Real-time Nondestructive Structural Health Monitoring using Support Vector Machines and Wavelets

Ahmet Bulut[a], Ambuj K. Singh[a], Peter Shin[b], Tony Fountain[b], Hector Jasso[b], Linjun Yan[c], and Ahmed Elgamal[c]

[a]Department of Computer Science, UC Santa Barbara, Santa Barbara, CA 93106-5110;
[b]San Diego Supercomputer Center, UC San Diego, San Diego, CA 92093-0505;
[c]Department of Structural Engineering, UC San Diego, San Diego,CA 92093-0085

ABSTRACT

We present an alternative to visual inspection for detecting damage to civil infrastructure. We describe a real-time decision support system for nondestructive health monitoring. The system is instrumented by an integrated network of wireless sensors mounted on civil infrastructures such as bridges, highways, and commercial and industrial facilities. To address scalability and power consumption issues related to sensor networks, we propose a three-tier system that uses wavelets to adaptively reduce the streaming data spatially and temporally. At the sensor level, measurement data is temporally compressed before being sent upstream to intermediate communication nodes. There, correlated data from multiple sensors is combined and sent to the operation center for further reduction and interpretation. At each level, the compression ratio can be adaptively changed via wavelets.

This multi-resolution approach is useful in optimizing total resources in the system. At the operation center, Support Vector Machines (SVMs) are used to detect the location of potential damage from the reduced data. We demonstrate that the SVM is a robust classifier in the presence of noise and that wavelet-based compression gracefully degrades its classification accuracy. We validate the effectiveness of our approach using a finite element model of the Humboldt Bay Bridge. We envision that our approach will prove novel and useful in the design of scalable nondestructive health monitoring systems.

Keywords: Decision support systems, classification in sensor networks, signal processing of raw sensor data

1. INTRODUCTION

As of 2003, there are 286,196 bridges in the United States,[1] many of them has been deteriorating due to age. Bridge inspectors regularly rate the condition of these structures by visual inspection, but the ratings are usually inaccurate.[2] On the other hand, as new fabrication and integration technologies reduce the cost and size of micro-sensors and wireless interfaces, it becomes feasible to automate the bridge inspection process. An automated decision support system could read from a densely distributed wireless network of sensors and actuators placed on bridges and assess bridge health. This would reduce the variability and error that is quite common in visual inspection studies. It would also enable real-time monitoring of bridge structure health, which is necessary for immediate action in case of emergencies such as earthquakes.

In this paper, we present our prototype for realizing an emergency response system. We simulate healthy and damaged bridges using a physics-based Finite Element (FE) model. In order to locate structural damage, we apply a force to the center of the bridge, and read vibration signatures as real-time streaming data from the sensor network. We feed the data into a Support Vector Machine (SVM) classifier in order to determine the location of the damage. We further explore reducing data temporally and spatially using wavelet approximation methods, and quantify its effect on classification accuracy.

The remainder of the paper is organized as follows: In Section 2, we introduce our testbed and simulation model. We define the classification problem and present our approach using SVMs. Later, we present the sensornet in our monitoring framework and discuss various data reduction techniques using wavelets. In Section 3, we discuss the results of our experiments and their implications for practice. Finally in Section 4, we conclude and discuss avenues for future work.

Send correspondence to Ahmet Bulut, e-mail: bulut@cs.ucsb.edu

2. PROPOSED SOLUTION

Ideally, we need to train a classifier using data from sensors mounted on real bridges. However, this is impractical, because it would require damaging the structure at various places and levels. Therefore, we resort to a physics-based simulation model. This enables us to generate data easily and quickly. In addition, advanced computing power in our resort allows us to build a model that closely reflects the reality. We claim and prove that this simulation-based training and the resulting classification technique considered here is a promising research direction.

2.1. Environment characteristics

2.1.1. PEER Testbed: Humboldt Bay Middle Channel Bridge

The Pacific Earthquake Engineering Research (PEER) Center is a consortium of western universities working in partnership with business, industry, and government for identifying and reducing earthquake risks to life safety and economy through performance-based engineering. PEER's testbed project seeks to synthesize disparate university research products of PEER's research into a coherent methodology and to demonstrate and exercise that methodology on six real facilities: two buildings, two bridges, a campus of buildings, and a network of highway bridges.

The Humboldt Bay Middle Channel Bridge was selected as one of the testbeds, and all reported modeling efforts are motivated by its structural configuration. The bridge, as shown in Figure 1, is a 330 meters long, 9-span composite structure with precast and prestressed concrete I-girders and cast-in-place concrete slabs to provide continuity. It is supported on eight pile groups, each of which consists of 5 to 16 prestressed concrete piles in soils potentially vulnerable to liquefaction (under extreme earthquake shaking conditions). The bridge was designed in 1968, built in 1971, and has been the object of two seismic retrofit efforts by California Department of Transportation (CalTrans): the first one was designed in 1985 and completed in 1987, and the second was designed in 2001 and completed in 2002.

Figure 1. View of Humboldt Bay Middle Channel Bridge (Courtesy of Caltrans)

2.1.2. Simulation Model

Figure 2 shows the Finite Element (FE) model of the bridge. In this model, the bridge deck system is simulated by three continuous beams connected by two perfect hinges. There are eight piers labeled from 1 to 8. Experience has shown that damages to bridges usually occur at the lower ends of bridge piers, where the structure is subjected to large cyclic loading. Thus, we simplify the problem by assuming that damage can only happen at the lower end of each pier. The damage is represented by a plastic hinge, 1/8 up from the bottom of the pier in question. The plastic hinge effect is achieved by composing the pier of two beam-column elements (one 7/8ths of the height of the pier, and the other 1/8th), and varying the material properties of the shorter (bottom) element. In particular, we reduce Young's Modulus. Each pier is also connected to the bridge deck by a perfect hinge. The bridge is fixed at each pier base and at both ends. Material and geometric properties of different element types are listed in Table 1.

2.1.3. "Hammer-like" force

For the "hammer-like" force that creates the vibration used to determine the location of the damage, we considered a 1 KN impact force at the bridge center in the transversal direction. The simulation model calculates the dynamic response of the bridge under the impact load, and a simulated sensor located in the center of each span records the acceleration at 100Hz in the transversal direction for five seconds. The sensors are shown as double-pointed arrows in Figure 2. Figure 3 shows

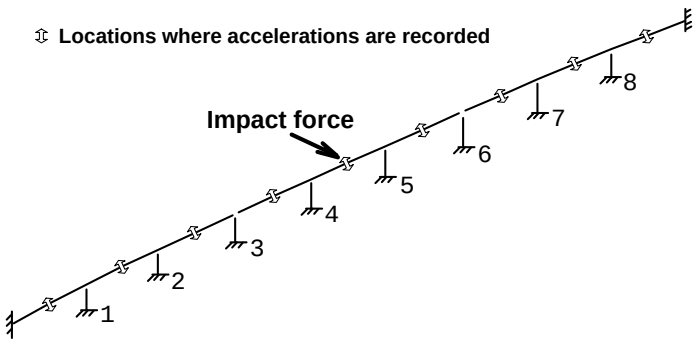

Figure 2. Finite Element Model of the Humboldt Bay Middle Channel Bridge

	Area m^2	Young's Modulus KN/m^2	Shear Modulus KN/m^2	Moment of Inertia (Y axis) m^4	Moment of Inertia (Z axis) m^4
Deck	9.0	3.0e+9	1.2e+7	7.0	70.0
Pier	3.6	3.0e+9	1.2e+7	2.7	0.432

Table 1. FEM Model Element Parameters

a recorded acceleration time history of the bridge middle point. In the real world, the acceleration time histories can be recorded by an accelerometer installed at corresponding locations.

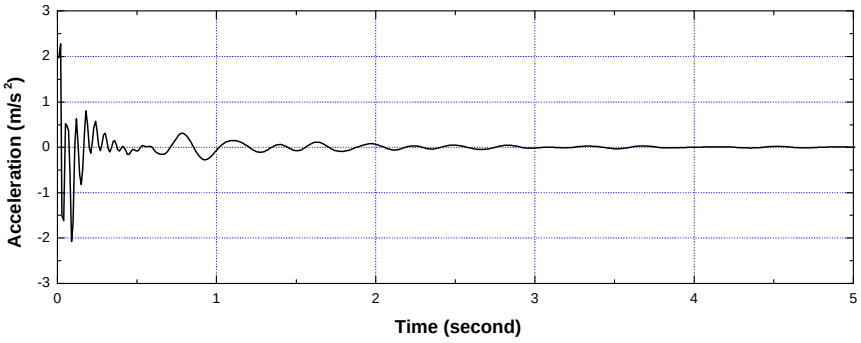

Figure 3. Acceleration time history of bridge middle point under impact load

2.1.4. Problem Definition and Dataset

As mentioned earlier, we assume that only plastic hinges appearing at the lower end of piers are considered as potential damage scenarios. For our initial efforts, we further refine the problem by assuming that only one pier is damaged at a time. The bridge contains eight piers, and therefore, eight potential plastic hinges locations corresponding to eight possible classes of damage. The problem that we consider in this paper can be restated as follows: classify a given vibration signature into one of these eight classes.

In this study, based on the common assumption by field engineers, we focus on reducing material strength by 10% to 50%. The reduction below 10% would be considered too low to be distinguished, and the reduction over 50% would be a structural failure rather than a damage. We generated 1700 cases of vibration signatures by reducing the Young's Modulus by 0%, 20% and 50%. Cases of 20% and 50% represent typical instances of small and large damage. These damage patterns are divided into two sets: (1) two thirds (1133 cases) of the dataset was used to train the SVM models, and (2) the other third (567 cases) was used as a test set. The accuracy of the classifiers is measured against the test set.

In order to study the noise tolerance of this methodology, the 1700 cases above contain synthetic noise, which is generated as follows: we assume that the noise for each time step follows an independently identical Gaussian distribution with mean 0. The real world contains a high ambient noise. Therefore, we used 20 times the typical resolution (0.0001 m/s^2) of an accelerometer as the standard deviation for the normal noise distribution.

2.2. Application of Support Vector Machines (SVM)

2.2.1. Related Work

Vibration signatures have been used to build non-linear neural network (NN) models of structures such as bridges.[3] The input to the model is a characterization of "hammer-like" forces, and the output is the response signature of the bridge. The neural network mimics the structure, and any potential damage is detected when the output differs from that of a healthy structure.

2.2.2. Description of SVMs

Our aim is to solve a nine-way classification problem: either conclude that the bridge is healthy, or determine the damage location among the eight possible alternatives. Given the high dimensionality of the input (4500 features corresponding to the vibration signature of nine sensors, each sensor sampling for 5 seconds at 100Hz), we chose SVMs for classification. SVMs solve a quadratic problem[4] with no local minimums, and therefore its response time is shorter compared to other algorithms like NNs. Furthermore, classifying new data takes a constant amount of time due to SVMs being linear discriminants.

We have trained eight SVMs, one for each possible damage location. Note that each location corresponds to a pier in the bridge. Since SVMs solve a two-way classification problem, each one gives an estimate to whether the bridge is damaged at the corresponding pier. If none of the SVMs signal a damage, the bridge is considered to be healthy. If two or more SVMs signal a potential damage, we classify the output as incorrect.

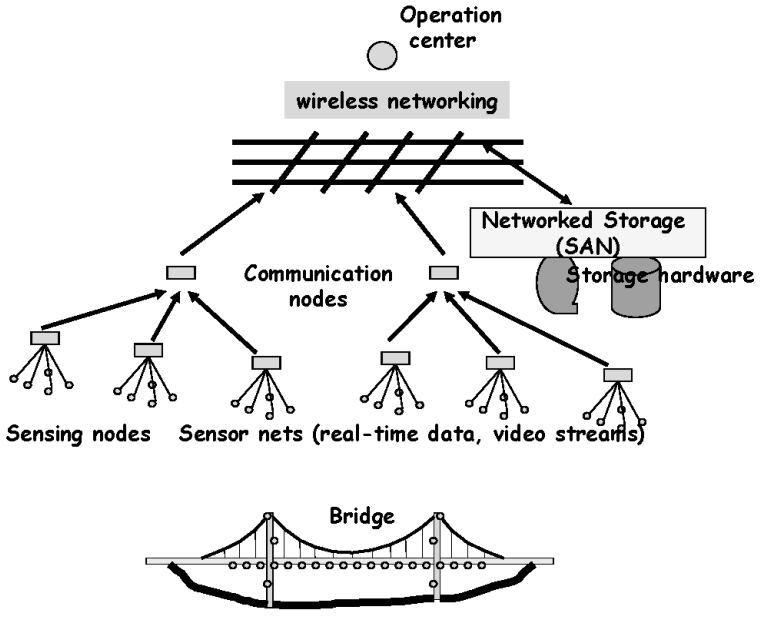

Figure 4. Elements of an Integrated Health Monitoring Framework

2.3. Sensor network

A significant challenge of our framework is the need to integrate multiple sensor streams to develop local and global health-state indicator variables that need to be queried and monitored by the system. The indicators may be defined either as user-specified aggregates (or other functions) over instantaneous values of several streams, or as pre-computed aggregates.

The sensor network shown in Figure 4 consists of a dense network of heterogeneous and homogeneous sensors with remote data collection, access, and control features. In this paper, we focus on one type of sensor data, i.e., vibration signatures. Analyzing data from thousands of sensors with long term and near real-time assessment requires building power-aware data mining system solutions. A majority of algorithms focus on hierarchical data mining, where sensor data are aggregated in a routing hierarchy towards a central processing center. Such in-network data compression approaches are well motivated by the nature of sensor data collected, which exhibit strong spatio-temporal correlations.[5] The processing center queries the sensor data using a drill-down approach. The utilization of our scheme in the sensor network is two-fold: (1) at the lowest level of the hierarchy, each *sensing node* removes the temporal redundancy in its stream of data. (2) In all other higher levels of the hierarchy, each *communication node* detects correlations among the information propagated from the lower level nodes and reduce the amount of information to push further up the hierarchy. The hierarchy in Figure 4 is a 2-level hierarchy. In general, the level of the hierarchy will increase with the scale of the monitoring framework, i.e., the number of sensing nodes and the number of communication nodes.

2.3.1. Sensing nodes

Sensing nodes are basic components in sensor networks. Each sensor integrated in a sensing node is a separate data source and monitors the physical environment by sampling physical signals. In other words, each sensor generates a discrete time series that contains temporally redundant information. Reducing the time series (data-stream) to include only potentially interesting events will help save the system energy: temporal data reduction is cheap since it involves only computation at a single sensor, and incurs no communication overhead. Therefore, data reduction is an essentially important in-network computation task. There has been a substantial amount of research work on data reduction especially in the context of sequence databases. Signal processing tools such as Discrete Fourier Transform[6–8] and Discrete Wavelet Transform[9, 10] are widely explored dimensionality reduction techniques. Piecewise constant approximation and its variations[11] are also straightforward and easy to implement, which give reasonably good results.

We use wavelets in our solution, since it allows summarization of a sensor stream incrementally at multiple resolutions. Multiple resolutions allow changing the compression ratio adaptively as will be explained later in this section. Wavelets are a mathematical tool for hierarchical decomposition of signals.[12] Among these, `Haar` wavelets constitute the simplest wavelet basis, as they are easy to understand and implement. Encoding a set of readings in a sensor network using `Haar` wavelets is addressed in.[13, 14]

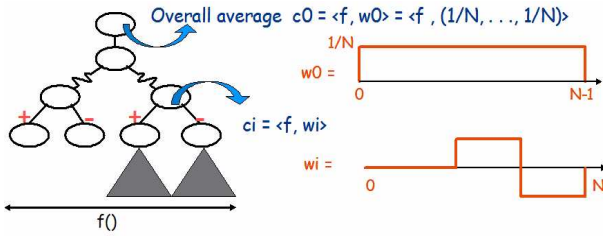

Figure 5. Each normalized coeffi cient c_i in the Haar decomposition tree is computed as $norm_i * [Avg(leftChildSubtree) - Avg(rightChildSubtree)]$, where $norm_i$ is an appropriate scaling factor for resolution i.

Resolution	Averages	Details
0	2 0 -1 3 1 5 4 -2	-
1	1 1 3 1	1 -2 -2 3
2	1 2	0 1
3	1.5	-0.5

Table 2. Haar Wavelet Decomposition. Resolution 0 shows the original signal. Pairwise averaging and differencing is performed on averages at resolution $i - 1$ to get averages and differences at resolution i.

We first summarize wavelet encoding using `Haar` wavelets. Haar encoding is the recursive pairwise averaging and differencing of the underlying data at multiple resolutions as depicted in Figure 5. Consider the following example, where we have the underlying signal $s = [2\ 0\ -1\ 3\ 1\ 5\ 4]$. We first perform pairwise averaging and differencing on s. For all

i, $1 \leq i \leq |s|/2 = 4$, we compute two values as $A_1[i] = (s[2i\text{-}1] + s[2i])/2$ and $D_1[i] = (s[2i\text{-}1] - s[2i])/2$. We get signals $A_1 = [1\ 1\ 3\ 1]$ and $D_1 = [1\ \text{-}2\ \text{-}2\ 3]$. In the next step, we perform pairwise averaging and differencing on A_1, and compute $A_2 = [1\ 2]$ and $D_2 = [0\ 1]$. Finally, we compute A_3 and D_3 using A_2. The signals A_3, D_3, D_2, and D_1 together constitute the `Haar` wavelet decomposition for signal s. The whole decomposition is given in a tabular format in Table 2. In the rest of the paper, we will focus on approximation coefficients A_i such that the higher the resolution i is the higher the compression ratio. As demonstrated in,[15] this simplification achieves an excellent accuracy in the statistical analysis of raw sensor data.

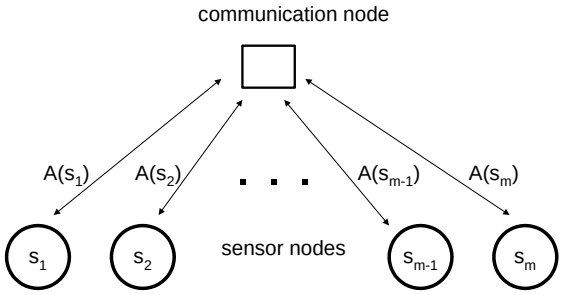

Figure 6. Sensors numbered from 1 to m are connected with a communication node. For each sensor, s_i denotes the raw signal, and $A(s_i)$ denotes a high-resolution approximation.

Once the approximation at an appropriate resolution is computed, it is propagated to the parent communication node. From this point on, we drop the subscript and use the capital letter A to denote an approximation at a pre-determined resolution i.

2.3.2. Communication nodes

Typically the number of sensors deployed in field increases with the size of the infrastructure being monitored. Our testbed is an example of a small infrastructure. However, large monitoring networks consist of thousands of sensing nodes. The size of the sensing network adversely affects the computational capability of the overall low-bandwidth, battery-powered network of sensors, since the amount of streaming data to deal with grows linearly with the number of sensors. Fortunately, the measurements across sensors are expected to exhibit correlations.[5] For this purpose, we employ a correlation detection filter at intermediate communication nodes to reduce data spatially. Reducing the number of sensors renders our algorithms more scalable for large sensor networks consisting of thousands of sensor nodes. This approach is similar to in-network aggregation.

Consider the example shown in Figure 6, where s_i denotes the raw signal at sensor node with identifier i. The communication node receives m different sensor approximations as $A(s_1)$,...,$A(s_m)$. Each approximation $A(s_i)$ denotes a vector of a pre-determined length. We denote all measurements collected as a matrix $X = [A(s_1) \ldots A(s_m)]$. On this matrix X, we compute CC, the $m \times m$ matrix of correlation coefficients. Let C denote the covariance matrix, then

$$CC(i,j) = C(i,j)/\sqrt{(C(i,i) * C(j,j))} \tag{1}$$

Once we compute CC, we can cluster approximations such that in each cluster G_i, every sensor pair is correlated within a given threshold τ, i.e.,

$$G_i = \{j | \tau < CC(i,j) \tag{2}$$

We can safely disregard all but cluster centers $A(s_i)$. The retained approximations are propagated further up the hierarchy.

2.4. Reducing the complexity of computing correlations

Computing a correlation matrix is expensive especially if the number of sensors m is large. Therefore, we use a faster, but a less accurate algorithm for computing this matrix. The computation problem is an all-pairs query such that given a threshold value τ, we have to find all sensor pairs, whose correlation coefficient is greater than or equal to τ. From this point on, we refer to this problem as ϵ-correlation, where $\tau = (1 - \epsilon^2/2)$. Note that this definition restricts us to only

positive correlations. The correlation coefficient between s_i and s_j can be reduced to the Euclidean distance between their corresponding normalized sequences.[16] We normalize a sequence $s[1], s[2], s[3], \ldots, s[N]$ as follows:

$$\hat{s}[k] = \frac{s[k] - \mu_s}{\sigma_s} \qquad k = 1, 2, \ldots, N \tag{3}$$

where μ_s is the arithmetic mean and $\sigma_s = \sqrt{\sum_{k=1}^{N}(s[k] - \mu_s)^2}$.

THEOREM 2.1. *The correlation coefficient of two sequences s_i and s_j is $corr(s_i, s_j) = 1 - \frac{1}{2}d^2(\hat{s}_i, \hat{s}_j)$, where $d(\hat{s}_i, \hat{s}_j)$ is the Euclidean distance between $\hat{s}_i$ and $\hat{s}_j$.*

Proof. The Euclidean distance between $\hat{s}_i$ and $\hat{s}_j$ is,

$$d^2(\hat{s}_i, \hat{s}_j) = \sum_{k=1}^{N}(\hat{s}_i[k] - \hat{s}_j[k])^2 = \sum_{k=1}^{N}\hat{s}_i[k]^2 + \sum_{k=1}^{N}\hat{s}_j[k]^2 - 2\sum_{k=1}^{N}\hat{s}_i[k]\hat{s}_j[k] \tag{4}$$

where $\sum_{k=1}^{N}\hat{s}_i[k]^2 = \sum_{k=1}^{N}\hat{s}_j[k]^2 = 1$ due to unit normalization. The correlation coefficient $corr(s_i, s_j)$ is equal to the inner product $\sum_{k=1}^{N}\hat{s}_i[k]\hat{s}_j[k]$ as shown in.[16] Therefore, $corr(s_i, s_j) = CC(i, j) = 1 - d^2(\hat{s}_i, \hat{s}_j)/2$ □

A lower bounding property $d(A(\hat{s}_i), A(\hat{s}_j)) \leq d(\hat{s}_i, \hat{s}_j)$ holds since wavelet transformation is an orthogonal transformation.[12] We use this lower bounding property in pruning out some of the candidate pairs because sensors s_i and s_j cannot be ϵ-correlated if

$$\epsilon < d(A(\hat{s}_i), A(\hat{s}_j)) \leq d(\hat{s}_i, \hat{s}_j) \tag{5}$$

At each communication node, approximations are inserted into an R*-Tree,[17] a variant of R-Tree[18] as shown pictorially in Figure 7. An R*-Tree is a spatial access method, which splits feature space in hierarchically nested, possibly overlapping, boxes called Minimum Bounding Rectangles (MBR). For notational convenience, we denote each approximation as s_i. The ϵ-correlation can be computed by performing a self-join on this tree. We enlarge each of the MBRs in the tree by $\frac{\epsilon}{2}$ as shown in Figure 8 and perform the so-called plane sweep algorithm.[19–21] We post process the results in order to discard false alarms. The system incurs false alarms due to the lower bounding property in distance computations. However, it does not incur any false dismissals.

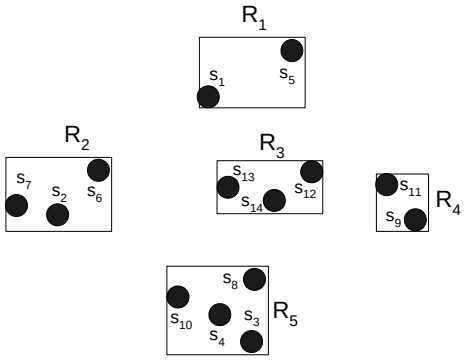

Figure 7. Index structure constructed at a communication node. Each individual point denotes a compressed signal propagated by a sensing node. For notational convenience, we denote each approximation as s_i.

This approach is more efficient than a base line approach that considers $m \times (m-1)/2$ pairs for correlation. Our approach reduces the computational complexity by filtering out all non-potential candidates using the index structure as follows: we consider intra-MBR-correlations and inter-MBR-correlations separately. If two sensors reside in the same MBR, no-pruning is performed, and therefore, they are included in the candidate set. However, if they reside in different MBRs R_i and R_j, they are considered to be a candidate only if the enlarged MBRs R_i' and R_j' overlap. For example in Figure 8, none of the sensors in MBR R_5 can be ϵ-correlated with sensors in MBR R_4. However, a sensor in R_5 can be ϵ-correlated with a sensor in R_3 since the enlarged MBRs overlap.

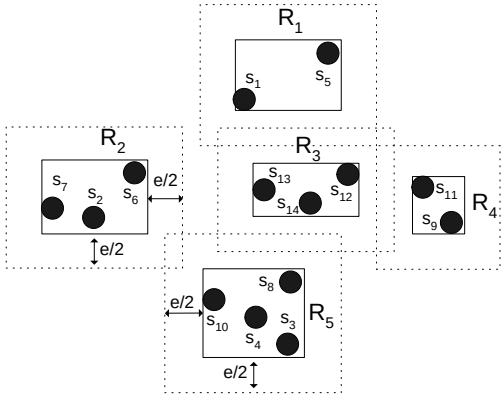

Figure 8. Index structure after MBRs are enlarged by $\epsilon/2$. We perform a self-join on this modified structure and identify all correlated sensors.

3. EXPERIMENTAL RESULTS

In this section, we study the classification accuracy of our approaches on the test-bed network. The dataset we use is the sensor data collected on the Humboldt Bay bridge. The original data is a matrix X of dimensions 1700×4500. Each row corresponds to a scenario in which one of the piers is broken. Each sensor records an acceleration time history of 500 time units length. For each row i, the data at columns $(j-1) \times 500$ to $j \times 500$ denotes the raw data recorded by sensor j (j goes from 1 to 9). The approximate dataset X_i consists of approximations at resolution i, and is a matrix of dimensions $1700 \times \lceil 4500/2^i \rceil$.

3.1. SVM results on exact data

As mentioned earlier in Section 2.1.4, the accuracy of SVMs is measured on the hold-out validation set, which contains 567 scenarios of a total of 1700. All classifiers have 100% accuracy in identifying damage on each pier. This reassures us that the SVMs scale well with high dimensional data. We quantify how SVM accuracy degrades with reducing the size of dataset using wavelets. The reduction is performed on two separate dimensions: the first is on the temporal dimension, and the second is on the spatial dimension. We present our results on these two approaches next.

	Broken pier number							
level	1	2	3	4	5	6	7	8
0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.94
3	0.94	1.00	1.00	1.00	1.00	1.00	0.94	0.94
4	0.94	1.00	1.00	1.00	1.00	1.00	0.94	0.94
5	0.94	1.00	1.00	1.00	1.00	1.00	0.94	0.90
6	0.94	0.98	1.00	1.00	0.99	1.00	0.94	0.88
7	0.94	0.94	1.00	1.00	0.94	1.00	0.94	0.88
8	0.88	0.88	0.94	0.94	0.94	0.94	0.88	0.88
9	0.88	0.88	0.88	0.88	0.94	0.94	0.88	0.88

Table 3. SVM results on multi-resolution data: Level 0 represents the exact data $X = X_0$. In general, level i represents X_i that contains wavelet approximation coefficients at resolution i.

3.2. SVM results after temporal reduction

The classification accuracy changes as we reduce the dimensionality of the data by increasing the resolution (higher levels). Table 3 shows the results for using `Haar` wavelets with varying resolution. As the resolution increases, the number of

approximation coefficients decreases exponentially. A smaller number of features results in a lower classification accuracy. However, the classifiers still achieve an excellent accuracy. Even though the largest resolution (level 9 in Table 3) reduces the total number of 4500 features down to 9 features, the classification accuracy decreases only to 88%.

We perform a similar experiment using different wavelet families such as `Daubechies`, `Symlets`, and `Coifflets`. However, the experiments show that all wavelet bases show an almost equal classification accuracy provided that the number of coefficients retained is the same.

sensor #	s1	s2	s3	s4	s5	s6	s7	s8	s9
s1	1.00	0	0	0.71	0	0	0.81	0	0
s2	-	1.00	0	0	0.84	0	0	0	0.72
s3	-	-	1.00	0	0	0.73	0	0	0
s4	-	-	-	1.00	0	0	0.94	0	0
s5	-	-	-	-	1.00	0	0	0	0.76
s6	-	-	-	-	-	1.00	0	0	0
s7	-	-	-	-	-	-	1.00	0	0
s8	-	-	-	-	-	-	-	1.00	0
s9	-	-	-	-	-	-	-	-	1.00

Table 4. Correlation coeffi cients matrix computed on X_9. The entries $(i, j) < \tau = 0.7$ are shown as 0. Since the correlation matrix is symmetric, we only show the upper right half.

3.3. SVM results after spatial reduction

Table 4 shows the correlation matrix on X_9 for a correlation threshold of $\tau = 0.7$. As one can confirm, there are four different clusters of correlated sensors: $G_1 = \{1, 4, 7\}$, $G_2 = \{2, 5, 9\}$, $G_3 = \{3, 6\}$, and $G_4 = \{8\}$. Instead of working with all nine sensors, we can work with a subset of sensors that includes a sensor from each cluster, e.g., $S = \{1, 3, 5, 8\}$. Other choices are possible as well. We test the classification accuracy on this reduced dataset and compare it with the accuracy achieved on the whole dataset. As Table 5 shows, the accuracy is the same (except for piers 4 and 5), validating our approach experimentally.

	Broken pier number							
	1	2	3	4	5	6	7	8
All	0.88	0.88	0.88	0.88	0.94	0.94	0.88	0.88
S	0.88	0.88	0.88	0.94	0.88	0.94	0.88	0.88

Table 5. SVM results on the dataset X_9 that contains level-9 approximation coeffi cients from all nine sensors and the reduced dataset S that contains level-9 approximation coeffi cients only from sensors $1, 3, 5$, and 8.

4. CONCLUSION AND FUTURE WORK

We have built a system that could replace the visual inspection of bridges, and that could reduce the current variability and error in measurements. The system eliminates inspection costs and requires only a fraction of time to assess structural damage. The proposed health monitoring framework consists of a multi-level communication hierarchy that can scale to large sensornets consisting of thousands of sensors. We selected the Humboldt Bay Bridge as a testbed, and simulated it using a finite element model. We first applied a force to the center of the bridge, recorded the resulting vibration signatures, and then trained an SVM classifier on these recordings that detects the location of potential damage. Signals from sensing nodes travel to the operation center for final inspection through communication nodes in a three-level communication architecture. We used Discrete Wavelet Transformation for temporal reduction of the signals at sensor level, and for spatial reduction throughout communication nodes and the operation center. The approach trades off accuracy for efficient handling and processing of sensor data in order to save the communication overhead and power-consumption.

The study findings show that the proposed system is a feasible approach for structural damage detection. Using low-dimensional streaming data, the SVM classifier is able to detect the location of damage with a high degree of accuracy. Furthermore, the SVM has proven to be robust in the presence of noise. Compression via wavelets results in a graceful

degradation of classification accuracy, thus providing a scalable solution for ever growing data in sensor network applications.

The system presented here is an important step towards building a more sophisticated system working on real bridges and with other types of structural damage. We are planning a number of extensions and enhancements to our current approach. These include explicit consideration of relevant utility measures associated with resources (e.g., bandwidth, power, and inspections) in order to provide a principled approach to decision making in this application. This study will guide us to determine optimal sensor density and distribution as well as operational characteristics. In addition to modelling enhancements, we intend to explore the application of these methods on larger and more complex bridges. The results from these investigations will move us closer to realizing our ultimate goal of real-time decision support for large bridges and other civil infrastructures.

REFERENCES

1. "Better roads for the government project team," in *http://www.betterroads.com/articles/bridgeinv03.htm*,
2. G. Washer, "Reliability of visual inspection for highway bridges," *USDOT, FHWA* **1**(FHWA-RD-01-020), 2001.
3. S. F. Masri, J. P. Caffrey, R. W. Wolfe, and A. W. Smyth, "An overview of some local and global approaches for vibration based damage assessment of structures,"
4. K. P. Bennett and C. Campbell, "Support vector machines: hype or hallelujah?," *SIGKDD Explor. Newsl.* **2**(2), pp. 1–13, 2000.
5. D. Ganesan, B. Greenstein, D. Perelyubskiy, D. Estrin, and J. Heidemann, "An evaluation of multi-resolution storage for sensor networks," in *Sensys*, pp. 89–102, 2003.
6. R. Agrawal, C. Faloutsos, and A. Swami, "Efficient similarity search in sequence databases," in *FODO*, pp. 69–84, 1993.
7. Y. Moon, K. Whang, and W. Han, "General match: A subsequence matching method in time-series databases based on generalized windows," in *SIGMOD*, pp. 382–393, 2002.
8. D. Rafiei and A. O. Mendelzon, "Efficient retrieval of similar time sequences using DFT," in *FODO*, 1998.
9. T. Kahveci and A. K. Singh, "Variable length queries for time series data," in *ICDE*, pp. 273–282, 2001.
10. I. Popivanov and R. J. Miller, "Similarity search over time series data using wavelets," in *ICDE*, 2002.
11. E. Keogh, K. Chakrabarti, S. Mehrotra, and M. Pazzani, "Locally adaptive dimensionality reduction for indexing large time series databases," in *SIGMOD*, pp. 151 – 162, 2001.
12. S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 2 ed., 1999.
13. A. Bulut and A. K. Singh, "SWAT: Hierarchical stream summarization in large networks," in *ICDE*, pp. 303–314, 2003.
14. J. M. Hellerstein, W. Hong, S. Madden, and K. Stanek, "Beyond average: Toward sophisticated sensing with queries," in *IPSN*, 2003.
15. A. Bulut and A. K. Singh, "A unified framework for monitoring data streams in real time," in *ICDE*, 2005.
16. Y. Zhu and D. Shasha, "Statstream: Statistical monitoring of thousands of data streams in real time," in *VLDB*, pp. 358–369, 2002.
17. N. Beckmann, H. P. Kriegel, R. Schneider, and B. Seeger, "The R*-tree: An efficient and robust access method for points and rectangles," in *SIGMOD*, pp. 322–331, 1990.
18. A. Guttman, "R-trees: A dynamic index structure for spatial searching," in *SIGMOD*, pp. 47–57, 1984.
19. T. Brinkhoff, H. P. Kriegel, and B. Seeger, "Efficient processing of spatial joins using R-trees," in *SIGMOD*, pp. 237–246, 1993.
20. Y. Huang, N. Jing, and E. A. Rundensteiner, "Spatial joins using R-trees: Breadth-first traversal with global optimizations," in *The VLDB Journal*, pp. 396–405, 1997.
21. T. Kahveci, C. A. Lang, and A. K. Singh, "Joining massive high-dimensional datasets," in *ICDE*, pp. 264–276, 2003.