



# Personalised federated learning framework for damage detection in structural health monitoring

Ali Anaissi<sup>1</sup> · Basem Suleiman<sup>1</sup> · Widad Alyassine<sup>1</sup>

Received: 17 January 2022 / Revised: 29 June 2022 / Accepted: 29 July 2022 / Published online: 13 August 2022  
© The Author(s) 2022

## Abstract

Data-driven machine learning models, compared to numerical models, demonstrated promising improvements in detecting damage in structural health monitoring (SHM) applications. In such approaches, sensors' data are used to train a model either in a centralized model (server) or locally inside each sensor unit node (client). The centralized learning model often leads to computing and privacy issues such as wireless transmission costs and data-sensitive vulnerability, especially in real-time settings. The decentralized model also poses different challenges such as feature correlations and relationships loss in decentralized learning settings. To handle the shortcomings of both models, we propose a new Personalized federated learning (FL) model augmented with tensor data fusion to learn and detect damage in SHM. Our approach employs FL which enables the central machine learning model to gain experience from diverse datasets located at different sensor locations. Furthermore, our proposed model addresses the problems associated non-i.i.d. data by employing the Moreau envelopes as a regularized loss function in the learning process of client's models. Our methods help in decoupling the client models from the central one which improves personalized in FL. Our experimental evaluation on real structural datasets demonstrates promising damage detection accuracy without the need to transmit the actual data to the centralized learning model. The results also show that the data correlations and relationships from all participating sensors are preserved.

**Keywords** Personalised federated learning · Structural health monitoring · Damage detection · Tensor · Data fusion

## 1 Introduction

There has been an exponential growth of data generated from sensors and computing devices connected to the Internet known as the Internet of Things (IoT). The IoT has penetrated pervasively into most aspects of human life everywhere such as civil infrastructures, health-care centres, transportation, etc., wherein smart services are utilized to continuously monitor every activity at all times and in real-time. In the field of civil infrastructures, IoT has provided flexibility and added value to Structural Health Monitoring (SHM) applications to generate actionable insights.

The applications of SHM aim to provide an automated process for damage detection in complex structures such as bridges using sensing data collected through multiple networked sensors attached to it [1]. This data is then utilized

to gain insight into the health of a structure and make timely and economic decisions about its maintenance. One of the traditional approaches for structural damage detection is known as model-driven which constructs a numerical model for the structure based on finite element analysis [2]. However, a numerical model can be impractical as it cannot always sensibly capture the behavior of the real structure. On the other hand, a modern technique known as the data-driven approach has been successfully adopted in SHM and its applications. The data-driven approach uses machine learning algorithms to construct a model from measured data and then makes predictions for new measured responses to detect structural damage. This approach has brought a concrete aspect to IoT in SHM and enabled IoT smart applications. Bridges are critical to our society as they connect various separated locations to allow the flows of people and goods within cities. Bridges are influenced by several factors such as environmental conditions (wind, ambient temperature change, etc) and various loads (constant and temporary loads), which makes them prone to damage and potential failures. Any problem with such a

✉ Ali Anaissi  
ali.anaissi@sydney.edu.au

<sup>1</sup> School of Computer Science, The University of Sydney, Street, Camperdown, NSW 2006, Australia

structure from small damage to catastrophic failures would result in significant economic and life losses. Most of the structural maintenance approaches are time-based, visual inspections are carried out at a predefined regular schedule which can be either too early or too late to detect damage. However, SHM is a condition-based approach; it uses data sensed continuously to provide real-time monitoring so that necessary maintenance actions can be taken once damage, or abnormal change in the structure behaviour, occur.

There are two data-driven mechanisms that have been commonly known for damage detection in complex structures. The first mechanism relies on a centralized machine learning model which requires transmitting the generated sensing data from the deployed sensors to a central processing unit to assess the structural condition. The data then is either aggregated or fused in one data structure to capture the correlation and relations between the measured variables from all sensors, and to learn the different aspects of the data (temporal, spatial, and feature) at the same time [3, 4]. This mechanism allows capturing the underlying structural aspects using multi-way sensing data which has made it successful as it achieved good accuracy in terms of damage detection. However, the use of such centralized model has its drawbacks, especially for SHM. It is not practical in the context of real-time monitoring and resource-constrained environment since sensors collect vibration measurements using accelerometers at high frequency during a time period, thus, contain a sequence of thousands of data points to be transmitted very frequently to the central model. Moreover, wireless transmission costs more energy than local processing, thus poses several challenges for battery-powered wireless nodes.

The alternative mechanism to handle the centralized model restrictions is to perform the learning from the sensed data in a distributed environment similar to the promising edge or distributed computing paradigms [5]. The distributed learning approach has a number of benefits including reducing the intensive data transmission over the network, conserving the energy of sensor nodes and, reducing the workload overhead on the central server on which data processing (learning) occurs.

One challenge that emerges with the distributed learning approach is that the measured data in SHM are often multi-way and highly redundant and correlated, i.e., many sensors at different locations simultaneously collect data over time. Therefore, a single sensor node analysis cannot capture all of these correlations and relationships together in a distributed learning model. In contrast, the centralized learning analysis allows learning from such data in multiple modes at the same time. The work proposed by Mehri et al. [5] has thoroughly investigated and compared the performance of the centralized prediction models versus a single sensor node model (decentralized) prediction. Their experimental

results show that the centralized model was able to successfully detect the presence of very small damage in a structure and to monitor its progress over time. On the other hand, the distributed learning models which were constructed for each sensor not only reduced the sensitivity of the models but also failed to monitor the progress of damage in the monitored structure.

Therefore, developing an effective and efficient damage detection model for SHM applications requires information derived from many spatially-distributed locations throughout large infrastructure covering various points in the monitored structure. However, consolidating this data in a centralized learning model can often be computationally complex and costly. This motivates for developing a more advanced model that utilizes the centralized learning model but without the need to transmit the frequently measured data to a single processing unit. In this study, we propose a novel approach to overcome the above-discussed challenges of centralized and decentralized learning mechanisms in SHM. Our approach is developed based on federated learning (FL) augmented with Tensor Data Fusion for damage detection in complex structures such as bridges.

Our approach is derived from an auto-encoder neural network (ANN) as a damage detection model and employs tensor data analysis to perform data fusion for a wired connected sensor in SHM applications to reduce the communications in the FL network. The emerging Federated Learning (FL) concept was initially proposed by Google for improving security and preventing data leakages in distributed environments [6]. FL allows the central machine learning model to build its learning from a broad range of data sets located at different locations. It aims to train a shared centralized machine learning model using datasets stored and distributed across multiple devices or sensors. In the context of our study, we devise an FL-approach that enables multiple IoT devices (sensors) to collaborate on the development of a central learning model, but without needing to directly share or pool all data measured from several sensors with each other. Our approach can work efficiently and effectively by sharing the model coefficients of each client model only rather than the whole data collected by all participating sensors at each period of time. The effectiveness of the model learning continues to improve over the course of several training iterations during which the shared models get exposed to a significantly wider range of data than what any single sensor node possesses in-house. Our proposed FL-based approach decentralizes machine learning by removing the need to pool data into a single location and through training the centralized model in multiple iterations at different locations (where sensors are deployed).

Although FL results in reducing data transmission and improving data security, it raises significant challenges in how to deal with non-IID (Independent and Identically

Distributed) data distribution and statistical diversity. In fact, Stochastic Gradient Descent (SGD), which is widely used in training models, may lean towards one distribution than the other leading to a non-generalized model. The work proposed in [7, 8] has thoroughly investigated the performance of the global model on clients' local data by showing how the accuracy decreases when the data diversity increases. To address the non-IID challenge in FL, we employed Moreau envelopes [9] as a regularized loss function in the learning process of the clients' models. The rationale idea behinds Moreau envelope is to leverage the central model in optimizing the clients' models not only using Federated Averaging (FedAvg), but personalize it w.r.t its local data distribution. The contribution of the work in this study is twofold.

- This paper addresses the problem of non-IID distribution of data in FL for better generalization of the clients' models.
- We extensively evaluated the performance of personalized FL augmented with tensor data fusion method to demonstrate the effectiveness of the proposed method in constructing generalized clients' models

The remaining of this paper is organized as follows. Section 2 reviews the related work and Sect. 3.1 provides a short overview of the original Auto-encoder Neural Network method. Our novel online personalized FL with tensor algorithm is described in Sect. 4. Section 6 then presents experimental results. Finally, we summarize our contributions and conclude this paper in Sect. 7.

## 2 Related work

Federated learning gained a lot of interest in recent years and as a result, it has attracted AI researchers as a new and promising machine learning approach [10, 11]. This federated learning approach attracts several well-suited practical problems and application areas due to its intrinsic settings where data needs to be decentralized and privacy to be preserved. However, only a few applications, that have been reported in the literature, utilized the federated learning approach to construct a global model. For instance, Bonawitz et al. [12] adopted the federated learning settings to develop a system that solves the problem of next-word-prediction in mobile devices. On the other hand, several other studies focused on addressing the training challenges of a central model to support all local data training especially when the distribution of data across clients is highly non-IID (independent and identically distributed). For example, Hanzely et al. [13] claim that a central model is still too far mature from the typical usage of one client. Similarly, Kairouz et al. [10] discuss the

broad challenges and open problems in the field of applying federated learning to different problems.

McMahan et al. [14] propose the first federated learning-based algorithm named FedAvg. It uses the local Stochastic Gradient Descent (GSD) updates to build a global model by taking average model coefficients from a subset of clients with non-IID data. Subsequently, Guha et al. [15] propose a new method to learn a global model in one single round of communication between the client and the central model. Other studies such as [16, 17] address the limitations of communication among distributed models in federated learning settings by performing periodic averaging, partial device participation, and quantized message-passing. Meanwhile, other research studies such as [18–20] suggest multiple local optimization rounds before reporting the learning from the distributed models to the central server.

Several other methods have been proposed to achieve personalization in federated learning. Recently, Smith et al. [21] propose a new algorithm named MOCHA based multi-task learning (MTL) framework to address the non-IID challenge in federated learning. However, the author in [13] proposes an L2GD algorithm that combines the optimization of the local and global models. Similarly, Dng et al. [8] develop an adaptive personalized federated learning (APFL) algorithm which mixes the user's local model with the global model. Another rationalization method called FedPer proposed by [22] which divides the layers in a neural network into a base and personalized, while both layers are trained by clients but only the base layers are trained by the server.

In this paper, we address the communication problem that arises in a federated learning environment where several distributed models (clients) communicate with the central model to report its learning to the central model (the sever). Our approach is centered around employing a tensor as a data fusion method for wired connected clients in SHM applications. Tensor analysis has been successfully applied in many application domains such as civil engineer, social network analysis, and computer vision [23–25], and produced promising results. In our proposed approach, we also address the non-IID challenge in federated learning settings by developing a personalized model which is optimized for each distributed data model. To the best of our knowledge, our federated-based approach and its application the structural health monitoring (SHM) domain create new research contributions.

## 3 Background

### 3.1 Autoencoder neural network

Autoencoder neural network (ANN) is an unsupervised learning process which has the ability to learn from one

class data. It is an extension of the traditional neural network which basically designed for supervised learning when the class labels are given with the training examples. The rational idea of an autoencoder is to force the network to learn a lower dimensional space for the input features, and then try to reconstruct the original feature space. In other words it sits the target values to be approximately equal to its original inputs. In this sense, the main objective of autoencoders is to learn reproducing input vectors  $\{x^1, x^2, x^3, \dots, x^m\}$  as outputs  $\{\hat{x}^1, \hat{x}^2, \hat{x}^3, \dots, \hat{x}^m\}$ . The architecture of ANN is made up of  $L$  layers ( $L = 3$  for simplification) denoted by input, hidden and output layers. Each layer consists from a set of nodes. Layer  $l_1$  is the input layer represents features which encoded into the hidden layer  $l_2$ , and then decoded into the output layer  $l_3$ . The learning process of ANN starts by successively computing the output of each node in the network. For a node  $i$  in layer  $l$  it calculates the output value  $z_i^{(l)}$  by computing the total weighted sum of the input values which also includes the bias term using the following equation:

$$z_i^{(l)} = \sum_{j=1}^n \theta_{ij}^{(l-1)} a_j^{(l-1)} \quad (1)$$

where  $\theta$  is the coefficient weight written as  $\theta_{ij}$  when associated with the connection between node  $j$  in layer  $l - 1$ , and node  $i$  in layer  $l$ , and  $a_j^{(l-1)}$  is the output value of node  $j$  in

layer  $l - 1$ . The resultant output denoted by  $a_i^{(l)}$  is then processed through an activation function as follows:

$$a_i^{(l)} = \mathcal{F}(z_i^{(l)}) \quad (2)$$

where  $\mathcal{F}()$  is the activation functions. The most common activation functions in the hidden layers are sigmoid and hyperbolic tangent. However, in autoencoder settings a linear function is used in the output layer since we don't scale the output of the network to a specific interval.

The autoencoder uses back propagation algorithm to learn the parameters  $\theta$  which approximate  $\hat{x} \approx x$ . In each iteration of the training process, we calculate the cost error  $\mathcal{L}(\theta_{ij}^{(l)}; x_i)$  using Eq. (3) and then propagate it backward to the network layer.

$$\mathcal{L}(\theta_{ij}^{(l)}; x_i) = \frac{1}{M} \sum_{m=1}^M \left( \frac{1}{2} \|x^m - \hat{x}^m\|^2 \right) \quad (3)$$

In this setting, we perform a stochastic gradient descent step to update the learning parameters  $\theta_{ij}^{(l)}$ . This is done by computing the partial derivative of the cost function  $\mathcal{L}(\theta_{ij}^{(l)}; x_i)$  (defined in Eq. 3) as follows:

$$\theta_{ij}^{(l)} := \theta_{ij}^{(l)} - \eta \frac{\partial}{\partial \theta_{ij}^{(l)}} \mathcal{L}(\theta_{ij}^{(l)}; x_i) \quad (4)$$

The complete steps are summarized in Algorithm 1.

---

#### Algorithm 1 Autoencoder training algorithm

---

**Require:** A set of  $n$  positive samples  $X = \{x_{(i)}\}_{i=1}^n$

**Ensure:** Initialize  $\theta_{ij}^{(l)}$  to a small random value  $\mathcal{N}(0, \epsilon^2)$

**while** not converged

• **for**  $i \leftarrow 1$  to  $n$

• Perform a feedforward pass to compute all nodes activations using Equations 1 and 2.

• Compute the cost function  $\mathcal{L}(\theta_{ij}^l; x_i)$  using Equation 3

• Update  $\theta_{ij}^l$  using Equation 4

**return**  $\theta_{ij}^l$

---

Once we successfully trained the autoencoder, the network will be able to reconstruct an new incoming positive data, while it fails with anomalous data. This will be judged based on the reconstruction error (RE) which is measured by applying the Euclidean norm to the difference between the input and output nodes as shown in Eq. (5).

$$RE(x) = \|x_i - \hat{x}_i\|^2 \quad (5)$$

The measured value of RE is used as anomaly score for a given new sample. Intuitively, examples from the similar distribution to the training data should have low reconstruction error, whereas anomalies should have high anomaly score.

## 4 Personalised federated learning framework for SHM

Our approach comprises of a number components which are detailed in this section. First, we described how the sensor data is monitored and modelled in distributed environment (in SHM such as bridges). Second, we define how the personalized federated learning is formulated as distributed models (clients and central server) and the communication among them. In our approach, we also propose aggregating the data points measured from multiple sensors into one client node. Third, we present our tensor data fusion method to perform this step. Fourth, describe our formal model of updating the learning experienced by the clients and server to detect structural damage.

### 4.1 Data modelling

Consider a set of sensor nodes  $S$  mounted on different locations of a bridge to measure and transmit sensing data related to a structural event. Each sensor node  $S$  can perform computation on sensed acceleration data to detect the damage in their vicinity. The data points collected concerning to the vibration responses are assumed to be a vector as  $X_i = [x_1, x_2, \dots, x_n]$ ; where  $n$  is the total number of data points sensed by a sensor node  $S$  over a time duration. Due to the lack of available data from the damaged state of the structure in most cases, the acceleration measurements we collect from a bridge is often corresponding to the healthy condition of the bridge. This data covers various environmental and ambient conditions as well as operational conditions, such as traffic loading. Therefore, in the training phase, we construct a one-class model by extracting the statistical features from raw acceleration data in the healthy condition of the bridge. The trained model will be used later to classify the raw acceleration measurement from unknown conditions of the bridge as either healthy or damaged. Each healthy training sample  $x_i \in \{X_i\}_{i=1}^n$  is an  $m$ -dimensional feature vector  $x^j = x^1, x^2, \dots, x^m$ , where  $j = 1, \dots, m$  are the statistical features extracted from sensed acceleration data in healthy condition of the bridge. The total number of features  $j$  depends on the sampling frequency and sampling window, and the total number of data points ( $n$ ) in  $X_i$  depends on the number of events. In our approach, we use an auto-encoder neural network as an anomaly detection method to fit a one-class model using healthy data. However, we need now to formulate this model in a federated learning (FL) setting with the help of tensor data analysis as a data fusion method.

### 4.2 Problem formulation in personalized federated learning

In FL setting, a set of clients  $S$  are connected to a central server to solve the following problem:

$$\min_{w \in \mathbb{R}^d} f(w) := \frac{1}{S} \sum_{s=1}^S f_s(\theta), \quad (6)$$

where  $f_s$  is the expected loss over the data distribution corresponding to a sensor node client  $S$  which defined as follows:

$$f_s(\theta) := \mathbb{E}_{x_i} [\mathcal{L}_s(\theta; x_i)] \quad (7)$$

where  $\mathcal{L}_s(\theta; x_i)$  measures the error of model  $\theta$  given the input  $x_i$  defined in Eq. (3).

The stochastic gradient descent (SGD) method solves the above problem defined in Eq. (7) by repeatedly updates  $\theta$  to minimize  $\mathcal{L}_s(\theta; x_i)$ . It starts with some initial value of  $\theta^{(t)}$  and then repeatedly performs the update as follows :

$$\theta^{(t+1)} := \theta^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial \theta}(x_i^{(t)}, \theta^{(t)}). \quad (8)$$

In FL, each client performs a number of  $E$  epochs at each round to compute the gradient of the loss over its local data and send the model parameters  $\theta_s^{t+1}$  to the server. The central sever aggregates these gradients and applies the global model parameters update by taking the average of the resulting models parameters as follows:

$$w^{(t+1)} := \frac{1}{S} \sum_{s=1}^S \theta_s^{(t+1)}. \quad (9)$$

The server then sends  $w^{(t+1)}$  to all clients in which each one performs another round to update  $\theta^{(t+1)}$  but with setting  $\theta^{(t)} = w^{(t+1)}$  as defined in traditional FL-FedAvg. However, such simple averaging method may not be practical in real world applications where some clients may have very few local data points. Therefore, we propose an ensemble learning method to combine multiple learners by selecting a random subset of clients under the condition that clients will only share their local models if they reach a minimum accuracy performance on their local validation data.

We further investigate the problem of non-i.i.d in FL settings. In fact, our current approach may work well when clients (sensors) have similar i.i.d. data. However, it is unrealistic to assume that since data may come from different environments and contexts in FL settings thus have non-i.i.d. Therefore, we employ Moreau envelope (ME) [9] as



a regularized loss function which helps decouple personalized model optimization from the global model learning in a bi-level problem stylized for personalized FL. With this scheme, instead of updating the clients’ models  $\theta^{(t+1)}$  entirely based on the server model  $w^{(t+1)}$ , we force the clients to pursue their own models based on  $\theta^{(t+1)}$  but with adding an  $l_2$ -norm regularized loss function i.e ME based on  $w^{(t+1)}$  so that  $\theta_s$  will not deviate far from  $w$ . Geometrically, the global model can be considered as a “central point”, where all clients agree to meet, and personalized models are the points in different directions that clients follow according to their heterogeneous data distributions. In this setting, the client model update will be as follows:

$$\theta^{(t+1)} := \theta^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial \theta}(x_i^{(t)}, \theta^{(t)}) + \lambda \|\theta^{(t)} - w^{(t+1)}\|^2, \quad (10)$$

where  $\lambda$  is a regularization parameter controls the penalty of  $l_2$ -norm.

### 4.3 Tensor data fusion

Our proposed approach also incorporates a data fusion step that merges data from a set of connected sensor nodes  $\mathcal{S}$  into one client node. In fact, sensors’ measurements usually have a high redundancy and correlation, which two-way matrix analysis may fail to capture all of these correlations and relationships together. A naive approach would simply concatenate the features obtained from different connected clients. However, unfolding the multi-way data and analyzing them using two-way methods may result in information loss and misinterpretation since it breaks the modular structure inherent in the data. Therefore, we present a method for data fusion using a tensor data structure that arranges the data from a set of connected sensor nodes as one single client node  $\mathcal{T}$  we call it a tensor node. This tensor node  $\mathcal{T}$  has data in a form of a three-way tensor  $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$  where  $I$  represents the number of connected clients,  $J$  represents the number of features in each client, and  $K$  is the total number

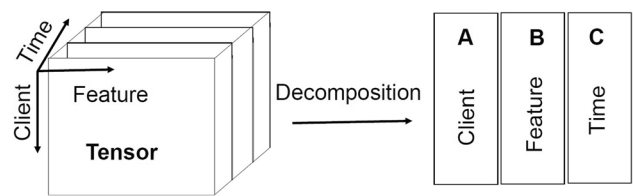


Fig. 1 Connected clients fused in a tensor

of data points sensed by a sensor node  $\mathcal{S}$ . The structure of this tensor is shown in Fig. 1

Once we arrange the data in a tensor form, we apply a tensor decomposition to extract latent information in each dimension of tensor  $\mathcal{X}$ . This work adopts the CP decomposition (CANDECOMP/PARAFAC decomposition) method for tensor decomposition due to its ease of interpretation compared with the Tucker method [26]. CP decomposes  $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$  into three matrices  $A \in \mathbb{R}^{I \times R}$ ,  $B \in \mathbb{R}^{J \times R}$  and  $C \in \mathbb{R}^{K \times R}$  where  $R$  is the latent factors. Each matrix represents latent information for each mode or dimension. It can be written as follows:

$$\mathcal{X}_{(ijk)} \approx \sum_{r=1}^R A_{ir} \circ B_{jr} \circ C_{kr} \quad (11)$$

where “ $\circ$ ” is a vector outer product.

We formulate the problem as follows:

$$\min_{A,B,C} \|\mathcal{X} - \sum_{r=1}^R \lambda_r A_r \circ B_r \circ C_r\|_f^2, \quad (12)$$

where  $\|\mathcal{X}\|_f^2$  is the norm value which is the sum squares of all elements of  $\mathcal{X}$ , and the subscript  $f$  denotes the Frobenius norm.  $A_r, B_r$  and  $C_r$  are  $r$ th columns of component matrices  $A \in \mathbb{R}^{I \times R}$ ,  $B \in \mathbb{R}^{J \times R}$  and  $C \in \mathbb{R}^{K \times R}$ .

We applied the alternating least square (ALS) technique to solve the CP decomposition problem. It iteratively solves each factor matrix by fixing the other two matrices using a least-square technique until it meets a convergence criterion [27]. The ALS technique is described in Algorithm 2.

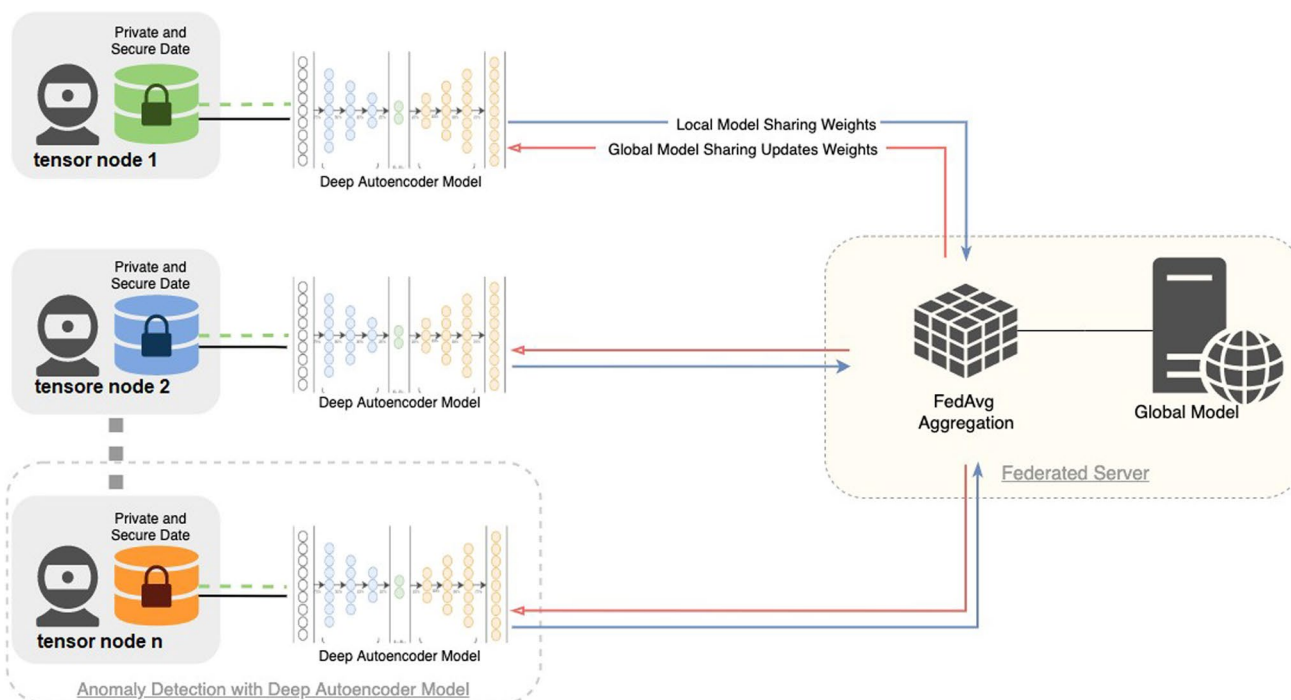
**Algorithm 2** CP Decomposition Using Alternating Least Squares

**Require:** : Tensor  $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ , latent factors  $R$   
**Ensure:** : Initialize  $A, B, C$   
**repeat**  
 $A \leftarrow \arg \min_A \frac{1}{2} \|X_{(1)} - A(C \odot B)^T\|^2$   
 $B \leftarrow \arg \min_B \frac{1}{2} \|X_{(2)} - B(C \odot A)^T\|^2$   
 $C \leftarrow \arg \min_C \frac{1}{2} \|X_{(3)} - C(B \odot A)^T\|^2 \quad \rightarrow \odot \text{ is the Khatri-Rao}$   
*product and  $X_{(i)}$  is an unfolding matrix of  $\mathcal{X}$  in mode  $i$*   
**until** convergence criterion is met **return** Matrices  $A \in \mathbb{R}^{I \times R}$ ,  $B \in \mathbb{R}^{J \times R}$ ,  
 and  $C \in \mathbb{R}^{K \times R}$

Once the convergence criteria are met, the ALS algorithm returns the three matrices  $A, B,$  and  $C$ . As mentioned before, the matrix  $C \in \mathbb{R}^{K \times R}$ , which is associated with the time mode, will be used later for constructing the central model. This matrix has  $K$  rows, each of which represents a data instance aggregated from all the clients given in a tensor node  $\mathcal{T}$  at a specific time.

**4.4 The client–server learning phase**

Based on the FL problem formulation and tensor data fusion described above, we present our structural damage detection approach. Our method uses the FL approach which is augmented with a tensor data fusion method, and an auto-encoder neural network model for structural damage detection. Each tensor node  $\mathcal{T}$  on the client-side will



**Fig. 2** The architecture of the federated learning network with deep auto-encoder model

initially fuse the sensors data in a tensor  $\mathcal{X}$  and apply CP algorithm using ALS to decompose  $\mathcal{X}$  into three matrices  $A$ ,  $B$ , and  $C$ . The matrix  $C$ , which represents the time mode, will be then used in the learning process. Our auto-encoder neural network uses the stochastic gradient

descent algorithm to learn reconstructions  $\hat{C}$  that is close to its original input  $C$ . At each round, each client  $\mathcal{T}$  performs a number of  $E$  epochs to update the model parameters and report them to the central server. Algorithm 3 explains the learning phase at given a tensor node  $\mathcal{T}$ .

---

**Algorithm 3** Client Side Learning
 

---

**Require:**  $I, E, \eta, \lambda, R$  and  $X_{i=1}^I \in \mathbb{R}^{K \times J}$

**Ensure:** Initialize  $A, B, C$  and  $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$

**Ensure:** Get  $w^{(t+1)}$  from the server and initialize  $\theta^{(t)}$

**for**  $i \leftarrow 1$  to  $I$

- Append  $X_i \in \mathbb{R}^{K \times J}$  to  $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$

Compute  $A, B$ , and  $C$  using Algorithm 2

**for**  $i \leftarrow 1$  to  $E$

- Compute  $\hat{C}$  using Equations 1 and 2.
- $\mathcal{L}(\theta_{ij}^{(l)}; x) = \frac{1}{v} \sum_{i=1}^v \left( \frac{1}{2} \|x^i - \hat{x}^i\|^2 \right)$
- $\theta^{(t+1)} := \theta^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial \theta}(x_i^{(t)}, \theta^{(t)}) + \lambda \|\theta^{(t)} - w^{(t+1)}\|^2$

**return**  $\theta^{(t+1)}$

---

**Fig. 3** Side view of the cable-stayed bridge from our first case study, Western Sydney, Australia (source: Google Earth)





Each client with a tensor node  $\mathcal{T}$  will report the model parameters  $\theta^{(t+1)}$  back to the central server. Once the updates are received, the central server will aggregate them using Eq. (9) and send them back to all client tensor nodes  $\mathcal{T}$ . Figure 2 gives an overview architecture of our federated learning network and Algorithm 4 explains the learning phase at the central server.

**Table 1** Number of samples in each joint of the arch bridge dataset

Dataset	Number of samples	Training	Test
Joint 1	6329	4430	1899
Joint 2	7237	5065	2172
Joint 3	4984	3488	1496
Joint 4	6886	0	6886
Joint 5	6715	4700	2015
Joint 6	4801	3360	1441

**Algorithm 4** Server side learning

**Require:**  $K$  number of client nodes  $\mathcal{T}$  **while** not converged

1. **for**  $s \leftarrow 1$  to  $S$  **in parallel**
  - (a) Read  $\theta_s^{(t+1)}$  which is calculated in Algorithm 3
2.  $w^{(t+1)} := \frac{1}{S} \sum_{s=1}^S \theta_s^{(t+1)}$

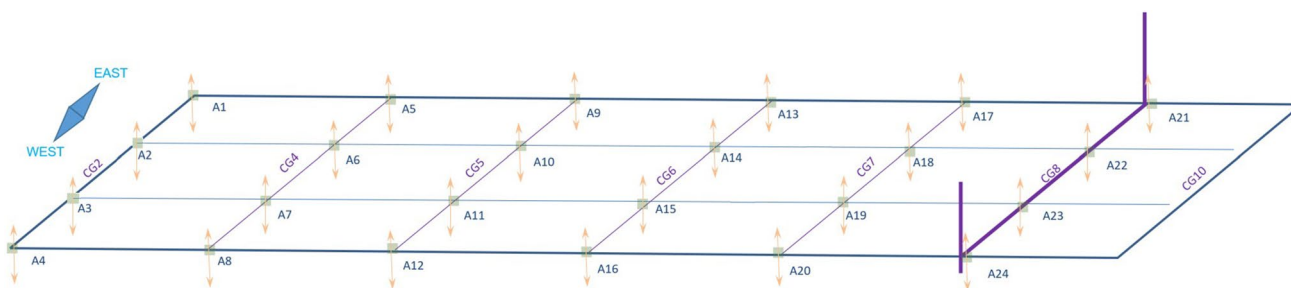
**return**  $w^{(t+1)}$

### 5 Experimental setup

#### 5.1 Data collection

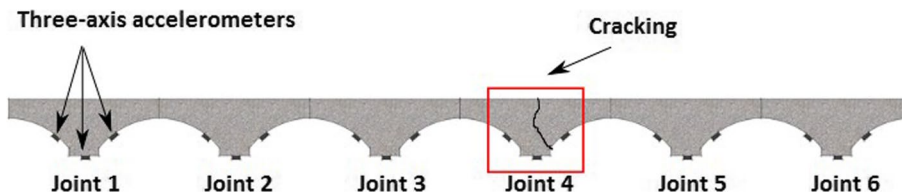
We conducted experiments on two case studies using structural vibration based datasets acquired from a network of accelerometers mounted on two bridges in Australia, cable-stayed bridge (see Fig. 3) and Arch Bridge. For all experiments, three hidden layers were used in our ANN and the accuracy values were obtained using the F-Score (FS),

defined as  $F\text{-score} = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$  where  $\text{Precision} = \frac{TP}{TP + FP}$  and  $\text{Recall} = \frac{TP}{TP + FN}$  (the number of true positive, false positive and false negative are abbreviated by TP, FP and FN, respectively). The core consistency diagnostic technique (CORCONDIA) technique described in [28] was used to determine the number of rank-one tensors  $R$  when it decomposed using CP-ALS method described



**Fig. 4** The locations on the bridge’s deck of the 24  $A_i$  accelerometers used in this study. The cross girder  $j$  of the bridge is displayed as  $CG_j$  [4]

**Fig. 5** Evaluated joints on the arch bridge



in 2. The CORCONDIA suggests  $R = 2$  for all experimented data sets.

### 5.1.1 The cable-stayed bridge

This bridge is 46 m long carries one traffic lane and one pedestrian lane. It is composed of single deck which is 0.16 m thick and 6.3 m wide. This deck is supported by four I-beam steel girders, and 16 stay cables. These cables are connected to the 33 m mast of the cable-stayed bridge. Figure 3 shows a side view of this bridge. We instrumented the Cable-Stayed Bridge with 24 uniaxial accelerometers and 28 strain gauges. However, we only used accelerations data collected from sensors  $A_i$  with  $i \in [1;24]$ . Figure 4 shows the locations of these 24 sensors on the bridge deck. These sensors are connected to a HBM Quantum-X data logger attached to an embedded computer on one side of the bridge. This embedded device provides time synchronization to the data, and stores them temporarily before forwarding via WiFi to a gateway on a nearby building. This gateway then forwards the data over a VPN to our laboratory. The acceleration data are collected at 600 Hz, with a range of 2G and a sensitivity of 2 V/G. Each set of sensors on the bridge along with one line (e.g A1: A4) is connected to one client node and fused in a tensor node  $\mathcal{T}$  to representing one client in our FL network. It results in six tensor nodes  $\mathcal{T}$ .

For the sake of experiments, we emulated two different types of damage on this bridge by placing a large static load (vehicle) at different locations of a structure. Thus, three scenarios have been considered which includes: no vehicle is placed on the bridge (healthy state), a light vehicle with the approximate mass of 3 t is placed on the bridge close to location A10 (“Car-Damage”) and a bus with the approximate mass of 12.5 t is located on the bridge at location A14 (“Bus-Damage”). This emulates slight and severe damage cases which were used in our evaluation Sect. 6.1.

This experiment generates 262 samples (a.k.a events) each consists of acceleration data for a period of 2 seconds at a sampling rate of 600 Hz. We separated the 262 data instances into two groups, 125 samples related to the healthy state and 137 samples for the damage state. The 137 damage examples were further divided into two different damaged cases: the “Car-Damage” samples (107) generated when a stationary car was placed on the bridge, and the “Bus-Damage” samples (30) emulated by the stationary bus.

For each reading of the uni-axial accelerometer, we normalized its magnitude to have a zero mean and one standard variation. The fast Fourier transform (FFT) is then used to represent the generated data in the frequency domain. Each event now has a feature vector of 600 attributes representing its frequencies. The resultant data at each sensor node  $\mathcal{T}$  has a structure of 4 sensors  $\times$  600 features  $\times$  262 events.

### 5.1.2 The arch bridge

This case study used acceleration data captured by a network of accelerometers deployed on the Arch Bridge. It has 800 jack arches to support its joints, which were each instrumented by three tri-axial accelerometers mounted on the left, middle and right side of the joint, as shown in Fig. 5. We conducted two different experiments using this data. The first experiment uses six joints (named 1–6) where only one joint (number four) was known as a cracked joint. The data used in this study contains 36,952 events as shown in Table 1 which were collected for three months. Each event is recorded by a sensor node when a vehicle passes by a jack arch for 1.6 s at a sampling rate of 375 Hz resulting in a feature vector of 600 attributes in the time domain. All the events in the datasets (1, 2, 3, 5, and 6) are labelled positive (healthy events), where all the events in dataset 4 (joint 4) are labelled negative (damaged events). For each reading of the tri-axial accelerometer ( $x, y, z$ ), we calculated the magnitude of the three vectors and then the data of each event is normalized to have a zero mean and one standard variation. Since the accelerometer data is represented in the time domain, it is noteworthy to represent the generated data in the frequency domain using Fourier transform. The resultant six datasets (using the middle sensor of each joint) have 300 features that represent the frequencies of each event. The data collected from the three sensors are fused in one tensor node  $\mathcal{T}$  which represents one client node in our FL network.

The second experiments involve a large set of 85 joints located on different structural locations on the bridge. In addition to the event responses, each node at the joint also collects continuous ambient vibration data at 1500 Hz during midnight. This data is filtered to a 1-min continuous record of ambient responses i.e. a period where no vehicle was driving over the joint. Note that for ambient data

**Table 2** Comparison of the TP, the FP, and the F-score between our FL-Tensor and FL-FedAvg

	PFL-Tensor	FL-Tensor	FL-FedAvg
Cable			
TP	<b>0.96 <math>\pm</math> 0.01</b>	0.91 $\pm$ 0.02	0.93 $\pm$ 0.04
FP	<b>0.12 <math>\pm</math> 0.01</b>	0.16 $\pm$ 0.02	0.29 $\pm$ 0.04
F-score	<b>0.92 <math>\pm</math> 0.01</b>	0.87 $\pm$ 0.02	0.81 $\pm$ 0.04
Arch: six-joints			
TP	<b>0.92 <math>\pm</math> 0.02</b>	0.82 $\pm$ 0.03	0.83 $\pm$ 0.06
FP	<b>0.18 <math>\pm</math> 0.02</b>	0.21 $\pm$ 0.03	0.32 $\pm$ 0.06
F-score	<b>0.88 <math>\pm</math> 0.02</b>	0.81 $\pm$ 0.03	0.78 $\pm$ 0.06
Arch: 85-joints			
TP	<b>0.93 <math>\pm</math> 0.02</b>	0.85 $\pm$ 0.03	0.80 $\pm$ 0.05
FP	<b>0.20 <math>\pm</math> 0.02</b>	0.25 $\pm$ 0.03	0.36 $\pm$ 0.05
F-score	<b>0.87 <math>\pm</math> 0.02</b>	0.80 $\pm$ 0.03	0.71 $\pm$ 0.05

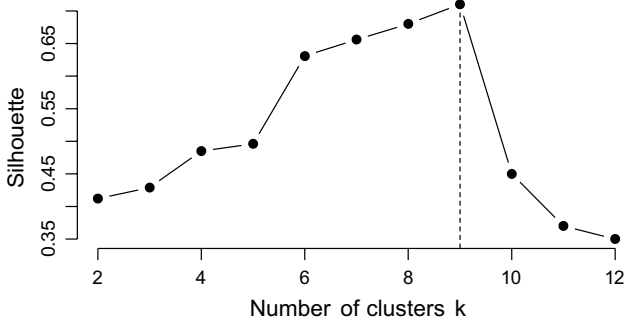


Fig. 6 Silhouette for different  $k$  parameters

collection, only one sensor (middle one) in each node collects data. In the scheme of the FL network, we want to use this ambient data to group bridge substructures with similar behavior into one client and then see whether our personalized FL-tensor with ME method can learn from non-i.i.d. This study leads us to apply  $k$ -means clustering technique which is described in the following section.

### 5.2 Substructure grouping using $k$ -means

$k$ -means clustering is a popular clustering technique. It partitions data into  $k$  clusters so that the within-cluster sum of squares are minimized. This optimization problem is typically addressed using an iterative technique. Convergence is reached when the centroids no longer change. The  $k$ -means algorithm is sensitive to the centroids initialization and the algorithm may converge to a sub-optimal solution. We used  $k$ -means++ method proposed by Arthur and Vassilvitskii [29] to optimize centroids initialization before applying the standard  $k$ -means technique. For selecting the number of clusters  $k$  we use the Silhouettes coefficient proposed by [30]. For each cluster group, we obtain from this process, we fuse all the events data for each joint exits in the same cluster in one tensor node  $\mathcal{T}$  which represents one client node in our FL network.

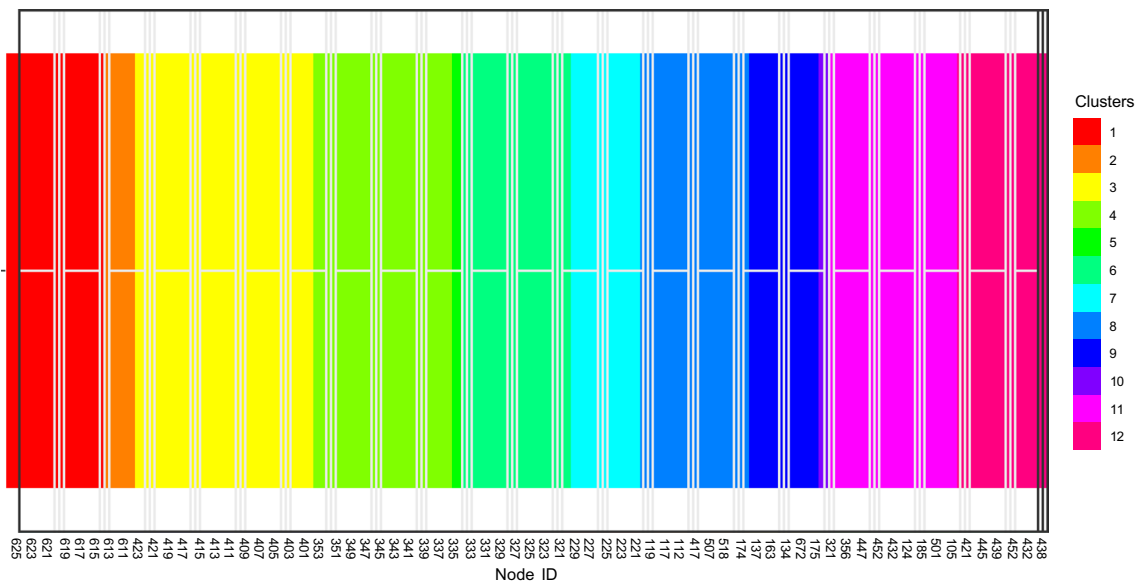


Fig. 7 Node ID and substructure ID, ordered from South to North

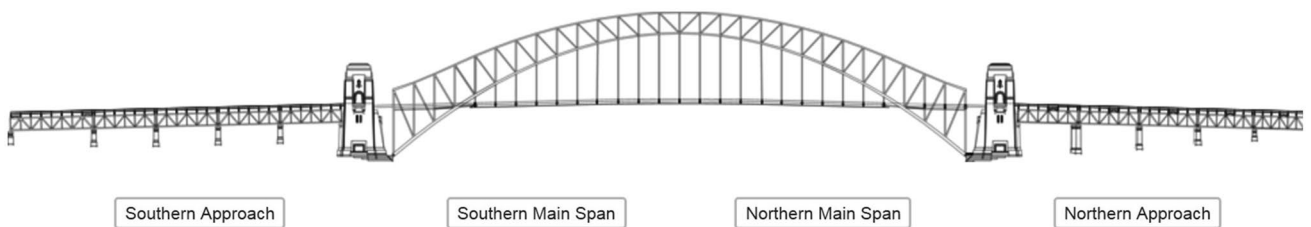


Fig. 8 Overview of the arch bridge

## 6 Experimental results and discussions

In this section, we present the performance evaluation of our proposed federated learning method in terms of detecting damage in civil structural. Our experimental valuation is based on the two real datasets; the cable-stayed bridge and the Arch bridge.

### 6.1 The cable-stayed bridge

We separated the 262 data instances into two groups, 125 samples related to the healthy state and 137 samples for the damage state. The 137 damage examples were further divided into two different damaged cases: the “Car-Damage” samples (107) generated when a stationary car was placed on the bridge, and the “Bus-Damage” samples (30) emulated by the stationary bus.

We randomly selected eighty percent of the healthy events (100 samples) from each tensor node  $\mathcal{T}$  to form training multi-way of  $\mathcal{X} \in \mathbb{R}^{4 \times 600 \times 100}$  (i.e. training set). The 137 examples related to the two damage cases were added to the remaining 20% of the healthy data to form a testing set, which was later used for the model evaluation.

At each client node  $\mathcal{T}$ , we initially applied Algorithm 2 to decompose the tensor  $\mathcal{X}$  into three matrices  $A$ ,  $B$ , and  $C$  which was used to construct learn the ANN model at each client as well as the central model using Algorithms 3 and 4. Although no data from the damaged state has been employed to construct the central model, each personalized local client model was able to identify the damage events related to “Car-Damage” and “Bus-Damage” with an average F-score accuracy of  $0.92 \pm 0.01$ . We compared the results of our PFL-tensor with tensor FL-tensor [31] and traditional FedAvg [14]. Table 2 show the resultant FP, TP and F-score accuracies. As can be seen, PFL-tensor produced better results than FL-tensor and FedAvg. Furthermore, the personalized client’s model was also able to separate the two damage cases (“Car-Damage” and “Bus-Damage”) where the reconstruction error values were further increased for the samples related to the more severe damage cases related to “Bus-Damage”. This is what we anticipated discovering from tensor which can extract damage-sensitive features. Moreover, it reduces the time communication by reducing the number of clients in our FL network.

### 6.2 The arch bridge

#### 6.2.1 Six joints experiments

For each dataset, we randomly selected 80% of the positive events for training and 20% for testing in addition to

the unhealthy events in dataset 4. At each client/joint  $\mathcal{T}$ , we fused data from the three sensors in a multi-way tensor of  $\mathcal{X} \in \mathbb{R}^{3 \times 300 \times n}$  where  $n$  represents number of the training samples defined in Table 1. Similar to the last case study, we applied the three Algorithms 2, 3 and 4 to decompose the tensor  $\mathcal{X}$  and to learn the ANN model at each client as well as central model, respectively. Each personalized client’s model was able to identify its local healthy samples with an average F-score equal to  $0.88 \pm 0.02$ . The model at client/joint 4 was also able to identify 0.86 of the damage samples. These results demonstrate that the PFL-tensor approach without data sharing is still able to identify damage events even-though these events were not involved in the training process. There is no doubt that this work also demonstrates that learning from massively decentralized data is still challenging and needs improvement especially in the prediction accuracy of damage events at joint 4. Table 2 shows the resultant FP, TP, and F-score accuracies of PFL-tensor compared to FL-tensor and FedAvg methods. Similarly, PFL-tensor outperformed these two methods in damage prediction accuracy with less false alarm rates (Table 2).

#### 6.2.2 85 joints experiments

Our initial experiments here were to perform substructure grouping of the 85 joints using  $k$ -means++ method with Silhouette coefficient to select the best values of  $k$ . We run  $k$ -means++ at different values of  $k$  in the range of [2, 12]. This selection was guided by domain knowledge of SHM applications. In practice, the maximum  $k$  value could be set equal to the number of structural spans of a bridge (see Fig. 8). For example,  $k$  could be set to 12 for a bridge that has 12 different structural spans. Our experimental results suggest that joints can be clustered into 9 groups with the highest Silhouette coefficient of 0.71 as shown in Fig. 6. By analyzing these clusters, we found that joints with similar substructure such as joints located on the middle of the bridge were clustered in one group and most of the substructures located on the edges were clustered in another group as shown in Fig. 7.

For each resultant cluster, we fuse all the events vibration data of each joint in one tensor node  $\mathcal{T}$  to represent one client node in our FL network, and we randomly selected 80% of the positive events for training and 20% for testing in addition to the unhealthy events related to joint 4 which was grouped with other joints in cluster 1. Each client/cluster now exists in a form of multi-way tensor  $\mathcal{X} \in \mathbb{R}^{\mathcal{J} \times 300 \times n}$  where  $n$  represents the smallest number of the training samples among joints within the same cluster, and  $\mathcal{J}$  is the number of joints within the same cluster. We applied here again the three Algorithms 2, 3 and 4 to decompose the tensor  $\mathcal{X}$  and to learn the ANN model at each client as well as central model, respectively. Our personalized client model was able

to identify its local healthy samples with an average F-score equal to  $0.87 \pm 0.02$ . The model at client 1 which contains damage data related to joint 4 was also able to identify 0.85 of the damage samples (Fig. 8).

## 7 Conclusions

In this paper, we present a novel machine learning approach for detecting damage in structural health monitoring. Our method employs a Federated Learning (FL) approach and tensor data fusion technique. We model the damage detection as a FL problem where data collected from several distributed sensors attached to a complex structure (clients) learn continuously in local settings without the need to share the data into a centralized learning model (server). Furthermore, our approach models the correlations and relationships among the various sensor nodes and shares the learning with a central model. Our experimental evaluation on two real bridge structure datasets showed promising damage detection accuracy by considering different damage scenarios. In the “Cable-Stayed Bridge” dataset, our FL-based method achieved an accuracy of 94–97%. Our centralized model based on shared models learning also showed that we were able to monitor the progress of damage in the structure by providing increasing reconstruction error values for the samples related to “Bus-damage” events. In the Arch Bridge dataset, our FL-based method achieved 86% damage detection accuracy. The experimental results of these case studies demonstrated the capability of our FL-based damage detection approach with the incorporation of tensor data fusion method to improve the damage detection accuracy and to avoid the problems of transmitting sensed data over the network (network traffic, the low energy consumption of the sensor nodes and vulnerability of the data). Our future work aims to improve our prediction accuracy and better optimisation of the personalized federated learning approach in our FL network, and further to apply it onto different application domains.

**Acknowledgements** The authors wish to thank the Roads and Maritime Services (RMS) in New South Wales, New South Wales Government in Australia and Data61 (CSIRO) for provision of the support and testing facilities for this research work. Thanks are also extended to Western Sydney University for facilitating the experiments on the cable-stayed bridge.

**Author Contributions** The authors contributed equally to this work.

**Funding** Open Access funding enabled and organized by CAUL and its Member Institutions. Not applicable.

**Availability of data and materials** Not applicable.

## Declarations

**Conflict of interest** Not applicable.

**Ethics approval and consent to participate** Not applicable.

**Consent for publication** Not applicable.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Farrar CR, Worden K (2012) Structural health monitoring: a machine learning perspective. Wiley, New York
- Doebbling SW, Farrar CR, Prime MB, Shevitz DW (1996) Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: a literature review. Technical report, Los Alamos National Laboratory, NM, USA
- Anaissi A, Lee Y, Naji M (2018) Regularized tensor learning with adaptive one-class support vector machines. In: International conference on neural information processing. Springer, pp 612–624
- Anaissi A, Makki Alamdari M, Rakotoarivelo T, Khoa N (2018) A tensor-based structural damage identification and severity assessment. *Sensors* 18(1):111
- Makki Alamdari M, Anaissi A, Khoa NL, Mustapha S (2019) Frequency domain decomposition-based multisensor data fusion for assessment of progressive damage in structures. *Struct Control Health Monit* 26(2):2299
- Konečný J, McMahan HB, Yu FX, Richtárik P, Suresh AT, Bacon D (2016) Federated learning: strategies for improving communication efficiency. arXiv preprint [arXiv:1610.05492](https://arxiv.org/abs/1610.05492)
- Li D, Wang J (2019) Fedmd: heterogenous federated learning via model distillation. arXiv preprint [arXiv:1910.03581](https://arxiv.org/abs/1910.03581)
- Deng Y, Kamani MM, Mahdavi M (2020) Adaptive personalized federated learning. arXiv preprint [arXiv:2003.13461](https://arxiv.org/abs/2003.13461)
- Moreau JJ (1963) Propriétés des applications. *Comptes rendus hebdomadaires des séances de l'Académie des Sci* 256:1069–1071
- Kairouz P, McMahan HB, Avent B, Bellet A, Bennis M, Bhagoji AN, Bonawitz K, Charles Z, Cormode G, Cummings R et al (2019) Advances and open problems in federated learning. arXiv preprint [arXiv:1912.04977](https://arxiv.org/abs/1912.04977)
- Dinh TC, Tran N, Nguyen TD (2020) Personalized federated learning with moreau envelopes. *Adv Neural Inf Process Syst* 33:21394–21405
- Hard A, Rao K, Mathews R, Ramaswamy S, Beaufays F, Augenstein S, Eichner H, Kiddon C, Ramage D (2018) Federated learning for mobile keyboard prediction. arXiv preprint [arXiv:1811.03604](https://arxiv.org/abs/1811.03604)
- Hanzely F, Richtárik P (2020) Federated learning of a mixture of global and local models. arXiv preprint [arXiv:2002.05516](https://arxiv.org/abs/2002.05516)



14. McMahan B, Moore E, Ramage D, Hampson S, Arcas BA (2017) Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. PMLR, pp 1273–1282
15. Guha N, Talwalkar A, Smith V (2019) One-shot federated learning. arXiv preprint [arXiv:1902.11175](https://arxiv.org/abs/1902.11175)
16. Reisizadeh A, Mokhtari A, Hassani H, Jadbabaie A, Pedarsani R (2020) Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In: International conference on artificial intelligence and statistics, pp 2021–2031
17. Dai X, Yan X, Zhou K, Yang H, Ng K.K, Cheng J, Fan Y (2019) Hyper-sphere quantization: Communication-efficient sgd for federated learning. arXiv preprint [arXiv:1911.04655](https://arxiv.org/abs/1911.04655)
18. Wang J, Joshi G (2018) Cooperative sgd: a unified framework for the design and analysis of communication-efficient sgd algorithms. arXiv preprint [arXiv:1808.07576](https://arxiv.org/abs/1808.07576)
19. Lin T, Stich SU, Patel KK, Jaggi M (2018) Don't use large mini-batches, use local sgd. arXiv preprint [arXiv:1808.07217](https://arxiv.org/abs/1808.07217)
20. Stich SU (2018) Local sgd converges fast and communicates little. arXiv preprint [arXiv:1805.09767](https://arxiv.org/abs/1805.09767)
21. Smith V, Chiang C-K, Sanjabi M, Talwalkar A (2017) Federated multi-task learning. arXiv preprint [arXiv:1705.10467](https://arxiv.org/abs/1705.10467)
22. Arivazhagan MG, Aggarwal V, Singh AK, Choudhary S (2019) Federated learning with personalization layers. arXiv preprint [arXiv:1912.00818](https://arxiv.org/abs/1912.00818)
23. Khoa NLD, Anaissi A, Wang Y (2017) Smart infrastructure maintenance using incremental tensor analysis. In: Proceedings of the 2017 ACM on conference on information and knowledge management. ACM, pp 959–967
24. Ho JC, Ghosh J, Sun J (2014) Marble: high-throughput phenotyping from electronic health records via sparse nonnegative tensor factorization. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 115–124
25. Bader BW, Harshman RA, Kolda TG (2007) Temporal analysis of semantic graphs using alsan. In: Seventh IEEE international conference on data mining, 2007. ICDM. IEEE, pp 33–42
26. Kolda TG, Bader BW (2009) Tensor decompositions and applications. *SIAM Rev* 51(3):455–500
27. Anaissi A, Suleiman B, Zandavi S.M (2020) Necpd: An online tensor decomposition with optimal stochastic gradient descent. arXiv preprint [arXiv:2003.08844](https://arxiv.org/abs/2003.08844)
28. Bro R, Kiers HA (2003) A new efficient method for determining the number of components in parafac models. *J Chemom* 17(5):274–286
29. Arthur D, Vassilvitskii S (2006) k-means++: The advantages of careful seeding. Technical report, Stanford
30. Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math* 20:53–65
31. Anaissi A, Suleiman B, Naji M, Braytee A (2021) Intelligent structural damage detection: a federated learning approach. In: International symposium on intelligent data analysis. Springer, 326–338

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.