

Online Data Fusion Using Incremental Tensor Learning

Nguyen Lu Dang Khoa¹, Hongda Tian², Yang Wang², and Fang Chen²

¹ Data61, CSIRO, Eveleigh, NSW, Australia khoa.nguyen@data61.csiro.au

² University of Technology Sydney, Broadway, NSW, Australia
{[hongda.tian](mailto:hongda.tian@uts.edu.au),[yang.wang](mailto:yang.wang@uts.edu.au),[fang.chen](mailto:fang.chen@uts.edu.au)}@uts.edu.au

Abstract. Despite the advances in Structural Health Monitoring (SHM) which provides actionable information on the current and future states of infrastructures, it is still challenging to fuse data properly from heterogeneous sources for robust damage identification. To address this challenge, the sensor data fusion in SHM is formulated as an incremental tensor learning problem in this paper. A novel method for online data fusion from heterogeneous sources based on incrementally-coupled tensor learning has been proposed. When new data are available, decomposed component matrices from multiple tensors are updated collectively and incrementally. A case study in SHM has been developed for sensor data fusion and online damage identification, where the SHM data are formed as multiple tensors to which the proposed data fusion method is applied, followed by a one-class support vector machine for damage detection. The effectiveness of the proposed method has been validated through experiments using synthetic data and data obtained from a real-life bridge. The results have demonstrated that the proposed fusion method is more robust to noise, and able to detect, assess and localize damage better than the use of individual data sources.

Keywords: Data fusion · Incrementally-coupled tensor learning · Online learning · Anomaly detection.

1 Introduction

Civil infrastructures are critical to our society as they support the flows of people and goods within cities. Any problem on such a structure from small damage to catastrophic failures would result in certain economic and potential life losses. Currently most of structural maintenances are time-based, e.g. visual inspections at predefined regular schedules. Structural Health Monitoring (SHM) is a condition-based monitoring using sensing system which provides actionable information on the current and future states of infrastructures. SHM systems built on advanced sensing technologies and data analytics allow the shift from time-based to condition-based maintenance [3].

In SHM, measured data are often in a multi-way form, i.e. multiple sensors at different locations simultaneously collect data over time. These data are highly

redundant and correlated, which are suitable to be analyzed using tensor analysis [7, 6]. There were efforts to apply tensor CANDECOMP/PARAFAC (CP) [7] decomposition in SHM for damage identification [10, 6, 5]. However, these approaches are confined to a fusion from sensors of the same type to guarantee the data can be formed in a single tensor. In many SHM systems, data come from heterogeneous sources due to an availability of different types of sensors (e.g. accelerometers, strain gauges and thermometers). Additionally, existing methods for data fusion from heterogeneous sources using tensor analysis [1, 12] mainly work in an offline manner, which is not practical for SHM applications. In this paper, we propose a method to fuse data online from heterogeneous sources based on incremental tensor learning, which is then used for damage identification in SHM. Our contributions are summarized as follows:

- We propose a method for online data fusion from heterogeneous sources using incrementally-coupled tensor learning. Specifically, our method collectively and incrementally updates component matrices for CP decomposition from multiple tensors when new data arrive.
- We develop a case study used in SHM for sensor data fusion and online damage identification. In the case study, the SHM data are formed as multiple tensors to which an incremental tensor fusion is applied, followed by a one-class support vector machine (OCSVM) for damage detection.
- We demonstrate the effectiveness of the proposed method through experiments using synthetic data and real data obtained from a bridge in Sydney.

In this paper, we represent a tensor as a three-way array, which is a typical case in SHM. However, all the theories could be generalized for a n -way array. The remainder of the paper is organized as follows. Section 2 summarizes the related work. Section 3 describes our novel method to incrementally update component matrices from multiple tensors at the same time and its uses for online damage identification in SHM. Section 4 presents the experimental results. We conclude our work in Section 5.

2 Related Work

Incremental tensor analysis, which is used for online applications, mainly focuses on Tucker decomposition [13, 8] since it makes use an extensive literature of incremental singular value decomposition (SVD). There are a few works [9, 15, 5] on an incremental learning for CP decomposition. Nion and Sidiropoulos [9] proposed a method to incrementally track the SVD of the unfolded tensor for CP. However, this technique scales linearly with time, which is impractical to use for large datasets. Zhou et al. [15] discussed a method to incrementally track CP decomposition over time. It follows an alternating least square (ALS) style: update a component matrix while fixing all the others. However, the update only occurs once instead of an iterative process, which makes the approximation sometimes ineffective. Khoa et al. [5] extended this method in a proper ALS style, resulting in more accurate updated component matrices.

Data fusion using coupled matrix/tensor decomposition has become popular recently [1, 12, 14]. Instead of using ALS algorithms, Acar et al. [1] proposed an all-at-once optimization approach for coupled matrix and tensor factorization. Sorber et al. [12] presented a framework where the type of tensor decomposition, the coupling between factorizations and the structure imposed on the factors can all be chosen freely without any changes to the solver. In [14], the authors proposed a method to learn a clustered low-rank representation for multiview spectral clustering using structured matrix factorization. However, these methods work offline which limits their applications.

3 Online Damage Identification Using Incrementally-Coupled Tensor Learning

The proposed method to identify damage online using incrementally-coupled tensor learning is depicted in Fig. 1. In SHM, vibration responses of a structure are measured over time by different types of sensors (e.g. accelerometers and strain gauges). The data from each type of sensors when the structure is in a healthy condition can be considered as a three-way tensor (*feature* \times *location* \times *time*). Thus we have tensor \mathcal{X}_1 for accelerometers and \mathcal{X}_2 for strain gauges. Feature is the information extracted from the raw signals; location represents sensor positions; and time indicates data snapshots at different timestamps. Each slice along the time axis shown in Fig. 1 is a frontal slice representing all features across all locations at a particular time.

Training tensors \mathcal{X}_1 and \mathcal{X}_2 are jointly decomposed into matrices of different modes using coupled tensor-tensor decomposition as described in Section 3.1. When new data arrive, these matrices are jointly updated using our proposed incremental tensor analysis as in Section 3.2. A monitoring of these factor matrices over time will help identify the damage in the structure (Section 3.3).

3.1 Data Fusion Using Coupled Tensor-Tensor Decomposition

Two typical approaches for tensor decomposition are CP decomposition and Tucker decomposition [7]. After a decomposition of a three-way tensor, three component matrices can be obtained representing information in each mode. In the case of SHM data as in Fig. 1, they are associated with feature (matrix A_1), location (matrix B_1) and time modes (matrix C) (for tensor \mathcal{X}_1). We also obtain component matrices A_2 , B_2 and C (for tensor \mathcal{X}_2). Note that C is the same for \mathcal{X}_1 and \mathcal{X}_2 since time information is shared between these two types of sensors.

In CP method, the decomposed matrices are unique provided that we permute the rank-one components [7]. Therefore it is easy to interpret the artifact in each mode separately using its corresponding component matrix. Thus, CP method is used in this paper for our SHM applications.

The problem to jointly decompose \mathcal{X}_1 and \mathcal{X}_2 using CP can be formulated as

$$f(A_1, B_1, C, A_2, B_2) = \frac{1}{2} \|\mathcal{X}_1 - [A_1, B_1, C]\|^2 + \frac{1}{2} \|\mathcal{X}_2 - [A_2, B_2, C]\|^2 \quad (1)$$

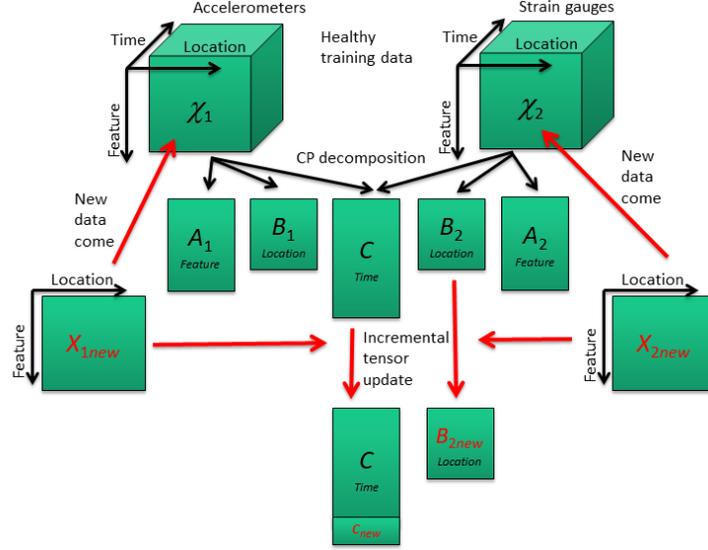


Fig. 1: A flowchart of incremental tensor fusion for online damage identification.

where $\mathcal{X}_i = [A_i, B_i, C]$ represents the CP decomposition and can be formulated as $X_{i(1)} = A_i(C \odot B_i)^\top$, $X_{i(2)} = B_i(C \odot A_i)^\top$ and $X_{i(3)} = C(B_i \odot A_i)^\top$ ($X_{i(j)}$ is an unfolding matrix of \mathcal{X}_i in mode j and \odot is the Khatri-Rao product) [7]. Equation 1 can be solved using ALS and it is summarized in Algorithm 1. In our SHM application (as in Section 4.2), the time matrix C is used for damage detection in time mode while location matrix B_2 is used for damage localization.

Algorithm 1 Coupled Tensor-Tensor Decomposition

Input: Tensors $\mathcal{X}_1, \mathcal{X}_2$, number of components R

Output: Component matrices A_1, B_1, C, A_2 and B_2

- 1: Initialize A_1, B_1, C, A_2 and B_2
 - 2: **repeat**
 - 3: $A_1 = \arg \min_{A_1} \frac{1}{2} \|X_{1(1)} - A_1(C \odot B_1)^\top\|^2$ (fixing B_1 and C)
 - 4: $B_1 = \arg \min_{B_1} \frac{1}{2} \|X_{1(2)} - B_1(C \odot A_1)^\top\|^2$ (fixing A_1 and C)
 - 5: $A_2 = \arg \min_{A_2} \frac{1}{2} \|X_{2(1)} - A_2(C \odot B_2)^\top\|^2$ (fixing B_2 and C)
 - 6: $B_2 = \arg \min_{B_2} \frac{1}{2} \|X_{2(2)} - B_2(C \odot A_2)^\top\|^2$ (fixing A_2 and C)
 - 7: $C = \arg \min_C \frac{1}{2} \|[X_{1(3)} \ X_{2(3)}] - C[(B_1 \odot A_1)^\top (B_2 \odot A_2)^\top]\|^2$ (fixing A_1, B_1, A_2 and B_2)
 - 8: **until** convergence
-

3.2 Incremental Tensor Update

OnlineCP-ALS [5] was proposed to incrementally update the component matrices of a tensor when new data arrive, which was shown to be better than other incremental CP decomposition methods. Using similar ideas, we propose a tech-

nique to jointly and incrementally update component matrices from different tensors over time as follows.

Update temporal mode C Due to an arrival of new information (new frontal slices in time mode), additional rows will be added to component matrix C . By fixing A_1 , B_1 , A_2 and B_2 , we can solve C from Equation 1 as:

$$\begin{aligned} C &= \arg \min_C \frac{1}{2} \left\| [X_{1(3)} \ X_{2(3)}] - C [(B_1 \odot A_1)^\top \ (B_2 \odot A_2)^\top] \right\| \\ &= \arg \min_C \frac{1}{2} \left\| \begin{bmatrix} [X_{1old(3)} \ X_{2old(3)}] - C_{old} [(B_1 \odot A_1)^\top \ (B_2 \odot A_2)^\top] \\ [X_{1new(3)} \ X_{2new(3)}] - C_{new} [(B_1 \odot A_1)^\top \ (B_2 \odot A_2)^\top] \end{bmatrix} \right\|. \end{aligned}$$

Thus,

$$C = \begin{bmatrix} C_{old} \\ C_{new} \end{bmatrix} = \begin{bmatrix} C_{old} \\ [X_{1new(3)} \ X_{2new(3)}] [(B_1 \odot A_1)^\top \ (B_2 \odot A_2)^\top]^\dagger \end{bmatrix}, \quad (2)$$

where \dagger is a matrix pseudo-inverse. Therefore, new rows added to C can be estimated using only new information appending in time mode.

Update non-temporal mode A_1 , B_1 , A_2 and B_2 By fixing B_1 and C for updating A_1 , the Equation 1 can be written as $\frac{1}{2} \|X_{1(1)} - A_1(C \odot B_1)^\top\|^2$. Using the approach as in [15], by taking the derivative of this function with regard to A_1 and setting it to zero, we have:

$$A_1 = \frac{X_{1(1)}(C \odot B_1)}{(C \odot B_1)^\top(C \odot B_1)} = P_1 Q_1^{-1},$$

where $P_1 = X_{1(1)}(C \odot B_1)$ and $Q_1 = (C \odot B_1)^\top(C \odot B_1)$.

Directly calculating P_1 and Q_1 is costly since $(C \odot B_1)$ is a big matrix. By representing $X_{1(1)}$ and C with *old* and *new* information, we can have $P_1 = P_{1old} + X_{1new(1)}(C_{new} \odot B_1)$ and $Q_1 = Q_{1old} + C_{new}^\top C_{new} \odot B_1^\top B_1$ (\odot is the Hadamard product). Therefore, A_1 can be computed as:

$$A_1 = P_1 Q_1^{-1} = \frac{P_{1old} + X_{1new(1)}(C_{new} \odot B_1)}{Q_{1old} + C_{new}^\top C_{new} \odot B_1^\top B_1}. \quad (3)$$

Similarly we can derive the update for B_1 as:

$$B_1 = U_1 V_1^{-1} = \frac{U_{1old} + X_{1new(2)}(C_{new} \odot A_1)}{V_{1old} + C_{new}^\top C_{new} \odot A_1^\top A_1}, \quad (4)$$

where $U_1 = X_{1(2)}(C \odot A_1)$ and $V_1 = C^\top C \odot A_1^\top A_1$ [15].

Likewise, A_2 and B_2 can be updated as:

$$A_2 = \frac{P_{2old} + X_{2new(1)}(C_{new} \odot B_2)}{Q_{2old} + C_{new}^\top C_{new} \odot B_2^\top B_2}; \quad (5)$$

$$B_2 = U_2 V_2^{-1} = \frac{U_{2old} + X_{2new(2)}(C_{new} \odot A_2)}{V_{2old} + C_{new}^\top C_{new} \circ A_2^\top A_2}. \quad (6)$$

We can see that by storing information from previous decomposition (i.e. P_1 , Q_1 , U_1 , V_1 , P_2 , Q_2 , U_2 and V_2), component matrices A_1 , B_1 , A_2 and B_2 are updated using only new information arriving in time mode.

onlineCP-Fusion For two three-way tensors that grow with time (shared C mode), a two-staged procedure is proposed to jointly incrementally update tensor component matrices. The technique, which is called onlineCP-Fusion, is described in Algorithm 2.

Algorithm 2 Incrementally-Coupled Tensor Update: onlineCP-Fusion

Input: Training tensors \mathcal{X}_{1train} and \mathcal{X}_{2train}

Output: Component matrices C , A_1 , B_1 , A_2 and B_2 when new data arrive

1: Initialization/training stage:

C , A_1 , B_1 , A_2 and B_2 are obtained using Algorithm 1 on training tensors

$P_1 = X_{1train(1)}(C \odot B_1)$ and $P_2 = X_{2train(1)}(C \odot B_2)$

$Q_1 = C^\top C \circ B_1^\top B_1$ and $Q_2 = C^\top C \circ B_2^\top B_2$

$U_1 = X_{1train(2)}(C \odot A_1)$ and $U_2 = X_{2train(2)}(C \odot A_2)$

$V_1 = C^\top C \circ A_1^\top A_1$ and $V_2 = C^\top C \circ A_2^\top A_2$

2: Update/test stage: when new data arrive as new slices appended to time mode

Repeat

Update P_1 , Q_1 , U_1 , V_1 , P_2 , Q_2 , U_2 and V_2 using ‘old’ information

C is updated using Eq. (2) (fixing A_1 , B_1 , A_2 and B_2)

A_1 is updated using Eq. (3) (fixing B_1 and C)

B_1 is updated using Eq. (4) (fixing A_1 and C)

A_2 is updated using Eq. (5) (fixing B_2 and C)

B_2 is updated using Eq. (6) (fixing A_2 and C)

Until convergence

Complexity analysis As in [15], the time complexity of onlineCP for a n -way tensor $\mathcal{X}_1(I_1 \times \dots \times I_{n-1} \times K)$ and $\mathcal{X}_2(J_1 \times \dots \times J_{n-1} \times K)$ are $O(nR \prod_{i=1}^{n-1} I_i)$ and $O(nR \prod_{i=1}^{n-1} J_i)$ where I_i and J_i are sizes of non-temporal modes. So it takes $O(nR(\prod_{i=1}^{n-1} I_i + \prod_{i=1}^{n-1} J_i)t)$ for onlineCP-Fusion, where t is the number of iterations for an ALS update. Since in a tensor growing in time mode normally $I_i, J_i \ll K$ (the size in time mode) and n, R, t are very small, the complexity for an update from onlineCP-Fusion can be consider as constant.

3.3 Online Damage Identification

Building a benchmark model In practice, events corresponding to damaged states of structures are often unavailable for a supervised learning approach. In this work, we use OCSVM [11] with Gaussian kernel as an anomaly detection method. The technique in [4] is adopted to tune σ in the kernel.

In this step, C , A_1 , B_1 , A_2 and B_2 are obtained using Algorithm 1 on training tensors. Each row of the component matrix C represents an event in time mode.

We build a benchmark model using healthy training events which are represented by rows of C by means of OCSVM.

Damage identification Due to an arrival of a new time event, an additional row C_{new} is added to the component matrix C , and matrices A_1, B_1, A_2 and B_2 are incrementally updated as described in Algorithm 2. The new row C_{new} will be checked if it agrees with the benchmark model built at the training stage, answering the condition of the structure. In the case of OCSVM, a negative decision value indicates that the new event is likely a damaged event.

Location matrices B_1 and B_2 , where each row captures meaningful information for each sensor location, could be used for damage localization. By analyzing these matrices when each new data instance arrives, it is able to find anomalies, which correspond to damaged locations. In this work we only use B_2 (which represents sensor locations for one type of sensors) for damage localization due to the specific sensor instrumentation for the bridge tested in the experiment. An average distance from a sensing location (a row in B_2) to k nearest neighboring locations ($k = 2$) is regarded as an anomaly score to localize damage.

To estimate the extent of damage, we analyze the decision values by the OCSVM model. The rationale is that a structure with a more severe damage (e.g. a longer crack) will behave more differently from a normal situation. Different ranges of the decision values may imply different severity levels of damage.

4 Experimental Results

Experiments were conducted using both synthetic data and data collected from a real bridge in operation. For all experiments, we compare our onlineCP-Fusion (for fusion \mathcal{X}_1 and \mathcal{X}_2) with onlineCP-ALS [5] (as baselines which learn from \mathcal{X}_1 and \mathcal{X}_2 separately). Another baseline called naive-Fusion to fuse data by concatenating the features from all strain gauge and cable sensors in each time instance as a feature vector, followed by random projection (with dimension size $k = 50$) to reduce the feature dimension. Then self-tuning OCSVM (with $\nu = 5\%$) was used for anomaly detection on feature spaces obtained by all these methods. About 80% of healthy data used for training and the rest for testing. All reported results were averaged over 10 trials.

4.1 Synthetic Data

We generated 5 matrices randomly from standard normal distribution with different means and variances: $A_1(50 \times R), B_1(20 \times R), A_2(25 \times R), B_2(10 \times R)$ and $C(500 \times R)$. They were considered as latent factors decomposed from two tensors as in Fig. 1. Then 5% of data instances in C were replaced with data generated randomly from a uniform distribution (as damage/anomaly). All the matrices were then normalized to have unity norm for all their columns. Then a tensor \mathcal{X}_1 was constructed from A_1, B_1, C and a tensor \mathcal{X}_2 was constructed from A_2, B_2, C using CP. Next Gaussian noise was randomly added to 35% of data along third dimension of tensor \mathcal{X}_1 and to 50% of data along third dimension of

tensor \mathcal{X}_2 . The purpose is to check if a data fusion of two tensors can eliminate the adverse effects of noise from individual tensors.

$R = 5$ was selected for tensor construction and decomposition in the experiment. $F1$ -score was adopted to measure the accuracy of OCSVM for anomaly detection in the learned time matrix C . Factor similarity was used to estimate the similarity between decomposed latent factors and the real ones we generated (this is not applicable to naive-Fusion). The similarity score for each column of each component matrix is computed as $\frac{|\hat{a}_r^\top a_r|}{\|\hat{a}_r\| \|a_r\|}$ after finding the best matching permutation of the columns (\hat{a}_r and a_r are a real latent factor column and its decomposed one respectively). The product of all these scores for all columns of all component matrices represents the final similarity score.

Factor similarities and $F1$ -scores based on different methods are shown in Table 1, indicating that fusing data from two noisy tensors by means of onlineCP-Fusion overall yields better result than all other baselines.

Table 1: Factor similarities and $F1$ -scores based on different methods.

	Factor Similarity	$F1$ -Score
onlineCP-ALS for \mathcal{X}_1	0.86	0.85
onlineCP-ALS for \mathcal{X}_2	0.68	0.72
naive-Fusion	N/A	0.54
onlineCP-Fusion for \mathcal{X}_1 and \mathcal{X}_2	0.86	0.87

4.2 Real Bridge Data

A cable-stayed bridge (Fig. 2a) in Sydney, Australia was considered as a case study in this work. It has a steel tower with a composite steel-concrete deck. The deck is supported by four I-beam steel girders, which are internally attached by a set of equally-spaced cross girders (CG). A dense array of sensing system has been deployed on the bridge since 2016. All the sensors are timely synchronized and are continuously measuring the dynamic response of the bridge under normal operation at 600 Hz. Each cable has been instrumented with a full axial Wheatstone bridge to measure the dynamic strain response of the cables (namely SA1 to SA8 which are, respectively, installed on cables 1 to 8 as in Fig. 2b. They are also aligned with CGs 4-7). After a test, it was realised that sensor SA4 was not operational and thus it was eliminated from the analysis. In our work, we used two sets of sensors for data fusion experiments: a set of 6 strain gauge sensors mounted to the bridge deck and a set of aforementioned 7 cable sensors.

We emulated damage by locating stationary mass on the bridge at different locations as real damage was not available. This additional mass can be treated as a damaged event for evaluation purpose since the increment of mass results in a similar effect on the bridge dynamic properties as the decrement of stiffness caused by an actual damage. Two extensive field experiments were conducted on this bridge which are referred to as Bus Damage Test and Car Damage Test. The Bus Damage Test was conducted in a way that a 13t three-axle bus was placed at a stationary location at mid-span of the bridge. Due to the distributed

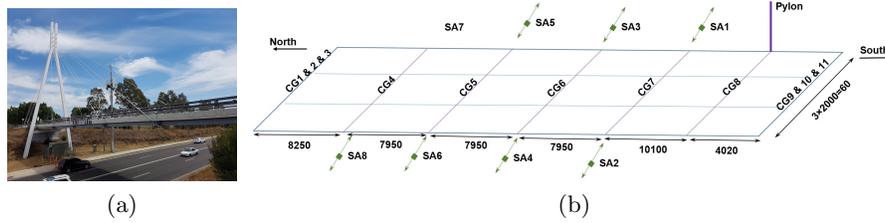


Fig. 2: The cable-stayed bridge and the longitudinal and lateral girders under the deck.

effect of mass in this case, this dataset is not suitable for damage localization and it will be solely adopted for detection and assessment of damage. In the Car Damage Test, a 2.4t car was utilized. In each damage case (i.e. Car Damage 1 to Car Damage 4), the vehicle was placed in each cross girder (CGs 4-7 respectively, where the 4 pairs of cable sensors are placed) and the dynamic response of the bridge was recorded under ambient excitation. The Car Damage Test could be used to verify whether the proposed method is capable of locating damage.

Feature extraction The change in the cable-forces was adopted for damage identification as any damage in the structure changes the distribution of the cable-forces. Ambient strain responses from each cable sensor in both healthy and damaged cases were split into events of 2 seconds for analysis. Then the following steps were applied to extract features for our damage identification. First, the dynamic strain responses due to the live load effects from each cable (except SA4 due to the sensor issue) were normalized by subtracting the average strain of the healthy training data from the same cable. Then the absolute normalized strain was transformed into a unique direction by taking into account the orientation of each cable. This resulted in seven time series responses for seven cable sensors (i.e. SA1, SA2, SA3, SA5, SA6, SA7 and SA8). Since each strain response had 1200 samples (2 seconds at 600 Hz) and there were 7 locations of cable sensors, the data formed a cable tensor of $1200 \text{ features} \times 7 \text{ locations} \times 187 \text{ events}$ where 187 indicates the total number of healthy events and damaged events (including 4 Car Damage cases and 1 Bus Damage case). For 6 strain gauges on the bridge deck, the feature extraction is the same except their orientations were not used as in the cable sensors. Similarly, we have a strain gauge tensor of $1200 \text{ features} \times 6 \text{ locations} \times 187 \text{ events}$.

Damage detection and severity assessment Similar to experiments for the synthetic data, Gaussian noise was randomly added to 35% of data along third dimension of strain gauge tensor and to 50% of data along third dimension of cable tensor. Anomaly detection using self-tuning OCSVM was applied on the feature spaces learned by onlineCP-Fusion and all baselines. The number of latent factors R was selected as 2 using core consistency diagnostic technique (CORCONDIA) [2]. $F1$ -scores of 0.99, 1 and 0.87 were achieved by onlineCP-ALS for the strain gauge tensor and cable tensor, and naiveFusion, respectively. Data fusion from two tensors using the proposed onlineCP-Fusion led to an $F1$ -

score of 1, which improved the overall performance from the approach without tensor data fusion. The results in Fig. 3 (obtained C matrices, $R = 2$) indicate the proposed method (Fig. 3c) is not only more capable to distinguish between healthy and damaged data for damage detection, but also between Bus Damage and Car Damage cases for severity assessment (i.e. Bus Damage samples were further away from the healthy data compared to those of Car Damage).

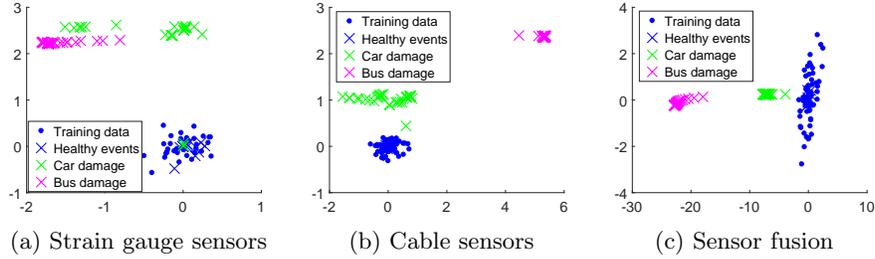


Fig. 3: Damage detection and severity assessment.

Damage localization Using the techniques in Section 3.2, component matrix B_{2new} was incrementally updated for every new test event. Sensor scores at each location were computed as in Section 3.3. We used the change of this score as an indicator to localize damage. Specifically the sensor which has the most change of this score is likely located near to the damage location. The results on damage localization for 4 Car Damage cases are shown in Fig. 4. As noticed, each damage case corresponds to a pair of figures for comparison purpose. Taking Figs. 4a and 4b as an example, Fig. 4a measures score changes (the difference or gap between the average sensor score of the healthy and damaged test data) of cable sensors when we only considered cable tensor. Fig. 4b measures score changes of cable sensors when a data fusion for cable and strain gauge sensors was used. It is shown in Figs. 4a and 4b that the damage location was close to SA7/SA8 (in CG4), which is true as the car was in CG4 in Car Damage 1. We achieved similar results for Car Damage 3 (CG6, SA3) and 4 (CG7, SA1/SA2), except in Car Damage 2 where SA3 was the sensor with the most change. Even though the use of data fusion achieved similar results with the use of only cable sensors, the score changes of localized sensors using onlineCP-Fusion are more pronounced compared with those obtained from only cable sensors.

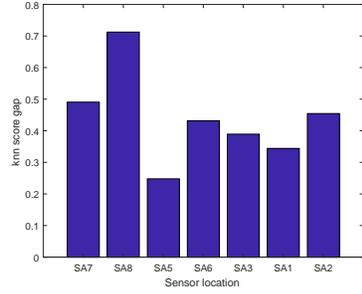
5 Conclusion

This paper has proposed a novel method for online data fusion from heterogeneous sources based on incrementally-coupled tensor learning, where component matrices from multiple tensors are updated collectively and incrementally when new data arrive. The method has been applied to a developed case study in SHM for sensor data fusion and online damage identification. Experiments using synthetic data and data obtained from a real-life bridge have verified the effectiveness of the proposed method for data fusion. The results show that the

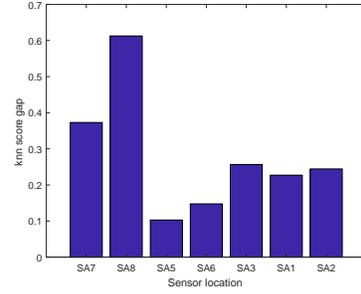
proposed data fusion approach is more robust to noise than the approach using individual data sources and has a potential for data fusion for damage identification in SHM.

References

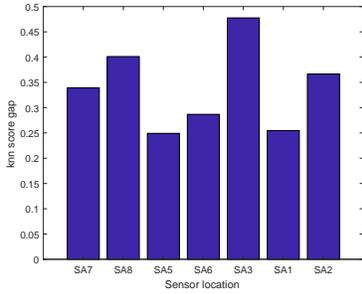
1. Acar, E., Kolda, T.G., Dunlavy, D.M.: All-at-once optimization for coupled matrix and tensor factorizations. In: *MLG'11: Proceedings of Mining and Learning with Graphs* (August 2011)
2. Bro, R., Kiers, H.A.L.: A new efficient method for determining the number of components in parafac models. *Journal of Chemometrics* **17**(5), 274–286 (2003)
3. Farrar, C.R., Worden, K.: An introduction to structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **365**(1851), 303–315 (2007)
4. Khazai, S., Homayouni, S., Safari, A., Mojaradi, B.: Anomaly detection in hyperspectral images based on an adaptive support vector method. *IEEE Geoscience and Remote Sensing Letters* **8**(4), 646–650 (2011)
5. Khoa, N.L.D., Anaissi, A., Wang, Y.: Smart infrastructure maintenance using incremental tensor analysis: Extended abstract. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. pp. 959–967. *CIKM '17*, ACM, New York, NY, USA (2017)
6. Khoa, N.L.D., Zhang, B., Wang, Y., Liu, W., Chen, F., Mustapha, S., Runcie, P.: *PAKDD 2015, Vietnam, May 19-22, 2015, Proceedings, Part I*, chap. On Damage Identification in Civil Structures Using Tensor Analysis, pp. 459–471. Springer International Publishing, Cham (2015)
7. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. *SIAM Review* **51**(3), 455–500 (September 2009)
8. Liu, W., Chan, J., Bailey, J., Leckie, C., Kotagiri, R.: Utilizing common substructures to speedup tensor factorization for mining dynamic graphs. In: *Proceedings of the 21st ACM international conference on Information and knowledge management*. pp. 435–444. *CIKM '12*, ACM, New York, NY, USA (2012)
9. Nion, D., Sidiropoulos, N.D.: Adaptive algorithms to track the parafac decomposition of a third-order tensor. *Trans. Sig. Proc.* **57**(6), 2299–2310 (Jun 2009)
10. Prada, M.A., Toivola, J., Kullaa, J., Hollmn, J.: Three-way analysis of structural health monitoring data. *Neurocomputing* **80**(0), 119 – 128 (2012), special Issue on *Machine Learning for Signal Processing 2010*
11. Schölkopf, B., Williamson, R.C., Smola, A.J., Shawe-Taylor, J., Platt, J.C.: Support vector method for novelty detection. In: *NIPS*. pp. 582–588 (1999)
12. Sorber, L., Barel, M.V., Lathauwer, L.D.: Structured data fusion. *IEEE Journal of Selected Topics in Signal Processing* **9**(4), 586–600 (June 2015)
13. Sun, J., Tao, D., Papadimitriou, S., Yu, P.S., Faloutsos, C.: Incremental tensor analysis: Theory and applications. *ACM Trans. Knowl. Discov. Data* **2**(3), 11:1–11:37 (Oct 2008)
14. Wang, Y., Wu, L., Lin, X., Gao, J.: Multiview spectral clustering via structured low-rank matrix factorization. *IEEE Transactions on Neural Networks and Learning Systems* **29**(10), 4833–4843 (Oct 2018)
15. Zhou, S., Vinh, N.X., Bailey, J., Jia, Y., Davidson, I.: Accelerating online cp decompositions for higher order tensors. In: *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 1375–1384. *KDD '16*, ACM, New York, NY, USA (2016)



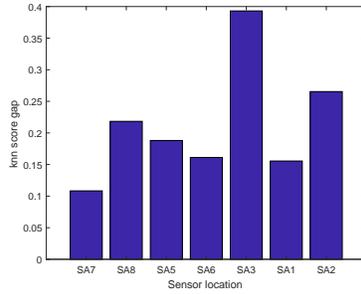
(a) Car Damage 1: sensor score



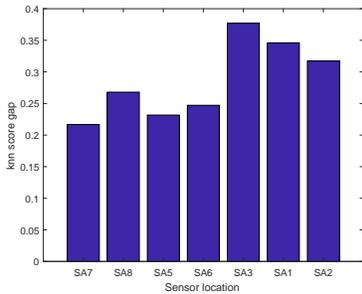
(b) Car Damage 1: sensor score (fusion)



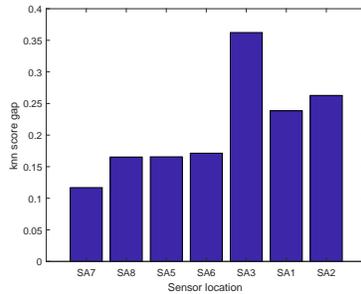
(c) Car Damage 2: sensor score



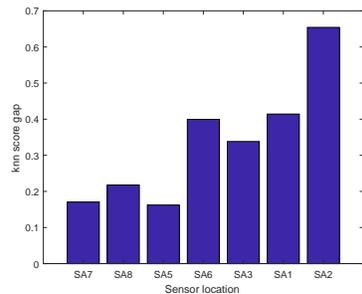
(d) Car Damage 2: sensor score (fusion)



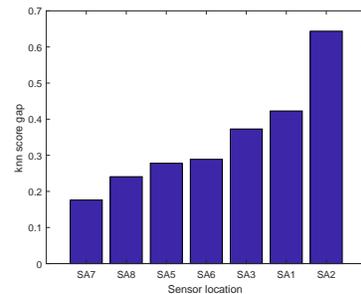
(e) Car Damage 3: sensor score



(f) Car Damage 3: sensor score (fusion)



(g) Car Damage 4: sensor score



(h) Car Damage 4: sensor score (fusion)

Fig. 4: Results on damage localization.