An extensible decision support system for bridge maintenance

Jeremy Barisch-Rooney

October 16, 2019

Contents

1 Introduction

The probability of a bridge to fail increases over time until it is no longer considered safe for use. Maintenance of a bridge is typically carried out when something goes wrong or according to a preventative maintenance schedule based on expert knowledge, neither approach making the best use of limited maintenance resources. Sensors can provide real-time information without the delay or cost of a manual maintenance check. The question of how to build an extensible decision support system (DSS) for bridge maintenance is answered in this thesis. What techniques do and don't work? And what are the costs and benefits of installing such a system on a bridge?

A **DSS for bridge maintenance** (BDSS) is a software system that provides the user of the system with information on the current status of a bridge. The provided information should enable the user of the system to make a more informed decision about when and/or where maintenance should be carried out. The provided information can include real-time sensor data and an analysis thereof. The primary aim of the analysis techniques applied to sensor data in this thesis is to classify data in two states, a normal or healthy state, and an abnormal or damaged states.

In the development of a BDSS it is **necessary to have sensor data** cor-

responding to both the damaged and undamaged states of the bridge in question. Due to the expensive nature of a bridge it is usually not allowed to impose a damaged state on the bridge. The use of finite element (FE) software allows us to create simulated sensor data via a finite element model (FEM) of a bridge on which different damage states can be imposed.

A FEM is a model of a structure in software. All models are wrong and thus the simulated sensor data generated by the FEM will **differ from sensor data collected from sensors installed in real life**. The simulated sensor data should be close enough to reality to provide confidence that the anaysis techniques in this thesis could also work with real sensor data, while acknowleding that some additional work would likely be needed to tune the anaysis techniques once real sensor data is available. After all, "in theory there is no difference between theory and practice, while in practice there is", TODO

A finite element simulation of a bridge can produce sensor data that is necessarily different from data collected from real installed sensors. It is necessary to **validate the simulated data against some ground truth** in order to have some confidence in the accuracy of the simulated data, and it is also desirable to **test the developed analysis techniques on real data** such that we have some confidence in the techniques, for when a BDSS would eventually be. installed in real life.

A validated FEM of bridge 705 in Amsterdam was available at TNO for the FEM program Diana. This allows the programmatically generated FEM for OpenSees to be validated by comparing sensor data collected under a specified load from the OpenSees model of bridge 705 to sensor data collected from the validated Diana model under the same load.

While we do not have bridge data availabe from a normal and abnormal state we do have data from viaducts for which there are two states available, high and low temperature. In this thesis the **analysis techniques are tested** on this data to provide, an albeit limited, test that the techniques can perform a classification between states.

Extensibility is a measure of the ability to extend a software system. According to TODO

'An important kind of reuse is extensibility, i.e., the extension of software without accessing existing code to edit or copy it." The research in this thesis is not just reproduceable but also extensible. The benefit of this research being extensible is that an interested party should be able to download the

software and perform research on another bridge or investigate how a new classification technique performs on sensor data, without having to start from scratch. By creating an extensible system it is my hope to facilitate further research in this area and prevent duplication of effort.

For a software system to be extensible, the source code must be available to any user wishing to extend said software. The benefits of **open source software** are well known, in particular open source software allows *any individual with an interest* to develop or *extend* the software. Open source software can thus leverage the knowledge of the community and prevent duplication of efforts which can occur when software is developed behind closed doors. Open source software also provides transparency to anyone wishing to investigate the software and may produce more reliable software due to more people having eyes on it.

OpenSees (The Open System for Earthquake Engineering Simulation) is an open source finite element software framework that anyone with a Windows, macOS or Linux machine, and an internet connection, can download and install. Depending on open source finite element software enables users to explore or extend the decision support software without requiring an expensive proprietary licence. In contrast to the Diana finite element software package, also used in this thesis, OpenSees does not require a licence for use and is additionaly available for macOS users.

This thesis first gives an overview of existing research into machine learning approaches for structural health monitoring (SHM), decision support systems and classification techniques. The methods section presents an in-depth description of how an extensible system is created for the collection of simulated sensor responses, how the inputs to this system should be structured, and what form the data-driven classification experiments will take. In the results section we take a look at the data generated by the data collection system, analyze the results of the classification experiments, and finally, present the costs and benefits of installing a decision support system for bridge maintenance.

2 Abbreviations & Terminology

3 Background & Motivation

{This is where I tell the story, why I am doing this (motivation) and what has been and not been done before (background)}

3.1 Extensibility

In order for the developed DSS to be truly extensible it is not limited to depend on a single finite element program. The system has as a parameter a method of communication with a finite element program, such that data can be collected and analyzed from different finite element programs, in this case OpenSees and Diana.

Due to the expensive nature of installing sensors in real life and of damaging a bridge which is likely prohibited, the software system includes a component for simulating sensor responses from reinforced concrete bridges. In order for this simulation to be extensible and allow for further research on bridges other than bridge 705, the specification of the bridge is simply a parameter of the system.

The developed decision support system has a number of **parameters** such that users wishing to extend the software further are not limited to focus on bridge 705 or to use a specific finite element program. The specification of a bridge is a parameter of the system, as is the type and intensity of traffic on the bridge. Furthermore, as mentioned earlier, different finite element programs can be integrated with this system, which may be useful if a finite element model of a bridge for a different finite element program is already available to the user.

3.2 Existing Work

A more in-depth look at what has been done before

This section contains a review of the most relevant material studied during this thesis work. The section begins with an overview of related works followed by a more in-depth look at the most relevant material. The aim of this section is to place the thesis in context and to provide background information to the reader on employed techniques. The section concludes by relating the reviewed material back to this thesis.

3.2.1 An overview

TODO: overview of related works

3.2.2 The application of machine learning to structural health monitoring

cite:worden2006application illustrates the utility of a data-driven approach to structural health monitoring (SHM) by a number of case studies. In particular the paper focuses on pattern recognition and machine learning (ML) algorithms that are applicable to damage identification problems.

The question of *damage detection* is simply to identify if a system has departed from normal (i.e. undamaged) condition. The more sophisticated problem of *damage identification* seeks to determine a greater level of information on the damage status, even to provide a forecast of the likely outcome of a situation. The problem of detection and identification can be considered as a hierarchy of levels as described in cite:rytter1993vibrational.

- Level 1. (Detection) indication that damage might be present in the structure.
- Level 2. (Localization) information about the probable position of the damage.
- Level 3. (Assessment) an estimate of the extend of the damage.
- Level 4. (Prediction) information about the safety of the structure.

This paper argues that ML provides solutions to these problems at upto level 3, and that in general level 4 cannot be addressed by ML methods.

Applying ML for the purpose of SHM is usually only a single step in a broader framework of analysis. Figure 1 shows the waterfall model (cite:bedworth2000omnibus) which begins with sensing (when to record responses) and ends with decision making. ML methods are only step four in this model. An important part of this entire process is feature extraction, step three, which can be regarded as a process of amplification, transforming the data to keep only information that is useful for the ML analysis. Another aim of feature extraction is to reduce the dimensionality of the data, to avoid the explosive growth of the data requirements for training with the data dimensions, known as the *curse of dimensionality* TODO:REF.

An experiment was setup to identify damage on the wing of a Gnat artefact. Damage scenarios for testing were created by making a number of cuts into copies of the wing panel. Transmissibility between two points was chosen as a measurement based on success in a previous study TODO:REF, it is the ratio of the acceleration spectra between two points $A_j(\omega)/A_i(\omega)$. This

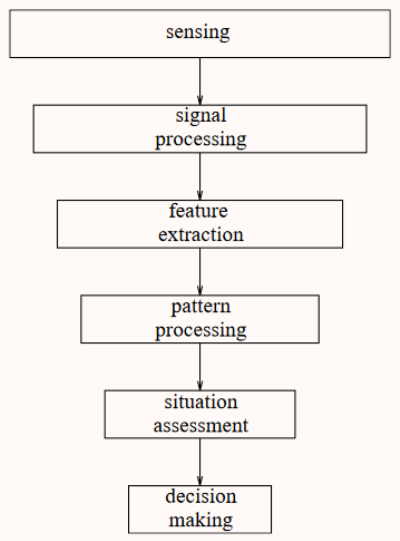

Figure 1: The *waterfall* model.

was measured for two pairs of perpendicular points on each wing; in the frequency range 1-2kHz, which was found to be sensitive to the type of damage investigated. The measurements were transformed into features for novelty detection by manual investigation of 128-average transmissibilities from the faulted and unfaulted panels, selecting for each feature a range of spectral lines as shown in TODO:FIG. 18 features were chosen.

To address the first level of Rytter's hierarchy, damage detection, an outlier analysis was applied. This outlier analysis calculates a distance measure (the squared Mahalanobis distance) for each testing observation from the training set. 4 of the 18 features could detect some of the damaged scenarios and could detect all of the unfaulted scenarios, other features produced false positives and were discarded. Two combined features managed to detect all damage types and raised no false positives.

The second level of Rytter's hierarchy is damage localization. This problem can be approached as a regression problem, however here it is based on the classification work done for damage detection where transmissibilities are used to determine damage classes for each panel. A vector of damage indices for each of the panels is given as input to a multi-layer perceptron (MLP) which is trained to select the damaged panel. The paper argues that "it may be sufficient to classify which skin panel is damged rather than give

a more precise damage location. It is likely that, by lowering expectations, a more robust damage locator will be the result". This approach has an accuracy of 86.5%, the main errors were from two pairs of adjacent panels, whose damage detectors would fire when either of the panels were removed. The approach depends on the fact that damage is local to some degree, and the damage detectors don't fire in all cases, which was true in this case.

, the assessment was based on the previous detection technique.

3.2.3 Neural Clouds for monitoring of complex systems

In one-class classification, a classifier attempts to identify objects of a single class among all objects by learning from a training set that consists only of objects of that class. One-class classifiers are useful in the domain of system condition monitoring because often only data corresponding to the normal range of operating conditions is available. Data corresponding to the class of abnormal conditions, when a failure or breakdown of a system has occurred, is often not available or is difficult or expensive to obtain.

The Neural Clouds (NC) method presented in cite:lang2008neural is a one-class classifier which provides a confidence measure of the condition of a complex system. In the NC algorithm we are dealing with measurements from a real object where each measurement is considered as a point in n-dimensional space.

First a normalization procedure is applied to the data to avoid clustering problems in the subsequent step. The data is then clustered and the centroids of the clusters extracted. The centroids are then encapsulated with "Gaussian bells", and these Gaussian bells are normalized to avoid outliers in the data.

The summation of the Gaussian bells results in a height `h` for each point `p` on the hyperplane of parameter values. The value of `h` at a point `p` can be interpreted as the probability of the parameter values at `p` falling within the normal conditions represented by the training data.

In comparison to other one-class classifiers, the NC method has an advantage in condition monitoring in that it creates this unique plateau where height can be interpreted as probability of the system condition. Figure 2 shows this plateau in comparison with other one-class classifiers, Gaussian mixture and Parzen-window.

It is important to note that when significant changes occur in the normal

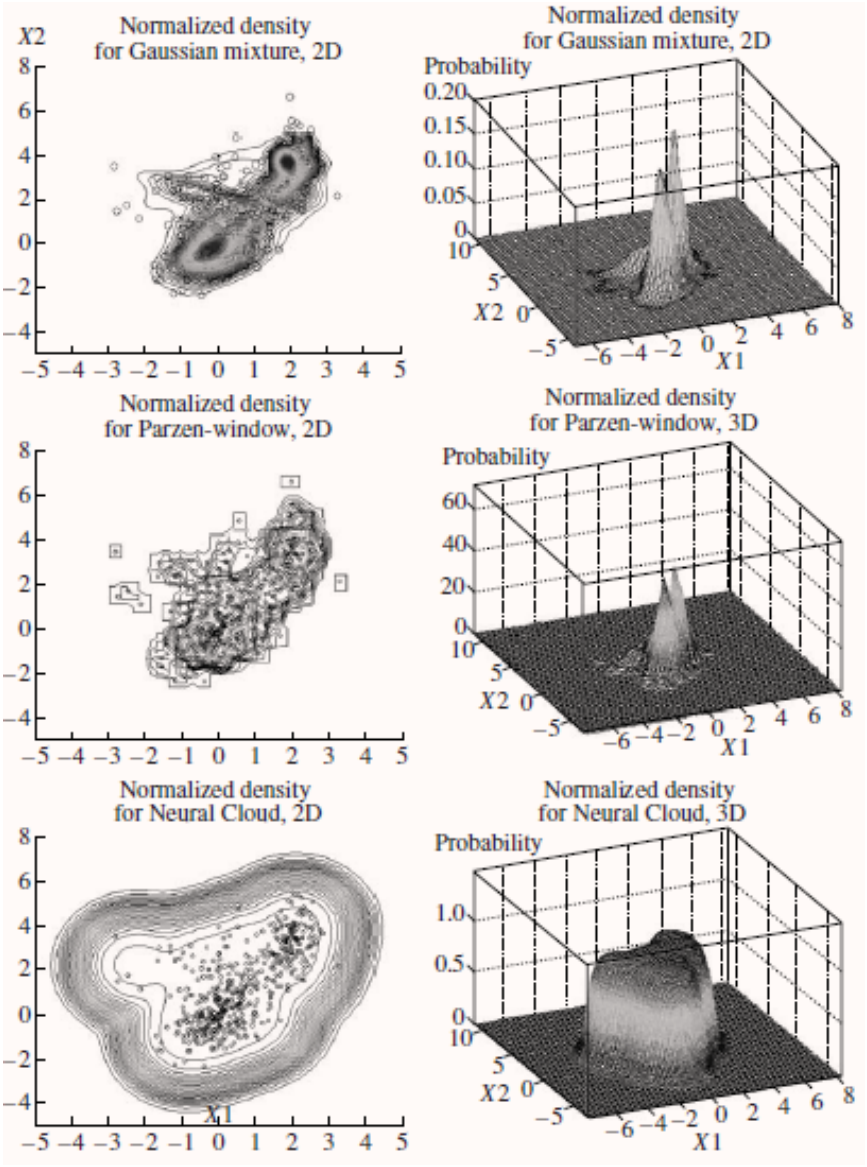

Figure 2: Comparison of Neural Clouds with other approaches, namely Gaussian mixture and Parzen-window. At the left side 2D contour line plots are pictures and at the right normalized density 3D plots.

state of the system, perhaps due to environmental changes, then the NC classifier should be retrained in order to avoid a false alarm. However, if a NC classifier is continually being retrained with real-time data then it may not detect a gradual long-term change to the system.

3.2.4 Combining data-driven methods with finite element analysis for flood early warning systems

In cite:pyayt2015combining a system for real-time levee condition monitoring is presented based on a combination of data-driven methods and finite-element analysis. Levee monitoring allows for earlier warning signals incase of levee failure, compared to the current method of visual inspection. The problem with visual inspection is that when deformations are visiable at the surface it means that levee collapse is already in progress.

Data-driven methods are model-free and include machine learning and statistical techniques, whereas finite-element analysis is a model-based method. One advantage of data-driven methods are that they do not require information about physical parameters of the monitored system. As opposed to finite-element analysis which in the case of levee condition monitoring requires parameters such as slope geometry and soil properties. The model-based methods provide more information about the monitored object, but are more expensive to evaluate and thus difficult to use for real-time condition assessment.

In this paper the data-driven and finite-element components of the system which were developed are referred to as the Artificial Intelligence (AI) and Computer Model (CM) respectively. The AI and CM can be combined in two ways. In the first case the CM is used for data generation. Data is generated by the CM corresponding to normal and abnormal conditions. The normal behaviour data is used to train the AI and both the normal and abnormal behaviour data can be used for testing the AI. In the second case shown in Figure 3 the CM is used for validation of the alarms generated by the AI. If the AI detects abnormal behaviour then the CM is run to confirm the result. If the AI was correct a warning is raised, else the new data point is used to retrain the AI.

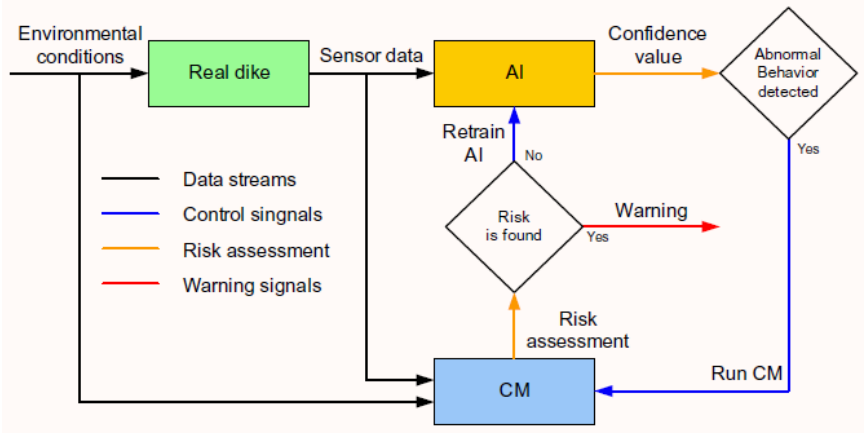

Figure 3: AI and CM...

3.2.5 Flood early warning system: design, implementation and computational modules.

In cite:krzhizhanovskaya2011flood a prototype of an flood early warning system (EWS) is presented as developed within the UrbanFlood FP7 project. This system monitors sensors installed in flood defenses, detects sensor signal abnormalities, calculates failure probability of the flood defense, and simulates failure scenarios. All of this information is made available online as part of a DSS to help the relevant figure of authority make an informed decision in case of emergency or routine assessment.

Some requirements that must be taken into account in the design of an EWS include:

- Sensor equipment design, installation and technical maintenance.
- Sensor data transmission, filtering and analysis.
- Computational models and simulation components.
- Onteractive visualization technologies.
- Remote access to the system.

Thus it is clear that the development of an EWS or DSS consists of much more than the development of the software components, but must also take into account the installation of hardware and the transmission of information between components of the system. These many interacting components are

shown in Figure 4 along with a description.

3.2.6 A clustering approach for structural health monitoring on bridges

In cite:diez2016clustering a clustering based approach is presented to group substructures or joints with similar behaviour and to detect abnormal or damaged ones. The presented approach is based on the simple idea that a sensor located at a damaged substructure or joint will record responses that are significantly different from sensors at undamaged points on the bridge.

The approach was applied to data collected from 2,400 tri-axial accelerometers installed on 800 jack arches on the Sydney Harbour Bridge. An *event* is defined as a time period in which a vehicle is driving across a joint. A pre-set threshold is set to trigger the recording of the responses by each sensor, each event is then represented by a vector of samples X.

Prior to performing any abnormality detection the data is preprocessed. First each event data is transformed into a feature $V_i = |A_i| - |A_r|$ where A_i is the instantaneous acceleration at the \$i\$th sample and A_r is the "rest vector" or average of the first 100 samples. The event data is then normalised as $X = \frac{V-\mu(V)}{\sigma(V)}$.

After normalisation of the event data, k-nearest neighbours is applied for outlier removal. One might consider that outliers are useful in the detection of abnormal conditions, since they represent abnormal responses. However if outlying data per joint are removed, then a greater level of confidence can be had when an abnormal condition is detected knowing that the result is not based on any outliers. In this outlier removal step the sum of the energy in time domain is calculated for event data as $E(X) = \sum_i |x_i|^2$. Then for every iteration of k-nearest neighbours, the k closest neighbours to the mean of the enery of the joint's signals μ_{joint} is calculated.

The event data is then transformed from the time domain into a series of frequencies using the Fast Fourier Transform (FFT), such that the original vibration data is now represented as a sequence that determines the importance of each frequency component in the signal. After this transformation a distance metric is calculated for each pair of event signals, this metric is used for k-means clustering of the data for anomaly detection. The distance metric used is the Euclidean distance: $dist(X, Y) = ||X - Y|| = \sqrt{\sum(x_i - y_i)^2}$.

Two clustering methods were applied, event-based and joint-based. In the

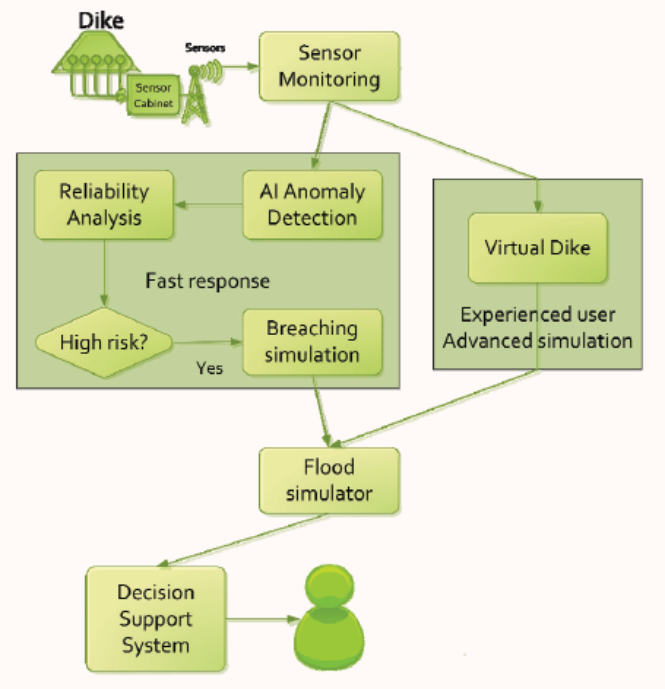

Figure 4: The *Sensor Monitoring* module receives data from the installed sensors which are then filtered by the *AI Anomaly Detector*. In case an abnormality is detected the *Reliability Analysis* calculates the probability of failure. If the failure probability is high then the *Breach Simulator* predicts the dynamics of the dike failure. A fast response is calculated beginning with the *AI Anomaly Detector* and ending with the *Breaching Simulator*. The *Virtual Dike* module is additionaly available for the purpose of simulation by expert users, but takes longer. The fast response and the response from the *Virtual Dike* module are both fed to the *Flood Simulator* which models the flooding dynamics, this information is sent to the decision support system to be made available to the decision maker.

event-based clustering experiment it was known beforehand that joint 4 was damaged. All event data was clustered using k-means clustering with $K = 2$ which resulted in a big cluster containing 23,849 events and a smaller cluster of 4662 events mostly located in joint 4. The percentage of events per joint in the big cluster are shown in Figure 5 where joint 4 is clearly an outlier.

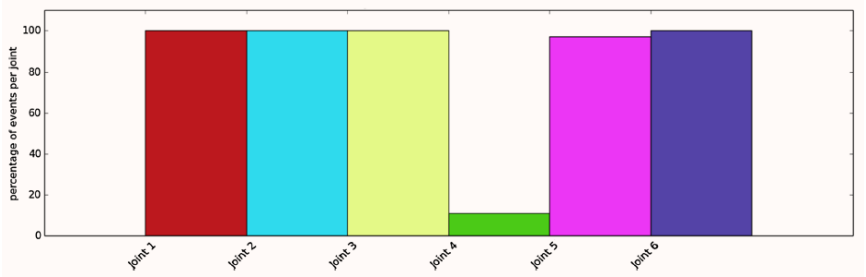

Figure 5: ...

A frequency profile of both the big and small cluster are shown in Figures 6 and 7. In case there is no knowledge of abnormal behaviour then this method can be used to separate outliers and obtain a profile of normal behaviour. In this research on SHB there was prior knowledge of a damaged joint. A frequency profile of an arbitrary joint and the damaged joint before and after repair is shown in Figure 8. The difference of the damaged profile to the other two is clear, which indicates that there is sufficient information in frequency information from accelerometers to detect abnormal joints.

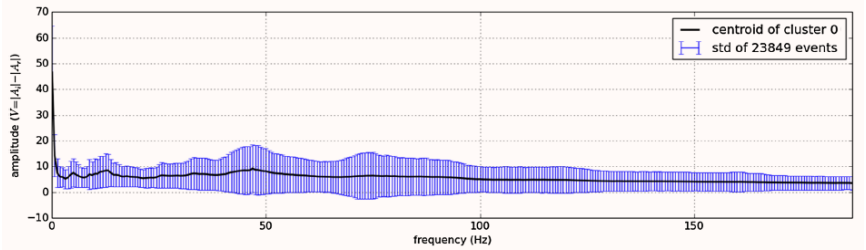

Figure 6: ...

In joint-based clustering a pairwise map of distances is calculated between each pair of joint representatives. A joint representative is calculated as the mean of the values of all event data for one joint, after the outlier removal phase. Two experiments were conducted. One experiment consisted only of 6 joints, including the damaged joint 4. The clustering method detected

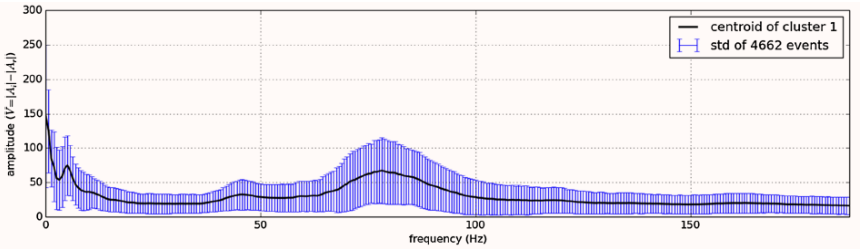

Figure 7: ...

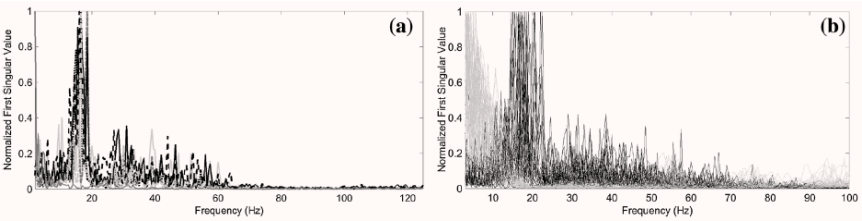

Figure 8: ...

the damaged joint as can be seen in 9. The second experiment was run on data from 71 joints. The resulting map can be seen in 10 which accurately detected the damaged joint 135. Damage was also detected in joint 131 but this result was not verified.

3.2.7 DSS

TODO: Overview of bridge DSS

3.2.8 Summary

TODO: conclude the literature review

4 Methods

4.1 Data Collection

This section describes the data collection system. This includes a brief overview, a description of the inputs and outputs of the system, the FE program used to run simulations, and a look at how the system operates, in particular focusing on the programatic interface to the system and the

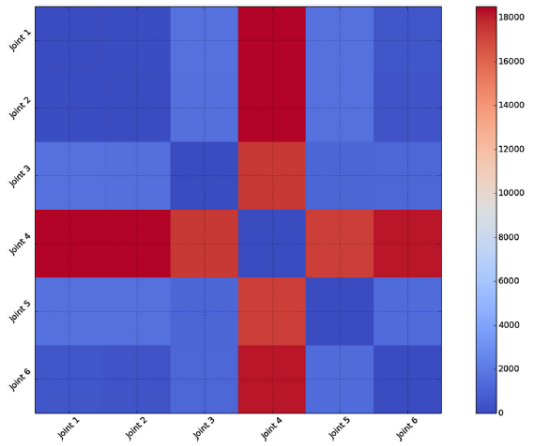

Figure 9: TODO:CAPTION

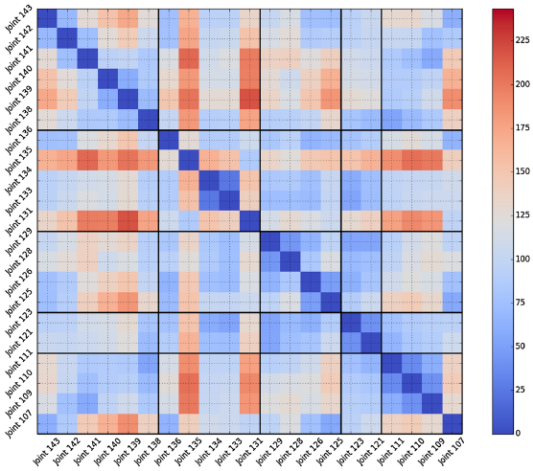

Figure 10: TODO:CAPTION

importance of the re-usability of the system.

The inputs of the system include a model of a bridge, a bridge scenario (healthy or damage state) and a traffic scenario (e.g. heavy or light vehicles). The outputs of the system are time series of responses from sensors distributed across the bridge model, these time series of responses we term *events*. Events are labelled by simulation scenario and simulation time.

For a given simulation scenario, a FEM of the bridge is built according to the given bridge scenario. A number of FE simulations are run, for each simulation a single concentrated load is placed at a single point on the bridge deck. The point is chosen such that it is on a track where a vehicle's wheels are expected to be when the vehicles are later "driven" along the bridge. Vehicles are then sampled from the given traffic scenario and driven along the bridge on a traffic lane, the vehicle's wheels moving on the previously mentioned tracks. Using the superposition property, responses collected from the simulations can be summed together, one for each vehicle's wheel, to determine a response at a point due to the vehicle being at a position on the bridge deck.

4.1.1 Bridge Modeling

A maintainable and extensible system was engineered for the purpose of data collection. At the center of this system is a programatic model of the problem domain. The model includes the types `Vehicle`, `Load`, `MovingLoad`, `{`Lane, `Fix`, `Bridge`, `Section`, `Patch`, `Layer`, `Material`, `Response` and `ResponseType`. These types are used to model traffic, bridges and sensor responses.

The data collection system is parameterized by the traffic on the bridge and a specification of a bridge itself. A bridge is modeled by length, width, `Fix`es, `Lane`s, and `Patch`es and`Layer`s that are combined to form a `Section`. The `Fix`es are used to define fixed nodes of the FEM which represent a bridge's piers. The `Lane`s define where vehicles are "driven" along in simulation. A `Section` determines the cross-sectional area of the bridge in terms of `Patch`es (rectangular patches with a number of fibers) and `Layer`s (a number of fibers along a line). Listing **??** shows the programatic specification of bridge 705. Figure 11 shows the bridge that is modeled based on the specification.

```
data Bridge = Bridge {
    length :: 102
  , width  :: 33.2
  , lanes  :: [Lane(4, 12.4), Lane(20.8, 29.2)]
```

```
, spans  :: [12.75, 15.3, 15.3, 15.3, 15.3, 15.3, 12.75]
, cross  :: Section(...)
}
```

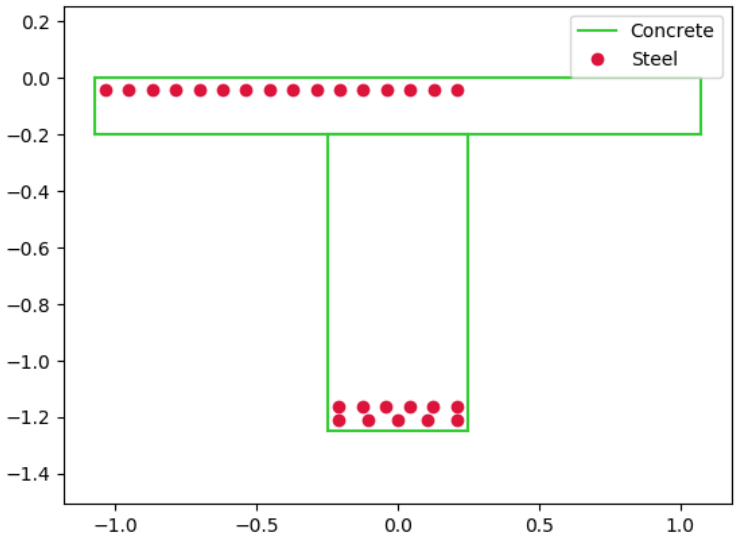

Figure 11: Cross section of bridge 705.

A vehicle is represented by a number of variables.

1. Discretization

   - Material properties may vary according to a continuous function on a real bridge while material properties in the FEM change at given discretization points.

     TODO

     itemize

4.1.2 Bridge Scenario

The goal of the damage identification model is to identify damage in a number of selected damage scenarios. Damage scenarios can be classified as short-term or long-term. Short-term events are defined as a change of the properties of structural materials

and elements, and of the behaviour of the whole structure, due to effects that occur during a very short period of time. Long-term events are time-dependent and may not only be related to external factors but also due to a change of state of materials with time. Tables 1 and 2 cite:sousa2019tool outline some of the predominant types of damage due to short-term and long-term events respectively.

TODO: Use table.el to fix tables

Table 1: Types of damage due to short-term events.

Event	Examples/Consequences
Collision	Impact by overweight vehicle or boat in the river
Blast	Impact by vehicle followed by explosion
Fire	Impact by vehicle followed by explosion and fire
Prestress loss	Sudden failure of a prestress tendon
Abnormal loading conditions	Loading concentration and/or overloading in a specific site along the
Excessive vibration	Earthquake
Impact	Impact pressure by water and debris during floods

Table 2: Types of damage due to long-term events.

Event	Examples/Consequences	Critical comp
Corrosion	Degradation of the bearings	Deck
	Loss of cross-section area in the prestressing tendons	Deck
Time-dependent properties of the structural materials	Excessive creep & shrinkage deformations	Deck
	Concrete deterioration	All
Low stress - high frequency fatigue	High frequency and magnitude of traffic loads	Deck
High stress - low frequency fatigue	Temperature induced cyclic loading	Abutment
Environmental effects	Freezing water leading to concrete expansion	All
Water infiltration/Leaking	Deterioration of the expansion joints; concrete degradation in the zone of the tendon anchorages	Deck
Pier settlement	Change in the soil properties	Deck

Of the damage scenarios listed in Tables 1 and 2, four scenarios are selected for identification by the DIM in addition to one unlisted damage scenario. These scenarios are chosen due to the practicality of simulating them in a FEM of bridge 705.

Pier settlement can be simulated by displacing a pier by a fixed amount, this is achieved in practice by applying an increasing vertical force known as a *displacement load* to the deck until the desired displacement is achieved.

Abnormal loading conditions can be simulated relatively easily by applying the heavy loads in the FE simulation. Care must be taken regarding the axle configuration because extreme heavy loads typically have a different axle configuration than less heavy vehicles.

Cracked concrete can be simulated by reducing the value of Young's modulus for the cracked concrete section. In practice, Young's modulus is often reduced to $\frac{1}{3}$ of its original value (cite:li2010predicting).

Corrosion of the reinforment bars can be simulated by increasing the size of the reinforcement bars TODO:WHY. Finally, a damage scenario is considered where it is not the bridge that is damaged but rather a sensor is malfunctioning.

A *malfunctioning sensor* can be simulated by adding a significant amount of noise to the simulated sensor responses or adding a constant offset to the responses TODO:LITERATURE. From discussions with Sousa TODO:REF, detecting malfunctioning sensors is useful to accomplish.

4.1.3 Traffic Scenario

To train a classifier to distinguish between normal and abnormal traffic conditions it is necessary to define normal traffic conditions.

4.1.4 FE Program

Two FE programs are used for the collection of sensor responses, OpenSees (cite:mazzoni2006opensees) and DIANA (cite:diana2019diana). OpenSees is used because it is open source software, such that anyone can download and use the software without a licence. On the other hand is proprietary software, if you want to do research with Diana a licence must be purchased. The reason Diana is supported is because a verified 3D FEM of bridge 705 is available for Diana. In this thesis the Diana FEM is used in limited ca-

pacity for the verification of results obtained via OpenSees. The focus is instead on OpenSees because it is software that anyone with a laptop can use for free to extend this research. In addition it is useful to have two FE programs available, one (OpenSees) can be used to run less accurate but faster 2D FE simulations, allowing for a more rapid research cycle. The results can then be compared and verified against results from more accurate but also more computationally expensive 3D FE simulations (Diana). It is noted that the 2D model will ignore some aspects in the transverse direction of the bridge deck. For example the 3D model of bridge 705 has two lanes, but the 2D model ignores the concept of lanes entirely.

OpenSees stands for the *Open Sysem for Earthquake Engineering Simulation*, it is "an open source software framework for creating applications for the nonlinear analysis of structural and soil systems using either a standard FEM or an FE reliability analysis. It is object-oriented by design andin addition to achieving computationally efficiencyits designed to be flexible, extensible, and portable" cite:mckenna2011opensees.

DIANA (**DI**splacement **ANA**lyzer) is developed by DIANA FEA BV which is a spin-off company from the Computational Mechanics department of TNO Building and Conctruction Research Institute in Delft, The Netherlands. DIANA is a FE software package that is dedicated to problems in civil engineering, inlcuding structural, geotechnical, tunnelling, earthquake and oil & gas.

TODO: Image of the 705 Diana model.

4.1.5 System Details

Talk about:

- 2D and 3D models
- influence lines
- avoiding unecessary simulations

A FEM of a bridge is a discretized representation of reality. The accuracy of the model depends on density of the mesh. A mesh

that consists of smaller and thus more shell elements will provide a more accurate response.

TODO

System Interface

4.1.6 Collected Data

4.1.7 Model Assumptions

- The movement of traffic on the bridge is All vehicles drive at a constant speed.
- All vehicles drive at the same speed.
- All vehicles drive along the center of a lane.
- All vehicles have the same axle-width.
- The behaviour of a bridge captured in FE simulation is sufficiently close to the real behaviour of a real bridge that the analysis techniques explored on the simulated data can also work on real data.

  This assumption is verified by (A) applying the analysis techniques explored on real data in addition to the simulated data and (B) verifying the collected responses against sensor measurements collected in real life.

  Note that the accuracy of the responses depends on the discretization density of the FEM. This is a trade-off of time versus accuracy which can be chosen by the user. Discretization of the FEM is covered in Section 1. The accuracy of the FEM is shown to converge for bridge 705 in TODO
- The simulated noise that is applied to responses from FE simulation is sufficiently close to noise from sensors in real life that the analysis techniques explored on the simulated data can also work on real data.

  This assumption is verified by (A) applying the analysis techniques explored on real data in addition to the simulated data and (B) verifying the simulated noise against noise from measurements collected in real life as shown in .

4.2 Anomally Detection

In this section the process of building the damage identification model is described. First there is an introduction to the damage scenarios that it is desirable for the model to identify, followed by a description of the setup for testing iterations of the model. After this an analysis is presented of the sensor responses with respect to the useful information in different sensor types for each damage scenario. Finally the damage identification model that is built is discussed.

4.2.1 Feature extraction

4.2.2 Test setup

4.2.3 Data analysis

4.2.4 Damage identification model

4.3 Decision Support System

4.3.1 Sensor placement

4.3.2 Cost-benefit analysis

4.3.3 Uncertainty

4.3.4 Generalizability

5 Results

6 Bibliography