Machine Learning for the Condition Assessment of Concrete Bridges

By

Fred Fayad

BSc. Civil Engineering, 2024, Swiss Federal Institute of Technology Lausanne

Submitted to the Department of Civil and Environmental Engineering in partial fulfillment of the requirements for the degree of
Master of Engineering in Civil and Environmental Engineering
at the

Massachusetts Institute of Technology

May 2025

Authored by: Fred Fayad
Civil and Environmental Engineering
May 9, 2025

Certified by: Oral Buyukozturk
George Macomber Professor of Civil and Environmental Engineering
Thesis Supervisor

Accepted by: Heidi Nepf
Donald and Martha Harleman Professor of Civil and Environmental Engineering
Chair, Graduate Program Committee

Machine Learning for the Condition Assessment of Concrete Bridges

by

Fred Fayad

Submitted to the Department of Civil and Environmental Engineering on May 9, 2025 in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Civil and Environmental Engineering

Abstract

The assessment of concrete bridge conditions is critical for ensuring structural integrity and public safety. Traditional inspection methods, which rely heavily on visual inspections and manual assessments, are time-consuming, subjective, and prone to human error. With the increasing number of aging bridges worldwide, there is a growing need for more efficient and accurate methods to assess bridge health. This thesis aims to explore the application of machine learning techniques for automating the bridge condition assessment process and improving the accuracy and reliability of bridge evaluations.

This study investigates the development and implementation of a model consisting of two machine learning algorithms to predict the condition of concrete bridges based on data collected from various public sources. The first algorithm appraises the structural health of a bridge based on bridge rating and the second algorithm assesses the condition of a bridge after a specific failure mechanism. Specifically, this work focuses on using classification algorithms such as Random Forest (RF), XGBoost, and Neural Networks (NN) in both algorithms to achieve their purpose.

The results of this study demonstrate that machine learning models can provide a decent performance in predicting bridge conditions. The overall model achieved a testing accuracy of 79%. This research contributes to the field of civil engineering by showcasing the potential of machine learning in infrastructure management. By automating the assessment process, the proposed models can help reduce the time and cost of inspections while providing more accurate data to guide maintenance planning and bridge rehabilitation efforts. Future work will focus on further optimizing the models, incorporating additional data sources, and deploying the system for real-time bridge monitoring.

Thesis supervisor: Oral Buyukozturk
Title: George Macomber Professor of Civil and Environmental Engineering

Acknowledgments

I want to express my heartfelt gratitude to my MIT mentors that made this research possible. Particularly, I would like to thank my thesis advisor Professor Oral Buyukozturk for his invaluable guidance and for sharing his incredible experience as an expert in the field. I would also like to thank my family for their unwavering support throughout this journey. A heartfelt acknowledgment goes to my father, Farid, whose inspiration have been a constant source of strength.

Contents

Abstract **2**

Acknowledgments **3**

1 Introduction **7**

2 Literature Review **8**

3 Methodology **10**
3.1 Creation of the Databases 10
3.1.1 Database 1 10
3.1.2 Database 2 12
3.2 Development of the two algorithms 14
3.2.1 Neural Network 14
3.2.2 Decision Trees 16
3.2.3 Random Forest 16
3.2.4 Extreme gradient booster XGB 16
3.2.5 K nearest neighbors 17
3.2.6 Stacking 18
3.2.7 Other Machine Learning Techniques 19

4 Results **20**
4.1 Performance 20
4.1.1 Algorithm 1 20
4.1.2 Algorithm 2 21
4.2 Assembly and Deployment of the model 23

5 Case Study **26**
5.1 Condition assessment: Sunshine Skyway Bridge 26
5.2 Design example: Reinterpretation of The Harvard Bridge 28

6 Closing Remarks and conclusions **31**

List of Figures

3.1 Architecture of algorithm 1 15
3.2 Architecture of algorithm 2 16
3.3 KNN Accuracy VS. K Value 18
3.4 Stacking model Algorithm 1 18

4.1 Confusion matrix for algorithm 1 21
4.2 Neural Network training process for algorithm 2 22
4.3 Confusion matrix for algorithm 2 23
4.4 Feature importance study 24
4.5 Website interface for bridge model 25

5.1 Picture of the Sunshine Skyway Bridge 26
5.2 Results of the assessment from the model 28
5.3 Picture of the Harvard Bridge 29

List of Tables

3.1 Feature Summary of Dataset 1 12
3.2 Classification grading system 12
3.3 Feature Summary of Dataset 2 13

4.1 Algorithm 1 accuracy per class 20
4.2 Algorithm 2 accuracy per class 22

5.1 Features of the Sunshine Skyway Bridge 27
5.2 Fixed design features of the Harvard Bridge 29

1 Introduction

Bridges are a critical part of the modern infrastructure and economy, yet studies show that 25% of the bridges in the United States are structurally deficient or functionally obsolete [1]. Reinforced concrete and Prestressed concrete bridges are facing multiple failures across the world due to many causes like steel corrosion, construction problems and extreme events [2]. Thus, there is a growing need for the development of a tool that would guide us in the design and assessment of bridges. Equipping every bridge with sensors and cameras is very challenging. Additionally, It can be very difficult to predict the response of a bridge under a specific failure mechanism. Running a finite element simulation would be very computationally heavy for each bridge, especially in the design phase or in the assessment of multiple bridge structures. Machine learning algorithms could be very useful for this purpose, research shows that this technology is very efficient in the field of monitoring concrete bridges and construction quality management, but they still need some maturing on the design side [3].
To address all these worldwide challenges related to critical infrastructure, I decided to take on this thesis that aims at creating a machine learning tool that can assess the structural condition of bridges and predict their response to an extreme event. This is a perfect topic to bridge the gap between machine learning and structural engineering, especially in the world of design and assessment of condition in construction management. For the sake of this thesis project, I will only focus on reinforced and prestressed concrete bridges to narrow the research. But it can be generalized to other types of buildings or even infrastructure.
The goal is to develop two machine learning algorithms, one that can appraise the structural integrity of a concrete bridge based on bridge rating and one that can predict the state of a bridge after an incident. Their output will then be combined to create a model that can assess the condition of a concrete bridge. It could also be useful to guide the design of a bridge, the servicing sequence and the implementation of precautions before an extreme event.

2 Literature Review

I started this initiative with a thorough literature review to see how Machine learning is being integrated into the world of bridge engineering. I found multiple articles that relate to my subject directly and indirectly.
Starting first with the condition assessment of bridges subject to failure events. In multiple of his papers, M. Z. Naser employs machine learning techniques to identify bridges that are vulnerable to extreme events. In his paper [4], he develops multiple machine learning algorithms that can predict the vulnerability of many types of bridges to extreme events. This classification task is based on a dataset of around 300 bridges. The algorithm outputs the state of the bridges after an extreme event (No/Minor Damage, Major Damage, Collapse). Even though he does not touch on the appraisal of the structural health of bridges, this research can be useful to complement one of the algorithms that we will be working on. We will first use most of the database that is provided to feed the training of the model. In addition, we could take inspiration of a lot of the features used, like span, age and others as well as some of the machine learning approaches he used like deep learning, genetic algorithm and many more. Finally, the algorithm that we will develop will also be a classification algorithm that outputs the vulnerability of a bridge to a specific failure cause, even though the classes won't be the same. In his other paper [5] where he only focuses on fire-vulnerable bridges, he combines the output of 2 algorithms, which is also a technique that could be explored for this work. This will inspire this thesis in different aspects. First, the development of this algorithm will be followed by a parametric study of the features that will help guide the design of RC and PC bridges to avoid these failures, potentially leading to recommending measures in the concept phase of the design. Some of the tools used to develop a proper algorithm for this task will also inspire this work.
Other pieces of literature were also very useful for this project, especially in terms of data collection. Starting first with a paper by W. Fan et al. [3], the authors discuss the use of Machine learning for multiple applications related to reinforced concrete bridges. They propose multiple algorithms and tools that could be in multiple domains like: Structural design, Construction quality management and inspection of reinforced concrete bridges. This study clearly highlights that machine learning still needs some progress to achieve maturity in the design aspect of these bridges. The key outcome of this study for this research is the need to develop an ML algorithm that could guide choices for the design of reinforced concrete bridges like structural system and reinforcing ratio. This paper also provides multiple databases that could be useful for the model alongside a catalog of machine learning algorithms that has been proven to work on many applications of structural engineering for bridges. Another interesting paper in terms of data collection for our project is the "Analysis of recent bridge failures in the United States" by Kumalasari Wardhana and Fabian C. Hadipriono [2]. It

consists of a huge statistical analysis of bridge failures in the United States between 1989 and 2000 operating on a dataset of around 500 bridges. Securing this much data will be very important to ensure the validity of the results.
The Key outcomes of this literature review are the multiple databases that are fit to use for the model, as well as the different types of machine learning algorithms that were used for different purposes of structural or bridge engineering. These studies also inspired the purpose of this algorithm in not only the condition assessment side but also the design aspect of the model.

3 Methodology

Before detailing the methodology followed to develop this model, it is important to clearly state its purpose. The model will have the task of assessing the condition of concrete bridges based on a two-steps approach, two criteria identified by two different machine learning algorithms: The first one appraises the structural health of a bridge by classifying it as good, fair, or poor and the second one predicts the condition of the bridge subject to multiple failure events indicating if it is Strongly damaged or not. Combined, the output of the two algorithms creates an efficient model to assess the state of the bridge infrastructure now and in the future.
There are multiple steps involved for the development of a proper Machine learning algorithm. They can be broken down into 2 major tasks: Data collection and assembly of a database including feature engineering; and development of the algorithm. As for the model assembly, it will be discussed in the Results section.

3.1 Creation of the Databases

First, the creation of a proper database is crucial to ensure that the results from the model are accurate. In this case, we need two separate databases to train each of our two algorithms. This is arguably the most important part for a machine learning model. After gathering the data, it is also crucial to apply feature engineering to make raw unfiltered data efficient and understandable to our different algorithms. This is a step further in guaranteeing the quality of the results. The database should be constituted of 3 major components: The training set for the training of the algorithm as its name suggests, the validation set for the tuning of the hyperparameters and a first estimation of the real performance and finally the testing set to assess the final performance of the model.

3.1.1 Database 1

For the algorithm that evaluates the structural health of a bridge, a very big database is required to correctly understand the underlying pattern behind every bridge feature. This database should consist of bridges that have not collapsed. An immense archive on existing bridges in the United States is publicly available on the Federal highway administration's website [6]. This archive is very useful for our algorithm because of the amount of data that is available, taking up nearly 300 mb of storage. It consists of around 650000 bridges; approximately 400000 of them being concrete, with 105 features for each bridge. Most of these features are not useful for our application and for the purpose of our model, there is no need to use this extensive amount of information on bridges. This is why the database had to be filtered heavily before being fed to our

algorithm. The dataset was very difficult to read directly, for storage purposes they had everything stored as a mix of numbers, letters and codes that could be decoded through a guide that was provided on their website.
After going through every feature and decoding it, the relevant features for our algorithm were initially determined through a series of statistical studies and educated engineering decisions, with around 37 features retained. Although this amount might sound overwhelming, this ensures accurate results and the most optimal performance of the algorithm. They are listed in Table 3.1 in the left column. Every feature can either take a discrete or continuous value, the second column of Table 3.1 consists of the number of categories for each discrete feature (The number of values that this feature can take), in case the feature is continuous, it is indicated by the "Continuous" label. For this algorithm, the features had to be label encoded instead of one-hot encoding because of the extensive number of options for each feature. There were very little missing values and outliers because this data was collected by hand every year by bridge inspectors at all departments of transportation across the United States.
The bridges are classified in 3 categories: Good, Fair and Poor based on the following grading system: 0-4: Poor, 5-6: Fair and 7-10: Good (Table 3.2). This grade is assigned every year by bridge inspectors based on the condition of the bridge in the specific year. They assign a grade for 4 different elements (Deck, Superstructure, Substructure and Culvert) of the bridge's structure and then choose the lowest one as the bridge rating. Flaws of this process are that it is very subjective and can sometimes vary between states or even counties and this could be a potential source of inaccuracies and poor performance of the algorithm.

Feature	**Number of Categories**
Latitude	Continuous
Longitude	Continuous
Skew	Continuous
Route Type	8
Level of Service	8
Position on Highway Network	2
Functional Classification	12
Bridge Age (initial construction)	Continuous
Number of Lanes on Structure	Continuous
Number of Lanes under Structure	Continuous
Average Daily Traffic (ADT)	Continuous
Design Load	10
Approach Roadway Width	Continuous
Historical Significance	5
Actual Operational Status	8
Type of Service on the Bridge	10
Type of Service Under the Bridge	10
Structural Material	4
Structural System	23
Number of Spans	Continuous
Length of Maximum Span	Continuous
Total Structure Length	Continuous
Deck Width	Continuous

Vertical Clearance Above Bridge	Continuous
Type of Structure Beneath	3
Vertical Clearance Under Bridge	Continuous
Direction of Traffic	4
Bridge Age (reconstruction)	Continuous
Deck Structure Type	10
Type of Wearing Surface	10
Type of Membrane	6
Deck Protection	9
Average Daily Truck Traffic (% of ADT)	Continuous
Scour Criticality	13
Future ADT	Continuous
Deck Area	Continuous
Restriction Above the Bridge	2

Table 3.1: Feature Summary of Dataset 1

Score	**Condition**
0-4	Poor
5-6	Fair
7-10	Good

Table 3.2: Classification grading system

A detailed explanation of each and every feature can be found on the NBI dataset guide that is published on their website [6].
It is important to mention that the database was divided in the following way to train and assess the performance of the algorithm: 80% of the data was used for training and validation and 20% was used for testing.

3.1.2 Database 2

For the algorithm that looks at bridge failure, we started gathering some of the data that M.Z. Nasser published in his papers [4] [5]. This data had a lot of interesting features related to the failure of a bridge like: Age, Span, Structural system, Number of Lanes, Location, Material, Incident type. In total, it had gathered data on around 79 reinforced and prestressed bridges that failed due to extreme events. In an effort to boost this very small database, we then went on a mission to find relevant data that could be useful to predict the condition of a bridge subject to a failure event. This took a considerable amount of time and struggle because of the lack of cooperation from the departments of transportation across the United States, most of them responded negatively to the data inquiry saying that these types of data collection were defunded in the early 2010's, not much data was available publicly online. Fortunately, with the help of Mr. Andrew Whittaker from the MCEER at University of Buffalo, I was able to acquire a database from one of their research paper that looked at bridge failures in the United States. The only issue with this database is that it is not as well informed as the original one since it was only used for a statistical study. Considerable data was missing even though the database had a lot more features; there were barely 4 or 5 concrete

bridges that had a complete row of data and that could be useful for our application. To qualify for our algorithm, all of this data had to be completed. Serious effort was put into completing this database, most of which was unsuccessful. Research online only led to articles that talked about the failure happening without mentioning any of the bridge features, unfortunately, only a handful of these accidents were truly documented, and these bridges didn't exist anymore in any of the public records. We also tried contacting multiple departments of transportation in the United States to gather more information about these failure events but the outcome was the same, they either failed to respond or we did not find a way to reach them.
Due to the lack of data and the failure to acquire more information, we decided to integrate bridges with other materials in the training of the algorithm, because even if the bridge is of a different material, the algorithm can still learn the underlying pattern in different features like Age, span number of lanes and location. It is crucial to mention that for features like Age and Span, the range of values are limited to the maximum and minimum values in the database to avoid highly inaccurate predictions in our models. Despite limiting the scope of this application, this is a necessary step to ensure that predictions are realistic.
For the feature engineering of this database, the features that proved to be useful for this algorithm are stored in Table 3.3. All of the noise and rows with missing values were deleted and the textual data was transformed into information using one hot encoding. We finally end up with a database of 315 rows and 7 features. To evaluate the condition of a bridge after a specific failure event, we made the choice to classify the bridges in 2 categories based on the optimization of the results: No/Minor Damage and Major Damage/Collapse. Even though they are not as specific as one would expect, they give a very accurate prediction on what will happen between these 2 events in the most conservative way possible.

Feature	**Number of Categories**
Span (m)	Continuous
Age	Continuous
Number of Lanes	Continuous
Location	3
System	5
Material	5
Failure Event	12

Table 3.3: Feature Summary of Dataset 2

Based on the data collected, we were able to characterize 12 different failure events for this application. Some of which are the following: Earthquake, Fire, Collision, Construction defect and much more. The downside of the data collected is that there is no intensity related to these failures, this means that we cannot accurately grasp and understand the vulnerability of the bridge structures to different magnitudes of a specific event. Subsequently, the failure event designated in this feature can be considered at the average intensity for this specific type.
For this algorithm, the dataset was split in the following manner: 80% for training and validation and 20% for the final testing.

3.2 Development of the two algorithms

Our model is composed of two algorithms which means two different tasks: The first algorithm is a classification problem with 3 classes: Good, Fair or Poor to appraise the structural health of the bridge, these 3 classes are based on the grading system mentioned above. There was also an attempt to try and predict the grade of each bridge with either regression or classification and then assign it to a class based on grade grouping. This approach has a lot of benefits because it can give a deeper understanding of the state instead of just labeling it qualitatively. It ended up not falling through for our overall model because the results were better if we directly targeted the class label, but the knowledge gained could still be useful for another application.
The second algorithm is also a classification problem but this time it is binary with only 2 classes: No/Minor Damage and Major Damage/Collapse to qualify the condition of a bridge after a specific failure event. As mentioned in the literature review, in his research [4] M.Z. Nasser tried separating Major Damage and Collapse to have 3 different classes, the algorithm ended up performing very poorly and couldn't correctly assess the bridges with major damage. Subsequently, we decided that for the sake of my model combining Major Damage and Collapse is a conservative approach that is safer overall for bridge assessment.
In typical Machine learning fashion, once the dataset is prepared it is time to develop a proper algorithm. For those not familiar with machine learning, the concept is the following: We have a dataset constituted of features and a target output, this target output can be a continuous for a regression problem or discrete if the task is classification (The case here); the algorithm basically "trains on the data", in other words, it is very similar to fitting a function between points, the model optimizes a bunch of weights to find a pattern in the features that results in an accurate prediction. Machine learning and artificial intelligence are often compared to a black box, this is why it is important to understand the way every algorithm works to develop a particular relationship between the features. In this section, there will be a simple explanation of a sample of the relevant algorithms trained for algorithm 1, 2 or even for both. The goal is not to detail the way these algorithms operate but familiarize the reader with simple explanations of why they are different.

3.2.1 Neural Network

Neural Networks are by far the most popular and widely used algorithms worldwide for multiple reasons. I use it for both my algorithms and the results are the best for algorithm 2 and second best for algorithm 1. They can be used for a variety of applications, from generative AI to visual recognition. They are very powerful for all kinds of tasks because of their high nonlinearity and capability to closely approximate patterns and relationships. A neural network can be broken down into this very simplistic explanation: It consists of an input layer and output layer with hidden layers in between, each of these layers is constituted of a finite number of neurons; the neuron itself has an input and output as well as an activation function with weights. The goal is to optimize the weights of every neuron with respect to the features and targets of our dataset by backpropagation to compute the gradient. The number of layers and neurons within each layer is completely dependent on the problem and is called architecture of the neural network. One common problem for neural networks is

overfitting, which means that the algorithm struggles to generalize to data outside of the training set because of its very high nonlinearity and complexity. There are multiple ways to combat this phenomenon such as early stopping which stops the training process when the training loss and testing loss diverges or Dropout which randomly drop nodes to force the network to have a redundant representation.
For algorithm 1, the architecture that proved to give me the best results at a testing accuracy of 75% is in Figure 3.1. Unfortunately, this was not enough to beat the best algorithm for this task. For this multiclass classification algorithm, we chose a pretty complex architecture because of the very high amount of data available, also to maximize the testing accuracy, we use a learning rate scheduler that can intervene in the training process to change the learning rate and make the process smoother.

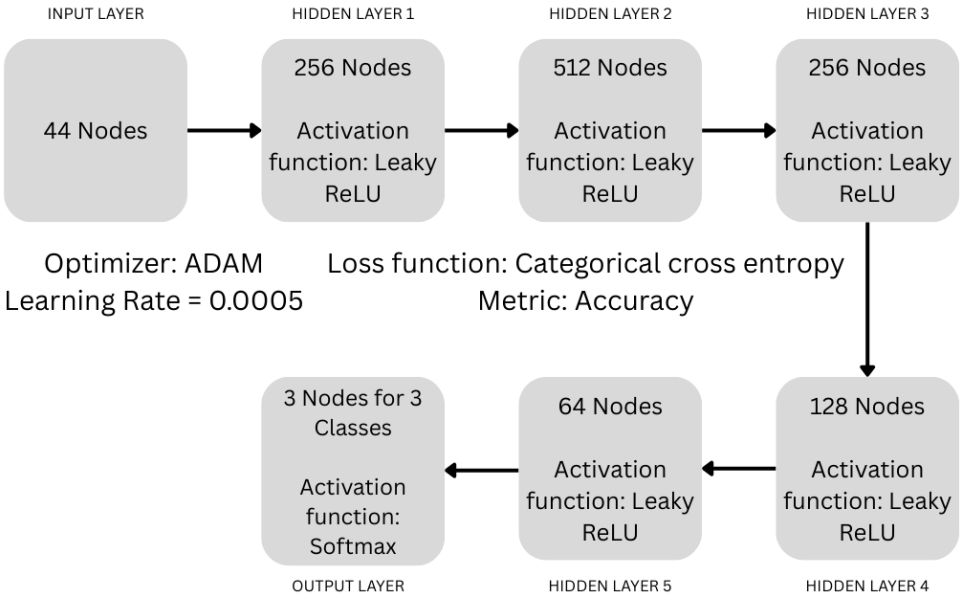

Figure 3.1: Architecture of algorithm 1

For algorithm 2, Figure 3.2 displays the architecture that worked best for this binary classification task. Even though there is more layers, the overall complexity of the algorithm is less than for the first task because of how small the training set is and to prevent overfitting. This was the best algorithm for this task with a testing accuracy of around 80%.

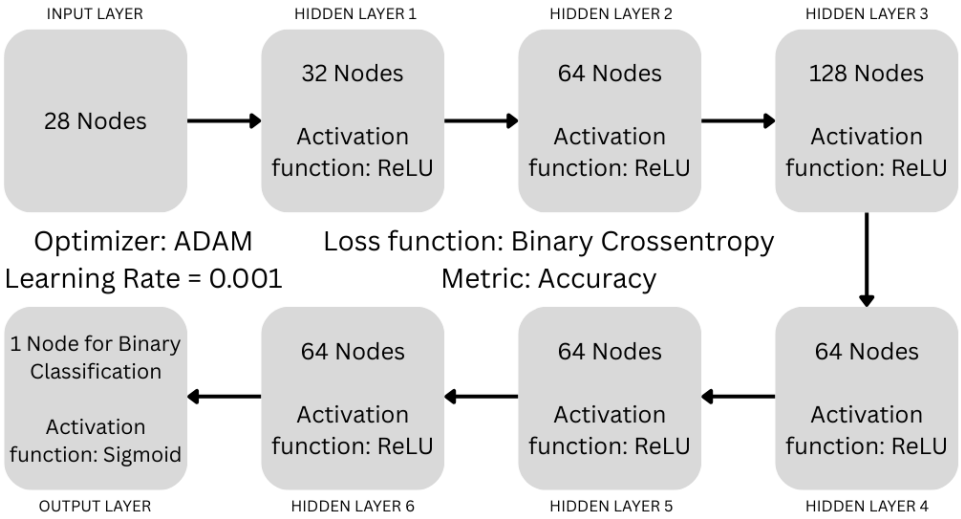

Figure 3.2: Architecture of algorithm 2

3.2.2 Decision Trees

Decision trees is a very popular machine learning algorithm because of how intuitive and straightforward it is to understand. As the name suggests, it splits knowledge in the form of if-then rules at each node, in simpler words it classifies the data based on the value of a feature. This learning model is what is known as a weak learner because of its struggle to grasp complex patterns and nonlinear relationships compared to more developed models like neural networks, they are also sensitive to small perturbations and tend to overfit. They are generally used in an ensemble methods approach which will be discussed later with Random forests and XGB.
Since it is usually outperformed by more complex algorithms, we only tried using algorithm 1 without spending much time on the optimization of the hyperparameters. Results were as expected, a bit less than all the algorithms with a testing accuracy of 72%.

3.2.3 Random Forest

Random forest is an ensemble learning method that assembles multiple decision trees and combines their results to optimize the final output. This algorithm is very efficient in improving accuracy and preventing overfitting because it averages the results from multiple trees. It uses a method called Bootstrap aggregating or bagging which consists of splitting the training set randomly and training it on multiple trees, feature randomness is also employed to ensure that no single feature dominates the model. This algorithm was tested for my first application with the very big dataset. Multiple hyper parameters were tested to optimize the results, and the best combination is the following: 100 estimators and a maximum depth of tree of 15. The results are better compared to Decision trees with a testing accuracy of 74%.

3.2.4 Extreme gradient booster XGB

Similar to random forest, XGB or extreme gradient booster is also an ensemble method that relies on a series of weak learners, mostly decision trees. Boosting is defined as

training learners on the filtered output of other learners. XGB is based on the following framework: it builds decision trees sequentially such that each subsequent tree minimizes the error of the previous trees using a gradient descent approach. This relatively new machine learning algorithm holds various advantages compared to other ensemble methods because of its ability to use deeper trees in the training process, meaning that it can learn the complex patterns better. It offers very high performance relatively quickly because of its parallel processing.
For algorithm 1, XGB held the best testing accuracy compared to its competitors at an impressive 78%. The following hyperparameters were chosen after optimization using grid search:150 estimators, a maximum tree depth of 15 and a learning rate of 0.1.
XGB also performed well for algorithm 2, pushing the overall testing accuracy at: 78% after optimizing the hyperparameters (200 estimators and a max tree depth of 10). Although this score is very good, it still does not beat the neural network for this application.

3.2.5 K nearest neighbors

K nearest neighbors is a machine learning algorithm that is very different from all of what is mentioned above because of its nonparametric approach. Effectively, this algorithm is very fast because it does not require any training since there is no weight that needs to be optimized. It is purposed around the idea that data points of similar classes exist in proximity, which means that they could be classified using majority vote amongst its k nearest neighbors with k being a hyperparameter of this algorithm. This method is very fast but is very restricted because it struggles to adapt to complex patterns. It also requires more computation time to predict an output to compute all of the distances.
Due to the complexity of algorithm 1 and the size of the database, KNN performed poorly scoring 68% testing accuracy with K=3.
Nevertheless, it performed better than expected for algorithm 2 despite the small size of the database, scoring 72% of testing accuracy for a value of K=31. This value of K was found by testing the algorithm for each value of K, and the extracting the model with the best accuracy, this process generates the graphic in figure 3.3:

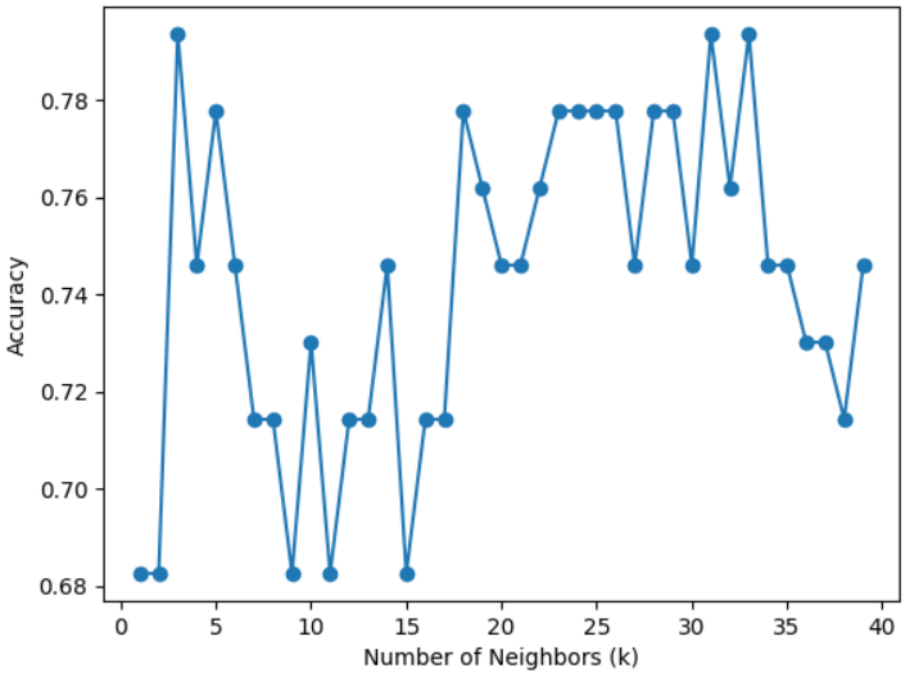

Figure 3.3: KNN Accuracy VS. K Value

3.2.6 Stacking

Stacking is another ensemble method that can boost the results of our model. This tool combines model outputs using a second stage learner like linear regression, it is very useful because sometime two different Machine learning techniques are each good at grasping one area of complexity which means that if we combine their output, we can represent the whole pattern better and end up with better accuracy.
This tool was used for algorithm 1 to combine the outputs of Random Forest and XGB using logistic regression, visualization in figure 3.4. It performed very similar to XGB with an overall accuracy of 77%.

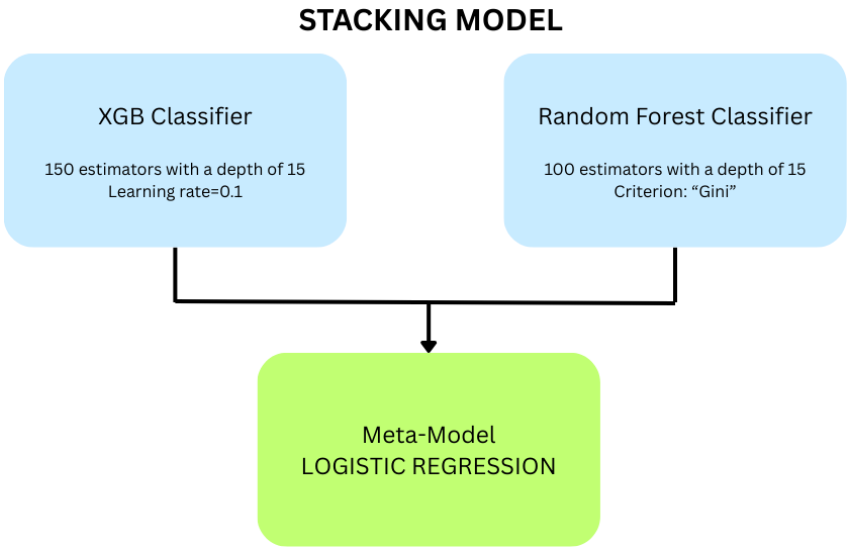

Figure 3.4: Stacking model Algorithm 1

3.2.7 Other Machine Learning Techniques

To boost the result and ensure that the output of both algorithms is optimal, state of the art machine learning techniques had to be implemented. One crucial element for all models is scaling, effectively an algorithm performs a lot better when the data is scaled. Cross-Validation is also a technique that was used to check that the performance of the algorithm was consistent throughout the entire database, the concept is straightforward: the training data is divided into n folds and the algorithm is trained on n-1 folds and validated on 1 for all of them. This method is also useful when the database is small which is the case for our second algorithm, it allows the combination of the training and validation data into one single dataset that is used for both tasks. Grid Search is also a technique that was very helpful to optimize the hyperparameters of every model. We also looked at data augmentation or feature expansion, but it did not work efficiently for any of the algorithms.

4 Results

The outcome of this thesis project is a Machine Learning model that can assess the condition of a bridge based on two different criteria embodied in the two different algorithms. The idea for this tool is to be used as a preliminary way to detect the structurally deficient bridges within a specific database.

4.1 Performance

Although this was already discussed in the development of the different algorithms, this section focuses on the performance of the final model itself instead of looking at individual machine learning techniques. The model is divided in two algorithms:

4.1.1 Algorithm 1

XGB or extreme gradient booster is the choice for algorithm 1. This classifier showed a performance of 78% testing accuracy which is considered very good for such application. The reason why I consider this performance to be enough is because the database used to train and validate this algorithm was rated and classified in a subjective way. Effectively, bridge inspectors had to assign grades for four different structural elements of the bridge and then choose the minimum value to assign a final grade and condition to the bridge, although their grading was based on fixed criteria, the judgement is still a subjective matter that can vary between bridge inspectors and classifying a subjective decision with an accuracy of up to 78% is definitely considered impressive.
Figure 4.1 shows the confusion matrix for algorithm 1 and Table 4.1 presents the accuracy per class.

Class	**Testing accuracy**
Poor	27.5%
Fair	78.1%
Good	80.1%

Table 4.1: Algorithm 1 accuracy per class

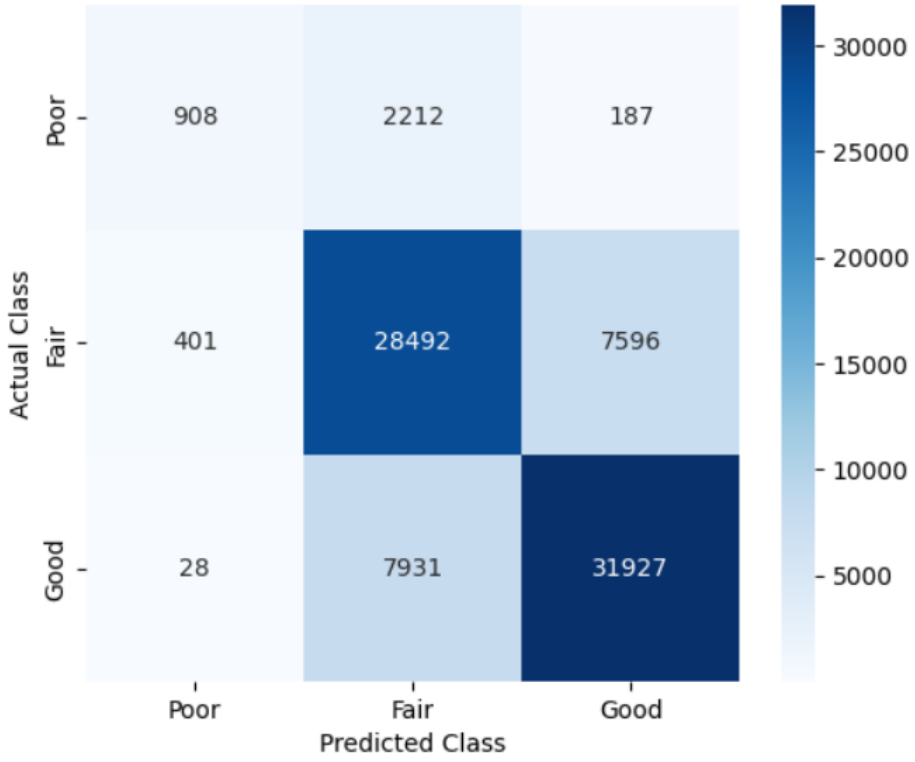

Figure 4.1: Confusion matrix for algorithm 1

Reading both these indicators, we can conclude that the algorithm struggles in correctly predicting bridges that are in poor condition, it only has a testing accuracy of 27.5% for this category . It tends to have a non-conservative approach by classifying bridges in poor condition in the fair category. This is definitely due to the very small amount of bridges in poor condition in the training data, which means that the algorithm does not have enough data to correctly represent the pattern in this class of bridges.
Nevertheless, the algorithm performs very well in other classes with up to 80% testing accuracy.
This performance definitely triggers a discussion on using an alternative distribution of classes based on the grading system. In other words, changing the way bridges are scored or even changing the thresholds that classifies them in the 3 classes could lead to a better differentiation and better understanding of their condition. Or even going as far as classifying them in two classes the same way it was done for algorithm 2 with the uncertainty in the Major Damage class.

4.1.2 Algorithm 2

For algorithm 2, Neural Networks performed the best at finding the patterns in this very small database with a testing accuracy of 80%. As mentioned previously, the main struggle with this algorithm was the very small amount of data available. This led to a lot of uncertainty at first with the accuracy of the results, but after applying multiple machine learning methods like early stopping to ensure that no overfitting is happening, and a learning rate scheduler to optimize this hyperparameter, the results appears to be consistent.
Figure 4.2 shows the evolution of training accuracy and testing accuracy in the training

process of the neural network. This graph is typical of a neural network training process regulated by multiple techniques such as dropout of neurons and learning rate scheduling to avoid overfitting and optimize the testing accuracy. To ensure the results were consistent, the database was shuffled randomly when the training process was started.

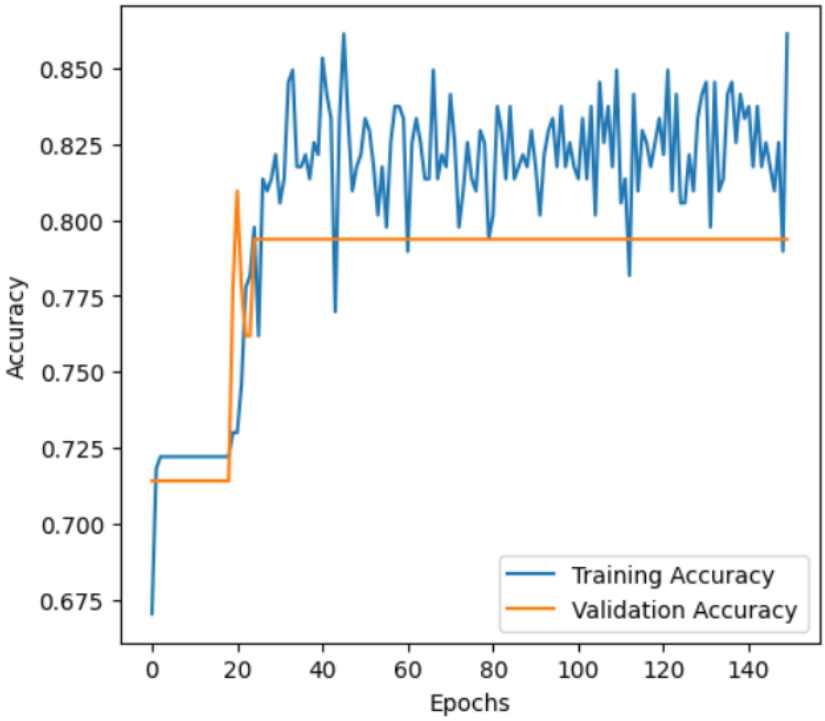

Figure 4.2: Neural Network training process for algorithm 2

Figure 4.3 shows the confusion matrix of algorithm 2 and Table 4.2 shows the accuracy by category. The two categories are No/Minor Damage and Major Damage/Collapse which are labeled on the website in the following way: No collapse for No/Minor Damage and Trouble for Major Damage/Collapse.

Class	**Testing accuracy**
No/Minor Damage	72.2%
Major Damage/Collapse	82.2%

Table 4.2: Algorithm 2 accuracy per class

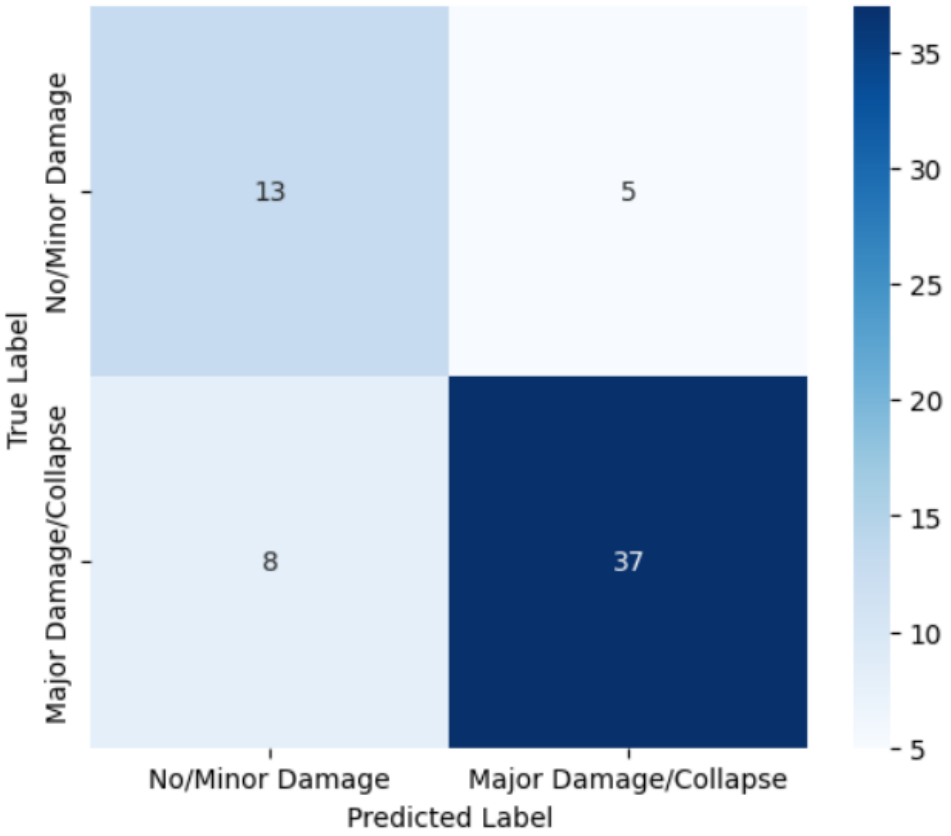

Figure 4.3: Confusion matrix for algorithm 2

It is important to mention that the choice to redistribute the classes this way compare to what was done in [4] was based on the accuracy per class. Effectively, after developing a model for the multi-class classification, the training and testing accuracy for the Major Damage class was extremely low and uncertain at around 20%, this lowered the overall accuracy to 64% and compromised the results. This led us to redistribute the classes in a more efficient way that is more significant for our purpose. These two indicators show that the results for this algorithm are satisfying for our purpose. The testing accuracy of this algorithm is at 80% which is very close to our first model and this leads us to say that the accuracy of our overall model can be estimated at 79%.

4.2 Assembly and Deployment of the model

Now that the two algorithms are developed and optimized, it is time to combine them into one model. The idea is to first appraise the structural condition of a bridge and then test it for a couple of relevant failure mechanisms, and then based on their output, have the user decide which bridges are safe and healthy and which ones are structurally deficient. We can think about the deployment of the model in two different ways: First, the assessment of individual bridges can be done using a website where the user inputs all the features for the bridge that they want to appraise and then predicts the condition based on what was discussed above.

Second, if we had to assess a group of bridges or a bigger database, the algorithm could just be deployed on the entire database at once and output a table with the results for every bridge, this output could also be color coded to flag the bridges that are structurally deficient based on our criteria.
To showcase the results of my work, I have created a website for the individual bridge condition assessment. In this website, the user must fill in the features for the specific bridge that they want to assess. To fulfill the intention of making the website very user friendly, we decided to use a slightly modified version of Algorithm 1. Effectively, with this new version the user would only have to input 27 features instead of 37, slightly compromising the overall testing performance to 77% now. The features that were removed are the following:

- Number of spans
- Total structure length
- Type of service on the bridge
- Level of service
- Deck width
- Type of service under the bridge
- Future average daily traffic
- Deck area
- Direction of traffic
- Type of structure under bridge

The choice of these 10 specific features was educated by an importance study of the features for our extreme gradient booster model. They were the least important features for our algorithm according to the F-score standard or gain measure. Figure 4.4 shows a plot of the importance of each feature for our model.

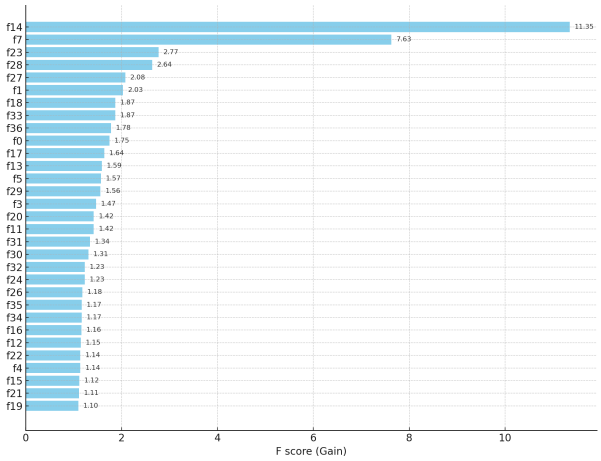

Figure 4.4: Feature importance study

After inputting the data, the user then presses a button to assess their bridge in two steps: first, they appraise the structural health of their bridge based on algorithm 1,

second they evaluate the condition of the infrastructure following the inputted failure mechanism based on algorithm 2. For the second algorithm, we used the XGB model developed instead of the most optimal Neural Network algorithm because of instabilities in implementing the trained NN model to our website. This also results in a slight compromise in the performance of algorithm 2, 78% testing accuracy instead of 80%. Figure 4.5 shows a picture of the interface. The section on the left allows the user to input the features, some of the features are continuous, other displays a selectbox with specific values. The main display showcases a button that must be pressed by the user after inputting all the features to output the predicted bridge condition for both algorithms.

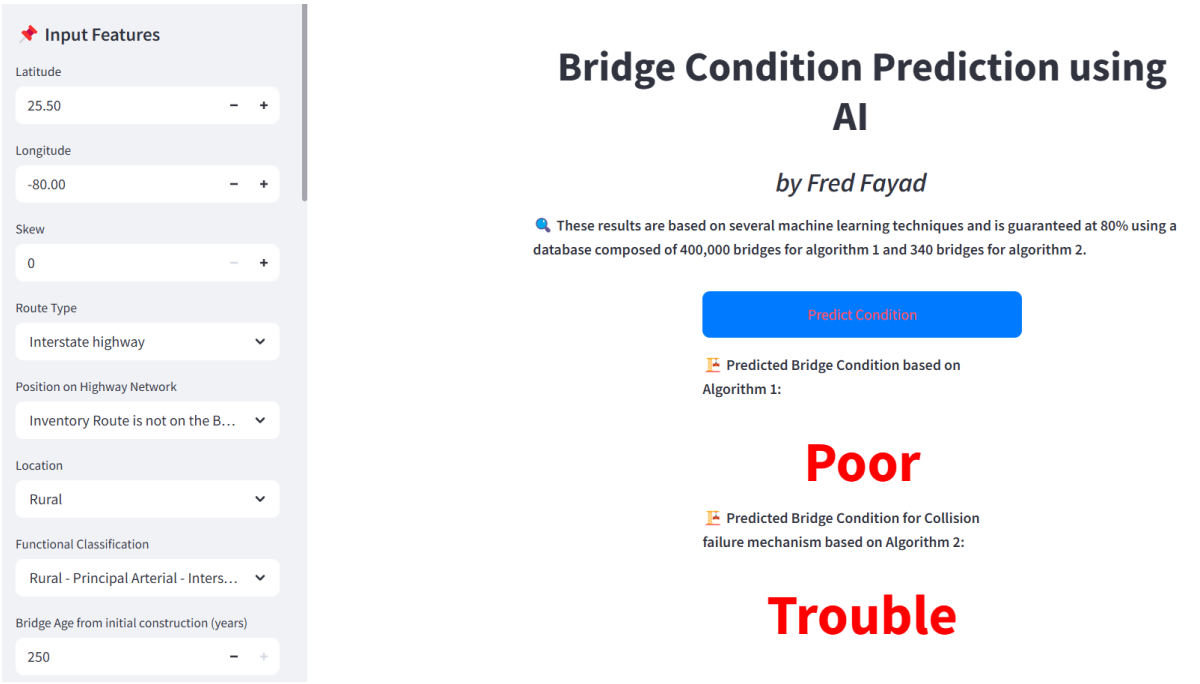

Figure 4.5: Website interface for bridge model

The tool can be explored on the Bridge condition assessment website.

5 Case Study

To show how this model is intended to be used, this section is dedicated to two different case study of concrete bridges in the United States. The goal is to first show how the tool could be used to assess the condition of the concrete bridge and what engineering decisions to base on that, second, a design example will show that the model could also be clever to guide design decisions in the conceptual design phase of a bridge.

5.1 Condition assessment: Sunshine Skyway Bridge

The Sunshine Skyway bridge is a perfect example to show how this model works. This bridge in Florida was rebuilt in 1986 after collapsing of the old one in 1980 because of a ship collision. The prestressed concrete bridge is a pair of long beam bridges with a central tall cable-stayed bridge, with an impressive total length of 4.14 miles.

Figure 5.1: Picture of the Sunshine Skyway Bridge

The features of this bridge required to get an output from our model can be found in Table 5.1.

Feature	**Value**
Latitude	27.620394
Longitude	-82.65579

Skew	0
Route Type	Interstate highway
Position on Highway Network	On the Base Network
Functional Classification	Urban - Principal Arterial - Interstate
Bridge Age (initial construction)	38
Number of Lanes on Structure	4
Number of Lanes under Structure	0
Average Daily Traffic (ADT)	65,215
Design Load	M 13.5
Approach Roadway Width	100.7
Historical Significance	On NHS
Actual Operational Status	Open, no restriction
Structural Material	Prestressed Concrete Continuous
Structural System	Stayed Girder
Length of Maximum Span	1200.1
Restriction Above the Bridge	No restriction above the bridge
Vertical Clearance Above Bridge	9999 meters (No restriction)
Vertical Clearance Under Bridge	0
Bridge Age (reconstruction)	0 (No reconstruction)
Deck Structure Type	Other
Type of Wearing Surface	None
Type of Membrane	None
Deck Protection	Epoxy Coated Reinforcing
Average Daily Truck Traffic (% of ADT)	7%
Scour Criticality	Foundations Stable

Table 5.1: Features of the Sunshine Skyway Bridge

The results are shown in Fig 5.2. We can see that the bridge condition based on algorithm 1 is Good, this means that the bridge does not need an immediate intervention and is not structurally deficient based on algorithm 1. If the bridge was flagged as poor or fair condition, there should be some type of intervention from a party involved with the bridge in the form of a complete structural health assessment to determine if any rehabilitation work needs to be done.
As for algorithm 2, the bridge was tested under a collision failure mechanism which revealed vulnerability and was flagged as troublesome. This means that further investigations and simulations should be done to check if the bridge could potentially withstand any type of impact. On the other hand, if tested under earthquake failure, the algorithm estimates the bridge to be subject to No/Minor Damage.
These results can be interpreted in multiple different ways. The main take from this case study is that this algorithm should be carefully used by competent authorities to run a preliminary check on their bridge infrastructure and potentially flag the structurally deficient bridges based on the two criteria mentioned above. This study should be followed up by more in depth analysis and structural health assessment of the bridges to be on the conservative side.

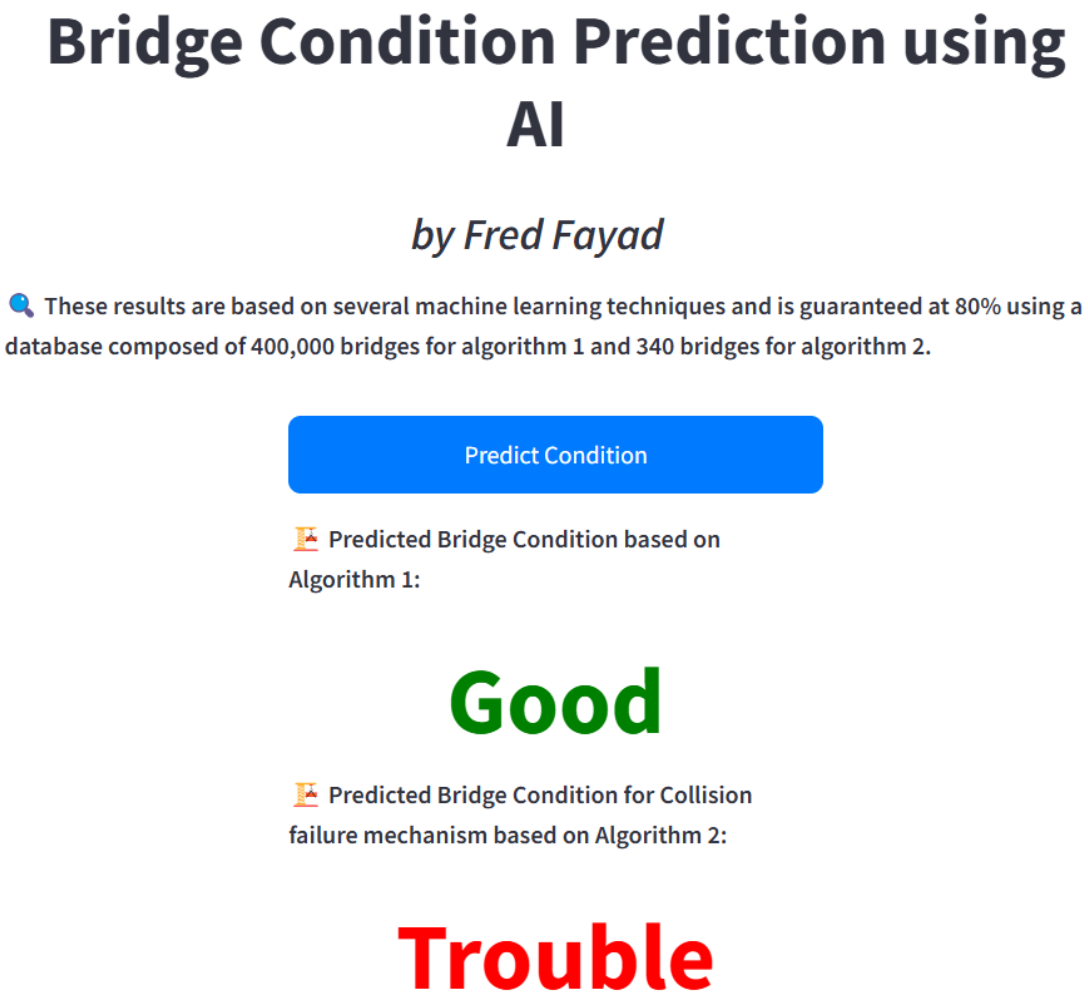

Figure 5.2: Results of the assessment from the model

5.2 Design example: Reinterpretation of The Harvard Bridge

This example is not exactly a case study but more a design guide and discussion on other potential applications of this algorithm. If the city decided to replace the Harvard bridge in Boston (Fig 5.3) by a concrete bridge, some of the features are fixed, but other are parameters based on design decisions which could be guided by our algorithm.

Figure 5.3: Picture of the Harvard Bridge

These are the features that can not be influenced by the design decisions:

Feature	**Value**
Latitude	42.35439
Longitude	-71.091194
Skew	0
Route Type	State Highway
Position on Highway Network	On base network
Functional Classification	Urban - Other Principal Arterial
Number of Lanes under Structure	0
Average Daily Traffic (ADT)	38,638
Approach Roadway Width	51.8
Historical Significance	On National Register
Actual Operational Status	Open, no restriction
Restriction Above the Bridge	No restriction
Vertical Clearance Above Bridge	-
Vertical Clearance Under Bridge	12.5
Bridge Age (reconstruction)	0
Average Daily Truck Traffic (% of ADT)	6%

Table 5.2: Fixed design features of the Harvard Bridge

On the other hand, there is still a very large amount of features that can be based on design decisions. These design choices can be educated by the model that we created in both its algorithms 1 and 2. It can be thought of in the following way: The owner of the bridge wants his structure to stay in the Good category for the next 30 years without doing any kind of major maintenance or service work, he/she asks the bridge designer to base the design on this criteria. Now that the owner fixed the Bridge Age feature, these are the parameters that the bridge designer has to tune to achieve the 30 years good category request:

- Number of Lanes on Structure

- Design Load
- Structural Material
- Structural System
- Length of Maximum Span
- Deck Structure Type
- Type of Wearing Surface
- Type of Membrane
- Deck Protection

Based on his engineering choices for every one of these features, the designer can use algorithm 1 of our model to see what combination can achieve the good category status after 30 years of service. The added value from algorithm 2 can also be extremely useful to guide the design of the bridge to withstand different possible failure scenarios.

6 Closing Remarks and conclusions

One of the key advantages of Machine learning models is that they can be considered as living algorithms. In other words, they are in constant improvement with the amount of data they are fed. We believe that if more data and understanding is fed into these algorithms in the future, their performance could significantly improve and their role can shift from preliminary assessment and conceptual design to actual final assessment and design tools. This is very promising for the future of such applications, especially that the growth of artificial intelligence has become exponential. Unfortunately, the civil engineering and bridge engineering industry is somewhat behind these technological advancements. The industry seems struggling to comply with and understand these new techniques.
This technology has also the potential to be applied to other types of infrastructure such as tunnels and buildings. Effectively, as long as the data is available, a model like this one can be reproduced and employed to assess the condition of other types of infrastructure. There is also a possibility to use the exact same model that we developed, it could be applied to a different dataset, however the hyperparameters would need to be re-tuned because they are very much dependent of the specific data that is fed to the model.
A major struggle in the development of this model was the search for an adequate database. Departments of transportation which are the primary target user of such a model are falling behind in the collection of data related to bridge failure and collapse. From various interactions, it has been put to light that the program related to the understanding of bridge collapse was defunded in the early 21st century.
The following conclusions can be stated from this research:

- Bridges and other types of infrastructure are in critical need for maintenance and rehabilitation to prevent further unfortunate events like failure and collapse.
- There is not enough data gathered around the collapse of bridges. This topic is not understood correctly and still lacks very important knowledge. It is important moving forward to document bridge collapses better to develop an understanding of this topic and help prevent such unfortunate events.
- Machine learning and artificial intelligence can definitely contribute to improving the world of bridge engineering, especially in the condition assessment part. The algorithm created in this paper is a living proof of this statement, bringing the two worlds closer while serving a very important purpose of flagging the structurally deficient infrastructure.

Bibliography

[1] A. Azizinamini, "Accelerated bridge construction," *J. Bridge Eng.*, vol. 25, no. 12, p. 02020002, Dec. 2020.

[2] K. Wardhana and F. C. Hadipriono, "Analysis of recent bridge failures in the united states," *J. Perform. Constr. Facil.*, vol. 17, no. 3, pp. 144–150, Aug. 2003.

[3] W. Fan *et al.*, "Machine learning applied to the design and inspection of reinforced concrete bridges: Resilient methods and emerging applications," *Structures*, vol. 33, pp. 3954–3963, Oct. 2021.

[4] M. Z. Naser, "Can past failures help identify vulnerable bridges to extreme events? a biomimetical machine learning approach," *Engineering with Computers*, vol. 37, no. 2, pp. 1099–1131, Apr. 2021.

[5] M. Abedi and M. Z. Naser, "Rai: Rapid, autonomous and intelligent machine learning approach to identify fire-vulnerable bridges," *Applied Soft Computing*, vol. 113, p. 107896, Dec. 2021.

[6] Federal Highway Administration, "National Bridge Inventory," https://infobridge.fhwa.dot.gov/Data, n.d., accessed: 2025-04-11.