Predicting Delamination in Concrete Bridge Decks from Ground Penetrating Radar Signals using Machine Learning

Authors

- **Ishfaq Aziz**
  0000-0001-5723-0331 · ishfaqa2

- **Jesus Castro**
  0000-0002-7684-7422 · jcastr54

- **Chirayu Kothari**
  0000-0003-4468-1138 · ck0103

- **Omar Abdelrahman**
  0000-0003-0618-8481 · oaa4

Introduction

Visual inspection of structures provides valuable information for QC/QA purposes, but its value is proportional to the knowledge and experience of the inspector. Still, visual inspections are limited to the features visible on accessible surfaces. Exploratory investigations (by destructive methods) have been conducted to ensure the internal conditions of structural elements under the assumption that the obtained results represent the conditions of an entire section (or region) in the structure. Then, the need for nondestructive methods arises. Nondestructive methods allow the evaluation of internal conditions without affecting the surface conditions. These methods usually rely on the acoustics and electrical properties of the materials.

Ground Penetrating radar (GPR) is a nondestructive method that relies on the dielectric properties of materials and can be used (in civil engineering practices) to locate steel and internal defects in concrete pavements. GPR is an effective tool for subsurface inspection and quality control on engineering construction projects. The survey method is rapid, nondestructive, and noninvasive. Interpretation of GPR data commonly helps to evaluate and measure different properties of a concrete structure.

The data collected from the GPR are usually presented in two different formats, A-scan, and B-scan. Most modern devices can present both formats simultaneously. The A-scan is the raw energy signal received by the antenna shown as a function of time and signal strength (amplitude). The received signal in Figure 1a is an example of an A-scan. The B-scan, also known as radargram, is constructed from the sequence of multiple A-scans related to the antenna's position, as shown in Figure 1b, where the depth is represented on the y-axis, and the survey distance is shown on the x-axis orthogonal to the y-axis. The amplitude of the received signal is often shown as a color-coded intensity plot, often in grey, as shown in Figure 1c. B-scans are usually visually inspected to identify and locate any delamination within the concrete or to locate the reinforcement rebars.
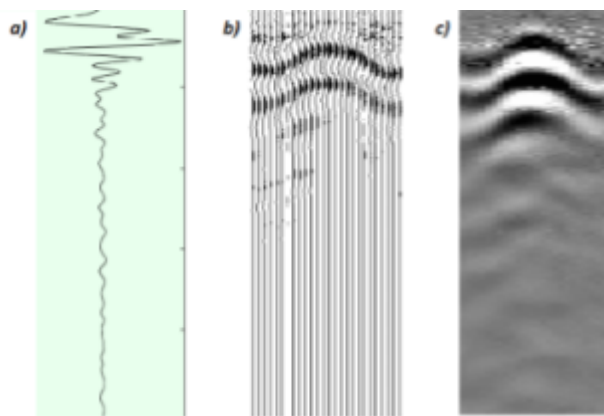

Figure 1: a) Single wave signal (A-scan). b) Collection of signals across a distance along the surface. c) Color intensity plot of (b) (B-scan). [2]

Usually, the GPR is used to detect delamination defects in the pavements. Delamination defects can be under or over the rebars. It is very difficult to observe the defects which are under the rebars. Here, we analyzed a GPR data set that was prelabelled for three classes containing no defects and delamination over the rebars and under the rebars. Models like decision tree, random forest, regression, and convolutional neural networks were used to predict the classes based on the A-scans. The input of the model was either the raw scans, principal components, or the dominant frequency and amplitude in the FFT domain.

Description of Dataset

The data was obtained from https://commons.und.edu/data/19/ [1], which contains annotated dataset for the structural defects. Data is present in the. xlsx format. Each column in the dataset has an approximate length of 512, where the first column represents the time (ns), and the rest of the columns are all amplitudes of GPR signals. Hence, each raw signal (columns) is directly annotated to one of the three possible classes. In addition, other information like the scan length and the coordinates of each signal scan is also provided.

The data set represents raw signals of Non-Destructive Evaluation (NDE) tests conducted by Ground Penetrating Radar (GPR) which were collected from in-service reinforced concrete bridge decks. The dataset was annotated using three classes by bridge deck repair: Class 1-No Delamination, Class 2-Delamination (delamination above top bar mat), and Class 3-Delamination (delamination below top bar mat).

Exploratory Data Analysis

Preliminary Analysis

The dataset consists of 16,384 GPR scans taken at different locations along the bridge. Each scan has a total of 512 points representing a scan time of 12 ns. The scans are categorized into class 1 – no delamination, class 2 – delamination above the top rebar, and class 3 – delamination below the top rebar, representing 8427, 6812, and 1144 scans, respectively, as shown in Table 1. Other properties of the GPR scans are summarized in Table 2.

Table 1. Summary of classes/labels of the dataset

Classes	Class 1	Class 2	Class 3
No of scans	8427	6812	1144
Meaning of each class	Sound Concrete	Concrete with delamination above top rebar	Concrete with delamination below top rebar
Peak amplitude of mean signal	10121	9813	9649

Table 2. Properties of the GPR scans

Parameter	Value
Total number of scans	16,383
Total Length of scan	363.3 ft.
Interval between subsequent scans	0.0222 ft.
Time duration of each scan	12 nano seconds
Data points per scan	512
Sample rate (calculated)	42.67 GHz

The amplitudes of the signals in the dataset were found to have high numerical values. The average amplitude of the whole dataset is around 33,000. So, to make the data more symmetric around the x-axis, the average of the first few nanoseconds of readings was subtracted from the whole dataset, resulting in scans that start with amplitudes close to zero. Figure 2 shows one randomly selected scan from each of the three classes.

Figure 2: Randomly selected signals from each of the three classes

Based on the reference paper used to obtain the data [1], the B-scan, a visual representation of a combination of individual scans stitched together, showed that the bottom rebar reflection was detected at around 7 ns. Thus, it was decided that the readings after around 8-ns would not be useful for our model, and the remaining 4 ns were removed. Also, it was observed that the initial reflection from the top surface of the concrete is detected around 2 ns, so the first 2 ns were also removed from our data. Sample scans from the resulting signal are shown in Figure 3.
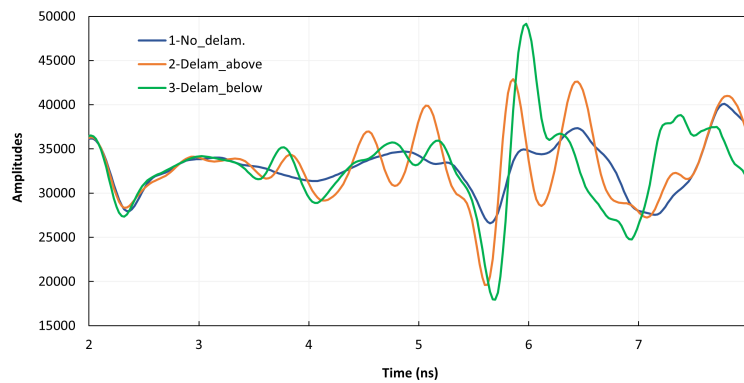

Figure 3: Randomly selected signals from each class, showing only the data points between 2 and 8 ns.

Mean Plots

To visually distinguish between the three different classes of data, the mean of all scans from each class is plotted in Figure 4. The figure shows that the three means look almost identical, so there is no distinctive feature in the time domain signal that can help assign new signals to any of the three classes. Thus, deeper levels of data analysis are required to identify any distinctive features that can help classify new data.
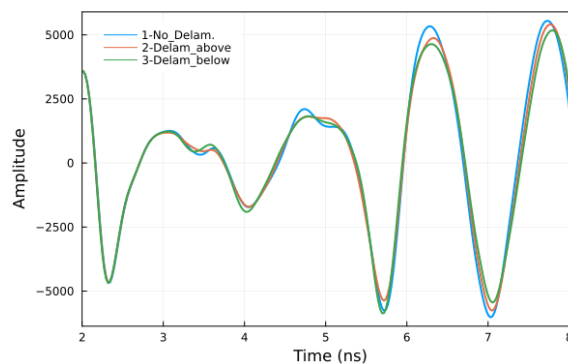

Figure 4: The mean of all signals for each class.

Fast Fourier Transform (FFT)

Another approach to visualizing sinusoidal time-series datasets is by identifying dominant cyclic patterns. This can be conducted with the Fast-Fourier Transform, a computationally efficient way to calculate the discrete Fourier Transform from a dataset. This algorithm transforms information on the time domain into the frequency domain.

The time-series datasets from the three classes (no delamination, delamination above rebar, and delamination below rebar) were all transformed into the frequency domain to identify patterns that could easily differentiate one class from another. As observed in Figure 5, most of the signal amplitude peaks are observed at a frequency of approximately 0.8 GHz, which is in the standard operating frequency range of GPR antennas for concrete testing.

Figure 5: Amplitude vs Frequency Class 1 with no delamination

In Figure 6, it is observed that in the sections where delamination was identified above the top rebar layer, the amplitude peaks were higher in magnitude than the observed peaks in the areas with non-delaminated sections at similar frequencies. On the other hand, the plot corresponding to the amplitudes of sections identified with delamination below the top rebar later (Figure 7.) displays amplitude peaks with magnitudes lower than in Figure 5 and Figure 6.

Figure 6: Amplitude vs Frequency Class 2 with delamination above the rebar

Figure 7: Amplitude vs Frequency Class 3 with delamination below the top rebar

From this data processing approach (FFT), a good differentiation from the three classes (beyond the observed magnitudes in amplitude at approximately 0.8 GHz) cannot be concluded. Furthermore, the maximum amplitude of the FFT data and the frequency corresponding to the maximum amplitude was plotted (Figure 8). It was observed that the maximum amplitude of most of Class 1 and Class 2 was similar. However, some of the FFT spectra in Class 2 had higher amplitude than that of Class 1. Also, some of the FFT spectra in Class 2 had amplitude close to the mean of Class 3. It can be attributed to the labeling of data.

Figure 8: Violin plots representing the variation in maximum FFT amplitude of the three classes.

Similarly, the frequency at maximum amplitude was observed for the FFT spectra in the three classes (Figure 9). The mean frequency at maximum amplitude for class 3 was lower than that of class 2 and class 1. The Violin plot for class 1 and class 2 look similar. However, some of the FFT spectra in class 2 have a frequency at a maximum amplitude lower than that of class 1's data.

Figure 9: Violin plot representing the variation in frequency at maximum FFT amplitude of the three classes.

It can be easier to separate Class 3 from Class 1 and Class 2 based on the FFT analysis.

Principal Component Analysis

Principal component analysis (PCA) was performed on the dataset to change the basis of the data and improve its interpretability. The number of modes was selected to be 2. The results of the first and second modes of the PCA data of the whole dataset are plotted in Figure 10. Since the number of data points for each class is different, normalizing the number of points might provide a better visual representation of the scattering of data in different classes. So, 1000 random points were selected from each class, and then PCA was performed on this sub-dataset. The results of the two modes of this PCA are shown in Figure 11.

Figure 10: The PCA data for all the datasets.

Figure 11: PCA data of two different 1000 randomly selected points from each class.

Predictive Modeling

Based on the exploratory data analysis, it can be concluded that the differentiation of the classes is challenging. The analytical approach to determining key features for distinguishing different classes is not conclusive, and overall, the data delineation is not observed. Hence, four different types of models, including the Convolutional Neural Network (CNN), were used to classify the delamination into three categories. The dataset was divided into training and testing data.

The four models used in the preliminary predictive modeling are 'Decision Tree', 'Random Forest', 'Regression', and '1-D Convolutional Neural Network'. These models' explanations and the corresponding results are discussed below.

1. Decision Tree

The first predictive model applied was the 'Decision tree' model, a non-parametric supervised learning method that can be used for classification or regression. Deeper trees lead to more complex decision rules and, therefore, better model fitting.

Decision tree in machine learning has the advantage of being simple to understand, interpret and visualize. The model can be visually represented by a flowchart of decisions. It can also implicitly perform variable screening or feature selection to determine the most dominant variables that can govern the classification more than the others. However, it also comes with some limitations. For instance, decision tree learners can create over-complex trees that do not generalize the data well, causing data overfitting. In addition, the predictions of decision trees are neither smooth nor continuous, so they do not perform well with extrapolation.

The decision tree classifier was created using 3 labels -one for each class- and 12 variables. These variables are the 10 first PCA modes of the data, the dominant FFT frequency, and its corresponding amplitude. The first 10 PCA modes were selected as they cover more than 90% of the variance. Then, leaves having more than 90% combined purity were merged before applying the model to avoid overfitting. About 60 % of the data from each class was used for training the model. Then, the model was tested on the complete dataset.

The summary of the results is shown in Table 3. The accuracy achieved on training data for classes 1, 2, and 3 were around 86%, 83%, and 46%, respectively. This resulted in an overall accuracy of about 82% on the training data. Due to the lower number of Class 3 data points, the model prediction accuracy is lower for class 3 than for the other two classes. The confusion matrices achieved by using this model are shown in Figure 12. When applied to the complete dataset, the achieved accuracy was around 67%, which means that the accuracy of the unseen 40% of data was about 45%.
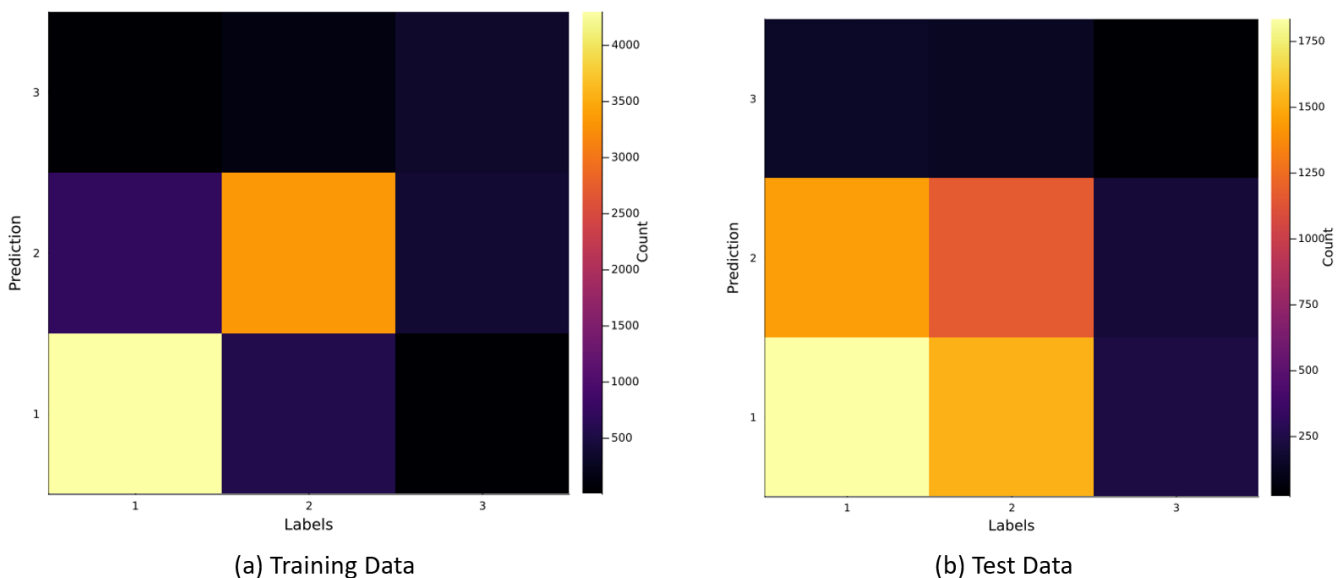

(a) Training Data   (b) Test Data

Figure 12:   Confusion matrix for the 'Decision Tree' model

2. Random Forest

To avoid the problem of overfitting, we have also tried to apply the 'Random Forest' classifier. A random forest is a meta-estimator that fits several decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The parameters used for building our model are listed below:

• Number of random features = 6

• Trees = 10

• Portions of samples per tree = 0.6

• Maximum tree depth = 17

The accuracy obtained by the model using the numbers mentioned above was around 64 %. Values of these parameters were changed to default values (default of the package: DecisionTree.jl), and then the new model was generated and tested. With the default values, the achieved accuracy was around 66 %. The confusion matrices obtained from this model is shown in Figure 13.
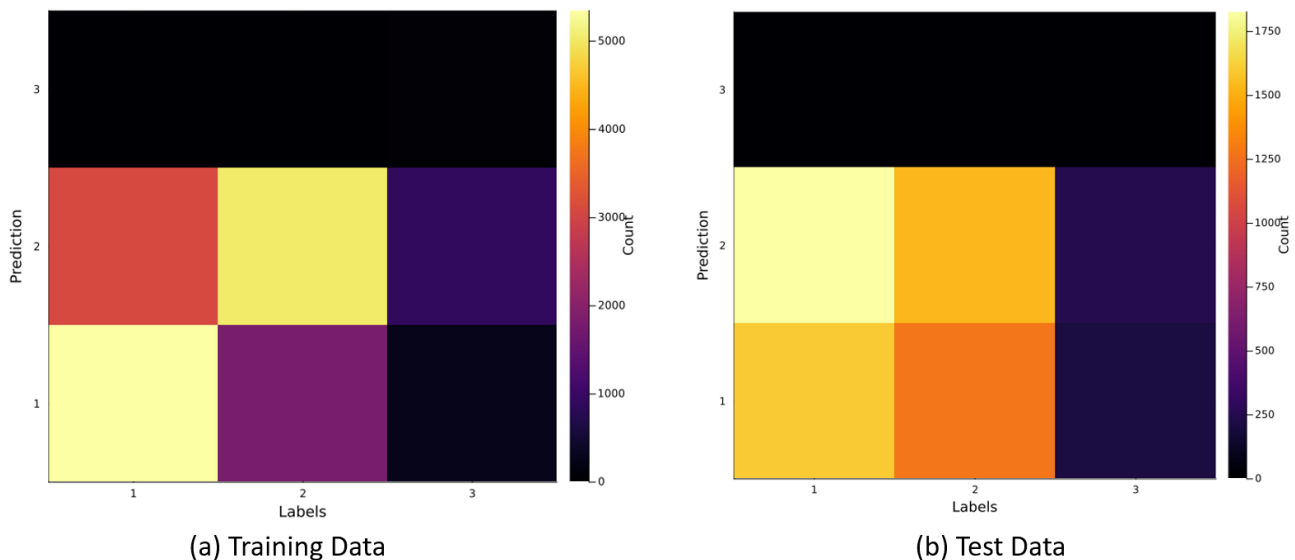

(a) Training Data   (b) Test Data

Figure 13: Confusion matrix for the 'Random Forest' model

3. Regression

Regression is a method for creating a model of the relationship between one or more independent variables and dependent variables, where variation in the dependent variables is used to explain variation in the dependent variables. The regression method prepared for this project considered a total of 260 independent variables and 1 dependent variable (classification). The regression model was optimized by minimizing a maximum square error function used a learning rate of 0.001 and 10,000 steps. 60% of the entire data set was used as training for the regression model.

Ideally, regression model results should match actual observations. As observed in Figure 14(a), similar values were predicted for the three observed classes, which enables this system alone to accurately classify the predictions. These findings, plus observations from the confusion matrix in Figure 14(b), confirm that the regression model is inadequate for a classification model.
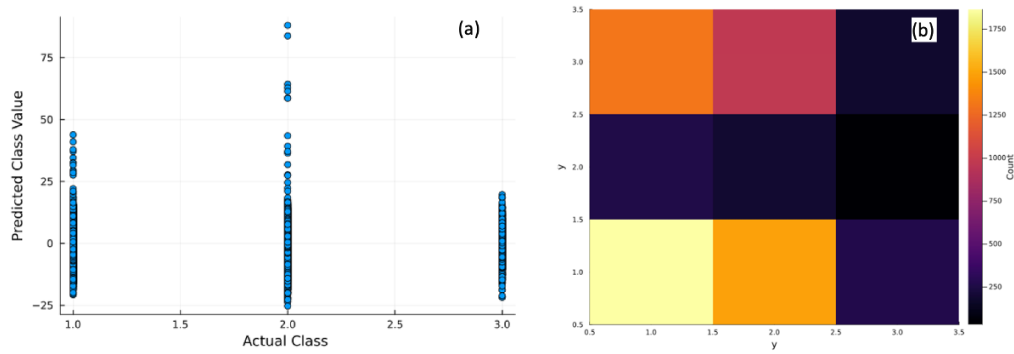
Figure 14: (a) Class Predictions vs Actual Classes, from regression-based machine learning model. (b)Confusion matrix for the regression-based model

4. Convolutional Neural Network

The Convolutional Neural Network (CNN) was created for the GPR data at different locations. Out of the approximately 16000 signals, around 9800 random signals were used to train the model. Each signal has 260 data points. The 1D CNN is composed of several layers. The first layer was a convolution layer with a filter size = 5 with 1 input channel to 8 output channels of training data. The padding was of size= 2 and Relu was the activation function. The next layer was a Maxpool layer of size= 4. The third layer is another convolutional layer with a filter of a size = 5 and padding of size= 2 with 8 channels to 3 channels. The fourth layer with a Maxpool layer of size =5 and the fifth layer is a dense layer of size (39,3) based on the input size and the number of channels. Finally, a Softmax layer is applied to discriminate between different classes. The batch size was 10 and learning rate was 0.001.

Figure 15 show some diagnostics of our model training, include loss and accuracy curves as a function of gradient descent step and a confusion matrix of the final results. The accuracy of 52% is achieved based on the convolutional neural network. As observed in the figure , the accuracy becomes constant after certain epochs and does not improve. The model was further executed for different channels as well as different activation functions like Relu, Tanh, and its combination. However, the output was close to 52% accuracy with no improvement. The rest 40% of the total data was tested based on the CNN model. The accuracy was again 52% and the correlation matrix for the testing data was similar to that of the correlation matrix for the training data(scales on the color bar are different as the number of datasets for training and testing data is different). Further modification and improvements in the CNN model are required.
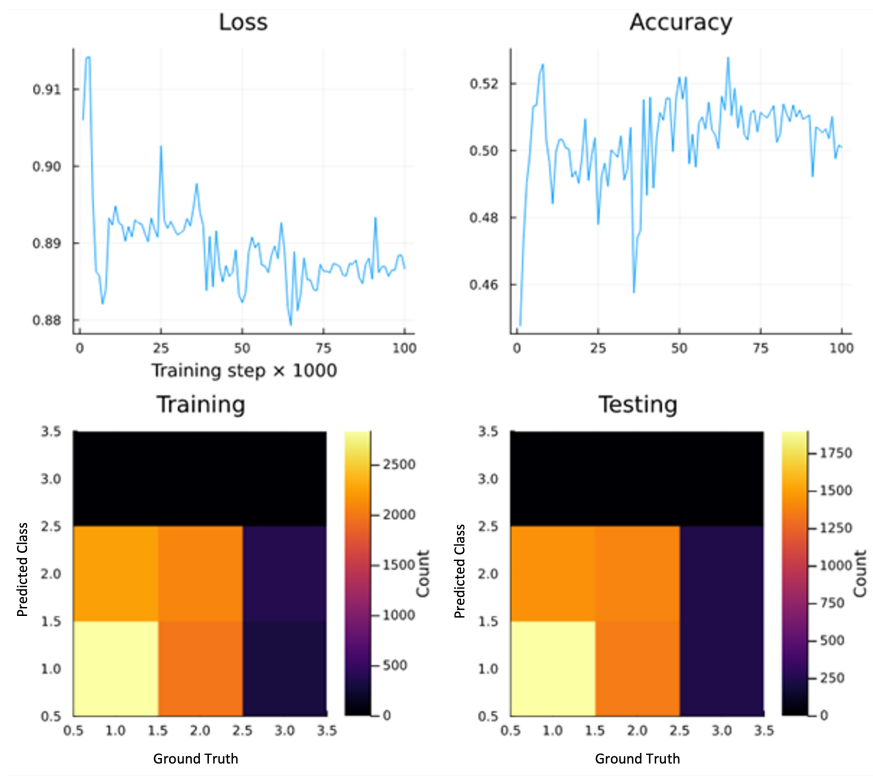
Figure 15: Performance of the CNN model. It shows the loss and accuracy curves as a function of gradient descent step and a confusion matrix for both the training data and the testing data

Discussions

As mentioned, the data is classified into three (3) different labels/classes. However, the amount of data samples for the three classes is not the same. For example, class 3 represents only about 7.5% of the entire dataset, while class 1 is approximately 50%. Table 3 provides an accuracy-based summary of the effectiveness of each of the studied predictive models. The accuracy of predictive models is linked to the sample size of each class. An imbalanced dataset may cause predictive models to be biased towards learning more about the dominant classes causing higher accuracy of prediction of these classes. Moreover, further improvement in CNN model by optimizing the model parameters and layers is required. The regression model's accuracy showed the lowest accuracy. Decision Tree and Random Forest models provided relatively high accuracy levels on training data (above 75%), 1-D CNN model provided the highest accuracy on unseen data with 52%.

Table 3. Summary of results of predictive modeling

Models	Training data	Accuracy On Training data	Accuracy On Unseen data
Decision Tree	60 %	82 %	45 %
Random Forest	60 %	76 %	49 %
Regression	60 %	33 %	34 %
1-D CNN	60%	52 %	52 %

Based on the accuracy values of the models created in this study, it can be concluded that these models need to be improved to reach satisfactory predictions. One of the future recommendations to improve the accuracy of the models is to train the models on larger datasets. This can be done by using other datasets that were obtained by scans performed on other bridge decks in the same project. Training the models on larger datasets can possibly reduce the effect of overfitting and can make them perform better when tested on unseen data. Another strategy is to increase the number of parameters used in the model as independent variables, for example, using a higher number of PCA modes. An example of the first strategy was tried out on two of the models: decision tree and random forest. Details of these trials can be found in the supplementary information section.

Supplementary Information

Modeling on a larger dataset:

As mentioned earlier, only one dataset file from the data source [1] (that contains 16,383 A-scans) was used for all our analysis and predictive modeling. As a part of further analysis, we used 30 dataset files that had 218,626 A-scans to build a 'Decision Tree' and a 'Random Forest' model. Of the total A-scans, approximately 71 % are from class 1, 23 % from class 2, and 6 % from class 3. Instead of the previously used 12 features, 22 features were used which include the first 20 PCA modes (covering 97 % variance), dominant FFT Frequency, and corresponding amplitude.

The default values of settings of the 'Decision Tree' package were used for modeling. The results are shown in Table 4, Figure 16, and Figure 17. It can be observed that both the models overfit the training data and the models have lower accuracy on the unseen data. However, the 'Random Forest' model performs better on the test data. The dataset is highly dominated by the class-1 (>70 %) data for which the models tend to be biased towards class-1. Similarly, the number of class 3 data is very small (5.6 %) for which the error in predicting unseen class 3 signals is very high.

Table 4. Results of the predictive modeling on a larger dataset

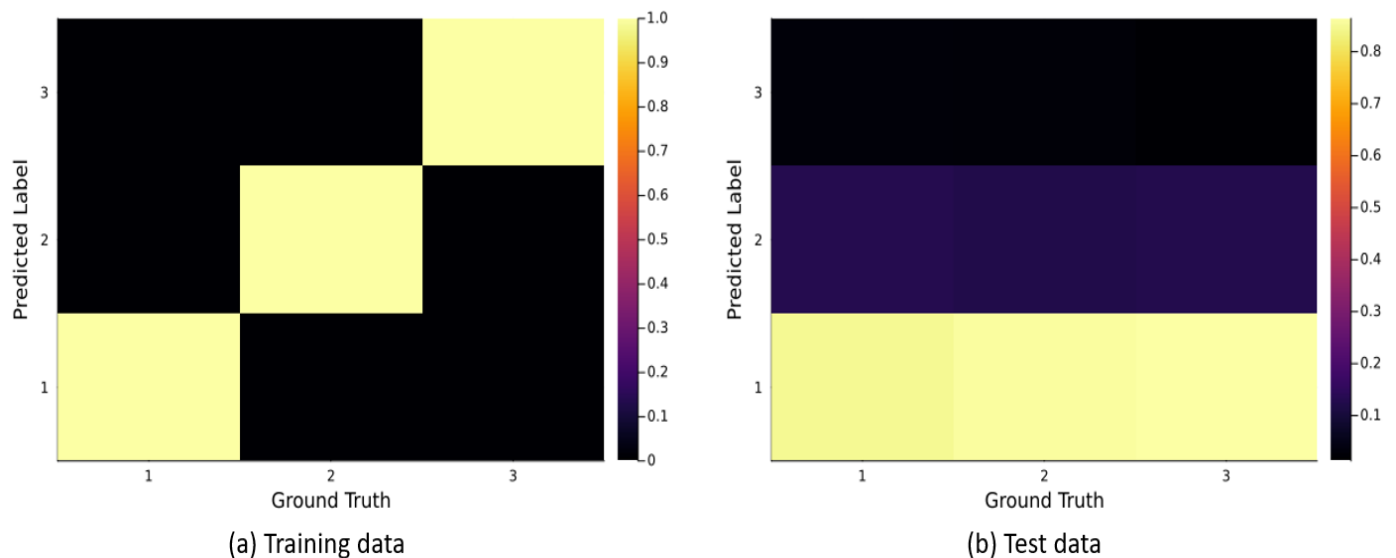| Models	Accuracy on Training data	Accuracy On Test/Unseen data
Decision Tree	100 %	58.2 %
Random Forest	94.7 %	71.3 %

(a) Training data   (b) Test data

Figure 16: **Normalized confusion matrix for the 'Decision Tree' model on the larger dataset**
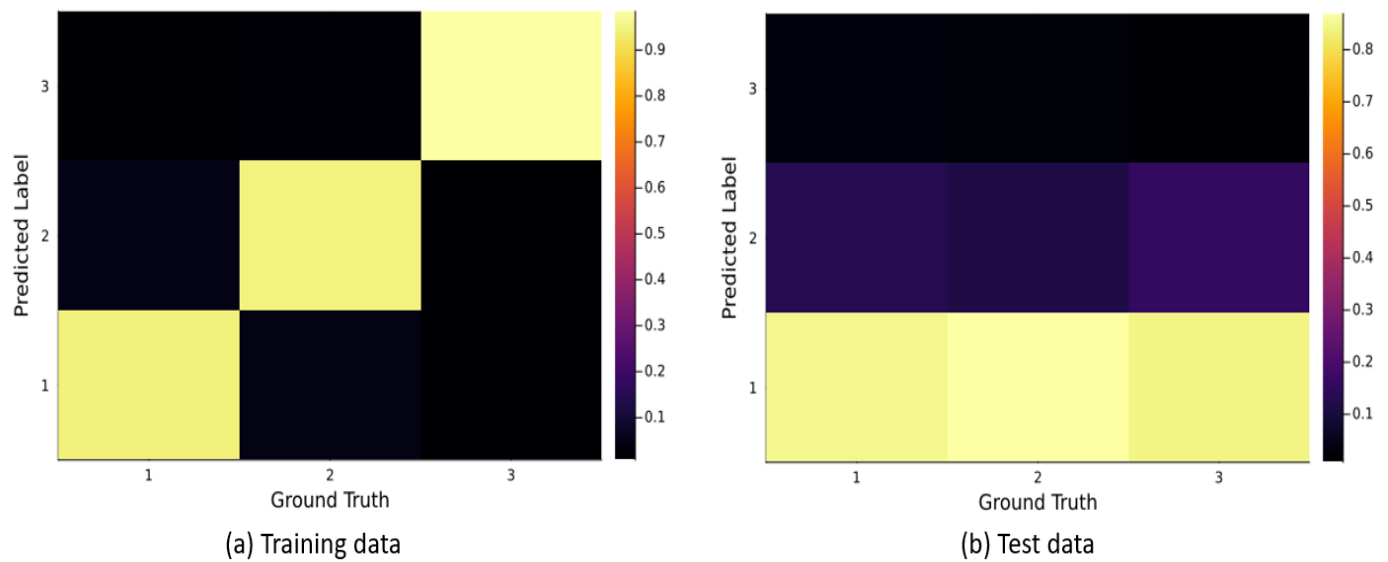
Figure 17: Normalized confusion matrix for the 'Random Forest' model on the larger dataset

References

[1] Ichi, E., & Dorafshan, S. (2021). SDNET2021: Annotated NDE Dataset for Structural Defects, 10.31356/data019

[2] Scheers, B., 2001. Chapter 2. Ground penetrating radar. In: Ultra-Wideband Ground Penetrating Radar, with Application to the Detection of Anti-Personnel Landmines. Brussels: s.n., pp. 2-1 -2-43.