
**Infrastructure Asset Management**

# Data-driven estimation of deterioration curves: a railway supporting structures case study

INAM-2021-006 | Paper

Submitted by: Saviz Moghtadernejad, Gerald Huber, Juergen Hackl, Bryan Adey

Keywords: ASSET FAILURE AND ANALYSIS, DATA, INFORMATION AND KNOWLEDGE MANAGEMENT, MAINTENANCE AND INSPECTION

RIVER VALLEY
TECHNOLOGIES

# Data-driven estimation of deterioration curves: a railway supporting structures case study

Saviz Moghtadernejad, PhD
Postdoctoral Fellow, Institute of Construction and Infrastructure Management, ETH Zurich, Zurich, Switzerland.
(Corresponding author: moghtadernejad@ibi.baug.ethz.ch.)
Orcid: 0000-0001-9860-5523

Gérald Huber, MSc
Institute of Construction and Infrastructure Management, ETH Zurich, Zurich, Switzerland.

Jürgen Hackl, PhD
Assistant Professor, School of Engineering, University of Liverpool, Liverpool, UK.
Orcid: 0000-0002-8849-5751

Bryan T. Adey, PhD
Professor and Head of the Infrastructure Management Group, Institute of Construction and Infrastructure Management, ETH Zurich, Zurich, Switzerland.

**Abstract:** A significant portion of railway network income is spent on the maintenance and restoration of the railway infrastructure to ensure that the networks continue to provide the expected level of service. The execution of the interventions, i.e. when and where to perform maintenance or restoration activities, depends on how the state of the infrastructure assets changes over time. Such information helps ensure that appropriate interventions are selected to reduce the deterioration speed and to maximize the effect of the expenditure on monitoring, maintenance, repair, and renewal of the assets. Presently, there is an explosion of effort in the investigation and use of data-driven methods to estimate deterioration curves. However, real-world time history data normally includes measurement errors and discrepancies that should not be neglected. These errors include missing information, discrepancies in input data, and changes in the condition rating scheme. This paper provides solutions for addressing these issues using machine learning algorithms and estimates the deterioration curves for railway supporting structures using Markov models and discusses the results.

**Keywords:** Maintenance & inspection; Data; Information & Knowledge management; Asset failure & analysis

## Notation

| | |
|---|---|
| $\Delta t$ | Time intervals |
| $\mu$ | Mean value |
| C(t) | The regression curve as a function of time |
| CS | Condition state |
| E(t) | The expected CS at time *t* based on the Markov chain and the estimated probabilities |
| P | Transition probability matrix |
| $P_{ij}$ | Probability of transition from state *i* to *j* |
| S | Condition state vector |
| $\sigma$ | Standard deviation |

## 1 Introduction

Modern societies depend on well-functioning transportation infrastructure. As infrastructure continually deteriorates, stakeholders have to be able to accurately predict its deterioration speed to determine the optimal maintenance programs. Although there are different types of models used for this purpose (Setunge and Hasan, 2011), Markov and semi-Markov models, have perhaps been used for management purposes the most extensively (Nam, 2009; Kobayashi, Kaito and Lethanh, 2012). For example, Madanat et al. (1995), Robelin and Madanat (2007), and Setunge and Hasan (2011) used Markov models to predict bridge deterioration curves, and Ortiz-Garcia et al. (2006) used them to model the pavement deterioration. Manafpour et al. (2018) used a semi-Markov time-based model to model concrete bridge deck deterioration and Edirisinghe et al. (2015) predicted the building deterioration using a Markov model.

In general, the Markov models use data from inspections of the condition state of the infrastructure over time, to estimate the deterioration curves. Consequently, the quality of data plays a significant role in the accuracy of deterioration prediction and the resulting lifecycle cost estimations. However, despite the recent progress in more frequent and accurate monitoring of the assets and storage of the related results, in practice, real-world data often does not exist in sufficient quantity, contains errors and discrepancies, and is not always suitable for estimating accurate transition probabilities. These issues and errors often result from not archiving the results of past inspections (lack of history), missing information or faulty entries, lack of a robust guideline for condition assessment, the discrepancies between the judgment of inspectors, and the measurement errors related to machines, equipment, sensors, etc.

Researchers have also done work to bridge these gaps for Markov models. Specifically, Mizutani et al. (2017) suggested improving the estimation of Markov transition probabilities using mechanistic-empirical models and Lethanh et al. (2017) used these models along with Monte Carlo simulations to estimate the transition probabilities for a reinforced concrete bridge element with chloride-induced corrosion. Humplick (1992), has studied methods to tackle the issues of measurement errors related to monitoring equipment and measurement locations. Park et al. (2008) and Hong and Prozzi (2006) have used a Bayesian approach to deal with the small population samples for pavement deterioration prediction. Chu and Durango-Cohen (2007, 2008) used Kalman filters to eliminate errors in pressure and deflection measurements for asphalt pavements and provided numerical examples to demonstrate how their framework accommodated the missing values. In general, most issues related to the errors in data, that are addressed in the literature are related to the equipment, item or location related errors, or measurement errors specified to humans such as faulty entry, discrepancies in the judgments of the inspectors (Kobayashi *et al.,* 2012). There are occasions, however, where there is a systematic discrepancy among the data entries, such as changes in the condition rating system, that have not been addressed in the literature.

This paper contributes to the literature by examining a real-world case study to predict the deterioration curves of the railway supporting structures using Markov models. In this study, state-of-the-art tools were used to clean the data and deal with the faulty/incomplete entries. Moreover, three classification algorithms, i.e. K-Nearest Neighbors algorithms (KNN), Neural Networks (NN), and random forest algorithms were used to adjust a portion of the data collected using an old condition rating scheme to the equivalents with a new condition rating scheme.

The structure of the paper is as follows: Section 2 provides an overview of the study. In section 3 the procedure for data preparation and structuring is discussed. Section 4 introduces the methodology to estimate the transition probabilities. The dwell times for different categories of supporting structures are estimated in section 5, and section 6 discusses the results. Finally, the summary and conclusions of the study are presented in section 7.

## 2 Study description

This paper developed the deterioration curves for railway supporting structures, from a data set with faulty/incomplete entries, inaccuracies related to the inconsistent monitoring programs and biased data, as well as the situation where there were changes in the condition state rating system. The supporting structures in this study were bridges and retaining walls that laterally support soil to restrain it at different levels on the two sides.

An inventory of the assets containing 4’988 bridges and 17’000 retaining walls were created in the years 1983 and 2000 respectively, and the results of regular inspections were entered in a database following that date. In the asset inventory, each object was assigned a unique identification number, and the document provided information on the type of construction and materials, the dimensions, the position of the object, the construction year, and the results of the inspections. The bridges were classified into three categories of masonry, steel, concrete, and composite. For the retaining walls, from a static point of view, they were divided into three categories: gravity walls that hold back the earth pressure with their weight alone, cantilever walls, and anchored walls. Material-wise they were divided into three categories: masonry, concrete, and natural stone walls. For all structures, the inspections were performed every six years on average, which resulted in a total of 26’106 status reports from 1983 to December 2018 for bridges, and 52’647 status reports from the year 2000 to February 2020 for the retaining walls. These status reports indicated the condition state of the objects at the time of inspection. The first change in the condition rating scheme occurred in 2009 and affected all objects. The second change occurred in 2013 and affected only natural stone retaining walls (see Figure 1).

The steps used to develop the deterioration curves are as follows. In the first step, the data was cleaned and prepared for analysis. In the second step, the transition probabilities were estimated. The deterioration curves and the dwell times, i.e. the duration that the structures stay in each condition state, were approximated using the transition probabilities and Monte-Carlo simulations. The following sections discuss these steps in more detail.

## 3 Data preparation

### *3.1 Data cleaning*

In the first step, the data was cleaned and structured to be used for the estimation of the transition probabilities. The faulty/incomplete entries were first corrected or deleted if unusable; with the aim of keeping as many entries as possible to have the most informative value for the models.

### *3.2 Dealing with changing rating schemes*

In the second step, the problem of the variations in the condition rating scheme was addressed. Four condition states were used prior to 2009 for all object types, and five condition states afterwards. Additionally, the classification criteria for natural stone walls became stricter in 2013, meaning that worse condition states were assigned to objects after 2013 than before. Hence, condition states before and after 2009 and 2013 could not be directly compared.

This problem was addressed by reclassifying the inspection results (condition states) that happened before the changes in the rating system. As this could be done in different ways, the performance of three classification algorithms, KNN, NN, and the random forest were compared together to adjust the past condition states to the currently used condition states. The work was based on the idea that the condition states could be estimated using other information available in the database, such as wall type and material, construction year, and damage type. The condition states of the natural stone retaining walls before 2009 were first adjusted for the first rating change, and then again for the second rating change in 2013. Brief overviews of the different classification methods that were used are provided in the appendix.

As can be seen in Figure 1, for bridge inspections before 2009, bridges with the condition state (CS) 1 and 2 maintained the same CS, those with CS3 could either be CS3 or CS4, and bridges with a CS4 would be classified as CS5. For the retaining wall inspections before 2009, walls with the condition state (CS) 1 maintained the same CS, those with CS2 could either be CS2 or CS3, walls with CS3 could be either CS3 or CS4, and walls with a CS4 would be classified as CS5. Additionally, for stone walls, condition states reported before 2013, were treated as follows. Walls with CS1 were still considered as CS1. Walls with the CS2 could be either CS2, 3, or 4, and walls with CS3 could be converted into CS3, 4, or 5. This meant that a total of five reclassifications were required:

1) Reclassify the old CS3 (all bridges before 2009)
2) Reclassify the old CS2 (all walls before 2009)
3) Reclassify the old CS3 (all walls before 2009)
4) Reclassify the old CS2 (natural stone walls before 2013)
5) Reclassify the old CS3 (natural stone walls before 2013)

After an initial assessment of the number of observations for each CS, it was observed that the distribution of data entries for each CS, for both bridges and walls, was very skewed and unbalanced. For example, the data record for the retaining walls contained 12,394 status entries of CS2 and 2,979 status entries of CS3. When dealing with an unbalanced dataset such as this one, the classification algorithms predict the CS too optimistically, since there are many more entries in CS2 than in CS3. This is because the frequency of entries in each CS is learned in the training dataset, such that the skewed distribution is reflected in the predicted CSs. To ensure that the classification only takes place based on the informative value of the features and not the relative frequency of occurrence of the individual CSs, the training dataset was modified (augmented) to have the same number of entries for each CS.

Two methods of data augmentation namely oversampling and Synthetic Minority Oversampling Technique (SMOTE) were used to deal with the problem of the skewed dataset. In oversampling, copies of the features of the minority class(es) were created until the number of entries in the minority class(es) were the same as the class with the highest number of data entries. SMOTE synthesized new entries for the minority class(es) rather than duplicating them. This algorithm uses the concept of the KNN, and selects data points in the minority class that are close in the feature space, draws a line between the data points, and adds a new entry at a point along that line. Figure 2 provides an insight into how the new data entry is generated.

For each of the algorithms, $\frac{2}{3}$ of the data points were used as the training set and $\frac{1}{3}$ as the evaluation set, and the features (input values) were normalized. The choice of features was tailored to each classification algorithm, i.e. with the random forest and the NN, all available features (related to the damage type, material, wall type, the year of construction, and the distance to the track axis) were used, as these classification algorithms use a weighting scheme for the features and as sufficient data points were available, their performance was not negatively affected when all features were used. With the KNN algorithm, a feature selection algorithm was first applied to only consider the most meaningful features. The reason is that the KNN algorithm can achieve higher accuracy when there are fewer features involved. In general, to reduce the number of features, either a Principal Component Analysis (PCA) or a “Feature Selection” procedure is carried out. In this study, a “backward elimination” technique for feature selection was used and the most significant features (i.e. the construction year, material, and damage mechanism) were selected to be used in the KNN classifier. These features were those that correlated most strongly with the condition states. For bridges, features based on the material (masonry, steel, concrete, composite), features based on the damage type (corrosion, damage to the cover, damage to the support structure), and the construction year had the highest correlations with the condition states. For retaining walls, features based on the material (natural stone and reinforced concrete), features based on wall categories (gravity walls, cantilever walls and anchored walls);

features based on the damage type (damage to the cover and damage to the support structure); and the construction year had the highest correlations with the condition states.

In the next step, a so-called "parameter tuning" was carried out for all classification algorithms, and the combination of parameters that resulted in the highest $f_1$ score was selected for each algorithm. The $f_1$ score is a measure that represents how good the classifier is performing, and is calculated as:

$$f_1 = 2 * \frac{(Recall * Precision)}{Recall + Precision} \quad [1]$$

where in a classification problem with two classes of positive and negative, precision is the ratio of the true positive observations to the total predicted positive observations; and recall is the ratio of the true positive observations to the sum of true positive and false negative observations (i.e. all observations in actual positive class). The results of the best combinations of parameters for each algorithm are summarized in Table 1.

Table 2 summarizes the performance of the algorithms on the evaluation set for each algorithm with original data (normal), and the data augmented with oversampling and SMOTE. It can be observed from Table 2, that the random forest algorithm delivered better classification results, followed by the NN and the KNN algorithm. Also, data augmented with oversampling led to better performance of the classifiers in comparison to SMOTE and the original training set (Normal). Figure 3 provides the results of the four classification tasks. The percentages on the arrows show the proportion of the data with the old rating scheme that is transferred to a new CS. For example, for retaining walls, for the inspections that were carried out before 2009, 69.6% of the CS2 states maintained the same CS, while 30.4% were demoted to CS3 according to the new condition rating scheme.

In the next step, the data was further refined and structured for the development of the deterioration curves for each category of bridges and retaining walls.

### *3.3 Data structuring*

Figure 4 illustrates the recorded condition states of the bridges and retaining walls after the data cleaning and preparation. The x-axis shows the condition state of the objects and the y-axis shows the number of observations (each data point represents an inspection result). After data preparation, a total of 20'022 (out of 26'106) status entries for bridges and 24'479 (out of 52'647) status entries for retaining walls remained. This shows that the retaining walls had lost more than half of their observations due to duplicated or incomplete entries in the retaining wall data inventory. Moreover, most of the observations belong to the first three CSs and only 3.15% and 0.3% of them are dedicated to CS4 and CS5 respectively.

The bridges and walls were then divided into suitable categories to develop more accurate and informative data-based decay curves. The aim was to group the objects based on the distinct features that could significantly influence their deterioration rate; while keeping the number of groups as small as possible to ensure sufficient data points per category. Figures 5 and 6 show the categories of bridges and retaining walls respectively along with the number of observations per CS for each category of objects.

In the next step, a sequence of the CS transitions was created for each bridge and retaining wall based on the results of the inspections, to develop the transition probabilities. These sequences could not contain improvements in the condition state. However, if all objects that were ever repaired were filtered out, too many data points would be lost. To avoid this issue, the sequences of CS transitions were adjusted as follows:

- If there was an improvement in the CS of an object and this observation was confirmed twice in succession, it was assumed that the object was renewed. Consequently, the sequence was divided into two separate ones as if the second sequence belongs to a newly constructed object. For example, the wall $i$ with condition state sequence $CS_i = \{1,1,2,3,1,1,2\}$, was split into two sequences $CS_i = \{1,1,2,3\}$ and $CS_{i+1} = \{1,1,2\}$.

- In a sequence, where the improvement in the CS only happened once and then it returned to a worse state, such as in $CS_i = \{1,2,1,2,3\}$, the improvement is considered as an inconsistency in the judgment of the inspectors and hence it is corrected as $CS_i = \{1,2,2,2,3\}$.

A further issue with the current dataset was related to the lack of history as the database only contained the results of inspections from the year 1983 to 2018 for bridges and 2000 to 2020 for the retaining walls. This denotes that information on the condition development of the objects before this period was missing. For example, a retaining wall was built in 1970 but was only added to the database in 2006 with a CS1. In 2010, a degradation to CS2 was reported. However, it is unknown whether between 1970 and 2006, the wall stayed in CS1, or underwent one or more condition improvement interventions. Hence, there is a lower bound of 4, and an upper bound of 40 years for the dwell time in CS1, depending on the condition history of this wall. Since assuming that no interventions were executed before the first inspection would result in optimistic and non-conservative results, the period between the year of construction and the initial inspection was not taken into account. This approach, although conservative, allowed for keeping the entries without the construction year as the data analysis could be carried out using Markov models. Figures 7 and 8 illustrate the recorded condition state transitions for each category of bridges and walls.

## 4 Transition probabilities

Markov models were used to estimate the deterioration curves. Markov models are stochastic processes that provide predictions of the future development of a process with limited knowledge of its history. Hence, they were very suitable for modeling the deterioration curves of the railway supporting structures (i.e bridges and retaining walls), as the information on the development of the condition states was only available from the initial inspection of the objects, and not from the year of construction. In these models, the probability of transition to a future state $\mathrm{X}_{n+1}$ depends only on the current state $\mathrm{X}_n$, and not on the previous states ($\mathrm{X}_{n-1}$, $\mathrm{X}_{n-2}$, ...) (Parzen, 1962). The future state of an event is estimated using the probability of transition from one state to another over multiple discrete intervals. As a convention, these transitions are time-homogeneous; meaning that the probability of transition from one state to another remains constant throughout the time (Howard, 1971). Such transition probabilities are represented by a $n \times n$ matrix, where $n$ is the number of possible states. Hence, the transition matrices for the bridges and retaining walls with 5 possible condition states, would be $5 \times 5$ matrices.

The elements in the transition probability matrix $p_{ij}$ indicate the probability that the object changes from initial state $i$ to state $j$ within a certain time interval. Since an object can only be in one condition state at any point in time, the sum of each row $i \in \{1,2,3,4,5\}$ of the matrix should be equal to 1. If a transition from $CS_i$ to $CS_j$ is not possible, then the corresponding element in the matrix $p_{ij} = 0$. Consequently, the bridge and wall deterioration transition matrices would be upper triangular (Eq. 2) since, in a natural deterioration process, there cannot be improvements in the condition states (without interventions).

$$P = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} & p_{15} \\ 0 & p_{22} & p_{23} & p_{24} & p_{25} \\ 0 & 0 & p_{33} & p_{34} & p_{35} \\ 0 & 0 & 0 & p_{44} & p_{45} \\ 0 & 0 & 0 & 0 & p_{55} \end{pmatrix} \quad [2]$$

A state $CS_i$ is called absorbing when it can no longer be left, i.e. when $p_{ii} = 1$. In the case of the deterioration process for bridges and retaining walls, state CS5 is an absorbing state, as there is no way to get out of this state without taking repairing measures.

A regression-based optimization was used to estimate the transition probabilities (Roelfstra *et al.*, 2004). The objective function of this approach (Eq.3) is to minimize the sum of the absolute differences between the condition state at time $t$ based on the regression curve $C(t)$, and the expected CS at time $t$ based on the Markov chain and the estimated probabilities, $E(t)$ (Bulusu and Sinha, 1997).

$$\min Z = \sum_{t=1}^{N} |C(t) - E(t)| \quad [3]$$

$$E(t) = P(t) \times S \quad [4]$$

where $N$ is the total number of the transition periods and $S$ is the condition state vector. Eq. 4.a and Eq. 4.b present the constraints that must be taken into account.

$$0 \le p_{ij} \le 1 \quad [4.a]$$

$$\sum_{j} p_{ij} = 1 \qquad [4.b]$$

For the bridges and retaining walls, the expected CS at time $t$, i.e. $E(t)$, can be expressed as:

$$E(t) = x(t-1) \times p_{ij} \qquad [5]$$

Assuming $x_{ij}$ denotes the proportion of objects in $CS_j$ at time interval $i$, the objective function of the optimization problem in Eq. 3 can be written as:

$$\min Z = \begin{bmatrix} x_{11} & \cdots & x_{1j} & 0 & \cdots & & 0 & \cdots \\ \vdots & & \vdots & 0 & \cdots & & 0 & \cdots \\ x_{(i-1)1} & \cdots & x_{(i-1)j} & 0 & \cdots & & 0 & \cdots \\ 0 & 0 & 0 & x_{11} & \cdots & x_{1j} & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & & \vdots & 0 & \cdots \\ & & & x_{(i-1)1} & \cdots & x_{(i-1)j} & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & x_{11} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} p_{11} \\ \vdots \\ p_{j1} \\ p_{12} \\ \vdots \\ p_{j2} \\ \vdots \\ p_{1i} \\ \vdots \\ p_{ji} \end{bmatrix} - \begin{bmatrix} x_{21} \\ \vdots \\ x_{i1} \\ x_{22} \\ \vdots \\ x_{i2} \\ \vdots \\ x_{2j} \\ \vdots \\ x_{ij} \end{bmatrix} \qquad [6]$$

The matrix $x(t-1)$ has a dimension of $[(i-1) * j; j^2]$, the vector of the transition probabilities $p_{ij}$ has a dimension of $[j^2; 1]$, and the vector $x(t)$ has a dimension of $[(i-1) * j; 1]$.

The process of estimating the transition probabilities starts with data aggregation within a certain time interval $\Delta t$. For the bridges since the age of the objects were known, the $\Delta t$ was selected as 5 over a period of 100 years (20 time intervals), and then the proportion of the number of observations in each condition state was calculated in each time interval, considering the age of the bridge. For the walls, as the information regarding the year of construction was unknown for almost half of the walls, the time of the first inspection for all walls was considered as $t_0$ and only the time differences between the inspections were considered. As the inspections in the database were conducted from the year 2000 to 2020, 20 time intervals ($\Delta t$ =1) were selected for data aggregation to estimate the transition probabilities for each category of retaining walls.

In this problem, $i \in \{1,2,3,\ldots,20\}$ denotes the interval number, and $j \in \{1,2,3,4,5\}$ represents the condition states. Subsequently, the transition probabilities were estimated using the regression-based approach described above. The optimal solutions to Eq. 6, i.e. the transition probability matrices for each category of the bridges and retaining walls are presented in Tables 3 and 4. It can be observed that due to the lack of sufficient observations in CS5, CS4 is the absorbing state for all wall categories except masonry gravity walls. Moreover, the number of observations in CS4 and CS3 were much lower than CS2 and CS1, which suggests that the bridges and the retaining walls were maintained regularly to avoid dangerous condition states.

## 5 Deterioration curves and dwell times

After data preparation and estimation of the transition probabilities, a fictitious portfolio of 12,000 objects was created for each category of bridges and retaining walls. The deterioration of the objects over time was then simulated using the estimated transition probabilities and the Monte Carlo simulations. The dwell times for each CS were then calculated, based on the results of the previous step (Figure 9 and 10). Tables 5 and 6 provide a comparison between the dwell times of bridges and retaining walls derived directly from the data (i.e. the max, min, and the mean time that each category of bridges and retaining walls was in each condition state based on the inspection data) and the estimated dwell times using the developed transition probabilities and the Monte-Carlo simulations.

## 6 Discussion

The comparison between the dwell times derived directly from inspection data and the dwell times estimated using transition probabilities and Monte-Carlo simulations shows that the wall and bridge categories with a higher number of observations produced more accurate estimations of the transition probabilities. This can especially be seen through the comparison of the dwell times for concrete and composite bridges, where the estimated dwell times are almost equal to the dwell times from inspection data (Figure 9). In these categories, although the minimum and mean dwell times are almost the same, the max dwell times from the simulations are considerably longer. Here it seems that the simulations likely provide a better reflection of reality than the data, because the max dwell times from the data are constrained due to the limited time period over which data was collected. For example, in Table 5 the max dwell time derived from data for concrete bridges in CS2 is 35.2, which is equal to the max inspection history. In contrast, the Monte Carlo simulations show a longer max dwell time, which is because the dwell time for a "fictitious" bridge is not constrained.

For the masonry bridges and steel bridges, there are discrepancies among the dwell times derived from data and those estimated using the simulations, which is mainly due to the lack of sufficient observations in these categories (Table 5). For the retaining walls (Table 6 and Figure 10), the discrepancies are more noticeable. For example, for all wall categories, the dwell times estimated for CS2 using the simulations are higher than those observed directly from data due to overestimation of the transition probability $p_{22}$. This is principally due to the fact that the number of CS2 observations were higher than the rest of the CSs and secondly due to the accumulation of aggregated observations in the first few time intervals, which occurred because the year of construction was unknown for almost half of the retaining walls. To deal with this issue as mentioned in section 3.3, when preparing the data, the initial inspections for all walls were set to $t = 0$; while this is not the case in reality as the walls are of different ages, and the first inspections were not actually carried out at the same time. This approach was selected as it allowed for keeping the entries without the construction year and also had the advantage of being "representative" for all available time intervals ($\Delta t$ =1), whereas for the other approach used for the bridges, where the chronological order of the first inspections in time were used (i.e. not set to be at $t = 0$), a "representative deterioration time interval" was selected to estimate the transition probabilities ($\Delta t$ =5) which is also consistent with the use of Markov models.

In general, it should be noted that the dwell times might seem to many experts as relatively short. The predominant reason for this is most likely due to the definition of the condition states on the object level and their use in practice. Although the condition states were defined to give a general impression of how the object is deteriorating over time and were meant to give a global view of the object, in practice the objects are assigned condition states associated with the worst state of an element, which alerts management to the fact that an intervention is required in the near future, even if it is small. Hence, the sum of the dwell times in each condition state for an asset does not correspond with the total amount of time that it is expected to be in service before it needs to be replaced. This difference is because in one case condition states are used to approximate the life of the asset and in the other, they are being used to trigger interventions.

## 7 Summary and conclusions

Determination of optimal intervention programs for infrastructure depends on the development of the condition state of the assets over time. Markov models are appropriate models that use the information from the condition state development of the assets over time and predict their deterioration rate and the remaining service life. This paper used a Markov model to predict the deterioration curves of the railway bridges and retaining walls and proposed solutions to address the challenges that normally exist when dealing with real-world data, including insufficient inspection history, incomplete/faulty entries, biased data caused by the lack of clear guidelines for the inspectors, and the changes to the condition rating scheme.

It is concluded that the following considerations can be made in other similar real-world situations:

- Clean the data and correct or delete the faulty/incomplete entries with the goal to keep as many entries as possible to increase the informative value of the models.
- While structuring the data, separate the situations where there are discrepancies between the judgment of inspectors, from when there have been improvements in the condition of the structures due to maintenance work.
- Adjust the old inspection data with the new rating scheme when there is a change in the condition rating system. In this study, KNN, NN, and random forest were used for reclassification purposes and to align the inspection data with the new condition rating schemes, so that the data from before and after the changes in the condition rating scheme could be compared with each other. The results suggested that the random forest was the most appropriate method for such classification problems.
- The transition probability matrices can be estimated using a regression-based optimization approach for each category of objects. The deterioration curves and dwell times can be estimated by using the transition probabilities in conjunction with Monte Carlo simulations.
- When there is no information on the construction year of the objects in the database, in using the Markov models, the period between the year of construction and the initial inspection can be neglected and the initial inspections for all objects should be set to $t = 0$. This would result in the

conservative estimation of the transition probabilities except for the condition state that had the highest concentration of the observations in the first few intervals.

- As the number of observations plays a significant role in the accuracy of the deterioration curves, it is necessary to mitigate data losses due to faulty entries as much as possible. This can be done by equipping the database with a drop-down menu rather than the manual insertion of the information. Additionally, to avoid biases in the inspection ratings, the condition rating scheme should be revised to be strict and clear, leaving no room for interpretations.
- To obtain more accurate estimates of the deterioration process with limited data, it might be more beneficial to determine mechanistic-empirical deterioration models for each individual retaining wall category and then calibrate the results using the available data. Finally, the application of automated structural health monitoring or drone-based surveillance systems can facilitate regular collection of data from all assets, for future developments of the deterioration curves.
- The difference between the different uses of condition states has to be considered, i.e. one cannot simply take the condition states reported in data and use them directly to estimate the expected lifetime of the asset.

**Acknowledgments**

The authors would like to thank the financial support of the Fonds de Recherche du Québec - Nature et Technologies (FRQNT), and the Natural Sciences and Engineering Research Council of Canada (NSERC).

**References**

Bulusu, S. and Sinha, K. C. (1997) ‘Comparison of methodologies to predict bridge deterioration’, *Transportation Research Record*. SAGE Publications Sage CA: Los Angeles, CA, 1597(1), pp. 34–42.

Chu, C.-Y. and Durango-Cohen, P. L. (2007) ‘Estimation of infrastructure performance models using state-space specifications of time series models’, *Transportation Research Part C: Emerging Technologies*. Elsevier, 15(1), pp. 17–32.

Chu, C.-Y. and Durango-Cohen, P. L. (2008) ‘Estimation of dynamic performance models for transportation infrastructure using panel data’, *Transportation Research Part B: Methodological*.

Elsevier, 42(1), pp. 57–81.

Edirisinghe, R., Setunge, S. and Zhang, G. (2015) ‘Markov model—based building deterioration prediction and ISO factor analysis for building management’, *Journal of Management in Engineering*. American Society of Civil Engineers, 31(6), p. 4015009.

Hong, F. and Prozzi, J. A. (2006) ‘Estimation of pavement performance deterioration using Bayesian approach’, *Journal of infrastructure systems*. American Society of Civil Engineers, 12(2), pp. 77–86.

Howard, R. A. (1971) *Dynamic Probabilistic Systems:@ Semi-Markov and decision processes.-1971.-1122 p*. J. Wiley.

Humplick, F. (1992) ‘Highway pavement distress evaluation: Modeling measurement error’, *Transportation Research Part B: Methodological*. Elsevier, 26(2), pp. 135–154.

Kobayashi, K., Kaito, K. and Lethanh, N. (2012) ‘A statistical deterioration forecasting method using hidden Markov model for infrastructure management’, *Transportation Research Part B: Methodological*. Elsevier, 46(4), pp. 544–561.

Lethanh, N., Hackl, J. and Adey, B. T. (2017) ‘Determination of Markov transition probabilities to be used in bridge management from mechanistic-empirical models’, *Journal of bridge engineering*. American Society of Civil Engineers, 22(10), p. 4017063.

Madanat, S. M., Mishalani, R. and Ibrahim, W. H. W. (1995) ‘Estimation of infrastructure transition probabilities from condition rating data’, *Journal of infrastructure systems*. American Society of Civil Engineers, 1(2), pp. 120–125.

Manafpour, A. *et al.* (2018) ‘Stochastic analysis and time-based modeling of concrete bridge deck deterioration’, *Journal of Bridge Engineering*. American Society of Civil Engineers, 23(9), p. 4018066.

Mizutani, D. *et al.* (2017) ‘Improving the Estimation of Markov Transition Probabilities Using Mechanistic-Empirical Models’, *Frontiers in Built Environment*. Frontiers, 3, p. 58.

Nam, L. T. (2009) ‘Stochastic optimization methods for infrastructure management with incomplete monitoring data’. 京都大学 (Kyoto University).

Ortiz-García, J. J., Costello, S. B. and Snaith, M. S. (2006) ‘Derivation of transition probability matrices for pavement deterioration modeling’, *Journal of Transportation Engineering*. American Society of Civil Engineers, 132(2), pp. 141–161.

Park, E. S. *et al.* (2008) ‘A Bayesian approach for improved pavement performance prediction’, *Journal of Applied Statistics*. Taylor & Francis, 35(11), pp. 1219–1238.

Parzen, E. (1962) 'On estimation of a probability density function and mode', *The annals of mathematical statistics*. JSTOR, 33(3), pp. 1065–1076.

Robelin, C.-A. and Madanat, S. M. (2007) 'History-dependent bridge deck maintenance and replacement optimization with Markov decision processes', *Journal of Infrastructure Systems*. American Society of Civil Engineers, 13(3), pp. 195–201.

Roelfstra, G. *et al.* (2004) 'Condition evolution in bridge management systems and corrosion-induced deterioration', *Journal of Bridge Engineering*. American Society of Civil Engineers, 9(3), pp. 268–277.

Setunge, S. and Hasan, M. S. (2011) 'Concrete bridge deterioration prediction using Markov chain approach'.

# 8 Appendix

## *8.1 The K-Nearest-Neighbor*

The K-Nearest-Neighbor algorithm (KNN) is a simple, but very efficient classification method. The key idea behind the KNN classification is that similar measured values belong to the same classes. Thus, one only needs to know the class identifier of a certain number of the nearest neighbors to be able to estimate the class number of a data point (Figure 11). The class of an unknown data point is usually determined using the majority criterion. i.e. if the majority of the neighbors of an unknown point are from class $A$, it is very likely that the point is also from class $A$. The number of nearest neighbors $k$ should also be kept as small as possible since a large $k$ can lead to a bad classification if the individual classes are not well-separated.

## *8.2 Neural Networks*

Neural networks (NN) are computational models with the capacity to learn, generalize, or classify data. NNs are beneficial in approximating unknown non-linear functions that depend on a large number of variables (features) since their application eliminates the need to define that function. Figure 12 represents a typical single-layer neural network classifier. The model contains an input layer of variables (features), a hidden layer, and an output layer with the desired classes. In addition to the normal neurons, there are also bias neurons that are used to help ensure that the model has a good fit with existing data. The weights (connections between neurons) are used to model the relationships between the neurons.

The activation function is used to introduce non-linearity into the model to ensure that there is a good fit between the values predicted by the model and the expected output. To use these models, the network needs to be trained first, i.e. the weights are adjusted (by going through a certain number of forward and backward propagations) so that the error between the expected output and the generated output of the model is minimized.

In a trained NN classifier, an input data point with known characteristics (features) is passed to the first hidden layer, where the activation function in each neuron receives the input $x_i$ and generates the output. That output is then passed to the next hidden layer (if any) and the procedure continues till it reaches the output layer, where outputs from the previous layer are combined to yield a final class of the input data point.

### *8.3 Random Forest*

Random forests are built from decision trees and are widely used for classification and regression problems. Fundamentally, they structure multiple hierarchical sequences of yes/no questions that ultimately lead to a decision. The variety of decision trees is what makes random forests more effective than individual decision trees. The random forest algorithm determines the class of an unknown data point using the following steps:

1- Create a bootstrapped dataset
2- Create a decision tree using the bootstrapped dataset, only using a random subset of the variables (features) at each step.
3- Repeat steps 1 and 2, $i$ times ($i$ being the number of created decision trees or estimators).
4- Take the unclassified data and run it down each decision tree, to find the class of the data point using various decision trees.
5- The class of an unknown data point is determined using the majority criterion, i.e. the class that receives the most votes from the decision trees is chosen.

**Figure captions**

Figure 1. Changes in the condition-rating scheme of all objects in 2009 and the natural stone retaining walls in 2013

Figure 2. The 2D illustration of SMOTE

Figure 3. Redistribution of the condition states in accordance with the new rating schemes

Figure 4. Recorded condition states after the data preparation

Figure 5. Recorded condition states per category of bridges

Figure 6. Recorded condition states per category of walls

Figure 7. Recorded condition state transitions for each category of bridges

Figure 8. Recorded condition state transitions for each category of walls

Figure 9. Estimated dwell times for each category of bridges; solid lines indicate the mean dwell time while the dashed lines (--) show the minimum and the dotted dashed lines (-.-) show the maximum values.

Figure 10. Estimated dwell times for each category of walls; solid lines indicate the mean dwell time while the dashed lines (--) show the minimum and the dotted dashed lines (-.-) show the maximum values.

Figure 11. Schematic overview of the KNN algorithm

Figure 12. Visual representation of a NN classifier

**Table captions**

Table 1. Selected parameters for KNN, NN, and random forest algorithms

Table 2. Performance of the KNN, NN, and random forest algorithms

Table 1. Transition probabilities for each category of bridges

Table 2. Transition probabilities for each category of retaining walls

Table 3. Estimated dwell times for bridges

Table 4. Estimated dwell times for retaining walls

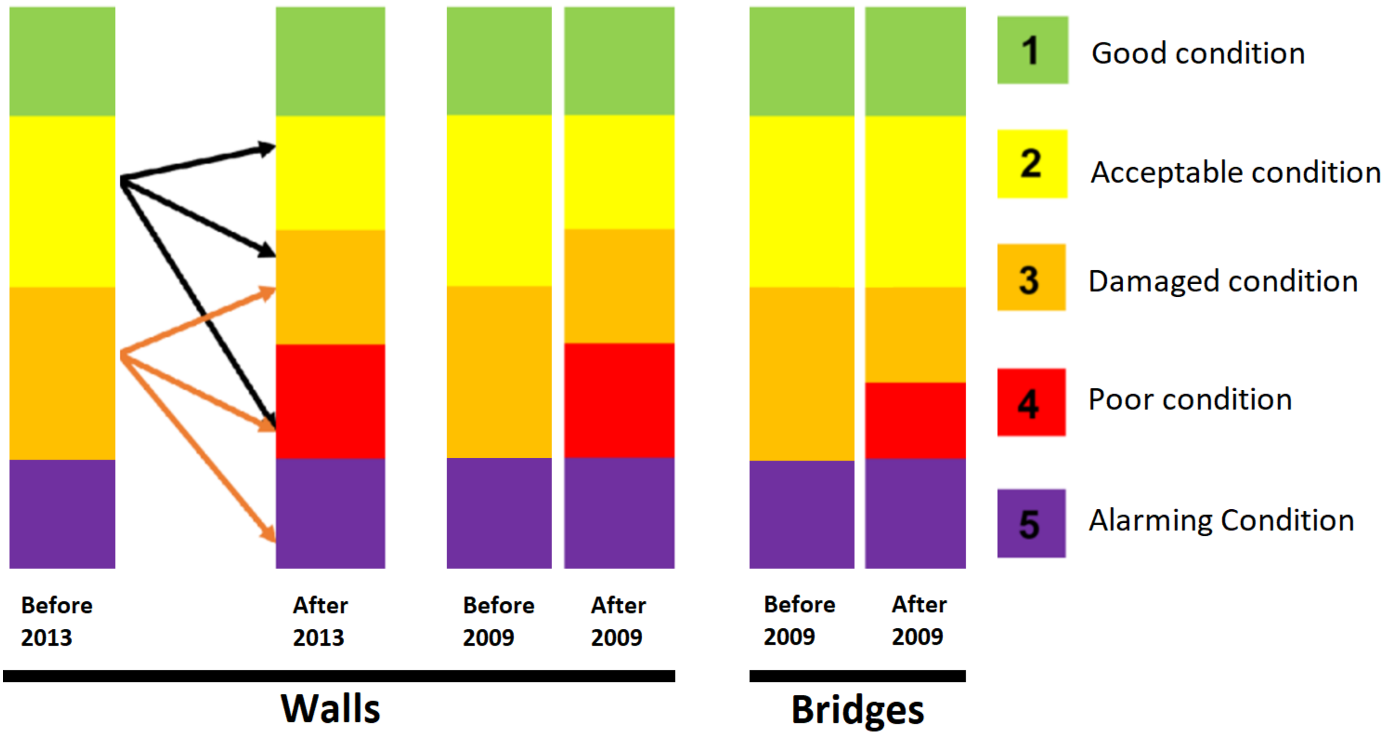
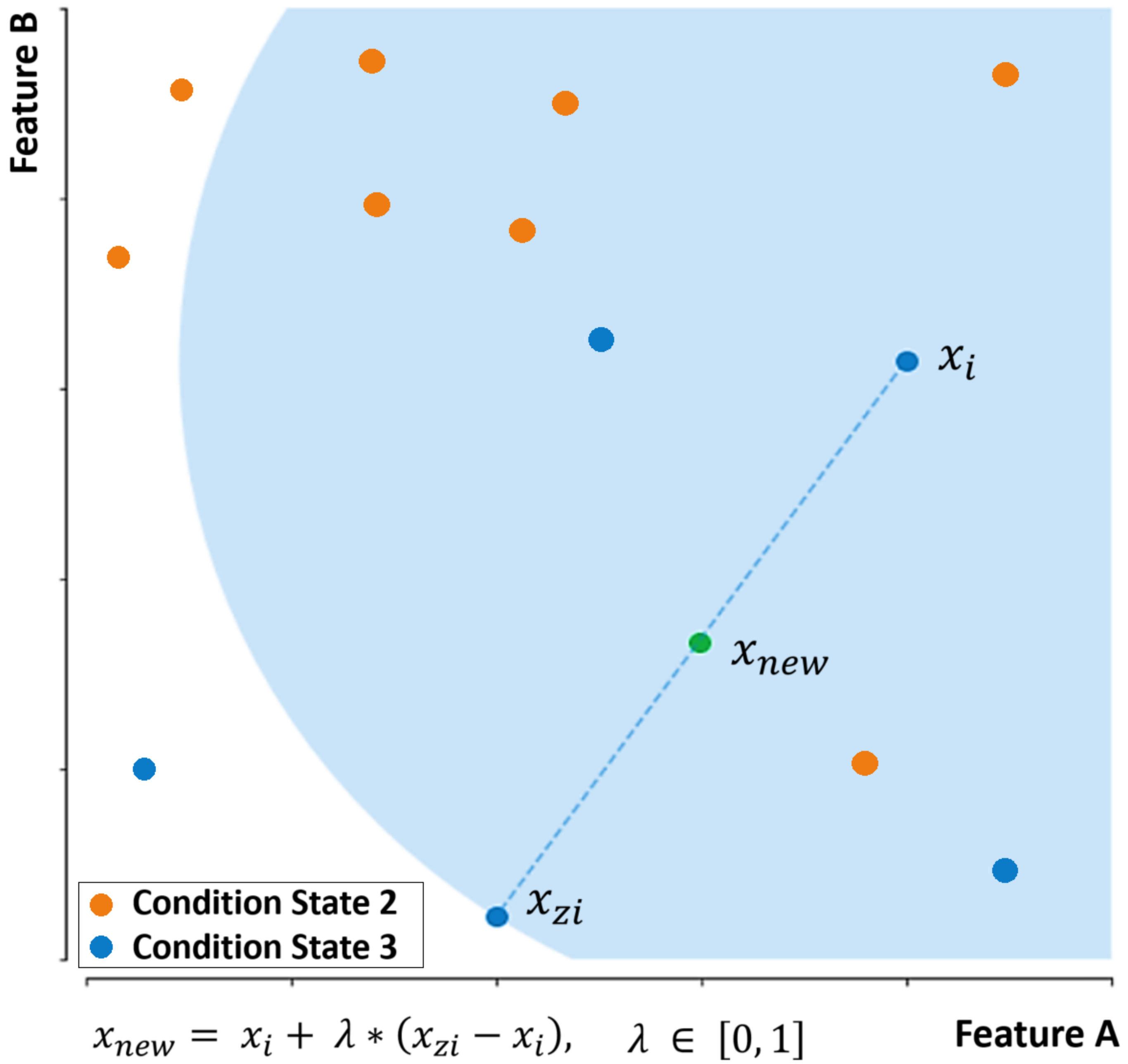
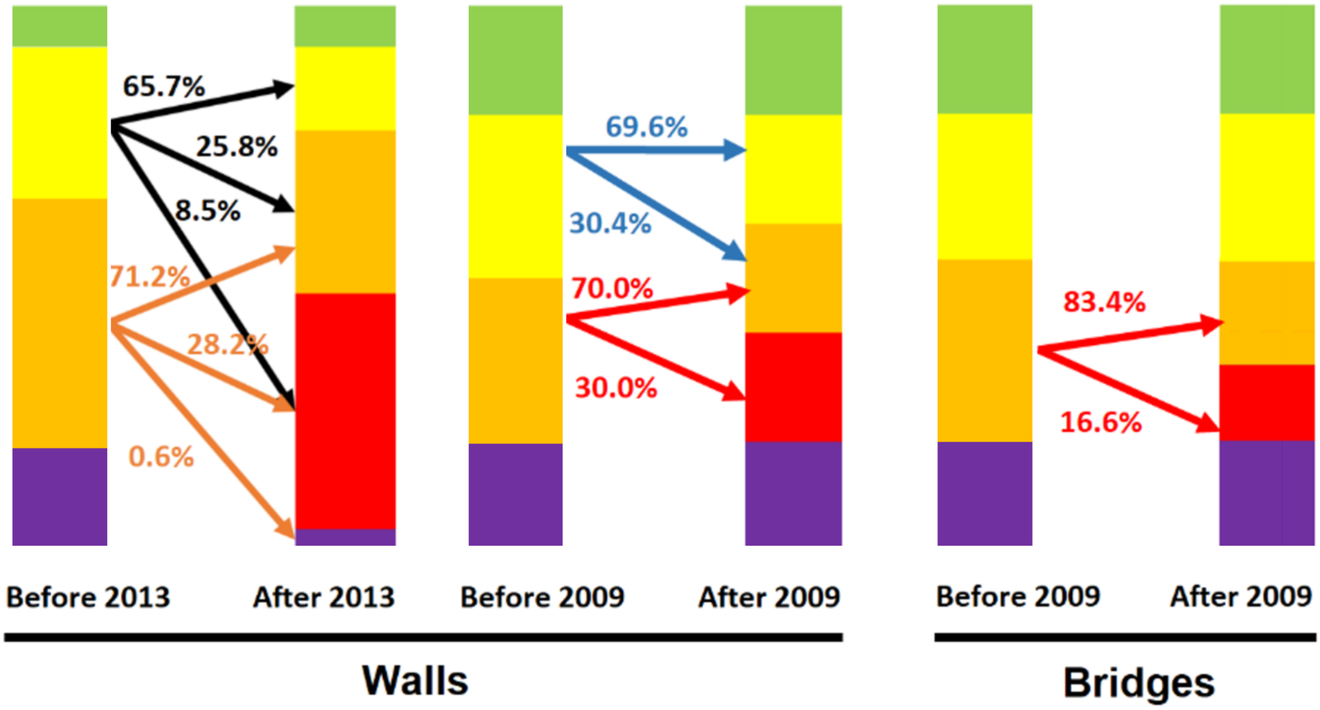



Figure 1.png Figure


Feature B
$x_i$
$x_{new}$
$x_{zi}$
Condition State 2
Condition State 3
$x_{new} = x_i + \lambda * (x_{zi} - x_i), \quad \lambda \in [0,1]$
Feature A


65.7%
25.8%
8.5%
71.2%
28.2%
0.6%
69.6%
30.4%
70.0%
30.0%
83.4%
16.6%
Before 2013
After 2013
Before 2009
After 2009
Before 2009
After 2009
Walls
Bridges

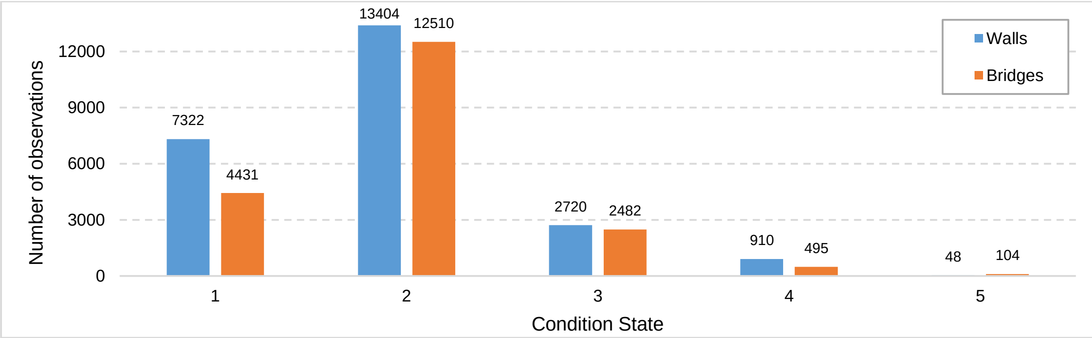



**Figure 4.** Recorded condition states after the data preparation

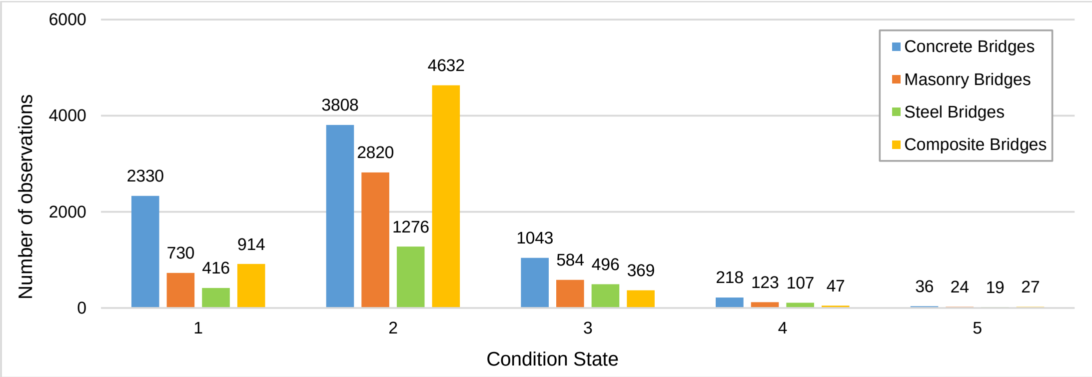



**Figure 5.** Recorded condition states per category of bridges

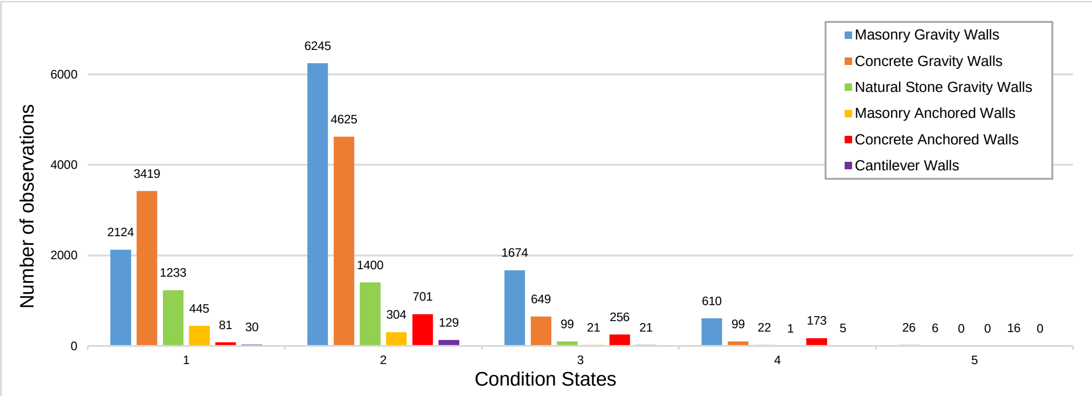



**Figure 6.** Recorded condition states per category of walls

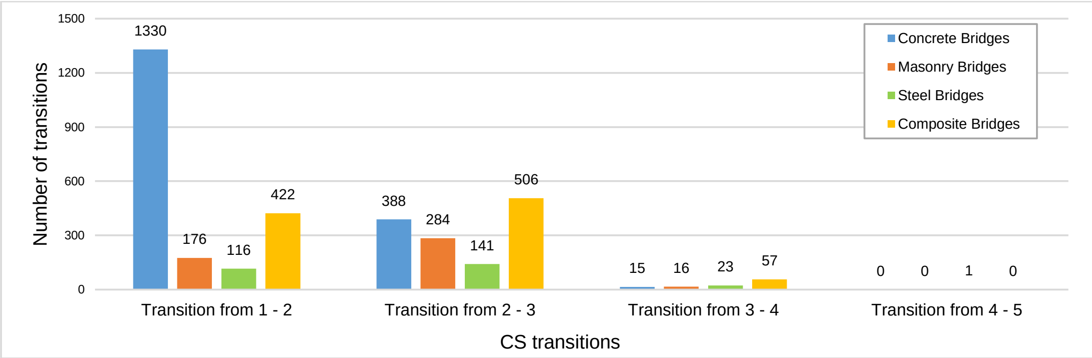



**Figure 7.** Recorded condition state transitions for each category of bridges

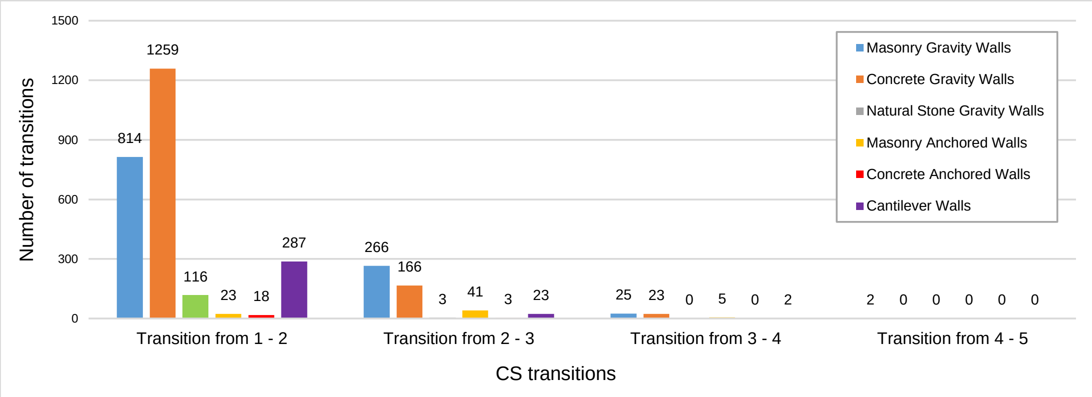
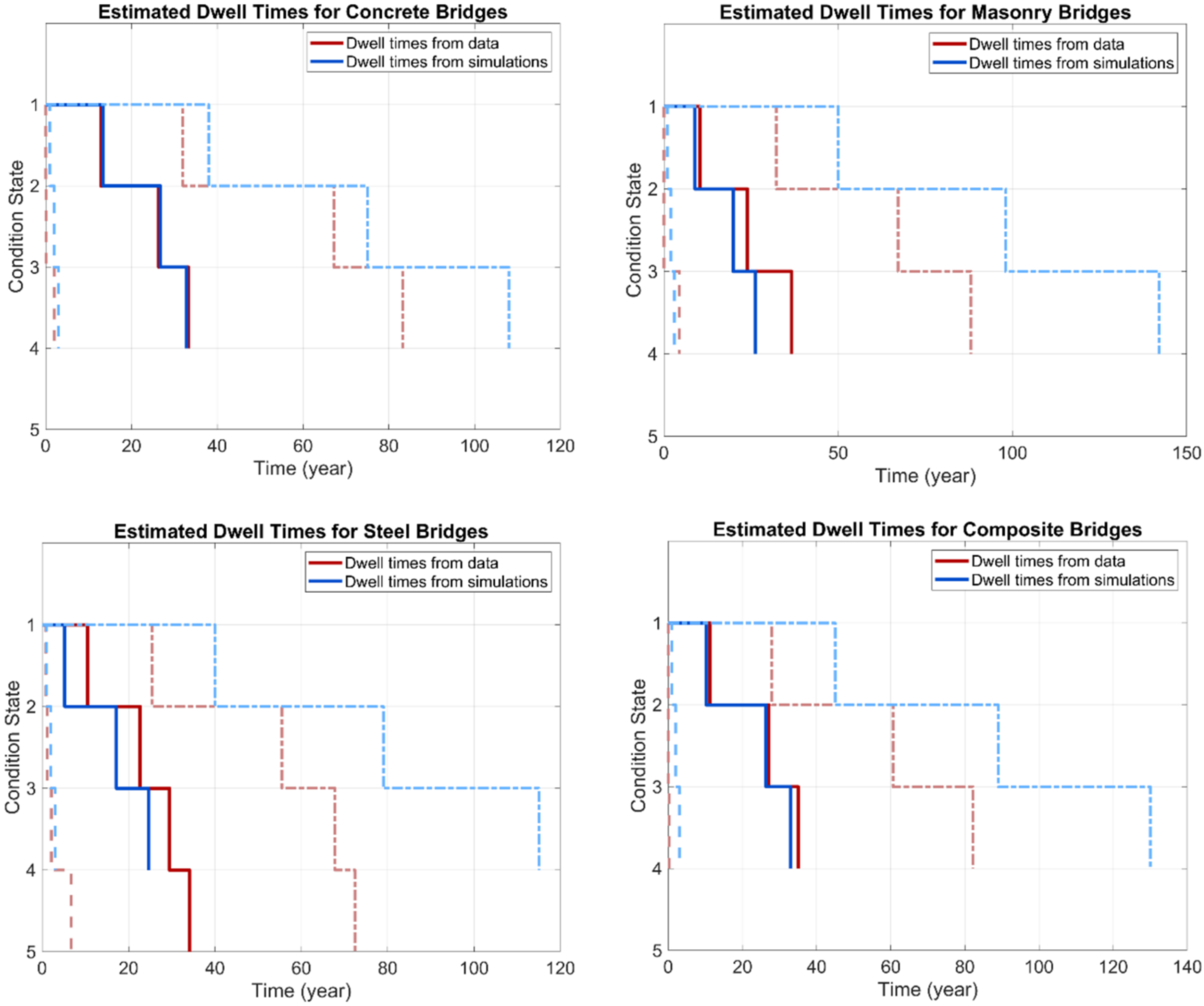
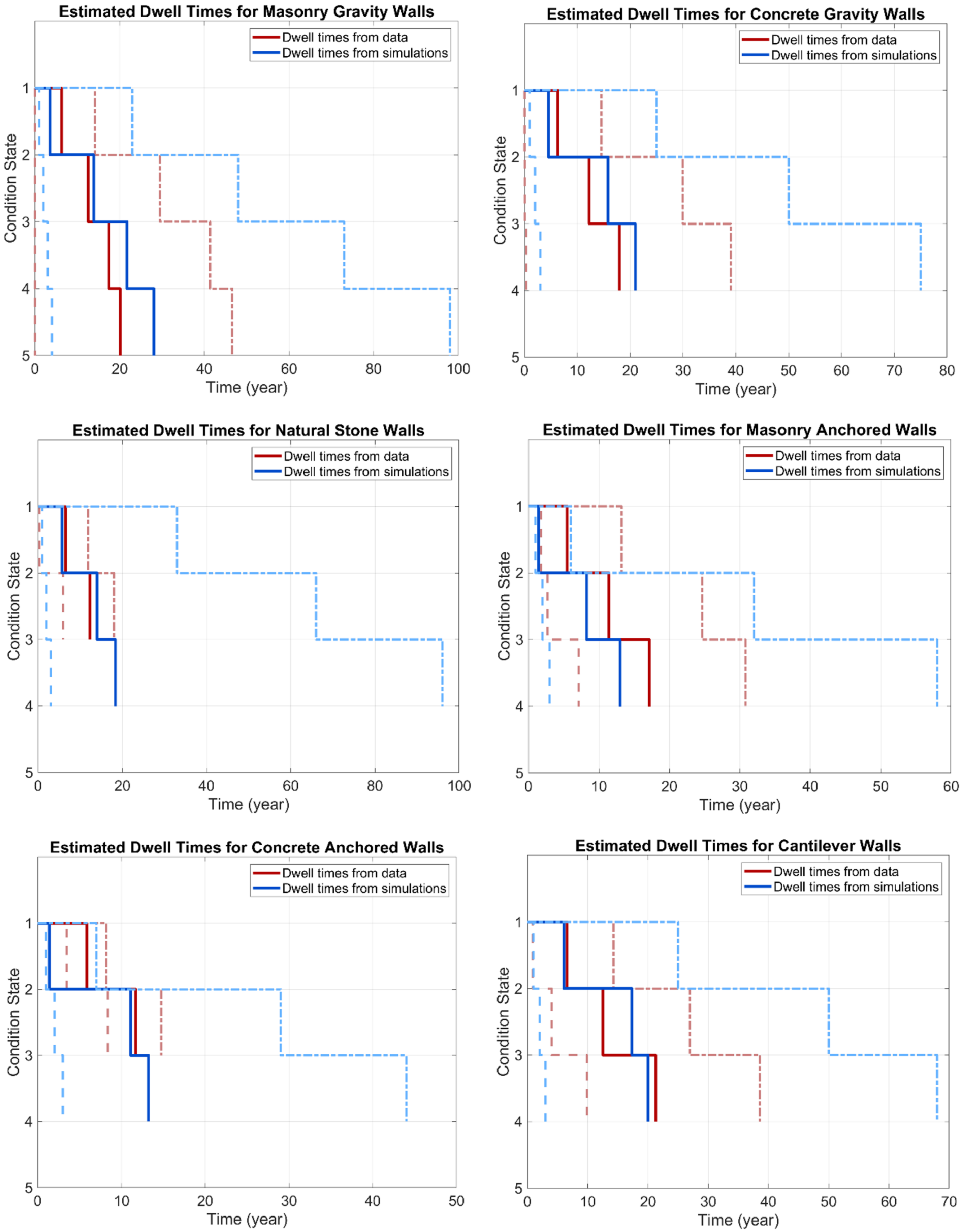
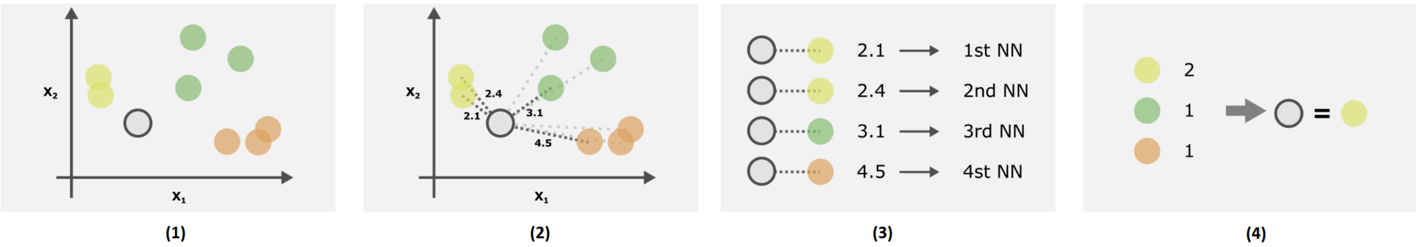



**Figure 8.** Recorded condition state transitions for each category of walls



Estimated Dwell Times for Concrete Bridges
Dwell times from data
Dwell times from simulations
Condition State
Time (year)
Estimated Dwell Times for Masonry Bridges
Dwell times from data
Dwell times from simulations
Condition State
Time (year)
Estimated Dwell Times for Steel Bridges
Dwell times from data
Dwell times from simulations
Condition State
Time (year)
Estimated Dwell Times for Composite Bridges
Dwell times from data
Dwell times from simulations
Condition State
Time (year)


Estimated Dwell Times for Masonry Gravity Walls
Estimated Dwell Times for Concrete Gravity Walls
Estimated Dwell Times for Natural Stone Walls
Estimated Dwell Times for Masonry Anchored Walls
Estimated Dwell Times for Concrete Anchored Walls
Estimated Dwell Times for Cantilever Walls
Dwell times from data
Dwell times from simulations
Condition State
Time (year)


x₂
x₁
(1)
x₂
x₁
2.4
2.1
3.1
4.5
(2)
2.1 1st NN
2.4 2nd NN
3.1 3rd NN
4.5 4st NN
(3)
2
1
1
=
(4)

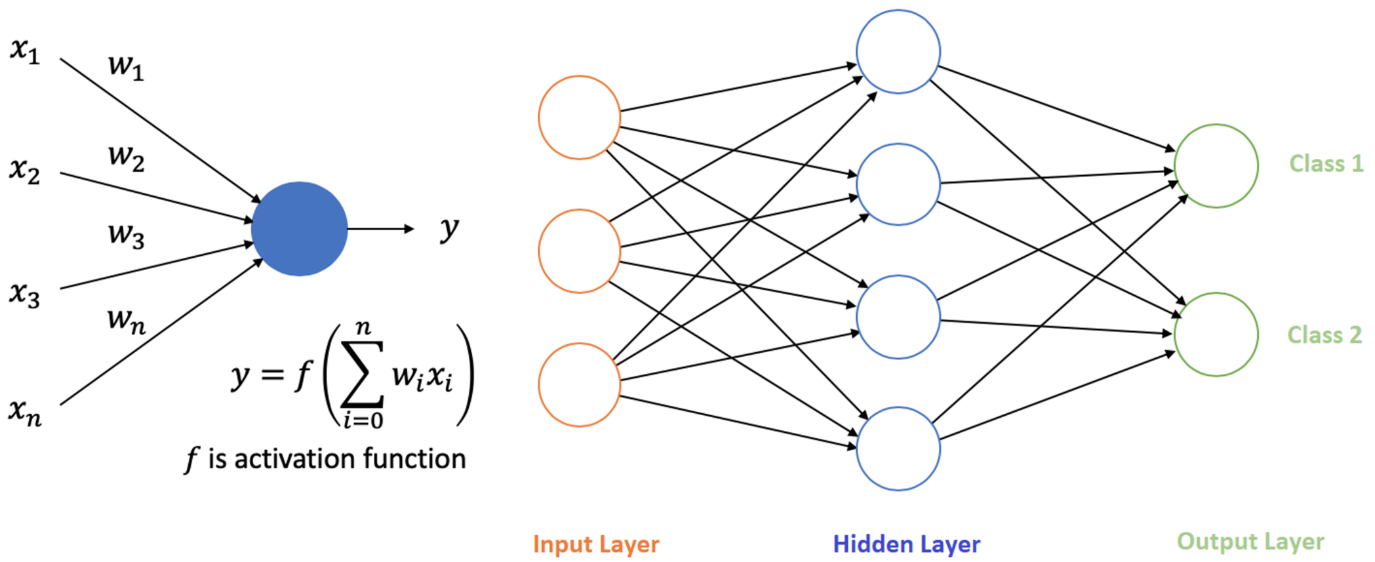

$x_1$
$w_1$
$x_2$
$w_2$
$w_3$
$x_3$
$w_n$
$x_n$
$y$
$y = f\left(\sum_{i=0}^{n} w_i x_i\right)$
$f$ is activation function
Class 1
Class 2
Input Layer
Hidden Layer
Output Layer

**Table 1.** Selected parameters for KNN, NN, and random forest algorithms

| Algorithm | The selected combination of parameters | Comments |
|---|---|---|
| KNN | $k = 15$ | number of closest neighbors for classification 1, 2 and 3 |
| | $k = 20$ | number of closest neighbors for classification 4 and 5 |
| NN | Number of hidden layers = 3 | The number of hidden layers |
| | hidden layer sizes = (50, 50, 50) | |
| | $\varphi$ = tanh | Activation function [logistic, tahn, relu] |
| | $\alpha$ = 0.01 | Learning rate |
| Random Forest | max_depth = 100 | The maximum depth of the tree |
| | max_features = sqrt | The number of features to consider when searching for a suitable split |
| | min_samples_split = 2 | The minimum number of samples required to split an internal node |
| | n_estimators = 100 | The number of trees in the forest |

**Table 2.** Performance of the KNN, NN, and random forest algorithms

| Classification Algorithm | | KNN | | | NN | | | Random Forest | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Aggregation of the training data set | | Normal | Oversampled | SMOTE | Normal | Oversampled | SMOTE | Normal | Oversampled | SMOTE |
| Classification 1 | $f_1$ Score | 0.569 | 0.547 | 0.584 | 0.587 | 0.603 | 0.612 | 0.723 | 0.763 | 0.745 |
| Classification 2 | $f_1$ Score | 0.538 | 0.592 | 0.592 | 0.564 | 0.614 | 0.591 | 0.714 | 0.749 | 0.730 |
| Classification 3 | $f_1$ Score | 0.530 | 0.566 | 0.565 | 0.680 | 0.684 | 0.650 | 0.809 | 0.811 | 0.800 |
| Classification 4 | $f_1$ Score | 0.401 | 0.456 | 0.444 | 0.473 | 0.460 | 0.434 | 0.682 | 0.699 | 0.695 |
| Classification 5 | $f_1$ Score | 0.377 | 0.429 | 0.335 | 0.551 | 0.632 | 0.572 | 0.793 | 0.862 | 0.795 |

**Table 3.** Transition probabilities for each category of bridges

$$P_a = \begin{bmatrix} 0.9456 & 0.0544 & 0 & 0 \\ 0 & 0.9796 & 0.0204 & 0 \\ 0 & 0 & 0.8890 & 0.1110 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**(a) Concrete bridges**

$$P_b = \begin{bmatrix} 0.8896 & 0.0752 & 0.0350 & 0 \\ 0 & 0.9206 & 0.0794 & 0 \\ 0 & 0 & 0.8489 & 0.1511 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**(b) Masonry bridges**

$$P_c = \begin{bmatrix} 0.8102 & 0.1898 & 0.2617 & 0 \\ 0 & 0.9376 & 0.0623 & 0 \\ 0 & 0.0 & 0.8954 & 0.1046 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**(c) Steel bridges**

$$P_d = \begin{bmatrix} 0.9063 & 0.0937 & 0 & 0 \\ 0 & 0.9774 & 0.0226 & 0 \\ 0 & 0.0 & 0.8791 & 0.1209 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**(d) Composite bridges**

**Table 4.** Transition probabilities for each category of retaining walls

$$P_a = \begin{bmatrix} 0.7229 & 0.2771 & 0 & 0 & 0 \\ 0 & 0.9557 & 0.0407 & 0.0036 & 0 \\ 0 & 0 & 0.9373 & 0.0627 & 0 \\ 0 & 0 & 0 & 0.9247 & 0.0753 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

**(a) Masonry gravity walls**

$$P_b = \begin{bmatrix} 0.7827 & 0.1878 & 0.0295 & 0 \\ 0 & 0.9852 & 0.0148 & 0 \\ 0 & 0 & 0.8469 & 0.1531 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**(b) Concrete gravity walls**

$$P_c = \begin{bmatrix} 0.8277 & 0.1723 & 0 & 0 \\ 0 & 0.8979 & 0.1021 & 0 \\ 0 & 0.0 & 0.7880 & 0.2120 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**(c) Stone gravity walls**

$$P_d = \begin{bmatrix} 0.3109 & 0.4274 & 0.2617 & 0 \\ 0 & 0.8672 & 0.1322 & 0 \\ 0 & 0.0 & 0.8006 & 0.1994 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**(d) Masonry anchored walls**

$$P_e = \begin{bmatrix} 0.2986 & 0.4486 & 0 & 0.2528 \\ 0 & 0.9568 & 0.0432 & 0 \\ 0 & 0.0 & 0.5414 & 0.4586 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**(e) Concrete anchored walls**

$$P_f = \begin{bmatrix} 0.8464 & 0.1204 & 0.0332 & 0 \\ 0 & 0.9908 & 0.0092 & 0 \\ 0 & 0.0 & 0.6440 & 0.3560 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**(f) Cantilever walls**

**Table 5.** Estimated dwell times for bridges

| Type | Condition State | $\mu$ | $\sigma$ | Median | Min | Max | Observations |
|---|---|---|---|---|---|---|---|
| Concrete Bridges | | | | | | | |
| Type | Condition State | $\mu$ | $\sigma$ | Median | Min | Max | Observations |
| Dwell times from data | 1 | 12.86 | 6.58 | 11.71 | 0.00 | 31.97 | 1330 |
| | 2 | 13.48 | 8.61 | 11.58 | 0.08 | 35.20 | 388 |
| | 3 | 6.94 | 3.89 | 5.81 | 1.94 | 16.11 | 15 |
| Dwell times from simulations | 1 | 13.39 | 9.94 | 11.00 | 1.00 | 38.00 | 10547 |
| | 2 | 13.35 | 8.88 | 12.00 | 1.00 | 37.00 | 3997 |
| | 3 | 6.09 | 5.04 | 4.00 | 1.00 | 33.00 | 2844 |
| Masonry Bridges | | | | | | | |
| Type | Condition State | $\mu$ | $\sigma$ | Median | Min | Max | Observations |
| Dwell times from data | 1 | 10.30 | 4.76 | 10.93 | 0.00 | 32.23 | 176 |
| | 2 | 13.62 | 7.96 | 11.59 | 0.00 | 34.96 | 284 |
| | 3 | 12.69 | 5.14 | 12.70 | 4.46 | 20.86 | 16 |
| Dwell times from simulations | 1 | 8.89 | 8.01 | 6.00 | 1.00 | 50.00 | 11968 |
| | 2 | 10.98 | 9.21 | 8.00 | 1.00 | 48.00 | 7812 |
| | 3 | 6.40 | 5.70 | 5.00 | 1.00 | 44.00 | 11178 |
| Steel Bridges | | | | | | | |
| Type | Condition State | $\mu$ | $\sigma$ | Median | Min | Max | Observations |
| Dwell times from data | 1 | 10.50 | 5.24 | 10.17 | 0.84 | 25.47 | 116 |
| | 2 | 12.17 | 6.67 | 11.59 | 0.33 | 30.04 | 141 |
| | 3 | 6.80 | 3.04 | 5.95 | 0.92 | 12.29 | 23 |
| | 4 | 4.62 | 0.00 | 4.62 | 4.62 | 4.62 | 1 |
| Dwell times from simulations | 1 | 5.22 | 4.65 | 4.00 | 1.00 | 40.00 | 11997 |
| | 2 | 11.93 | 9.02 | 10.00 | 1.00 | 39.00 | 10682 |
| | 3 | 7.46 | 6.16 | 6.00 | 1.00 | 36.00 | 9124 |
| Composite Bridges | | | | | | | |
| Type | Condition State | $\mu$ | $\sigma$ | Median | Min | Max | Observations |
| Dwell times from data | 1 | 11.26 | 5.46 | 11.03 | 0.00 | 27.95 | 422 |
| | 2 | 15.83 | 8.15 | 15.84 | 0.00 | 32.65 | 506 |
| | 3 | 7.96 | 4.24 | 5.98 | 0.23 | 21.57 | 57 |
| Dwell times from simulations | 1 | 10.22 | 9.00 | 7.00 | 1.00 | 45.00 | 11851 |
| | 2 | 16.15 | 10.92 | 14.00 | 1.00 | 44.00 | 6505 |
| | 3 | 6.65 | 5.57 | 5.00 | 1.00 | 41.00 | 5297 |

**Table 6.** Estimated dwell times for retaining walls

| Type | Condition State | $\mu$ | $\sigma$ | Median | Min | Max | Observations |
|---|---|---|---|---|---|---|---|
| Masonry Gravity Walls | | | | | | | |
| Dwell times from data | 1 | 6.27 | 2.64 | 5.87 | 0.00 | 14.17 | 814 |
| | 2 | 6.32 | 2.54 | 5.92 | 0.01 | 15.36 | 266 |
| | 3 | 4.94 | 2.81 | 5.40 | 0.04 | 11.86 | 25 |
| | 4 | 2.59 | 2.58 | 2.59 | 0.00 | 5.17 | 2 |
| Dwell times from simulations | 1 | 3.56 | 2.93 | 3.00 | 1.00 | 23.00 | 3944 |
| | 2 | 10.30 | 6.73 | 9.00 | 1.00 | 25.00 | 6190 |
| | 3 | 7.88 | 5.75 | 7.00 | 1.00 | 25.00 | 4347 |
| | 4 | 6.35 | 4.92 | 5.00 | 1.00 | 25.00 | 2887 |
| Concrete Gravity Wall | | | | | | | |
| Type | Condition State | $\mu$ | $\sigma$ | Median | Min | Max | Observations |
| Dwell times from data | 1 | 6.32 | 2.36 | 5.96 | 0.00 | 14.58 | 1259 |
| | 2 | 5.92 | 2.14 | 5.84 | 0.08 | 15.38 | 166 |
| | 3 | 5.74 | 2.14 | 5.94 | 0.22 | 9.11 | 23 |
| Dwell times from simulations | 1 | 4.54 | 3.87 | 3.00 | 1.00 | 25.00 | 6277 |
| | 2 | 11.26 | 6.84 | 11.00 | 1.00 | 25.00 | 2966 |
| | 3 | 5.18 | 4.33 | 4.00 | 1.00 | 25.00 | 3655 |
| Natural Stone Gravity Walls | | | | | | | |
| Type | Condition State | $\mu$ | $\sigma$ | Median | Min | Max | Observations |
| Dwell times from data | 1 | 6.49 | 2.27 | 5.93 | 0.26 | 11.87 | 116 |
| | 2 | 5.84 | 0.24 | 5.67 | 5.67 | 6.17 | 3 |
| Dwell times from simulations | 1 | 5.71 | 5.04 | 4.00 | 1.00 | 33.00 | 9887 |
| | 2 | 8.30 | 6.69 | 6.00 | 1.00 | 33.00 | 11185 |
| | 3 | 4.33 | 3.65 | 3.00 | 1.00 | 30.00 | 10717 |
| Masonry Anchored Walls | | | | | | | |
| Type | Condition State | $\mu$ | $\sigma$ | Median | Min | Max | Observations |
| Dwell times from data | 1 | 5.50 | 2.26 | 5.87 | 1.79 | 13.21 | 23 |
| | 2 | 5.90 | 2.09 | 5.90 | 0.93 | 11.44 | 41 |
| | 3 | 5.74 | 0.68 | 6.01 | 4.39 | 6.14 | 5 |
| Dwell times from simulations | 1 | 1.44 | 0.76 | 1.00 | 1.00 | 6.00 | 856 |
| | 2 | 6.80 | 5.53 | 5.00 | 1.00 | 26.00 | 8223 |
| | 3 | 4.74 | 3.94 | 4.00 | 1.00 | 26.00 | 10378 |
| Concrete Anchored Walls | | | | | | | |
| Type | Condition State | $\mu$ | $\sigma$ | Median | Min | Max | Observations |
| Dwell times from data | 1 | 5.85 | 1.21 | 5.81 | 3.44 | 8.20 | 18 |
| | 2 | 5.85 | 0.66 | 6.07 | 4.95 | 6.53 | 3 |
| Dwell times from simulations | 1 | 1.41 | 0.75 | 1.00 | 1.00 | 7.00 | 2710 |
| | 2 | 9.69 | 6.17 | 9.00 | 1.00 | 22.00 | 5136 |
| | 3 | 2.13 | 1.52 | 2.00 | 1.00 | 15.00 | 6332 |
| Cantilever Retaining Walls | | | | | | | |
| Type | Condition State | $\mu$ | $\sigma$ | Median | Min | Max | Observations |
| Dwell times from data | 1 | 6.57 | 2.56 | 5.93 | 0.87 | 14.26 | 287 |
| | 2 | 5.96 | 1.89 | 5.58 | 3.11 | 12.68 | 23 |
| | 3 | 8.74 | 2.87 | 8.74 | 5.87 | 11.61 | 2 |
| Dwell times from simulations | 1 | 6.04 | 5.08 | 4.00 | 1.00 | 25.00 | 5615 |
| | 2 | 11.30 | 6.83 | 11.00 | 1.00 | 25.00 | 1833 |
| | 3 | 2.68 | 2.17 | 2.00 | 1.00 | 18.00 | 3478 |