A Correlation Network Model for Structural Health Monitoring and Analyzing Safety Issues in Civil Infrastructures

Alexander Fuchsberger
University of Nebraska, Omaha
afuchsberger@unomaha.edu

Hesham Ali
University of Nebraska, Omaha
hali@unomaha.edu

Abstract

Structural Health monitoring (SHM) is essential to analyze safety issues in civil infrastructures and bridges. With the recent advancements in sensor technology, SHM is moving from the occasional or periodic maintenance checks to continuous monitoring. While each technique, whether it is utilizing assessment or sensors, has their advantages and disadvantages, we propose a method to predict infrastructure health based on representing data streams from multiple sources into a graph model that is more scalable, flexible and efficient than relational or unstructured databases. The proposed approach is centered on the idea of intelligently determining similarities among various structures based on population analysis that can then be visualized and carefully studied. If some "unhealthy" structures are identified through assessments or sensor readings, the model is capable of finding additional structures with similar parameters that need to be carefully inspected. This can save time, cost and effort in inspection cycles, provide increased readiness, help to prioritize inspections, and in general lead to safer, more reliable infrastructures.

1. Introduction

Structural Health Monitoring is the process of "determining and tracking structural integrity and assessing the nature of damage in a structure" [1]. After fifteen years of signal processing, new sensor technologies and control theory, damage detection and management is still ineffective. The Federal Highway Administration (FHA) inspects all national bridges with a span length of over 25 feet every two years, regardless of their status or urgency. The inspections are usually limited to visual inspections and tap tests. Tap tests are used to find voids or debonding in concrete structures through acoustic signals [1]. The outcome of the FHA bridge inspections is a database with safety and reliability evaluations of more than 600,000 highway bridges. The problem is that these ratings are entered manually and often based on best guesses. While visually detectable damage like cracks can be evaluated fairly easily, other types of damage, like shifts or changes in structural mode shapes are harder to detect. Additionally, environmental factors like temperature, wind, and traffic load can have significant influence over sensor readings and leave the measurements unreliable and situational.

Regular, scheduled inspections of civil infrastructures proved to be inefficient and costly, and the possibility remains that some bridges needing engineering renewal or replacement are not identified in time [1]. Additionally, the detection of a condition decrease is not affecting the future level of inspection detail and frequency. In this research, we propose a radically new concept to predict the safety and reliability of civil infrastructures in large scale with the help of graph algorithms.

Manually assessing damage on civil structures has been the original concept but due limitations in technology and data analytics it is still widely used, despite being labor intense. For twenty years, civil infrastructures are increasingly monitored autonomously through sensor technology now. This marks the next level of structural health monitoring. In this research, we compare the effectiveness, scalability, robustness and accuracy of these sensor-based methods. The highest level of structural health monitoring is through prediction. This area is least explored by scholars and practitioners and in this research we assemble current techniques for damage prediction and control and suggest a method for identifying underperforming structures based on prediction and graph theory. Recent improvements in Big Data management, computing power, algorithm efficiency and visualization tools enable a radically new way of assessing damage in civil structures without the costly need to send humans or place sensors.

Conclusively this paper addresses the following research questions:

- How can a graph-based prediction model identify 'unhealthy' infrastructures based on similarity to other problematic infrastructures?
- How does a new data representation reduce the volume of data and processing time?

URI: http://hdl.handle.net/10125/41271
ISBN: 978-0-9981331-0-2

In section 2, this research surveys current structural health monitoring techniques through assessment, sensors or prediction. Further structural damage types, measurement accuracy and limitations of these methods are described. In section 3, we link structural health monitoring to big data and graph theory, and explore methods from data science to manage similar challenges. We then propose a method for identifying potentially unreliable or insecure structures based on graph algorithms, which are suitable for large-scale and dynamic analysis (section 4). We proceed with a discussion on the applicability and limitations and future direction and a final conclusion (section 5).

2. Structural Damage and Health Monitoring

The need for advanced structural health monitoring and damage detection tools has been laid out by multiple scholars and practitioners [1]–[3]. Structural Health monitoring should ideally detect damage as it appears and evaluate location within the structure and severity. However most current damage detection methods can only determine whether damage is present in the entire structure at a specific point in time [1] or not. These methods are considered "global health monitoring" methods and are often sufficient to determine further action (examination, scuttle, reparation, replacement).

Local health monitoring methods, such as acoustic waves, X-rays or radar are much more likely to locate, quantify and determine the severity of damage but are currently not a realistic goal because they consume too much time and effort [1].

A third category of health monitoring tries to predict damage based on current and historical data and analytical methods. Chang et al. reviewed nondestructive and destructive damage detection methods in 2003. Since then much progress has been accomplished in sensor technology and data processing, enabling enhanced and improved structure health monitoring.

2.1. Health Monitoring through Assessment

Global health monitoring techniques are primarily applied to find shifts in resonant frequencies or changes in structural mode shapes [1], [4]. In concrete structures most of the stiffness is contributed by the concrete, which makes the effect of deterioration of the reinforcing steel hard to measure [5]. Steel bridges often do not reveal much damage to the point where the damage increases radically. Damage through corrosion, connection problems, material degradation or other means remains usually invisible with the exception of major cracks [1].

Mode shapes represent stress and vibrations of a structure when exposed to natural frequencies. During dynamic loading (earthquakes, vehicle movements, wind) structures resonate and with mode shape measuring techniques it is possible to identify statically weak elements of e.g. a bridge and its overall resistance to vibrations. Measuring mode shapes is difficult and research has shown that mode shapes are not much affected by local structural damage [6]. The matrix update method provides a mean to evaluate stiffness, mass and damping matrices of the structure through optimization techniques [1], [5], [9]. Damage detection methods are vulnerable to environmental effects and tend to perform more precise and reliable when the damage is severe and matches the underlying constraints [1]. There are methods to battle environmental noises through e.g. baseline signals reduction [11], wavelets [12] or Hilbert-Huang Transformation [13], [14]. Actuators and sensors can be used to circumvent the problem of noise [1]. Wu et al. introduced the concept of image processing and pattern recognition to detect surface cracks [15]. Grayscale images of the structure surface are filtered by the average gray level. This reveals parts of the image, which contrast highly from the average level. The shape and size of these forms allows for conclusions on the shapes and sizes of cracks. Structural damage changes the flexibility of the structure. This can be detected by the Damage Location Vector method [16].

Another class of damage assessment methods are those that use acoustic signals to determine inconsistencies within the structure. Examples are acoustic emissions, ultrasonic measurement, impact-echo and tap tests [1]. They are fairly robust to environmental effects and easy to deploy, however suffer from labor intense resource requirements [1].

X-rays and Gamma rays are also used to visualize the interior of civil structures. The setup is often very difficult to deploy because of the size of structures and the inaccessibility of sender and receiver positions. Some damage detection methods, especially the ones identifying stiffness or flexibility, compare measured data against prediction models or original specifications. This is a challenge in civil infrastructures because they are often not built with the accuracy as other fields (e.g. automotive industry). Reasons are on-site construction constrains and change orders and a concrete mixture is always unique [1]. This often leaves model-based detection methods restrained to assumptions and best guesses.

2.2. Health Monitoring through Sensors

When talking about SHM systems, permanently installed sensor systems are implied in the structural health monitoring process. Such systems serve a variety of tasks beside damage detection [17]: they have to provide real-time information for safety assessment immediately after disasters or information that may help to improve design specifications for future structures or provide data for scientific research. They further provide information to plan and prioritize structural inspection, rehabilitation, maintenance and repair [17]. The variety for available sensor technologies has exploded in recent years, while constantly getting more affordable and effectively. This enables a variety of applications like accelerometers, nuclear magnetic resonance capsules for chloride ion detection (deterioration indicator) and shearography for recognition of displacements [1]. Further sensors allow capturing 3D positions of objects and infrared thermography can be used to detect debonding [1].

A sensor usually targets a specific type of measure, for example, crack detection, cable breakage, steel reinforcement corrosion or debonding. Although individual sensors are limited and only provide a small fraction of the information needed to assess the health and reliability of an entire structure, they can be linked through wireless sensor networks (WSN), to provide powerful monitoring. In most cases, sensors are installed during construction and have an expected lifetime of many years, sometimes decades. This affects the construction of the bridge itself, when e.g. cables have to be placed and integrated in the structure. WSNs help to reduce planning and installation time drastically, especially when multiple sensors have to be placed. Some bridges carry hundreds of sensors.

While a higher amount of sensors would provide more detailed information on the structure, in practice, wide-scale distribution of SHM systems is usually limited by the cost of data acquisition systems and accessibility of such systems [17].

A recent development in SHM is that of “smart” sensors. On the sixth Australian Small Bridges Conference in Sydney 2014, it was concluded that the goal of smart infrastructures, where bridges generate real-time information that could be directly transformed into action is not yet reached [2]. Sensors of the future would need to asses more complex measures like the location extend and rate of corrosion on reinforcing bars within the concrete as well as cables, concrete strength measures and yield stress detection [2]. Further monitoring systems would have to shift from single point measures to reliable fatigue monitoring covering large areas [2]. This would help to reduce the immense data volume and variety generated by sensor networks and shift some of the computing power into an earlier stage and less complex stage of processing. Reducing the amount of data through smart sensors is a research goal also addressed at the “Bridge(ing) Data Workshop” in Omaha, 2015 [18].

Mobile sensing is another class of sensing methods and such sensors do not require static placement. Pictures for image processing can be taken through cameras placed on a moving vehicle. Even radar technology allows obtaining 3-dimensional pictures of the steel reinforcement within civil structures at traveling speeds [1].

2.3. Health Monitoring through Prediction

Structural health monitoring should reach beyond damage detection and aim to predict damage to prevent disasters before they happen. Predictive analytics tools can be used to better assess the safety levels of structures utilizing both historic and real-time data. Predicting, when structures will become suspect for maintenance based on integrating all available data is of particular interest in the industry. Having reliable and automated advanced analytics tools has the potential to save millions of dollars and potentially human lives.

First attempts to predict damage featured a statistical pattern-recognition approach using Bayes theorem by comparing probabilities of certain damage events [1], [20], [21]. The durability of a civil infrastructure is mainly dominated by fatigue behavior of the critical elements in the structure. In 2001, Li *et.al.* introduced a system to predict the service life of bridge deck sections through a permanent structural health monitoring system [22]. This system combined multiple sensor readings in order to evaluate fatigue damage.

Proven prediction models from other disciplines can be used to address this challenge. In this age, the constraints do not lie within data collection, but the processing and evaluation part. Collecting extensive amounts of data from a variety of sources like sensors, scans or web services is no longer a technological issue. Extracting useful information out of a continuous, massive stream of data is the challenge of our time. In our endeavor of finding a universal solution for civil infrastructure health monitoring, we inevitably stumble over the term Big Data. In our context, we refer to the definition of Madden: “Data that is too big, too fast or too hard for existing tools to process.” [23] The goal of big data analysis is to “turn data into meaningful knowledge

and support effective decision making and optimization" [24]. Chen et al. are giving a broad overview of the impact of Big Data on applications, analytics, technologies and emerging research [25].

We found that all model-based methods for assessing damage in structures are looking for anomalies, or outliers that indicate measurements differing from an optimal value or range. In computer science, anomaly detection is a branch of datamining concerned with discovering rare occurrences in datasets [26]. In infrastructure health monitoring, signs of damage do not necessarily have to be rare but at least the difference between a healthy and a damaged element must be quantifiable. As described earlier, a powerful machinery for effectively creating knowledge on the status of the infrastructure does have to combine the output of multiple data sources. Readings from one sensor may be insufficient to detect a change in the status of an element because of noise in the readings or failure of the sensor. Multiple sensors showing weak but consistent results may drastically amplify the reliability of readings. In such interconnected sensor networks, graph networks are a powerful alternative to manage this environment, which we will address in the next section.

Fan & Bifet address the current and future stage of Big Data Mining [27]. The bottleneck for efficient analysis are usually CPU power, memory capacity and the cluelessness to specify what exactly we are looking for. A good strategy is to prepare the data in a format through preprocessing that takes away load later. Padhy *et.al.* specify clustering as an outcome of a Big Data Mining method [28]. They found that clustering algorithm have the property of discovering patterns and rules, however the range of data mining covers many more tasks. Among them and sometimes parallel are exploratory data analysis, descriptive or predictive modelling or retrieval by content.

In the context of civil infrastructures, the ideal goal is to identify bridges based on their health and reliability, whether they have been inspected recently and have SHM systems deployed or not. Clustering is a technique that can identify patterns in data we already have and use it, to predict behavior on structures with limited information. There are five primary methods to obtain clusters or prediction data: Artificial neural networks, decision trees, genetic algorithms, the nearest neighbor method or rule induction. Artificial neural networks learn from predefined models and training sets and resemble biological neural networks in the structure. Decision trees are formed by generating rules for classification of datasets. Genetic algorithms are optimization techniques simulating natural evolution. The nearest neighbor method is a technique that allows classification of each record based on a combination of records most similar to it in a historical dataset. Rule induction extracts useful knowledge based on statistical significance.

While data collection does not need significant computing power, the issue becomes highly relevant in data processing, especially for enormous structural health monitoring datasets like that of the National Bridges Inventory. When implementing SHM systems this has to be kept in mind by optimizing the data load already at the site of the structure.

Once sensor and assessment data is flowing, monitoring has to manage high loads of processing operations that increase exponentially with the number of data entities (sensors and other sources). A common approach nowadays is to split the load between multiple processing units (CPUs). This process is called grid or parallel computing. Kečo & Subasi present an implementation of genetic algorithms using parallel computing and map/reduce/Hadoop as a programming paradigm. Genetic algorithms are heuristics mimicking natural evolution. To deal with the high processing power requirements there are two alternatives how parallel genetic algorithms can be implemented [29]:

1. Cluster nodes operate on same population
2. Each node in cluster has its own population

3. Health Monitoring through Graph Algorithms

To handle complex problems, we need to find a way to represent structural health monitoring data in a way we can understand it and a computer can process it. Many problems can be solved in converting the data into a graph notation since the science of graph theory offers a variety of solutions to common problems. Abstraction happens when real world problems are transformed into graphs and solution for the graphs imply solutions back to the real world.

Such graphs, represented by vertices (nodes) and edges (connections) present an opportunity to visualize clusters of data entities with similar characteristics and determine the relatedness between nodes. They can also be used to extract certain parameters and characteristics in order to find solutions for a greater set of problems. Graphs however suffer from the same constraints as database management systems. For most problems, the complexity increases exponentially with the number of data entities, especially when it comes to clustering. In many cases, a Heuristic (non-optimal

solution) is the best one can get. Graphs are still an excellent method to simplify large-scale problems to a level they can be methodically solved or understood. In this paper we also present a way to reduce complexity before the graph is processed which can significantly increase performance.

Simple vertices and edges sets may be sufficient to represent a graph but sometimes another dimension is required. Attributes are additional information attached to vertices and/or edges. When creating the graph structure, it is possible to avoid such additional attributes by integrating them into the decision whether to establish connections or not. This increases the preprocessing time to generate clusters but decreases the time to do the actual clustering. It is also possible to leave the attributes unchanged and therefore preprocessing time minimal; this will however result in increased clustering effort.

Graph networks can overcome some of the Big Data challenges mentioned before and are an excellent tool for anomaly detection. A variety of efficient algorithms is available to traverse, analyze and modify graphs, which makes it often an attractive alternative to relational database representations.

As of recent, researchers have increased efforts to use graph-based approaches for anomaly detection because they can handle the inter-dependent nature of data and provide a robust representation and an arsenal of efficient algorithms solving a variety of problems [26]. Akoglu et al. assessed in detail the effectiveness, scalability, generality, and robustness of current graph methods for anomaly detection and concluded that while static and/or plain graphs have been researched excessively, many open challenges for anomaly detection are waiting to be addressed in dynamic, attributed graphs [26].

One way to identify unrevealed anomalies in graph structures is through interactive graph querying [26]. The idea is to take a set of known anomalies and then compare their characteristics (attributes). Nodes with very similar attributes are connected closely in the graph to the anomaly nodes. This is a good indicator that something similar is going wrong in these nodes. Figure 1 provides a visual representation of the concept:

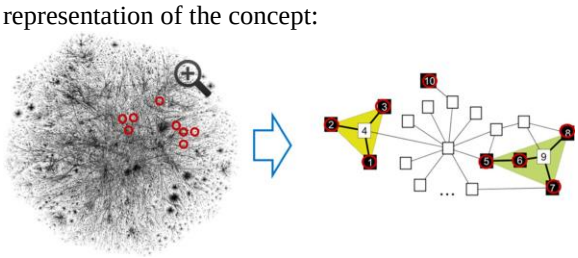

Fig. 1: Interactive Graph Querying [26] – given a set of detected anomalies, similar nodes can be identified through the connectedness to anomaly nodes in the graph.

When we apply this concept to civil infrastructures like bridges, we can assess and predict the health and reliability of bridges without inspecting them in the first place. By using the inspection data from similar bridges, we can identify characteristics and patterns for condition changes. Even relatively inflexible data like that of the National Bridges Inventory can reveal that the deck condition of bridges in warmer regions or with heavy load deteriorates faster, or that windy regions lead to intensified mode shifts or frequencies.

The difficulty in graph approaches is not to process or render the graph, but to define similarity between objects so that an edge can be created. Like in statistical regression, too many variables leave a function to determine similarity useless and imprecise. A workaround often used is to look at one attribute at a time – this eliminates the difficulty of defining similarity by introducing a simple threshold value. The resulting plain graph can easily be checked for anomalies, but does not consider anomalies, whose condition is defined by multiple attributes.

In our approach, we take interactive graph querying as a basis and describe a method for determining similarity based on multiple attributes. The result is a graph structure representing civil infrastructures as nodes and similarities as edges. With graph tools like Gephi or Cytoscape these graphs can be clustered, visualized, and used to identify “unhealthy” structures.

The next step is to decide whether to use existing clustering tools or write an own solution for that purpose. Some problems can be too specific or complex for the efficient generic algorithms provided by these tools. Graph tools offer the option to integrate procedures written in Languages like R or Python. In our test, the clustering algorithms “Fruchtermann Reingold” and “ForceAtlas2” in the Gephi Suite were sufficient to visualize a usable structure of healthy and unhealthy bridges but the algorithm to create the edges had to be shaped by ourselves. Yang & Kim proposed a prediction model for Big Data Analysis based on hybrid FCM clustering [32]. This method provides the advantage of automatic classification of the data without preprocessing. While FCM can classify properly it is unable to make precise predictions on numerical values. Supervised Learning has a high accuracy but several problems such as high requirements on the input data and difficult adoption if the data changes in structure. The hybrid FCM model combines the advantages of both models [32].

3.1. Increasing Efficiency

Efficiency can be gained by transforming the initial NBI data structure into a normalized database that can then be used by our algorithm to create the graph structure. Eventually the time it takes to do data mining and graph operations for the entire set of bridges is exponentially higher compared to the number of data entities (e.g. bridges). By reducing the size through efficient data structures and preprocessing, we can reduce the time for analysis exponentially because these analyses are usually polynomial. The effects are shown in Table 1:

TABLE I
INCREASING ALGORITHM EFFICIENCY THROUGH PREPROCESSING

N=100	Running Time [n]		In % of original time	
Linear	$O(n^2)$	$O(N^3)$	$O(n^2)$	$O(n^3)$
100	10,000	1,000,000	100	100
80	6,400	512,000	64	51
60	3,600	216,000	36	22
46	2,116	97,336	21	10
40	1,600	64,000	16	6
20	400	8,000	4	1
10,6	**112**	**1,191**	**1.12**	**0,12**

Example: If an algorithm takes 100 seconds for a linear process, it will take 10,000 seconds for an $O(n^2)$ process and 1 million seconds for an $O(n^3)$ process. A reduction of 20% of the nodes through preprocessing resolves already in a processing time of 64% of the original $O(n^2)$. For higher exponents the effect is even higher: 51% for $O(n^3)$.

The following example shows the process and effects of effective preprocessing on the data of the National Bridges Inventory. The original NBI data on approximately 600,000 bridges in the USA from 1995 to 2015 exceeds 6 GB. The first step in reducing the size is by simply transferring the raw data into a more efficient data structure.

The NBI data is provided in text-files where every line presents a bridge and a string at a specific position, with a specific length represents the value of an inspection parameter. Since the files are ASCI-encoded, every character requires a space of 8 bit. A parameter containing numbers between 0 and 255 would therefore require 24 bits. In a database, we can modify the parameter type to e.g. Integer and the size shrinks to 8 bit without losing any information. We can exclude qualitative or non-metric data entirely since they are of no use in our analysis. These steps reduce the size of the NBI database to 23% of the original size.

During the initial analysis, we found that much of this data is redundant. E.g. in every annual report, parameters like the year of construction or the location were included. By separating such data from the variable data, we can keep full integrity but further reduce the total data size to around 20% of the original. Around 20% of the parameters are static and therefore identical in every single report, so we can reduce the total dataset by another 15%.

The survey revealed another window of opportunity to reduce the amount of data. We found that the annual report actually summarizes data on 4 different inspections - the normal inspection and situational, special inspections: fracture critical details, underwater and other special inspection. All these inspections were assigned with a date and a frequency so we can tell exactly when the value of the parameters was taken. Most inspections are scheduled in intervals of two years, so we assume that most of the data actually is represented in year n and again in year n+1. By carefully comparing the data from a report and the next / previous report (containing the same information), we estimate to reduce the total size of the dataset by further 40%. We are currently at 10.6% of the original file size without losing any information. This results in a processing time reduction to only 1.12% of the original processing time for algorithms with a complexity of $O(n^2)$. For higher complexity algorithms, like $O(n^3)$, the effect is even greater (0.119% of the original processing time).

We conclude that the more complex the analysis the more benefit comes from early size reductions.

3.2. Clustering and Visualization

The idea of a predictive structural health management system is to analyze historical and contemporary data of known structures in order to predict the safety and reliability of all structures in a given population. Sophisticated prediction models use e.g. clustering to find similarities among data entities. A bridge belonging to a set of bridges with similar parameters will likely develop similarly over time. This argument is stronger, the more similar the bridges are, or, the better the clustering algorithm to define these similarities.

Since building a strong algorithm for clustering takes time and a lot of optimization effort, we initially used already existing clustering methods like Cytoscape's "K-Means" or "Fuzzy C" Clustering Heuristics in our first experiments. The created clusters were afterwards visualized.

Primarily the intension for this first analysis was not to find strongly similar bridges but to prove that certain parameters are suitable for determining an impact on the bridge health.

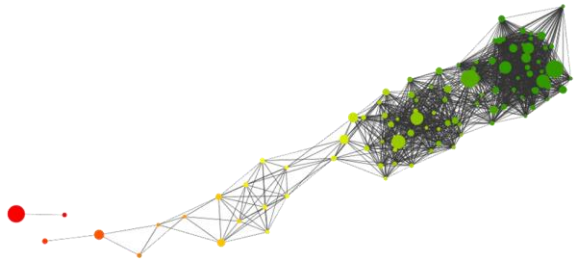

Fig. 2: Clustering of 100 random Nebraska bridges based on deck condition and date of last maintenance. We do see that that the deck condition worsens with the time passed since the last service. The outliers on the left are bridges that have been updated lately and still have a bad sufficiency rating.

We started by clustering bridges based on characteristics we already assumed a similarity relationship. For example, one underlying assumption was that the age of a bridge / last service on bridge (whatever is greater) or the deck condition affects the overall sufficiency rating. Figure 3 shows the visualization of 100 randomly selected bridges in Nebraska which were clustered by the deck condition, bridge age and sufficiency rating. To get additional information out we colored the nodes by the bridge's sufficiency rating and increased the size of the nodes for bridges with higher daily traffic. Since we included the output (sufficiency rating) also as an input (clustering parameter) we obviously have a rainbow effect on the node coloring. This graph can be used to find insufficient bridges, especially important bridges with a lot of traffic.

The next step was adding another parameter into the equation. The idea was to improve the clustering result. This would result in denser, more separated clusters. We found the inventory rating a fairly good parameter for improving clustering (Figure 4):

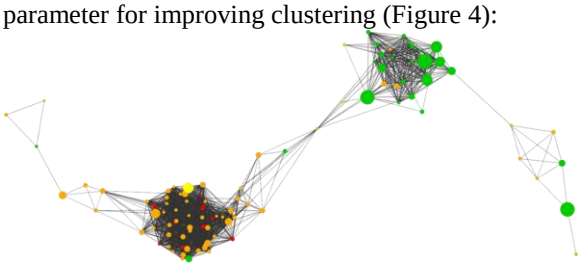

Fig. 3: Clustering Parameters of the same bridges from Figure 2 with added Inventory Rating results in improved clusters that provide clear information, which bridges should be focused on by maintenance.

The inventory rating describes the relationship of the total mass of vehicles crossing the bridge compared to the maximum safely capacity of the structure. This second graph had clearly two distinctive clusters, in which bridges with less traffic (smaller nodes) were more affected by the new conditions and therefore had lower average sufficiency ratings.

This first analysis revealed a number of problems: First, the time of the algorithm to compute the clusters increases exponentially; this limited visualization to a few hundred bridges. Second, we do not have enough information on some parameters and how they relate to each. Random selections of parameters showed that most of the 114 NBI parameters were not suitable for clustering. The result for selecting such parameters ends in a visualization of one big cluster. Third, the generated graphs are static and therefore only consider a prediction for a specific point in time.

A SHM prediction model ideally does not rely on predetermined assumptions of the relationships between parameters. The idea is to aim for a model that can find similarities without having pre-assumptions on the characteristics of such parameters because the goal is to generate new knowledge, not confirm already present knowledge.

All available clustering heuristics, no matter how sophisticated, require either one or a set of input variables for clustering. Since it is difficult to determine which ones are appropriate, all of them have to be considered as clustering parameters initially. None of the 20 Cytoscape clustering algorithms was able to create a result showing something different from a single large cluster. This is probably due to the high amount of parameters, rendering a similarity function useless.

The parameters should be weighted based on their total contribution to defining structures as similar. Ideally, the weighting could be also conditional, thus the value of certain parameters would define the choice between different weighting functions. Unfortunately, there is no known technique that can achieve that level of detail and accuracy yet. In the next section, we suggest a method to weight clustering parameters with the help of a genetic algorithm. Eventually an appropriate function to define similarity can be determined based on weighting parameters for clustering without the need to understanding their meaning in the first place.

4. Method – Dynamic Graph Clustering

The idea is to define a model that can be used to analyze any type of data for finding similarities among entities (data objects in the entire set). This is useful for prediction or pattern recognition and eventually as DSS (decision support systems). The model has support the following conditions:

1. scalable (expandable) and flexible
2. dynamic data from real-time sources
3. automatized (no specific skillset needed)

The general way to perform the analysis is that each entity is compared with each other for similarities. Several conditions evaluate the similarity or "closeness" in such a pair. A large set of variables related to each entity can be taken in consideration to evaluate the closeness. The algorithm finally calculates a value for each pair between 0 and 1. This value is created by transforming all variables into a numeric format and adding a constant for multiplication for each parameter. The final similarity value for pair (a,b) is therefore:

$$sim_{a,b} = \sum_{1...n}^{i} c_i * \frac{v_a}{v_b}$$

c_i is a constant factor from 0 to 1 which is used to manipulate the impact of a variable on the overall similarity. For attributes where clear clusters can be identified, the goal is to maximize their factors so that the parameter contributes maximally to the similarity function. For attributes with normal distributions or unclear clusters, factors should be minimized. Figure 4 shows two graphs A (left) and B (right). A is clearly more clustered than Graph B. With the help of graph algorithms, we can check the distance between nodes (similarity). If there are many short distances and optionally long distances this is an indicator for clear clusters (c_i). If the distances are mostly the same similarity is not clear for the given attribute and the factor c_i should be set low.

One limitation of that method is that more complex distributions cannot be simply represented in a single factor. It might occur that a part of the graph is in a clear cluster, while the remaining graph is normally distributed. This happens when for example the deck condition of a bridge is generally not representative in determining bridges similar unless the condition is severely bad.

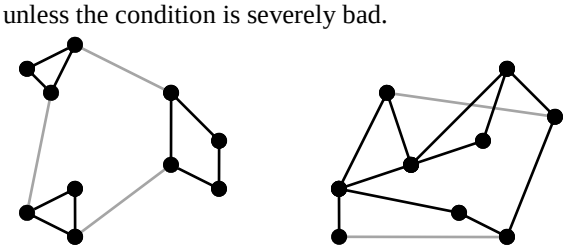

Fig. 4: The algorithm to detect similarities in civil infrastructures calculates a factor c_i for each attribute that determines the suitability of the attribute for determining similarity. Graph A (left) is more clustered than Graph B (right), therefore c_a will be higher than c_b.

To do this, clustering methods like distance-connectivity based algorithms can be used. This is done for all variables so that the overall similarity value $sim_{a,b}$ ranges within 0 and 1. The ideal goal is to find very dense and much separated clusters. Otherwise overlapping groups of similar entities would be created, which negatively affects the quality of the analysis. This model can be easily adopted in a Map/Reduce framework using Hadoop or a similar data structure.

We propose two approaches for finding the similarity threshold and therefore the basis for creating edges between data entities, which enables clustering of the graph. In the general approach, we do not start with predetermined assumptions, which parameters in the data will define similarity. In the specific approach, we start with rules that are likely going to define similarity and let the algorithm continue to learn from there.

4.1. General Approach

The algorithm is effectively trying to find a set of factors (c_1 to c_n), which define a similarity function and specify, how to cluster the dataset so that each cluster is clearly separated from each other. To get a good solution, the procedure has to be repeated multiple times until a satisfying result is reached. With each iteration, some of the factors are randomly adjusted and the algorithm evaluates the result. If the changes lead to an overall better outcome this process is repeated with these new values, otherwise it is repeated with the factors from the last iteration.

When repeated for many cycles, clusters can be derived which define closely related entities. With this knowledge, a series of different analysis can be performed quickly and automated:

- If a cluster has a problem, it is likely that each entity in this cluster has a similar problem
- If a cluster doesn't have a characteristic, it is unlikely that a node within the cluster has it
- The root of problems can be revealed because of a direct comparison of cluster members
- Problems could be automatically identified which the researcher is not even aware of

4.2. Specific Approach

By taking each variable in consideration like in the general approach, it is possible to identify similarities in any given dataset regardless of the type of data. Some issues however remain:

Repeating the algorithm until a satisfying result has been found, might take too long to be feasible. Further, some variables might not be compatible to each other and increasing the value for one might diffuse the result on another edge. This problem increases with the amount of variables for each entity. In such a case, the genetic algorithm would likely not reach a satisfiable result. One approach to

this dilemma is to select only specific parameters for a faster performance. This is done by locking the variables which are of less relevance for the analysis to a factor c_i=0. Therefore, they are excluded them from the equation. This is especially useful if a clear analysis goal is known.

Multidimensional analysis is possible too. For example, the task is to identify bridges that have both a bad deck conditioning and high maintenance costs. If analyzed over time researchers are able to derive information that bridges will require maintenance soon based on their current deck condition. They are even able to predict how much this will cost based on maintenance costs of similar bridges in terms of size, traffic, number of lanes or geographic location.

4.3. How to integrate real-time data

Since the impact of a variable (i) is modified by setting the factor ci to a value between 0 and 1, it is possible to exclude a variable for consideration at any given time by setting it to 0. This is useful when a dataset contains incomplete data. This also works the other way. If a new variable is added at any given time, it can be arranged that this new variable was already part of the old equation with c = 0.

Existing variables can be changed as well (while adding new data). E.g. a variable is given, which constantly counts the number of vehicles that pass a bridge. Instead of creating a new variable each time a vehicle passes, the algorithm increments a single variable which measures the traffic. The model changes the factors for variable, not the variables themselves. This lets a previously defined set of factors [$c_1 \ldots c_n$] still be valid because the factor determines the impact:

- If a variable is not suitable for identifying the similarity to other entities, changing the value will not have a big impact since c is small already.
- If a variable is important for identifying similarities, c has a high value and therefore changing the variable can reallocate the entity to another cluster.

We tested the concept on bridges based on the previously preprocessed dataset of the National Bridges Inventory (described in section 3). Parameters for evaluation ranged from deck or substructure condition, over the number of lanes, to safety ratings. Other data like average daily traffic, truck traffic and vertical clearance are incorporated as well, resulting in 84 parameters for each bridge. Parameters can be added at any given time to populate the network model for further advanced analysis. Figure 5 shows the final visualization of a sample dataset where the connections represent similarity and the color sufficiency rating as evaluated by the federal highway administration. 1000 random highway bridges in Nebraska were clustered based on similarity with the Fruchterman Reingold algorithm and colored based on the official FHA sufficiency ratings from 2015:

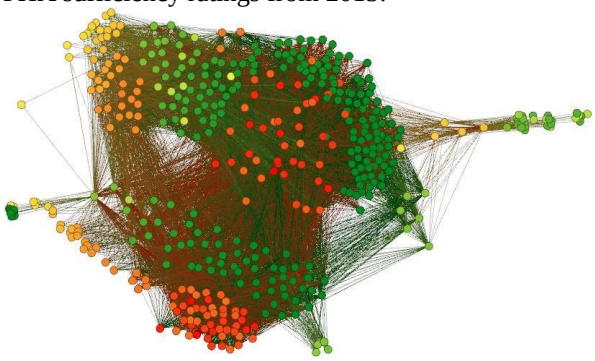

Fig. 5: Infrastructures were clustered based on similarity and colored based on sufficiency. Although generally healthy bridges (green), and unhealthy bridges (red) were separated during clustering some unhealthy bridges are in the health clusters. These outliers can now be identified and studied further, something not that easily achieved with a traditional database.

The graph shows clusters of healthy (green) and unhealthy (red, orange) bridges. However, it also shows that within the cluster of health bridges there are a few unhealthy outliers. This gives us directly an opportunity to have a closer manual look at these bridges and find out what is wrong with them. In contrast to traditional clustering based on only one or a few parameters (e.g. sufficiency rating) we identify more complex relationships between similar bridges. For example, we have four clusters of insufficient bridges (red) which obviously share different characteristics; otherwise, they would be located in one big cluster. In a second step, these differences can be identified by comparing the differences between clusters.

6. Conclusion

With sensor technology becoming more and more feasible, and advancements in computing efficiency, the path is paved for a new generation of structural health monitoring methodologies. Many researchers have laid out the challenges and opportunities for increasing safety and reliability in civil infrastructures. The critical step in the new SHM approaches is to build an Information System that fuels its information from the results of the analysis. This enables a graphical user interface to display critical bridges, expected problems, efficiency improvements in the inspection schedules and more.

In this study, we explored current approaches to structural health monitoring through assessment, sensor technology and prediction. Based on graph theory and genetic algorithms, we propose a method to cluster structures based on similarity. This allows data analysts to acquire in-depth knowledge on how combinations of SHM assessment and sensor parameters affect safety and reliability of civil structures. We orient our method on methods used in Big Data analysis and consider requirements like scalability, flexibility and incompleteness of the data. By using a genetic algorithm to improve the clustering over time, we try to take a step forward in filling some of the open challenges in anomaly detection within dynamic, attributed graphs.

Because this study is much exploratory in this stage we limited our analysis on smaller datasets. A comprehensive analysis of all 600.000 national bridges is out of scope for this paper. However, in further studies we want to expand on that concept and increase the variety of datasets outside the scope of infrastructure heath monitoring.

The literature review revealed that the performance and quality of such analysis could ultimately depend on only the available hardware. To minimize performance drawbacks and errors due missing, inaccurate or sparse data, choosing a proper data structure is essential. Relational database management systems provide little incentive for large data clusters, especially if they are mutating (data structure changes over time) or if they feed from sensor or live data.

7. References

[1] P. C. Chang, A. Flatau, and S. C. Liu, "Review paper: health monitoring of civil infrastructure," *Structural health monitoring*, vol. 2, no. 3, pp. 257–267, 2003.

[2] C. Middleton, P. Vardanega, G. Webb, and P. Fidler, "Smart Infrastructure–Are we delivering on the promise? Keynote Paper: 6th Australian Small Bridges Conference, Sydney, Australia, 27th to 28th May, 2014 (revised version).," 2015.

[3] O. Adarkwa, T. Schumacher, and N. Attoh-Okine, "Multiway Analysis of bridge structural types in the National Bridge Inventory (NBI): A tensor decomposition approach," in *Big Data, 2014 IEEE International Conference on*, 2014, pp. 1–6.

[4] S. W. Doebling, C. R. Farrar, M. B. Prime, "A summary review of vibration-based damage identification methods," *Shock and vibration digest*, vol. 30, no. 2, pp. 91–105, 1998.

[5] M. I. Friswell and J. E. T. Penny, "Is damage location using vibration measurements practical?," in *Proceedings of EUROMECH 365 international workshop: DAMAS*, 1997, vol. 97.

[6] C. H. J. Fox, "The location of defects in structures: a comparison of the use of natural frequency and mode shape data," in *PROCEEDINGS OF THE INTERNATIONAL MODAL ANALYSIS CONFERENCE*, 1992, pp. 522–522.

[9] D. C. Zimmerman and T. Simmermacher, "Model correlation using multiple static load and vibration tests," *AIAA journal*, vol. 33, no. 11, pp. 2182–2188, 1995.

[11] G. Chen and others, "Condition assessment of concrete structures by dynamic signature tests," in *Hopkins University*, 1999.

[12] Z. Hou and M. Noori, *Application of wavelet analysis for structural health monitoring*. 1999.

[13] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 1998, vol. 454, pp. 903–995.

[14] T. Schlurmann, "The empirical mode decomposition and the Hilbert spectra to analyse embedded characteristic oscillations of extreme waves," in *Rogue Waves*, 2001, pp. 157–165.

[15] M. Wu, X. Chen, and C. R. Liu, "Highway crack monitoring system," in *SPIE's 9th Annual International Symposium on Smart Structures and Materials*, 2002, pp. 293–299.

[16] D. Bernal, "Load vectors for damage localization," *Journal of Engineering Mechanics*, vol. 128, no. 1, pp. 7–14, 2002.

[17] T.-H. Yi and H.-N. Li, "Methodology developments in sensor placement for health monitoring of civil infrastructures," *International Journal of Distributed Sensor Networks*, vol. 2012, 2012.

[18] College of Engineering - University of Nebraska–Lincoln, "Bridge-ing Big Data Workshop," Omaha, 2015.

[20] H. Sohn, "A Bayesian probabilistic approach to damage detection for civil structures," Citeseer, 1998.

[21] S. Sankararaman and S. Mahadevan, "Bayesian methodology for diagnosis uncertainty quantification and health monitoring," *Structural Control and Health Monitoring*, vol. 20, no. 1, pp. 88–106, 2013.

[22] Z. X. Li, T. H. Chan, and J. M. Ko, "Fatigue analysis and life prediction of bridges with structural health monitoring data—Part I: methodology and strategy," *International Journal of Fatigue*, vol. 23, no. 1, pp. 45–53, 2001.

[23] S. Madden, "From databases to big data," *IEEE Internet Computing*, no. 3, pp. 4–6, 2012.

[24] S. J. Qin, "Process data analytics in the era of big data," *AIChE Journal*, vol. 60, no. 9, pp. 3092–3100, 2014.

[25] H. Chen, R. H. Chiang, and V. C. Storey, "Business Intelligence and Analytics: From Big Data to Big Impact.," *MIS quarterly*, vol. 36, no. 4, pp. 1165–1188, 2012.

[26] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 626–688, 2015.

[27] W. Fan and A. Bifet, "Mining big data: current status, and forecast to the future," *ACM sIGKDD Explorations Newsletter*, vol. 14, no. 2, pp. 1–5, 2013.

[28] N. Padhy, D. Mishra, R. Panigrahi, and others, "The survey of data mining applications and feature scope," *arXiv preprint arXiv:1211.5723*, 2012.

[29] D. Keco and A. Subasi, "Parallelization of genetic algorithms using Hadoop Map/Reduce," *SouthEast Europe Journal of Soft Computing*, vol. 1, no. 2, 2012.

[30] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[31] A. Verma, X. Llorà, D. E. Goldberg, and R. H. Campbell, "Scaling genetic algorithms using mapreduce," in *Intelligent Systems Design and Applications, 2009. ISDA'09. Ninth International Conference on*, 2009, pp. 13–18.

[32] S. Yang, J. Kim, and M. Chung, "A prediction model based on Big Data analysis using hybrid FCM clustering," in *Internet Technology and Secured Transactions (ICITST), 2014 9th International Conference for*, 2014, pp. 337–339.