Reinforcement Learning-based Bridge Inspection Management

XIN ZHANG[1], MANUEL SALMERON[1], BENJAMIN E. WOGEN[1], XIAOYU LIU[2], LISSETTE ITURBURU[1] and SHIRLEY DYKE[1,2]

ABSTRACT

Biannual inspections are required to assess the physical and functional condition of our nation's bridges. The Federal Highway Administration (FHWA) and various Departments of Transportation (DOT) in the United States periodically update specifications and techniques to normalize and advance the bridge inspection procedures. However, some ambiguity remains in these inspection requirements. One relevant example relates to inspection intervals and techniques. Currently, FHWA requires routine bridge inspection at least every two years, and if necessary, inspectors can adjust the inspection frequency. The details of how one would adjust the inspection frequency is not specified. And while many advanced techniques, e.g., ultrasonic surface wave and AI-based image inspection methods, can be applied to inspect bridges, the rationale to use these techniques relies on bridge inspectors' experience. This study focuses on developing a reinforcement learning-based method to assist inspectors in managing bridge inspection planning. In this method, a reinforcement learning algorithm is utilized to optimize the frequency of inspection and the selection of the inspection method. A physics-based damage development model is utilized to simulate the deterioration process of the bridge. The reward function designed in the reinforcement learning process considers both economic cost and inspection plan risk. After training, the reinforcement learning agent can rapidly determine an optimal bridge inspection policy based on a bridge's state, which can minimize both the cost and the risk of bridge inspection work. Thus, inspectors can refer to this agent to make a specific inspection plan for each bridge based on a bridge's design, history, and features.

[1]Lyles School of Civil Engineering, Purdue University, West Lafayette, IN 47907, USA; zhan3794@purdue.edu (X.Z.); salmeron@purdue.edu (M.S.); bwogen@purdue.edu (B.W.); liturbur@purdue.edu (L.I.); sdyke@purdue.edu (S.J.D)
[2]School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907, USA; liu1787@purdue.edu (X.L.)

INTRODUCTION

Bridge inspection has long been an emphasis by the Federal Highway Administration (FHWA) and the U.S. Department of Transportations (DOT) and thus researchers and engineers put plentiful effort to enhance the bridge inspection work. For example, a new bridge inspection manual Specifications for National Bridge Inventory [1] was released in May 2022 to further regulate biannual bridge inspection manners. The update for the new bridge inspection manual includes a new bridge condition evaluation procedure, with further refining the categories of bridge elements, etc. Such an update is aimed to help with consistency, robustness, and reliability of bridge inspection work. Besides the procedures, researchers are working to develop advanced bridge inspection methods, e.g., electrical resistivity [2] to support inspectors as visual inspection is subjective and highly depends on the inspectors' experience [3].

Though practitioners in the field have been working hard to improve bridge inspections, budget allocations are also constrained. The requirement to conduct biannual inspections has the primary goal of increasing the probability of finding damage so that a suitable monitoring or maintenance plan can be made. However, inspecting each bridge uniformly when an inspection is conducted every two years may not be feasible. One challenge is that different bridge types, locations, construction materials, among others, make the speed of deterioration vary considerably from bridge to bridge [4]. For example, bridges in mild environments will degrade more slowly than bridges in extreme environments when other conditions are similar. In addition, older bridges should be treated differently than newly constructed bridges. Thus, the argument can be made that bridge inspection should be scheduled dynamically based on each bridge's expected state. Manually designing such a schedule for each bridge is time-consuming due to the large number of bridges in the U.S. And there is no current standard for determining a bridge inspection schedule.

Another challenge is the criteria for adoption of suitable inspection methods. The selection of what inspection techniques may be applied in a given situation is determined by bridge inspectors, as currently there is no standard for this selection. As a result, the decision of applying those techniques is somewhat subjective. And as with the bridge inspection schedule, the adoption of a particular advanced bridge inspection method should rely on the current bridge's status. Thus, it is hard to achieve uniform standards across an inventory of bridges.

Considering the aforementioned issues plaguing current bridge inspection practices, this paper proposes a novel reinforcement learning (RL)-based approach for bridge inspection management. RL is a machine learning technique that involves an agent interacting with its environment to learn how to make optimal decisions [5]. Essentially, the agent in a RL algorithm observes the state of both the environment and itself, then takes an action based on that information. Following the action, the environment responds with a reward signal that reflects the quality of the decision made. The agent's objective is to learn a policy that maximizes the cumulative reward over time. The present study applies reinforcement learning to aid in the management of bridge inspections, encompassing both inspection scheduling and method selection. The components of a reinforcement learning algorithm - agent, environment, action, state, and reward - are deliberately designed to mirror the real-world scenario. The agent, similar to a human, will select the next inspection time and method (as action) based on current bridge state. And this action will react with the environment, which is bridge deterioration process and then the agent will obtain the reward, which is the cost spent for the selection of the inspection time and method. This ensures that the policies generated by the RL method can be applied to practical problems and yield effective solutions. The inspection of crack induced by the corrosion of rebar is selected as a representative situation with which to demonstrate the method. Due to a lack of data, analytical bridge crack development model [6] is employed to simulate the damage degradation process. The simulated damage degradation curves are utilized to train RL algorithm. After training, the results from our experiments demonstrate that the use of RL methods can optimize the bridge inspection management plan. This approach will likely save money when applied to an inventory.

This paper presents a practical application of reinforcement learning-based bridge inspection management method. The first section, Problem Identification, includes the scope of the inspection work to be solved using reinforcement learning. The second section, Technical Approach, includes the detailed setting of reinforcement learning algorithm to solve the problem presented in the first section. In the third section, Experimental Validation, the process of our experiment to validate the method will be introduced and an optimal inspection management plan will be obtained by comparing with current bridge inspection plan.

PROBLEM IDENTIFICATION

Bridge Inspection Management and Scope of Problem

Current FHWA procedures enforce a time-based bridge inspection. Under such a scheme, concrete deck bridges must be inspected every 24 months [7]. Even though several types of testing are available, such inspections are routinely visual [4]. Several researchers have pointed out the need for a new rationale to determine the frequency and type of inspections based on several factors including: the scope of the inspection, the condition of the bridge, and its estimated remaining life of service [8]. Despite the number of research studies proposing optimal inspection scheduling schemes [9-11], a lack of standards and specifications for such methodologies restrains their implementation.

This study offers a novel RL approach to determine the appropriate inspection schedule. The method, based on the use of a physics-based model for current state assessment and future states prognosis, can decide both the best time and type of inspection. The model is focused on common behavior of concrete decks. Chloride-induced corrosion, and the cracking associated with it, are deemed the most concerning type of deterioration for concrete decks, especially in coastal environments [12] and cold regions where deicing salt usage is necessary during winter [13]. The rust accumulation causes the eventual cracking of the surface of the deck. Such surface cracking is a critical parameter that is used to assign a condition ranking during the inspection, thus here it is viewed as important in the decision-making process for inspection scheduling. In the following, we will focus on arranging the inspection for this specific type of damage.

Crack Development Model

Predicting crack development in concrete is challenging due to its heterogeneous nature and uncertain fabrication process. Probability-based models [14-15] are commonly used but lack generalizability to new specimens. Analytical models offer an alternative, relying on concrete's physical properties, elasticity, and fracture mechanics theories. Uncertainty can be incorporated by randomizing model parameters.

The analytical model from [6] is employed, assuming concrete with embedded reinforcing steel as a thick-walled cylinder with a thin layer of pores covering the rebar, where r stands for the distance from the center of the rebar to an arbitrary point inside the cylinder, a is the distance from the center to the outer diameter of the pore layer, and b is the distance from the center to the cylinder's outer diameter, which is the concrete cover.

The cracking of the deck is divided into three phases: no cracking, partial cracking, and complete cracking. In the first phase, a ring of rust products starts to form around the rebar as shown in Figure 1 (b). The weight of this ring of rust is given by:

$$W_{rust}(t) = \sqrt{\frac{2k\pi D}{\alpha}[At(\ln t - 1) + Bt]} \tag{1}$$

where A and B are coefficients for the chosen corrosion law [16], k is a coefficient for units change, D is the rebar diameter, and α is the ratio of the molar masses of steel to rust products. The ring of rust exerts pressure on the concrete around the rebar, developing radial and tangential stresses according to the theory of elasticity.

The second cracking stage starts when the tangential stress at r=a becomes higher than the concrete's tensile strength. When this happens, a crack with tip at $r = r_0$ develops, as shown by Figure 1 (c). The cylinder is then divided into two sections: an inner, cracked circumference, and an outer, uncracked one, and the elastic stresses given by Equations (1) and (2) are no longer valid. The details on the calculation of r_0 can be found in [6] and [17]. Finally, the crack tip, determined by r_0, reaches the surface of the deck (Figure 1 (d)). After this point, the crack width in the surface becomes larger than 0 and is computed with the following Equation 2 [6]

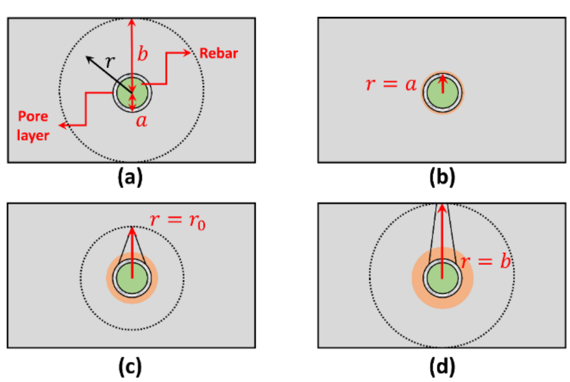


Figure 1. (a) Thick cylinder assumption. (b) No cracking. (c) Partial cracking. (d) Complete cracking.

$$\omega(t) = \frac{4\pi d_s(t)}{(1-\nu)\left(\frac{a}{b}\right)^{\sqrt{\alpha}} + (1+\nu)\left(\frac{b}{a}\right)^{\sqrt{\alpha}}} - \frac{2\pi b f_t}{E} \quad (2)$$

where α is a reduction factor for the tangential stiffness of the concrete (details in [17])

TECHNICAL APPROACH

Reinforcement Learning Introduction

RL is a machine learning technique used to solve intricate decision-making problems across various domains, including gaming, recommendation systems, and robotics. In RL, an agent learns to make optimal decisions by interacting with an environment through trial and error. The RL framework comprises of an agent, environment, state, action, and reward [18]. The agent can be a software program, robot, or any other system that receives input from the environment and takes actions. The environment can be either physical or virtual and provides feedback to the agent based on its actions. The state represents the current information of the system, and the action is the agent's decision based on the state. The reward is a scalar value that indicates whether the current action is beneficial or harmful to the agent's goal. RL provides a policy to solve the optimization problem, which maps states to actions using a set of rules, neural network, or other algorithms [19].

RL Framework for Bridge Inspection

Though RL algorithm has identical elements, the settings of those elements are different in different problems. This section will introduce how the RL algorithm is set to solve bridge inspection management optimization.

In this study, a convolutional neural network (CNN) [20] has been chosen as the primary agent. The CNN takes the state matrix of the bridge as its input and predicts the corresponding inspection action. Additionally, it also generates the corresponding value function for the input state every six months. To optimize the performance of the CNN, the weights are updated using the Stochastic Gradient Descent (SGD) [21] algorithm.

This bridge inspection management system uses three inspection methods to identify corrosion-induced cracks on bridges. The first method is AI-based image inspection [22], which provides quick and cost-effective results but is less reliable. The second method is visual inspection, which is slower and can only identify visible damage. However, the visual inspection method is assumed to be more reliable than the AI-based image inspection method because currently we think human is more reliable than AI [22]. The third method is ultrasonic wave inspection [23], which is more expensive but more reliable. The CNN agent outputs a four-length matrix indicating the probability of taking no action or using one of the three methods. The inspection method with the highest probability is chosen. The CNN also outputs the value function every six months, and its weights are updated accordingly using stochastic gradient descent.

The RL-based bridge inspection management system updates the state matrix and reward after determining the appropriate action (USW/visual inspection/AI-based image inspection) or no inspection. In this case, the environment of RL refers to the bridge environment, which determines the development of the crack. The bridge environment considers material constants, crack development speed, and maintenance actions. In this work, the model in the Crack Development Theory section is used to simulate the bridge's crack development process. Because each bridge has a unique environment, uncertainties are introduced to simulate a specific environment for each bridge. The uncertainties are involved in three aspects. The first uncertainty is material constant variability, which has been studied by others [24]. The second uncertainty is due to damage development, which may differ depending on the area's freeze-thaw cycle, temperature, deicing salt usage, etc. [25]. The third uncertainty in this work is from maintenance. Since there is no research studying the impact of bridge maintenance on damage, we assume the effect of bridge maintenance has uncertainty. With these uncertainties, different bridges will have their own crack development model with specific parameters. However, the exact crack development curve are unknown during the training, and inspectors can only estimate them using inspection data to obtain an estimated bridge crack development model. After an action is performed, the state matrix will be updated differently based on the specific crack development model.

The state matrix for the bridge includes structural information, damage status, and inspection history. It consists of a size-sixteen vector that represents the current state of the bridge. The first three elements correspond to the crack development model parameters and the uncertainty of crack development. The next element records the current crack width, followed by binary indicators of time since the last inspection. The remaining elements represent the age of the bridge in binary. If no inspection is conducted, the uncertainty, crack width, and age will be updated in the state matrix. If an inspection is conducted, the new inspection data will be obtained, and the parameters of the estimated crack development model will be calculated using the true crack development model.

The objective of this RL-based bridge inspection management system is to minimize the cost spent on bridge inspection work throughout the whole bridge life. To achieve this, the reward element in RL algorithm is composed of three terms, which is shown in Equation III:

$$\begin{aligned} Reward = {} & Cost_a(a) \times Rate_{cond}(cond, a) \\ & + Cost_c(c) \times Rate_{time-risk}(t, cond) \\ & + Cost_c(c) \times Rate_{action-risk}(cond, a) \end{aligned} \tag{3}$$

The first term here is the inspection action cost, which represents the cost spent for using a selected inspection method. The inspection action cost is determined by two factors: the price of the inspection action $Cost_a$ and the condition discount $Rate_{cond}$. The reason for the introduction of the condition discount is that the cost of an inspection action may vary depending on the condition of the bridge. For instance, if a bridge is in good condition, the visual inspection conducted may take less time and therefore cost less. The second term in Eq. 3 is the time-risk cost, which is to represent the cost due to waiting a long time with no inspection. This cost is composed of the price of component $Cost_c$ and the time-risk rate $Rate_{time-risk}$. The time-risk rate will increase with the length of time without inspection and in this work a Weibull distribution [26] is selected to describe such increase. The values of parameters λ and k for Weibull function are selected differently when a bridge is in different condition to reflect different impact of lasting no inspection on different stage of a bridge. The third term in Eq. 3 is the action risk cost. This cost reflects the risk of using different inspection actions. The action risk rate $Rate_{action-risk}$ is utilized to indicate the risk of an inspection action to the bridge at different conditions.

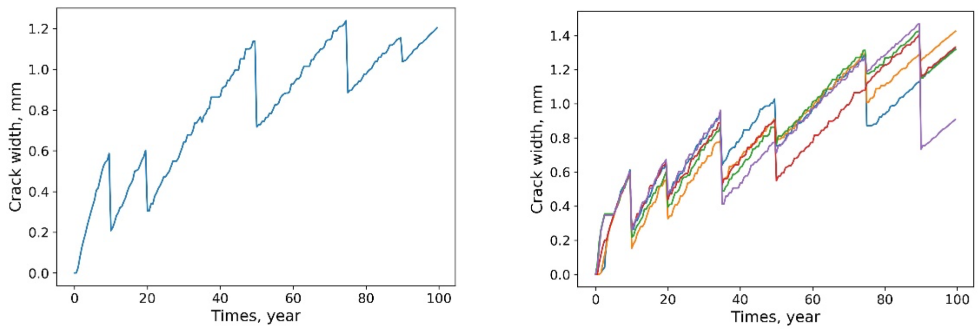
EXPERIMENTAL VALIDATION

Experiment Setting

As currently crack development data is not recorded, the simulated crack development data generated through the crack development model is utilized in this work for training the RL. As described in previous sections, three uncertainties are involved for each specific bridge.

The values and distributions of the initial crack development model are set as the following. Concrete cover of bridge deck follows normal distribution with $\mu = 63\ mm$ and $\sigma = 3.78\ mm$ [27]. Rebar diameter D follows lognormal distribution with $\mu = 19.05\ mm$ and $\sigma = 0.03 \times 19.05\ mm$ [24]. α_{rust} is a fixed value which is equal to 0.57 [6]. The initial values A and B for corrosion rate are set as exponentially distributed and the average values used for these exponential distributions are 0.3686 $\mu A/cm^2$ and 1.1305 $\mu A/cm^2$, respectively [6]. ρ_{rust} and ρ_{steel} are 0.36 and 0.785, respectively [6]. d_0 is assumed to be followed exponential distribution with average value is 0.0125 mm [6]. Tensile strength of the concrete, f_t follows a lognormal distribution with $\mu = 3.56\ MPa$ and $\sigma = 0.676\ MPa$ [27]. v_c is assumed to be exponentially distributed with an average value of 0.18 [6]. E_{ef} follows a normal distribution with $\mu = 30\ GPa$ and $\sigma = 3\ GPa$ [28]. γ used in the crack development equations is set as 8000.

The crack development uncertainty is set as 2% to start. And every half year, the simulated crack width is randomly generated based on a Gaussian distribution $X \sim N(w_{predicted}, uncertainty\_crack \times w_{predicted})$. In the distribution, $w_{predicted}$ is the predicted crack width from current crack development model. The crack development model's A and B parameters are updated by curve fitting after new crack width information is obtained. Uncertainty decreases 0.002% every 0.5 year, assuming crack development becomes more predictable with time. Maintenance is conducted after 10, 20, 35, 50, 75, and 90 years, reducing crack width by 30%-75%. Maintenance doesn't affect crack development, as we are assuming it doesn't affect rust thickness. When the bridge deck is over 100 years old, or the condition rating has dropped to 3, the bridge deck is replaced. A possible crack development curve of one bridge is shown in Figure 2 (1), with different simulated crack development curves in Figure 2 (2).



(1) (2)

Figure 2. Simulated crack development of (1) one bridge; (2) several bridges

In this work, we manually assign different values to the cost in the reward function due to the lack of relevant data recording regional variations. The action price for the USW, the visual inspection and the AI-based image inspection method are \$120, \$50, and \$10 per time, respectively. The price of the condition discount $Rate_{cond}$ is defined in TABLE I. The time risk rate $Rate_{time-risk}$ is defined in TABLE II and the action risk rate $Rate_{action-risk}$ is defined in TABLE III.

During training, an epsilon-greedy method [29] balances exploration-exploitation. The learning rate is 0.0001, the long-term benefits discount factor is 0.95, and training consists of 60,000 epochs. The RL environment is built using Tensorflow [30] and Keras [31], and training is performed using a NVIDIA GPU GeForce GTX Titan X.

Results

Under the settings introduced in previous sections, the RL algorithm can be trained. The training process is shown in Figure 3.

TABLE I. VALUES DEFINED FOR $Rate_{cond}$

Condition Rating	**9**	**8**	**7**	**6**	**5**	**4**	**3**
USW	0.7	0.8	0.9	0.9	1.0	1.0	1.0
Visual Inspection	0.4	0.4	0.5	0.7	1.0	1.0	1.0
AI-based method	0.7	0.8	0.9	0.9	1.0	1.0	1.0

TABLE II. VALUES FOR WEIBULL FUNCTION IN $Rate_{time-risk}$

Condition Rating	**9**	**8**	**7**	**6**	**5**	**4**	**3**
λ	40	20	15	10	5	3	3
k	3	3	3	3	3	3	3

TABLE III. VALUES DEFINED FOR $Rate_{action-risk}$

Condition Rating	**9**	**8**	**7**	**6**	**5**	**4**	**3**
USW	0.001	0.001	0.001	0.002	0.003	0.003	0.003
Visual Inspection	0.002	0.004	0.006	0.010	0.030	0.06	0.1
AI-based method	0.002	0.004	0.009	0.020	0.05	0.35	0.4

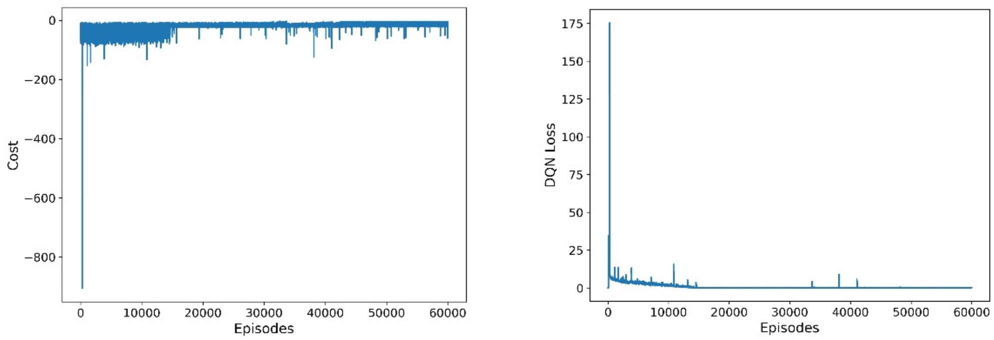


Figure 3. Training process. (1) Inspection cost; (2) DQN loss

The trained agent can make inspection decisions based on the current state of the bridge. For example, if a bridge deck has been in service for two years, with an estimated crack development model of A=0.1287 $\mu A/cm^2$ and B=1.3747 $\mu A/cm^2$, a crack width of 0.0866 mm, and a crack development uncertainty of 1.97%, the AI-based image inspection method will be recommended for inspection. However, for a bridge deck that has been in service for 69 years, with an estimated crack development model of A=1.0 $\mu A/cm^2$ and B=1.0 $\mu A/cm^2$, a crack width of 0.937 mm, and a crack development uncertainty of 0.63%, the USW method is recommended, even if the latest inspection was just six months earlier. The RL-based bridge inspection management plan tends to reduce inspection frequency and use cheaper methods for bridges in good condition and increase inspection frequency and use the USW method for bridges in poor condition. The plan can save approximately $10,000 under the simulated bridge crack development process as compared to time-based bridge inspection management plans based on tests of 1,000 simulated bridges.

CONCLUSION

In this paper, the bridge inspection management work has been encoded as an RL problem, and then RL techniques have been utilized and trained to help in this work. The conclusions of this work include:

- In our experiment, the RL method has been successfully trained. The trained RL agent can rapidly determine the inspection schedule based on the current bridge state. The bridge inspection plan generated by the RL method can save around $10,000 cost compared to current time-based bridge inspection plan.
- To improve the RL method's performance in generating better inspection plans, we found that more data is needed, as real-world data is lacking. These data should include material constants and their uncertainties, precise crack development curves for each bridge, the cost and risk associated with different inspection actions and bridge conditions, and the impact of maintenance on damage.
- The bridge inspection work includes other damage and bridge elements and thus, a more complex setting should be applied for future RL-based bridge inspection management research.

REFERENCES

1. Specifications for the National Bridge Inventory. Available online: https://www.fhwa.dot.gov/bridge/snbi.cfm (accessed on April 2023).
2. Gucunski, N., Nazarian, S., Imani, A., & Azari, H. (2014). Performance of NDT technologies in detection and characterization of reinforced concrete deck deterioration. In *Geo-Congress 2014: Geo-characterization and Modeling for Sustainability* (pp. 2436-2449).
3. Bu, G., Lee, J., Guan, H., Blumenstein, M., & Loo, Y. C. (2014). Development of an integrated method for probabilistic bridge-deterioration modeling. *Journal of Performance of Constructed Facilities*, 28(2), 330-340.
4. Abdallah, A. M., Atadero, R. A., & Ozbek, M. E. (2022). A state-of-the-art review of bridge inspection planning: Current situation and future needs. *Journal of Bridge Engineering*, 27(2), 03121001.
5. Wiering, M. A., & Van Otterlo, M. (2012). Reinforcement learning. *Adaptation, learning, and optimization*, 12(3), 729.
6. Chun-Qing, L., Melchers, R. E., & Jian-Jun, Z. (2006). Analytical model for corrosion-induced crack width in reinforced concrete structures. *ACI structural journal*, 103(4), 479.
7. FHWA, F. (2009). National bridge inspection standards (nbis). Federal Register, 69, 239.
8. Everett, T. D., Weykamp, P., Capers Jr, H. A., Cox, W. R., Drda, T. S., Hummel, L., ... & Washer, G. A. (2008). *Bridge evaluation quality assurance in Europe* (No. FHWA-PL-08-016).
9. Chung, H. Y., Manuel, L., & Frank, K. H. (2006). Optimal inspection scheduling of steel bridges using nondestructive testing techniques. *Journal of Bridge Engineering*, 11(3), 305-319.
10. Chung, H. Y. (2004). *Fatigue reliability and optimal inspection strategies for steel bridges*. The University of Texas at Austin.
11. Ilbeigi, M., & Pawar, B. (2020). A probabilistic model for optimal bridge inspection interval. *Infrastructures,* 5(6), 47.
12. James, A., Bazarchi, E., Chiniforush, A. A., Aghdam, P. P., Hosseini, M. R., Akbarnezhad, A., ... & Ghodoosi, F. (2019). Rebar corrosion detection, protection, and rehabilitation of reinforced concrete structures in coastal environments: A review. *Construction and Building Materials,* 224, 1026-1039.
13. Roelfstra, G., Hajdin, R., Adey, B., & Brühwiler, E. (2004). Condition evolution in bridge management systems and corrosion-induced deterioration. *Journal of Bridge Engineering*, 9(3), 268-277.
14. Vu, K. A., & Stewart, M. G. (2005). Predicting the likelihood and extent of reinforced concrete corrosion-induced cracking. *Journal of structural engineering*, 131(11), 1681-1689.
15. Riding, K. A., Poole, J. L., Schindler, A. K., Juenger, M. C., & Folliard, K. J. (2014). Statistical determination of cracking probability for mass concrete. *Journal of materials in civil engineering*, 26(9), 04014058.
16. Liu, T., & Weyers, R. W. (1998). Modeling the dynamic corrosion process in chloride contaminated concrete structures. *Cement and Concrete research*, 28(3), 365-379.
17. Li, C. Q., Zheng, J. J., Lawanwisut, W., & Melchers, R. E. (2007). Concrete delamination caused by steel reinforcement corrosion. *Journal of Materials in Civil Engineering*, 19(7), 591-600.
18. Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.
19. François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., & Pineau, J. (2018). An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning*, 11(3-4), 219-354.
20. O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:*1511.08458.
21. Ketkar, N., & Ketkar, N. (2017). Stochastic gradient descent. *Deep learning with Python: A hands-on introduction*, 113-132.
22. Zhang, X., Wogen, B. E., Liu, X., Iturburu, L., Salmeron, M., Dyke, S. J., ... & Ramirez, J. A. (2023). Machine-Aided Bridge Deck Crack Condition State Assessment Using Artificial Intelligence. *Sensors*, 23(9), 4192.
23. Lee, S., Kalos, N., & Shin, D. H. (2014). Non-destructive testing methods in the US for bridge inspection and maintenance. *KSCE Journal of civil engineering*, 18, 1322-1331.
24. Enright, M. P., & Frangopol, D. M. (1998). Service-life prediction of deteriorating concrete bridges. *Journal of Structural engineering*, *124*(3), 309-317.
25. Salmeron, M., Criner, N. M., Zhang, X., Dyke, S. J., Ramirez, J., Wogen, B., & Rearick, A. (2022). Physics-based modelling of construction defects in concrete decks. *In Bridge Safety, Maintenance, Management, Life-Cycle, Resilience and Sustainability* (pp. 200-206). CRC Press.

26. Garavaglia, E., Pavani, R., & Sgambi, L. (2021). The Use of Fragility Curves in the Life-Cycle Assessment of Deteriorating Bridge Structures. *Computation*, 9(3), 25.
27. Jaskulski, R., & Wiliński, P. (2016). Influence of concrete strength probability distribution on safety margin of concrete cross-section subjected to shear. *Procedia Engineering,* 153, 232-239.
28. Matos, J. C., Valente, I., & Cruz, P. J. (2010). Uncertainty evaluation of reinforced concrete structures behavior.
29. Wunder, M., Littman, M. L., & Babes, M. (2010). Classes of multiagent q-learning dynamics with epsilon-greedy exploration. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)* (pp. 1167-1174).
30. Pang, B., Nijkamp, E., & Wu, Y. N. (2020). Deep learning with tensorflow: A review. *Journal of Educational and Behavioral Statistics*, 45(2), 227-248.
31. Ketkar, N., & Ketkar, N. (2017). Introduction to keras. *Deep learning with python: a hands-on introduction*, 97-111.