Chen, L., and Bai, Q. (2019). "Optimization in decision making in infrastructure asset management: A review." *Applied Sciences*, 9(7), 1380.

Chen, L., Henning, T. F. P., Raith, A., and Shamseldin, A. Y. (2015). "Multiobjective optimization for maintenance decision making in infrastructure asset management." *Journal of Management in Engineering*, 31(6), 04015015.

Cheng, M., Yang, D. Y., and Frangopol, D. M. (2020). "Investigation of effects of time preference and risk perception on life-cycle management of civil infrastructure." *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, 6(1), 04020001.

Chollet, F. (2015). "KERAS." *GitHub*, <https://github.com/keras-team/keras> (Jan. 6, 2021).

Dong, Y., Frangopol, D. M., and Saydam, D. (2014). "Pre-earthquake multi-objective probabilistic retrofit optimization of bridge networks based on sustainability." *Journal of Bridge Engineering*, 19(6), 4014018.

FEMA-NIBS. (2003). “Hazus–MH 2.1: Technical manual.” Washington DC: Federal Emergency Management Agency, <www.fema.gov/plan/prevent/hazus> (Jan. 6, 2021).

FHWA. (2012). *Recording and coding guide for the structure inventory and appraisal of the nation’s bridges*. Federal Highway Administration, Washington, DC.

France-Mensah, J., and O’Brien, W. J. (2018). “Budget allocation models for pavement maintenance and rehabilitation: comparative case study.” *Journal of Management in Engineering*, 34(2), 05018002.

Frangopol, D. M. (2011). “Life-cycle performance, management, and optimisation of structural systems under uncertainty: accomplishments and challenges.” *Structure and Infrastructure Engineering*, 7(6), 389–413.

Frangopol, D. M., Dong, Y., and Sabatino, S. (2017). “Bridge life-cycle performance and cost: analysis, prediction, optimisation and decision-making.” *Structure and Infrastructure Engineering*, Taylor & Francis, 13(10), 1239–1257.

George, L., and George, L. (2018). “Brownian motion and stochastic processes.” *Energy Power Risk*, Emerald Publishing Limited, 3–32.

Ghodoosi, F., Abu-Samra, S., Zeynalian, M., and Zayed, T. (2018). “Maintenance cost optimization for bridge structures using system reliability analysis and genetic algorithms.” *Journal of Construction Engineering and Management*, 144(2), 04017116.

Ghosh, J., and Padgett, J. E. (2009). “Multi-hazard consideration of seismic and aging threats to bridges.” *Structures Congress 2009: Don’t Mess with Structural Engineers: Expanding Our Role*, 1–10.

Hawk, H., and Small, E. P. (1998). “The BRIDGIT bridge management system.” *Structural Engineering International*, 8(4), 309–314.

Hirsa, A., and Neftci, S. N. (2013). *An introduction to the mathematics of financial derivatives*. Academic press.

Ilbeigi, M., and Ashuri, Baabak Hui, Y. (2014). “A stochastic process to model the fluctuations of asphalt cement price.” *Construction Research Congress 2014: Construction in a Global Network*.

Ilker, K., D., G. D., and David, J. H. (2020). “Improving the accuracy of early cost estimates on transportation infrastructure projects.” *Journal of Management in Engineering*, American Society of Civil Engineers, 36(5), 4020063.

Jackson, B., Aidan, B., Emily, J., and Roshanak, N. (2020). “Characterizing the key predictors of renewable energy penetration for sustainable and resilient communities.” *Journal of Management in Engineering*, American Society of Civil Engineers, 36(4), 4020016.

Kandil, A., El-Rayes, K., and El-Anwar, O. (2010). “Optimization research: Enhancing the robustness of large-scale multiobjective optimization in construction.” *Journal of Construction Engineering and Management*, 136(1), 17–25.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. (2017). “On large-batch training for deep learning: Generalization gap and sharp minima.” *International Conference on Learning Representations*.

Kim, S., and Frangopol, D. M. (2018). “Decision making for probabilistic fatigue inspection planning based on multi-objective optimization.” *International Journal of Fatigue*, Elsevier, 111, 356–368.

Kim, Y., and Lee, E. B. (2018). “Optimal investment timing with investment propensity using fuzzy real options valuation.” *International Journal of Fuzzy Systems*, Springer Berlin Heidelberg, 20(6), 1888–1900.

Kim, Y., Shin, K., Ahn, J., and Lee, E. B. (2017). “Probabilistic cash flow-based optimal investment timing using two-color rainbow options valuation for economic sustainability appraisement.” *Sustainability*, 9(10), 1781.

Kingma, D. P., and Ba, J. (2015). “Adam: A method for stochastic optimization.” *3rd International Conference for Learning Representations, San Diego*.

Lagaros, N. D., Kepaptsoglou, K., and Karlaftis, M. G. (2013). “Fund allocation for civil infrastructure security upgrade.” *Journal of Management in Engineering*, 29(2), 172–182.

Law, A. M., and Kelton, W. D. (2000). *Simulation modeling and analysis*. New York: McGraw-Hill.

Li, Y., Dong, Y., Frangopol, D. M., and Gautam, D. (2020). “Long-term resilience and loss assessment of highway bridges under multiple natural hazards.” *Structure and Infrastructure Engineering*, 16(4), 626–641.

Miyamoto, A., Kawamura, K., and Nakamura, H. (2000). “Bridge management system and maintenance optimization for existing bridges.” *Computer-Aided Civil and Infrastructure Engineering*, 15(1), 45–55.

Mohamad, A., Jordan, S. F., and M., S. I. (2021). “Data-driven machine learning approach to integrate field submittals in project scheduling.” *Journal of Management in Engineering*, American Society of Civil Engineers, 37(1), 4020104.

Montazeri, N., and Touran, A. (2019). “Applied decision-making framework for maintenance scheduling in bridge management.” *Precedings on the Creative Construction Conference*, 547–555.

Ng, A. (2016). *Machine Learning Yearning*. *https://www.mlyearning.org/*, https://www.mlyearning.org/.

Nicholson, W., and Christopher, S. (2011). *Microeconomic theory: Basic principles and extensions*. South-Western College Pub.

Patidar, V., Labi, S., Morin, T., Thompson, P. D., and Sinha, K. C. (2011). “Evaluating methods and algorithms for multicriteria bridge management at the network level.” *Transportation Research Record*, 2220(1), 38–

47.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). “Scikit-learn: machine learning in Python.” *Journal of Machine Learning Research*, 12, 2825–2830.

Pindyck, R. S. (1993). “Investments of uncertain cost.” *Journal of Financial Economics*, 34(1), 53–76.

Rayan, A., and H., E. I. (2020). “Evaluation and prediction of the hazard potential level of dam infrastructures using computational artificial intelligence algorithms.” *Journal of Management in Engineering*, American Society of Civil Engineers, 36(5), 4020051.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). “You only look once: Unified, real-time object detection.” *Proceedings of the IEEE conference on computer vision and pattern recognition.*

Ross, S. M. (2010). *Introduction to probability models*. Academic Press.

Salimi, S., Mawlana, M., and Hammad, A. (2014). “Simulation-based multiobjective optimization of bridge construction processes using parallel computing.” *Proceedings of the 2014 Winter Simulation Conference*, 3272–3283.

Sánchez-Silva, M., Frangopol, D. M., Padgett, J., and Soliman, M. (2016). “Maintenance and operation of infrastructure systems: Review.” *Journal of Structural Engineering*, 142(9), 1–16.

Saydam, D., and Frangopol, D. M. (2015). “Risk-based maintenance optimization of deteriorating bridges.” *Journal of Structural Engineering (United States)*, 141(4), 1–10.

Sinha, K. C., Labi, S. A., McCullouch, B. G., Bhargava, A., and Qiang, B. (2009). “Updating and enhancing the Indiana bridge management system (IBMS).” *Publication FHWA/IN/JTRP-2008/30. Joint Transportation Research Program, Indiana Department of Transportation and Purdue University*, West Lafayette, Indiana, <https://docs.lib.purdue.edu/jtrp/1164/> (Jan. 6, 2021).

Soliman, A. A. S., Ahmed, M., and Tarek, Z. (2017). “Decision-making framework for integrated asset management.” *Proceedings, Annual Conference - Canadian Society for Civil Engineering*, 161–170.

TexasDOT. (2020). “Estimate of road user cost for personal vehicles and commercial trucks.” <https://www.txdot.gov/inside-txdot/division/construction/road-user-costs.html> (Jan. 6, 2021).

Thompson, P. D., P.Small, E., Johnson, M., and R.Marshal, A. (1998). “The Pontis bridge management system.” *Structural engineering international*, 8(4), 303–308.

Wang, R., Asghari, V., Hsu, S. C., Lee, C. J., and Chen, J. H. (2020). “Detecting corporate misconduct through random forest in China’s construction industry.” *Journal of Cleaner Production*, Elsevier Ltd, 268, 122266.

Xu, B., Wang, N., Chen, T., and Li, M. (2015). “Empirical evaluation of rectified activations in convolutional network.” *arXiv:1505.00853*.

Yang, D. Y., and Frangopol, D. M. (2020). “Life-cycle management of deteriorating bridge networks with network-level risk bounds and system reliability analysis.” *Structural Safety*, 83, 101911.

Yang, D. Y., Frangopol, D. M., and Teng, J. G. (2019). “Probabilistic life-cycle optimization of durability-enhancing maintenance actions: Application to FRP strengthening planning.” *Engineering Structures*, 188, 340–349.

Yang, I. T., Hsieh, Y. M., and Kung, L. O. (2012). “Parallel computing platform for multiobjective simulation optimization of bridge maintenance planning.” *Journal of Construction Engineering and Management*, 138(2), 215–226.

Yudi, C., Qi, W., and Wenying, J. (2020). “Rapid assessment of disaster impacts on highways using social media.” *Journal of Management in Engineering*, American Society of Civil Engineers, 36(5), 4020068.