



Spatial Damage Characterization in Self-Sensing Materials via Neural Network-Aided Electrical Impedance Tomography: A Computational Study

Lang Zhao,¹ Tyler Tallman² and Guang Lin^{3,*}

Abstract

Continuous structural health monitoring (SHM) and integrated non-destructive evaluation (NDE) are important for ensuring the safe operation of high-risk engineering structures. Recently, piezoresistive nanocomposite materials have received much attention for SHM and NDE. These materials are self-sensing because their electrical conductivity changes in response to deformation and damage. Combined with electrical impedance tomography (EIT), it is possible to map deleterious effects. However, EIT suffers from important limitations – it is computationally expensive, provides indistinct information on damage shape, and can miss multiple damages if they are close together. In this article we apply a novel neural network approach to quantify damage metrics such as size, number, and location from EIT data. This network is trained using a simulation routine calibrated to experimental data for a piezoresistive carbon nanofiber-modified epoxy. Our results show that the network can predict the number of damages with 99.2% accuracy, quantify damage size with respect to the averaged radius at an average of 2.46% error, and quantify damage position with respect to the domain length at an average of 0.89% error. These results are an important first step in translating the combination of self-sensing materials and EIT to real-world SHM and NDE.

Keywords: Damage characterization; Neural network; Machine learning; Structural health monitoring; Integrated nondestructive evaluation.

Received date: 30 June 2020; Accepted date: 18 September 2020.

Article type: Research article.

1. Introduction

Structural health monitoring (SHM) and integrated nondestructive evaluation (NDE) are promising methods of ensuring the safety and reducing the maintenance costs associated with potentially high-risk and high-value aerospace, civil, and energy engineering structures.^[1] Recently, so-called self-sensing materials have received much attention for application in SHM and integrated NDE. These materials are engineered to elicit a property change in response to environmental factors. For mechanical self-sensing (*i.e.* the detection of deformation and damage), the piezoresistive

effect has much potential. Piezoresistivity refers to a material having deformation and damage dependent electrical conductivity. Piezoresistive-based self-sensing is very appealing for SHM and integrated NDE because low-cost, non-invasive, and easily multiplexed electrical measurements can be used to monitor the mechanical state of the material or structure. Piezoresistivity is typically achieved by modifying a nonconductive matrix with highly conductive fillers. When sufficiently many conductivity fillers are added, an electrically connected or percolated filler network forms thereby allowing for electrical transport through the insulating matrix. Deformations which change the connectedness of the fillers or damage which severs the connection between fillers manifest as a conductivity change. The concentration of fillers needed to achieve percolation is closely tied with filler aspect ratio – very long and thin fillers percolate at much lower filler weight fractions. Because it is desirable to not modify the host matrix with excessively many fillers (potentially adversely affecting the load-carrying capability of the material), ultra-high aspect ratio fillers such as carbon nanotubes (CNTs) and carbon nanofibers (CNFs) have been thoroughly studied for use in

¹ School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907, USA.

² School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN 47907, USA.

³ Department of Mathematics, School of Mechanical Engineering, Department of Statistics (Courtesy), Department of Earth, Atmospheric, and Planetary Sciences (Courtesy), Purdue University, West Lafayette, IN 47907, USA.

*Email: Guanglin@purdue.edu (G. Linc)

self-sensing materials. For example, nanofiller modification has been explored for imparting piezoresistive self-sensing to fiber-reinforced polymers,^[2] polymer sensing skins,^[3] ceramics,^[4] and even cementitious materials.^[5] Electrical impedance tomography (EIT) is a natural complement for SHM in piezoresistive materials. Quite succinctly, EIT is a process by which the internal conductivity distribution of a domain can be spatially mapped via a series of voltage-current measurements taken at the boundary of the domain. EIT therefore allows us to detect and spatially map conductivity-changing artifacts such as deformations in nanofiller-modified polymers,^[6] impact damage in nanofiller-modified glass fiber/epoxy composites^[7] and sensing skins,^[8] and damage to cement with CNT-modified aggregate interfaces.^[9] Beyond nanofiller-modified materials, EIT has also been used for detecting and shaping damage in cementitious materials with silver or copper-based conductive coatings,^[10] metal particle-reinforced cement,^[11] and carbon fiber-reinforced composites.^[12] Because of this diversity of material systems and ability to shape and localize damage, the combination of piezoresistive self-sensing and conductivity mapping via EIT seemingly has much potential for SHM and integrated NDE. Despite this potential, EIT has some important limitations. First, depending on its formulation, it can be computationally expensive and therefore slow to produce an image (*e.g.* nonlinear difference imaging^[13]). Second, EIT images are blurry and indistinct. This can make definitive damage characterization difficult. This limitation is a consequence of the underlying physics of EIT being diffusion-based. And third, because of the diffuse nature of EIT, it can also miss multiple damages that are close together such as closely spaced holes by reproducing a single, large blob-like conductivity change that covers both damages. These second and third limitations – a lack of distinct shaping and indistinguishability of closely spaced damages – are readily apparent in Fig. 1. In this figure, EIT blurs together two closely spaced holes. Manual interpretation of EIT for prognostic purposes may therefore mistakenly assume that there are only two damages when there are, in fact, three. Furthermore, manual interpretation would be unable to provide precise characterization of these damages beyond occurring in roughly the same location as the conductivity change and being of roughly the same shape. Overcoming these limitations is important because real-time damage quantification (*i.e.* knowing the precise damage location, size, and shape) is essential to effecting meaningful diagnostics. In light of the preceding discussion, we herein present a novel method of integrating machine learning via neural networks with the EIT problem for damage detection, localization, and quantification. Indeed, the application of machine learning and neural networks to EIT has already received some attention particularly in the context of medical imaging.^[14–16] However, there are some important differences in the work herein presented. First, we apply network-supplemented EIT for damage detection in an engineered material system rather than for physiological imaging. And

second, we train the network to parameterize the damage. That is, some a priori information on the damage is provided to the network such that it can very accurately predict damage parameters including location, size, and number in a piezoresistive material with accuracy that is unapproachable by standard EIT. In the forthcoming we will first describe the material system by which our computational study was calibrated. Next, we will summarize the standard EIT formulation. Then, we will describe two approaches to training the neural network – via EIT conductivity maps (*i.e.* image data) and via EIT-predicted conductivity change vectors (*i.e.* numerical data). It will be shown that this is an important distinction which substantially affects the accuracy of the network. And lastly, we computationally validate the network using an experimentally calibrated simulation routine. Summarily, the intent of this work is to explore whether or not EIT-generated conductivity data (*i.e.* in the form of maps and vectors) can be integrated with existing machine learning architectures for spatial damage characterization.

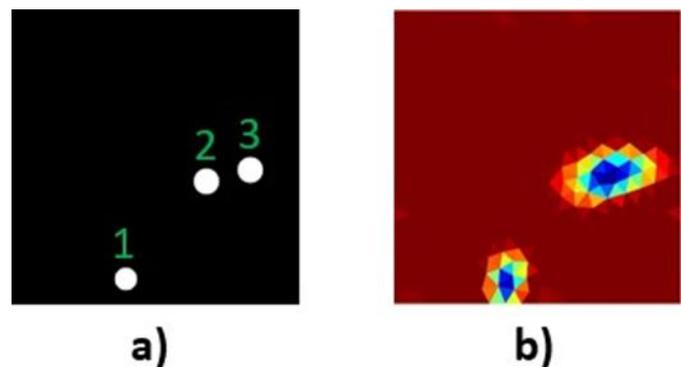


Fig. 1 Demonstration of important limitations of EIT for damage detection. a) Three holes are simulated. Two of the holes are quite close together. b) While the EIT reconstruction does capture artifacts collocating with the holes, the image is blurry making it difficult to extract meaningful diagnostic information such as hole size. Worse, EIT blurs together holes 2 and 3. This demonstrates an important limitation of EIT – it can fail to differentiate between close damages.

2. CNF/epoxy

The networks herein utilized were computationally trained using a model calibrated to experimental data for a CNF/epoxy nanocomposite. For the sake of completeness, the manufacturing procedure is briefly outlined here. However, interested readers are directed to the original work by Tallman and Hassan^[17] for a more expansive discussion on how the material was manufactured and its electrical properties were measured. To produce this material, CNFs (Pyrograf III PR-24-XT-HHT, Applied Sciences) were dispersed into an uncured liquid epoxy resin (Fibre Glast 2000) using a combination of high-energy mixing, bath sonication, a viscosity thinning agent, and chemical surfactant (Triton X-100, bioWORLD). Procedurally, desired amounts of CNFs were first weighed and added to the liquid epoxy resin along with acetone at an acetone-to-resin volume ratio of 1:2 and

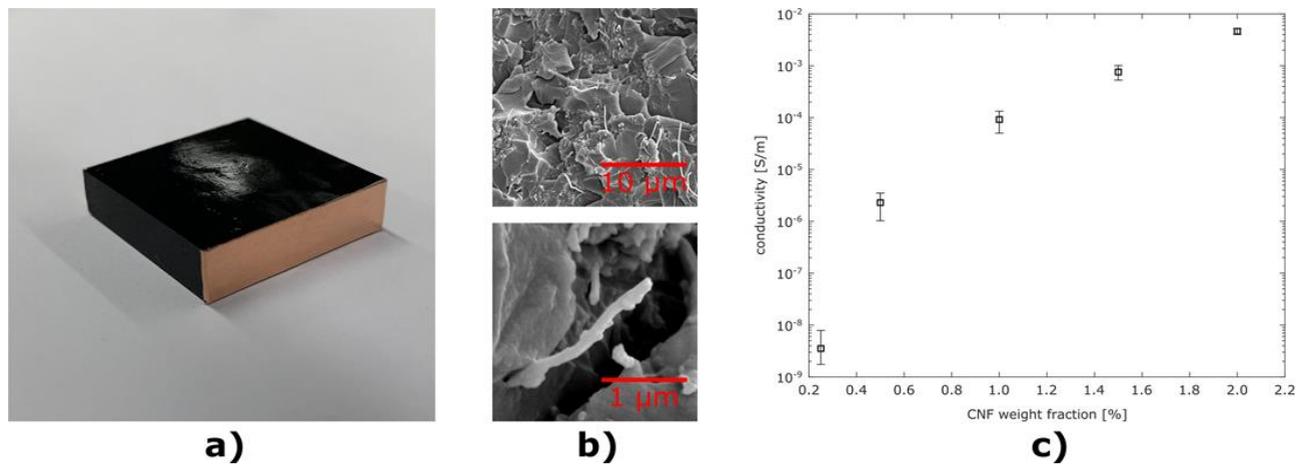


Fig. 2 Piezoresistive CNF/epoxy. a) Post-manufactured measurement specimen with electrodes applied. b) SEM images of CNF/epoxy fracture surface showing CNF dispersion above and close up of a single CNF protruding from the fracture surface below. c) Plot of CNF/epoxy conductivity versus CNF weight fraction. Boxes indicate mean conductivity and error bars indicate standard deviation.

surfactant at a surfactant-to-CNF weight ratio of 0.76:1. This mixture of CNFs, epoxy resin, surfactant, and acetone was then mixed in a planetary centrifuge mixer (Thinky-AR100) for three minutes. After this initial mixing, the mixture was sonicated for 4 hours in a bath sonicator. Sonication helps to break up CNF agglomerations. After the sonication process, a hotplate stirrer was used at 60 °C for 24 hours in order to eliminate the acetone from the mixture via evaporation. Next, curing agent and air release agent (BYK A-501) were added at weight ratios of 27:100 and 0.001:1, respectively. Lastly, the mixture was degassed for 30 minutes at room temperature before being poured in open rectangular molds and cured at 60 °C for five hours. A total of five specimens were made at 0.25, 0.5, 1, 1.5, and 2.0 wt.% CNFs. Post-cured CNF/epoxy specimens were then cut into smaller pieces measuring 1” × 1” × 0.25” via a water-cooled tile saw. These smaller pieces are referred to as measurement specimens. This resulted in a total of eight measurement specimens per CNF weight fraction. Electrodes were attached using a combination of conductive silver paste and copper tape such that electrical resistance could be measured. Post-manufactured and instrumented specimens and scanning electron microscope (SEM) images of the underlying CNF network are shown in Fig. 2. From the measured resistance and specimen dimensions, the electrical conductivity was calculated. Conductivity versus CNF weight fraction data is also shown in Fig. 2. This experimentally determined conductivity was used to calibrate the EIT model that is used to train the networks as described later.

3 Electrical impedance tomography

EIT is a process by which the internal conductivity of a domain is mapped. EIT has two important parts – i) the forward problem by which the voltage-current relationship in the domain is modelled and ii) the inverse problem which seeks to spatially map the internal conductivity distribution from observations of the voltage current relationship. These parts are both described in the forthcoming sub-sections.

3.1 EIT forward problem

The goal of the EIT forward problem is to solve for electrode voltages for a prescribed current injection magnitude and conductivity distribution. This process is governed by Laplace’s equation in the absence of internal current sources as is shown in equation (1), where σ is the conductivity distribution and ϕ is the electric potential.

$$\nabla \cdot \sigma \nabla \phi = 0 \quad (1)$$

In EIT, the complete electrode model (CEM) is used as boundary conditions on Laplace’s equation. The CEM boundary conditions account for the finite size of practical electrodes by adding an additional degree of freedom for the electrode voltage. As shown in equation (2), the CEM also accounts for a voltage drop between the domain and electrodes due to contact impedance. Charge conservation is enforced by equation (3) by requiring that the total current entering the domain is equal to the total current leaving the domain. In these equations, z_l represents the contact impedance between the l^{th} electrode and the domain, V_l is the voltage of the l^{th} measurement electrode, the total number of electrodes is denoted by L , E_l represents the size of the l^{th} electrode, and n represents a normal vector pointing outward from the domain boundary.

$$z_l \sigma \nabla \phi \cdot n = V_l - \phi \quad (2)$$

$$\sum_{i=1}^L \int_{E_i} \sigma \nabla \phi \cdot n \, ds_i = 0 \quad (3)$$

Equations (1)-(3) are most easily solved by discretization via the finite element method. This is shown in equation (4). In these equations, Φ is a vector representing electric potential in the domain, V is a vector of electrode voltages, and the vector I represents electrode currents. A_M is the stiffness matrix for steady-state diffusion and the remaining matrices are formed as shown as follows where w_i is the i^{th} finite element interpolation function. This work uses linear triangle elements.

$$\begin{bmatrix} A_M & A_Z \\ A_W^T & A_D \end{bmatrix} \begin{bmatrix} \Phi \\ V \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix} \quad (4)$$

$$A_{Zij} = \sum_{i=1}^L \int_{S_i} \frac{1}{z_i} w_i w_j ds_i \quad (5)$$

$$A_{Wij} = - \int_{S_i} \frac{1}{z_i} w_i ds_i \quad (6)$$

$$A_D = \text{diag} \left(\frac{E_i}{z_i} \right) \quad (7)$$

3.2 EIT inverse problem

The EIT inverse problem tries to spatially map the conductivity distribution of the domain for a given set of boundary voltage observations. This can be done by updating the conductivity distribution of the forward model until the model-predicted voltages match the experimentally collected voltages. However, recovering the conductivity distribution in this way is often impractical because the EIT inverse problem is ill-posed and therefore extremely susceptible to factors such as model-to-experiment electrode misplacement, geometry deviations, and noise. Therefore, difference imaging is most often used wherein the conductivity change between two states is sought (*i.e.* the change in conductivity before damage and after damage). Herein, we use a one-step linearization approach by linearizing the damage-induced conductivity change about an initial estimate of the background or unperturbed conductivity distribution. In order to do this, define the experimentally observed change in boundary voltage as shown next in equation (8).

$$\delta V = V_m(t_2) - V_m(t_1) \quad (8)$$

Above, the t_1 and t_2 arguments indicate that boundary voltages were measured before and after the damage event, respectively. We can likewise define this difference via the forward problem as $W = F(\sigma + \delta\sigma) - F(\sigma)$ where $F(\cdot)$ is the output of the forward problem evaluated at the conductivity in its argument and $\delta\sigma$ is the damage-induced conductivity change which we seek. To find $\delta\sigma$, we linearize $F(\sigma + \delta\sigma)$ about the unperturbed background conductivity, σ , via a truncated Taylor series as shown below in equation (9).

$$F(\sigma + \delta\sigma) \approx F(\sigma) + \frac{\partial F(\sigma)}{\partial \sigma} \delta\sigma \quad (9)$$

To actually solve for the conductivity change, we form a minimization problem in which we seek to minimize the difference between δV and the linearized form of W as shown in equation (10).

$$\delta\sigma^* = \underset{\delta\sigma \leq 0}{\text{argmin}} (\|\delta\sigma - \delta V\|_m^m + \alpha \|R(\delta\sigma)\|_n^n) \quad (10)$$

Above in equation (10), m and n represent the m or n^{th} vector norm to the m or n^{th} power. Herein, we use $m = n = 2$, but interested readers are directed to work by Tallman and Hernandez^[18] for a discussion on other norms. Note also that a regularization term, $R(\delta\sigma)$, has been incorporated into the minimization. This is necessary because of the ill-posedness of the EIT inverse problem. A discretized approximation of the Laplace operator is used for regularizing the inverse problem in this work due to its ability to filter out highly oscillatory noise artifacts. The extent of regularization is controlled by the scalar hyper-parameter, α . Lastly, note that the minimization

has been constrained such that $\delta\sigma \leq 0$. This is based on the physical realization that damage will cause a complete cessation or loss of conductivity. For more detailed solution strategies of equation (10), interested readers are directed to the engaging work of Smyl *et al.*^[19]

4. Problem setting

In this study, the goal is to detect the damage status on the self-sensing material specimen by analyzing the EIT-generated conductivity change distribution data with machine learning methods. In this preliminary, proof-of-concept study, we consider only simple through-hole damages. This task is divided into three sub-tasks: predicting the number, the radius, and the center position of the holes. For EIT images, we use convolutional neural networks (CNN) to predict the number of holes, the radius of holes, and use K-means to predict the center position of holes. For conductivity change vectors, we use fully connected neural networks (FCNN) to detect the damage status.

For the problem of predicting the number of holes, the principal idea is to build a model that can be used to predict the number of holes given the EIT generated conductivity change data. Since in our dataset, the number of holes on a specimen is between 1 to 3, this problem can be seen as a multi-class classification problem, the class 1, class 2, and class 3 represent that this specimen has 1, 2, and 3 holes, respectively. With the EIT images as input, we can train a CNN to predict the number of holes. On the other hand, with the conductivity change vectors as input, the FCNN is trained as a classification model to predict the number of holes.

In order to detect the degree of damage on the material specimens, the radius of holes is needed to be identified. A generic model is created to predict the radius of holes on material specimens given the EIT-generated conductivity change data. Since the distribution of radius is continuous, this task can be seen as a regression problem. With the EIT images as input, we can train a CNN to predict the radius of holes. On the other hand, with the conductivity change vectors as input, the FCNN is trained as a regression model to predict the radius of holes.

In order to localize the damage, we need to predict the center position of holes. From the EIT images, a hole can be seen as a cluster of pixels; therefore, the clustering algorithms can be applied to predict the center position of holes. Unlike the neural networks we introduced above, the clustering algorithms are unsupervised machine learning algorithms, so we do not need to provide an annotation of samples to the clustering algorithms. In our study, the K-means algorithm is used and the coordinates of the pixels that form the damaged domain on the image are fed to the K-means, K-means divides the coordinates into K clusters, K is the number of holes, and K-means can also output the centroids for the K clusters, we can approximately see the K centroids as the centers of holes. Using the conductivity change vectors as input, this problem

can be treated as a regression problem, so we apply the FCNN as a regression model to predict the center position of holes with the vectors.

5. Dataset

A large amount of data is required in our study, and all data used in our study is generated by an experimentally calibrated EIT simulation program. In these simulations, an adaptive mesh routine is used to simulate a random number of through holes within the specimen. Holes are simulated as voids in the mesh. The simulated specimen used in this work measured 0.9 m \times 0.9 m and has a background conductivity equal to the experimentally measured mean conductivity of CNF/epoxy at 1.0 wt.% CNFs, 9.16×10^{-5} S/m. A total of 16 electrodes, each measuring 1/9th of the domain width, were simulated using the CEM as described previously. These electrodes can be seen in Fig. 3. An adjacent injection and measurement scheme was used.

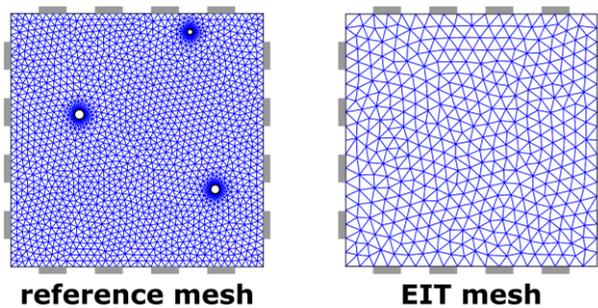


Fig. 3 Representative damaged reference mesh (note that damage is randomly generated such that no two reference meshes are the same) and the EIT reconstruction mesh. Electrodes are shown along the periphery of the domains in gray.

All damage in our study is circular and each specimen has at least 1 hole and at most 3 holes. For the balance of data, the numbers of generated samples with 1, 2, and 3 holes are the same. Each hole has a random radius between 0.03 m and 0.05 m. The EIT routine described previously is then used to reproduce the conductivity change distribution of the mesh with holes onto a different, hole-free mesh thereby avoiding a so-called ‘inverse crime’. Specifically, two reference meshes are used for the difference imaging process – a hole-free reference mesh and a reference mesh with 1-3 randomly generated holes such that a voltage change vector, δV , due to the holes can be formed. Further, the reference meshes are much more refined than the EIT reconstruction mesh. The reference meshes have on the order of 3,500 triangular elements (the exact number of elements depends on the number and size of the randomly generated through holes). The finite element mesh used for the EIT reconstruction, on the other hand, always has 772 triangular elements. Because finite element simulations are well known to converge with increasing number elements, utilizing a much coarser EIT mesh for the conductivity mapping ensures that intrinsic modeling errors exist between the reconstruction mesh and the simulated experimental data

(*i.e.* the reference mesh) as would be expected for actual experimental data. A representative reference mesh with three holes and the EIT reconstruction mesh are shown in Fig. 3. To more faithfully reproduce experimental conditions, voltage measurements were also contaminated with noise at a signal-to-noise ratio of 85 dB. These simulation parameters (*i.e.* EIT mesh refinement, signal-to-noise ratio, choice of regularization for EIT, *etc.*) were selected based on our previous experience with using EIT for imaging of CNF-based polymeric nanocomposites.^[6,20,21]

EIT generates two categories of data, conductivity change distribution vectors and images constructed from the vectors. For the vector, its length is 772, which equals to the number of cells in the EIT finite element mesh. Each element in the vector represents the conductivity change value of the corresponding cell. The images are stored as 538-by-538-by-3 data matrix that records the red, green, and blue color components for each individual pixel. Both the width and height of the image is 538 pixels. Both vector-based and image-based data represent the conductivity change distribution through a specimen and are considered as the inputs of our machine learning models in this study.

6. Methodology

6.1 EIT image-based machine learning approach

6.1.1 Convolutional neural networks

CNNs are a variant of neural networks that are frequently applied to image analysis. In this section, the most important theories and concepts of CNNs are described.

Convolutional layer

The convolutional layer is essential in CNN since it is used to detect features and has several kernels with learnable weights.^[22] Kernels work as filters and a kernel is a matrix of integers. Each kernel provides a measure for how close a patch of input resembles a feature. A kernel slides over the complete image and dot product is taken between the kernel and a patch of the image. The greater the result, the closer the patch of the image resembles the feature. The computed dot product values corresponding to the channels of each kernel are summed up with a bias to produce the results of each kernel. These results form the spatial feature maps of the convolutional layer.

Pooling layer

The pooling layer is where the features extracted by the convolutional layer are selected for down sampling, which reduces the computational cost. The function of the pooling layer is straight-forward; for example, the max pooling layer extracts the maximum in the feature maps; the average pooling layer extracts the averages in the feature maps.^[23]

Fully connected layer

The fully connected layer is an essential component of CNNs because it connects to all nodes in the previous layer. It takes the outputs of the previous layer, performs dot products between the weights of each node and the inputs, and

adds a bias in each node.^[24]

Softmax function

The softmax function is an activation function usually used in the classification model to predict the class of the input. The softmax function takes the output from the previous layer, estimates the probabilities for each class, and predicts the class that has the greatest probability as the prediction of the classification model.^[25]

6.1.2 K-means

The K-means algorithm is a clustering algorithm that is capable of partitioning a dataset into K distinct clusters^[26] – these clusters should not overlap each other such that each data point in the dataset belongs to only one cluster. K-means tries to make the data points inside a cluster as close as possible while keeping the data points in different clusters as far as possible. It randomly selects K data points as initial centroids then the sum of the squared distance between data points and all centroids are calculated. Each data point is assigned to the closest centroid, then centroids for the clusters are calculated by taking the average of the all data points that belong to each cluster. These steps are iterated until the centroids cannot be changed anymore.

6.1.3 Data Preprocessing

Noise is a common feature of EIT images, which influences the performance of our model. In order to mitigate this, the red color components of pixels are removed, the green and blue components whose value are smaller than 50 are also removed. In order to reduce the dimension of input and accelerate the model training process, the input images are resized to 100 pixels \times 100 pixels by applying interpolation. Pixel values of images are normalized by dividing all pixel values by 255 in order to improve the performance of the model and speed up the training process. The preprocessing steps are shown in Fig. 4.

6.1.4 Prediction of the number of holes and the radius of holes

Two CNNs are used to predict the number and radius of holes from EIT images. For the CNN used to predict the number of

holes, the first layer is the convolution layer that has 16 kernels with a kernel size of 3-by-3. The second layer is a max pooling layer used to reduce the complexity of computation by reducing the dimensionality of the input. The third layer is a flatten layer that flattens the input to a 1-dimensional array. The fourth layer is a fully connected layer that has 256 nodes to capture all the information contained in the input of this layer. The fifth layer is a fully connected layer with 32 nodes. An activation function called ReLU is applied in nodes in the convolutional layer and the first and second fully connected layer. Lastly, the output layer is a fully connected layer that has 3 nodes with a softmax activation function to work as a classifier. Fig. 5 presents the details of the structure of the CNN clearly.

The structure of the CNN for predicting the radius of holes is similar to that for predicting the number of holes, the only difference is that the output layer has no activation function since the convolutional neural network is a regression model.

6.1.5 Prediction of the center position of holes

K-means algorithm is employed to predict the center position of holes. The first step of using the K-means algorithm is specifying the number of clusters, K, which is the number of holes on the image in our study. A neural network model is employed first to predict the number of holes accurately as the value of K. After the EIT images are preprocessed, only the pixels that form the damage domain are reserved, and the coordinates of these pixels can be determined. The K-means algorithm with the coordinates is employed to predict the center position for the K number of holes.

6.2 Conductivity change vector-based machine learning approach

6.2.1 Data Preprocessing

The conductivity change vector is normalized to a unit vector by dividing every element by the norm of the vector. This normalization procedure can speed up the gradient descent process and improve the performance and stability of our model.

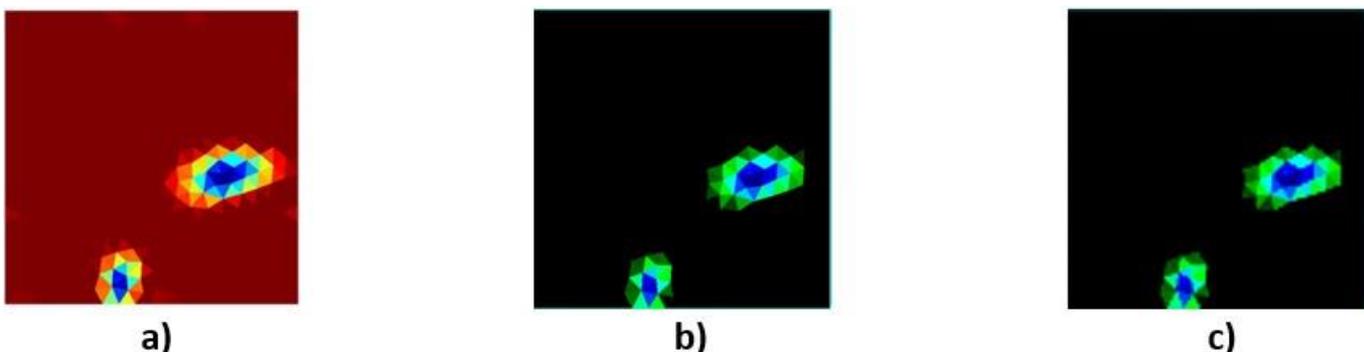


Fig. 4 Preprocessing of EIT images used for image-based training. a) An original EIT image. b) The image after being removed the color components. c) The image after being resized.

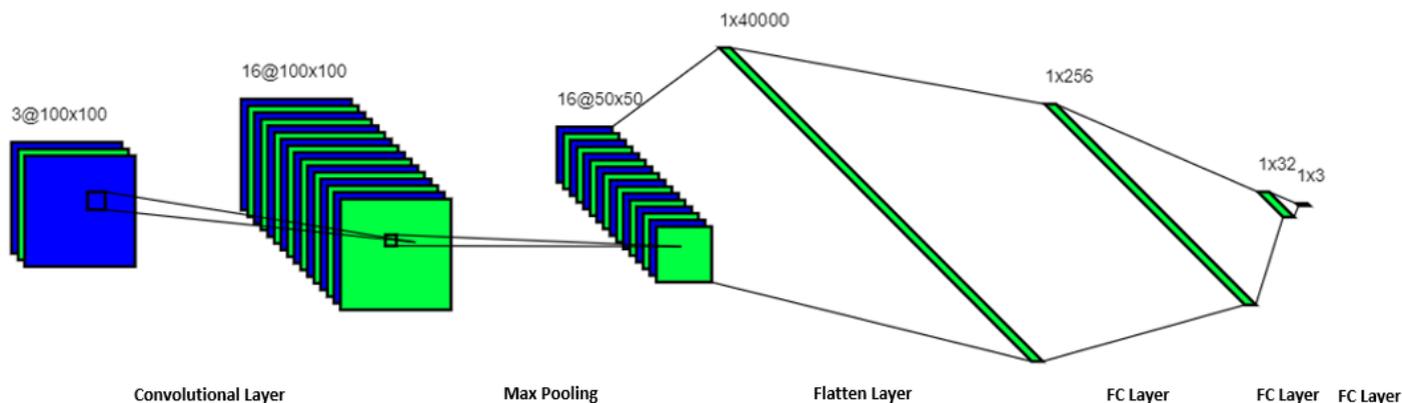


Fig. 5 The structure of the EIT image-based CNN for predicting the number of holes. 1 Convolutional layer, 2 Max pooling, 3 Flatten layer, 4 Fullyconnected layer, 5 Fully connected layer, 6 Fully connected layer + Softmax.

6.2.2 Prediction of the number, radius and center position of holes

Three FCNNs are employed to predict the number, the radius, and the center position of holes separately. First, a FCNN is built as a multi-class classification model to predict the number of holes in three classes (*i.e.* classes with 1, 2, and 3 holes) from the conductivity change vectors. The proposed network is composed of 1 input layer, 3 hidden layers, and 1 output layer. The input layer contains 772 nodes. Each of them represents an element in the conductivity change vector (recall that the EIT reconstruction mesh has 772 elements). The number of nodes in the first hidden layer is 256, which is enough to understand all the information contained in the input layer. There are 64 and 16 nodes in the next two hidden layers. The ReLU activation function is applied in nodes in hidden layers. In the output layer, there are three nodes and the activation function is again a softmax function since our model is a three-classes classifier. Fig. 6 presents the structure of the FCNN.

Second, a FCNN was built as a regression model to predict the radius of holes with the conductivity change vectors. The structure of the FCNN is almost the same as that for predicting the number of holes except that the third hidden layer has 32

neurons and no activation function is applied on the output layer in this network since the model is a regression model.

Third, a FCNN is constructed as a regression model to predict the center position of holes from the conductivity change vectors. This FCNN is similar to the one for predicting the number of holes except that there are 128 and 32 nodes in the second and third hidden layers and there are six neurons in the output layer with no activation function.

7. Results using the machine learning approaches with experimentally calibrated simulation data

7.1 Results obtained from the EIT image-based machine learning approach

The first experiment is conducted using the previously proposed CNN for predicting the number of holes. The network is trained and tested with EIT images. 22,950 EIT images are used to train and validate the CNN. In the training process, 50 epochs and batch size 128 have been used when feeding in the training samples. After training, the model’s performance on 4,050 testing images gives the accuracy of 0.911. Fig. 7 shows the training accuracy and the validation accuracy plots against epochs. From Fig. 7, we can see the training error and validation error converge after 50 epochs.

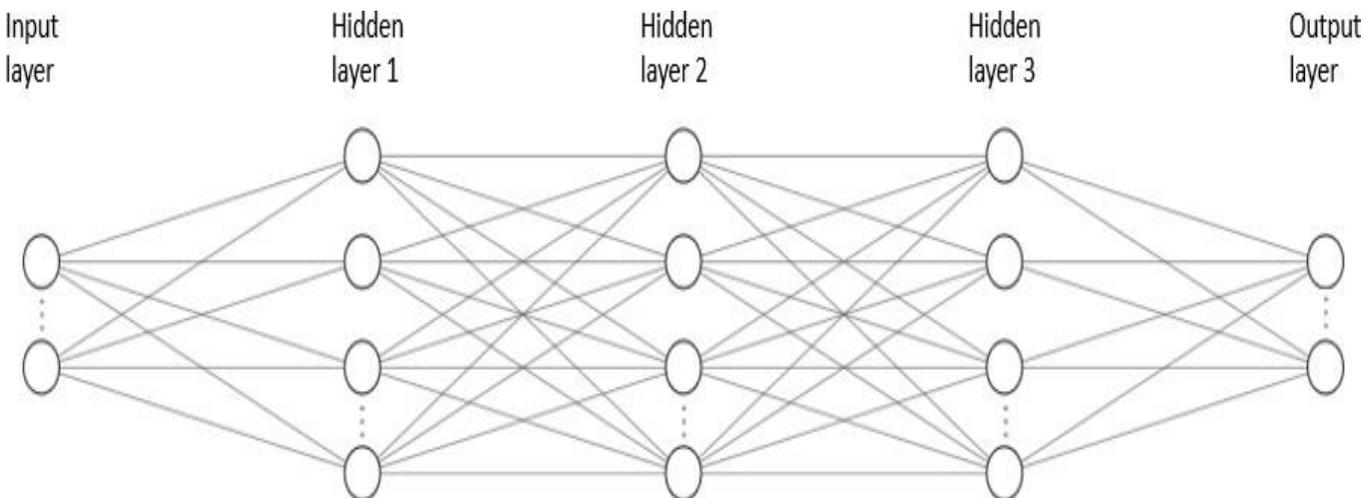


Fig. 6 The structure of the conductivity change vector-based FCNN for predicting the number of holes.

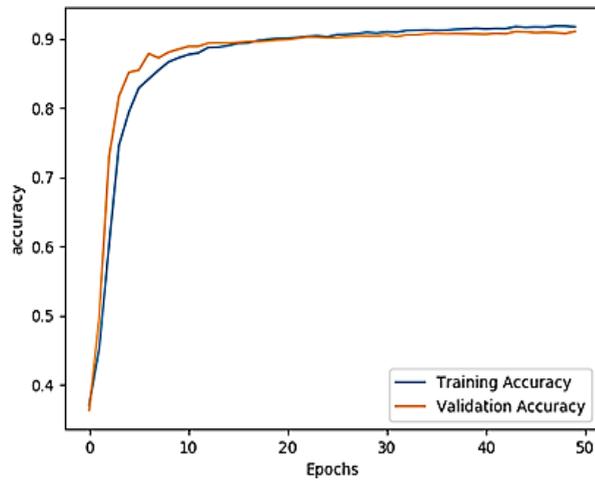


Fig. 7 Training process of EIT image-based CNN for predicting the number of holes.

The second experiment is conducted using the previously proposed CNN for predicting the radius of holes. 22,950 EIT images are used to train the neural network. After the training, the model is tested on 4,050 images. The performance using mean squared error (MSE) gives a testing error of 1.16×10^{-4} . By comparing the predicted output and the annotation, the average difference between the predicted and true radius is 0.00511 m. Fig. 8 shows the training process. From Fig. 8, we can see the training error and validation error converge – the validation error is relatively large compared to the training error.

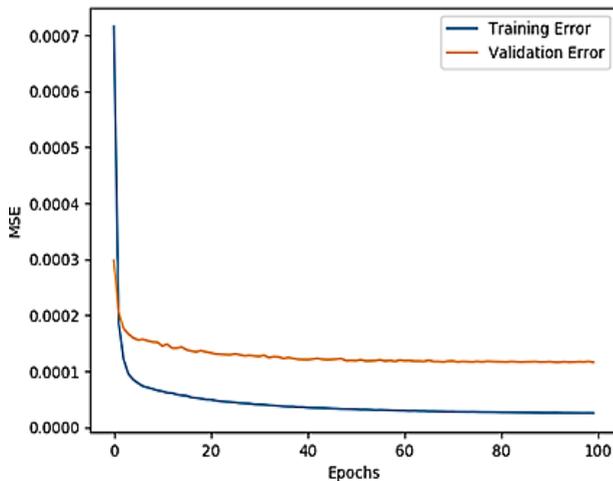


Fig. 8 Training process of EIT image-based CNN for predicting the radius of holes.

The third experiment is conducted using the K-means model with 27,000 images to predict the center position of holes. By comparing the predicted output and annotations, the average distance between predicted and true center position is 0.0172 m.

7.2 Results obtained from the conductivity change vector-based machine learning approach

The first experiment is conducted using the previously proposed FCNN for predicting the number of holes. 21,600

conductivity change vectors are used to train and validate the fully connected neural network. 5,000 epochs and batch size 128 have been applied in the training process. After the training, the model’s performance on 5,400 testing samples is evaluated with an accuracy of 0.992. Fig. 9 shows the training accuracy and the validation accuracy as a function of epochs. From this result, we can see a clear improvement in accuracy compared to the result of using EIT images as the inputs.

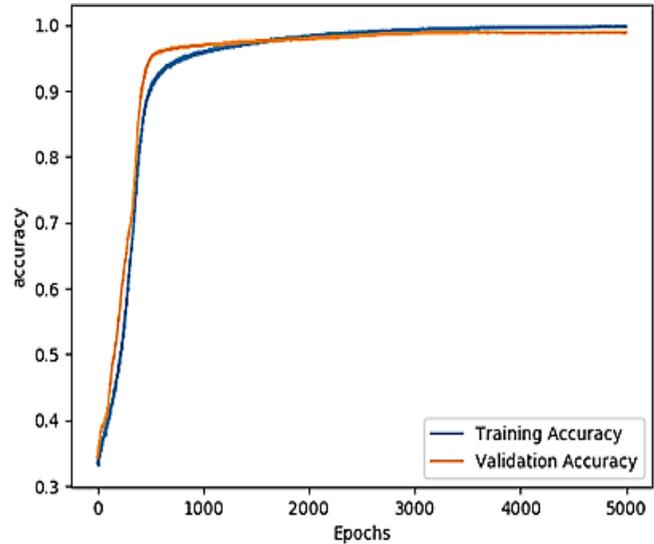


Fig. 9 Training process of the conductivity change vector-based FCNN for predicting the number of holes.

The second experiment is conducted using the previously proposed FCNN for predicting the radius of holes. 21,600 conductivity change vectors are used to train and validate the FCNN. 1,000 epochs and batch size 256 have been applied in the training process. After the training, the model’s performance on 5,400 testing samples using MSE is evaluated with the testing error of 6.11×10^{-6} . By comparing the predicted output and the annotation, the average difference between the predicted and true radius is 0.000987 m. Fig. 10 shows both the training error and validation error decrease as increasing the epochs.

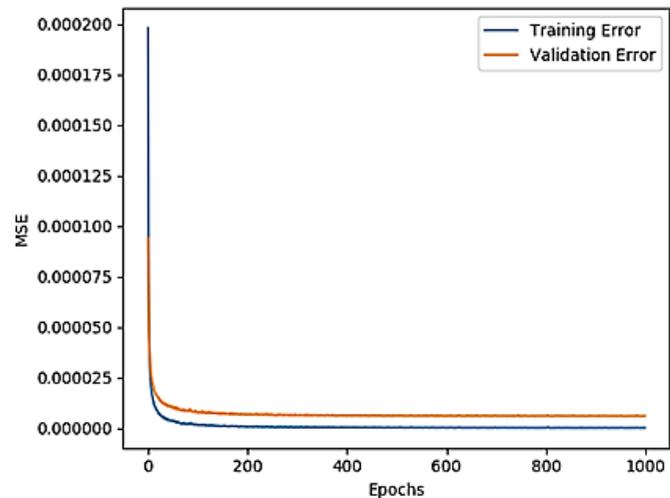


Fig. 10 Training process of the conductivity change vector-based FCNN for predicting the radius of holes.

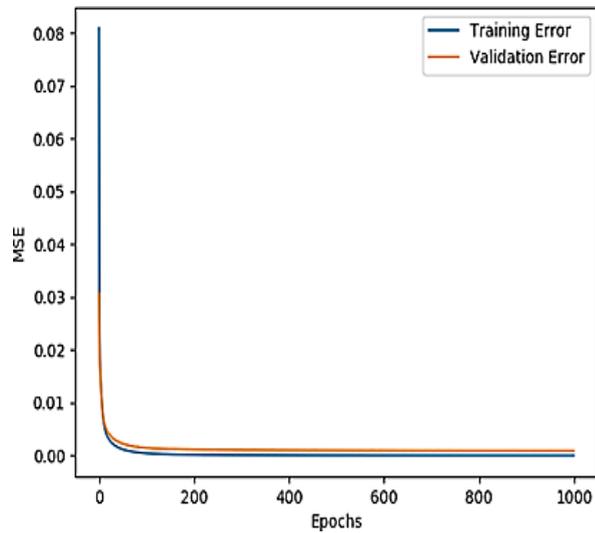


Fig. 11 Training process of the conductivity change vector-based FCNN for predicting the center position of holes.

The third experiment is to use the proposed FCNN for predicting the center position of holes. 21,600 conductivity change vectors are used to train and validate the FCNN, 1,000 epochs and batch size 64 have been applied in the training process. After the training, the model’s performance on 5,400 testing samples is evaluated using MSE with the testing error is 3.21×10^{-4} . By comparing the predicted output and the annotation, the average difference between the predicted and true center position is 0.00808 m. Fig. 11 shows the training error and validation error decrease as increasing epochs during the training process of the neural network.

7.3 Comparison of results with EIT images and conductivity change vectors as input

According to our experiments, machine learning methods, especially neural networks, can be built for the damage prediction and quantification from EIT-generated conductivity change data. In our study, we use EIT images and conductivity change vectors as input separately to predict the number, the radius and the center position of holes. The comparison between the performance of models with different inputs is shown in the Table 1 and a few representative images are shown in Fig. 12. From these results, we can clearly see that the conductivity change vector-based machine learning approach is a better option. The main reason may be that the process of constructing images from vectors causes the loss of information.

Table 1. Comparison between the performance of models using EIT images and conductivity change vectors as input.

Prediction Goal	Metrics	EIT Images	Conductivity Change Vectors
Number of holes	Testing accuracy	0.911	0.992
Hole radius	Average difference between predicted and true radius (m)	0.00511	0.000987
Center position	Average distance between predicted and true center positions (m)	0.0172	0.00808

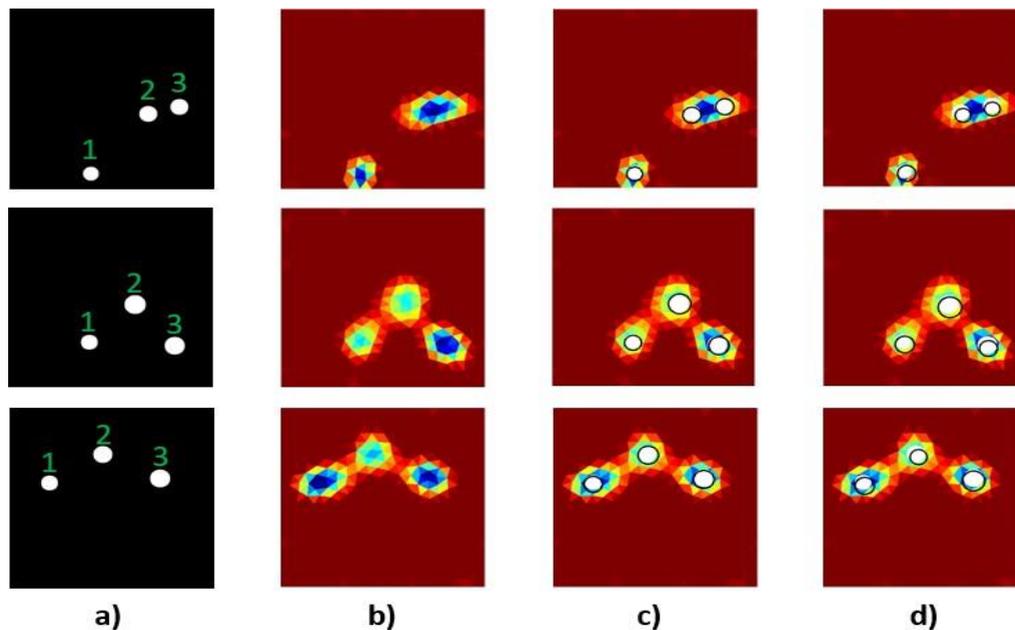


Fig. 12 Three representative examples of the performance of damage detection using machine learning approaches. a) Real damage status. b) Image generated by EIT. c) Damage predicted by the conductivity change vector-based machine learning approaches. d) Damage predicted by the EIT image-based machine learning approaches. In c) and d), the white filled circles are the real damages and the black circles are the predicted damages. Note again the top row where the proximity of the holes causes EIT to produce only a single region of conductivity change encompassing both holes. The method herein proposed, however, can adeptly recognize this as due to two distinct holes.

8. Conclusion

This study is one of the first attempts at combining neural networks with the EIT method for damage detection in self-sensing materials. And from the results presented herein, the proposed approach seems to have good performance. Compared to the traditional ways of using EIT for damage detection, our method not only detects the existence of damage but also gives a much more quantitative description of the damage status, including the number, radius, and center position of damage. Our results show that the machine learning approach can indeed predict the number of damages with up to 99.2% accuracy, quantify damage size with respect to the averaged radius at an average of 2.46% error, and quantify damage position with respect to the domain length at an average of 0.89% error. And because all of this can be done in virtually real-time, this method seemingly has great potential for application to the field of SHM. However, utilizing the conductivity change vector directly considerably outperformed image recognition-based methods. This is likely a consequence of reducing the image quality prior to employing image recognition. Encouraged by this success, several directions of future work are suggested. First, the size of damage studied in our research is not very small compared to the size of the specimen, particularly for damage detection thresholds of interest to practical SHM. Future work should therefore look to improve the sensitivity of our method for detecting smaller damage. Second, the number of holes in our study is limited between 1 to 3. Moving forward, we need to increase the number of holes that our method can handle. Third, in this proof-of-concept study, damage was limited to simple circular through-holes. Real structural damage is obviously much more complex. Consequently, future work should seek to generalize the shaping capabilities of the basic approach herein presented. Fourth, as described in section 5 Dataset, care was taken to ensure the simulation parameters are representative of real experimental data (*i.e.* by using a greatly refined reference mesh to ensure system modeling errors between the simulated data and the EIT mesh, including noise, *etc.*). However, the potential of the preliminary proof-of-concept results herein presented will need to ultimately be experimentally validated using real data. This will be especially important for understanding how modeling errors corrupt or bias the networks ability to predict actual damage. And fifth, the performance of neural network-aided damage characterization via EIT should be benchmarked against manual interpretation (*i.e.* EIT images interpreted by a person) in order to conclusively prove the advantage of machine learning over human inspection in this application.

Acknowledgements

We gratefully acknowledge the support from the National Science Foundation (DMS-1555072, DMS-1736364, CMMI-1634832, and CMMI-1560834), Brookhaven National

Laboratory Subcontract 382247, ARO/MURI grant W911NF-15-1-0562 and Department of Energy DE-SC0021142.

Conflict of Interest

There is no conflict of interest.

Supporting Information

Not applicable

References

- [1] C. R. Farrar and K. Worden, *Philos. T. R. Soc. A*, 2007, **365**, 303–315, doi: 10.1098/rsta.2006.1928.
- [2] E. T. Thostenson and T.W. Chou, *Adv. Mater.*, 2006, **18**, 2837–2841, doi: 10.1002/adma.200600977
- [3] K. J. Loh, T.-C. Hou, J. P. Lynch and N. A. Kotov, *J. Nondestruct. Eval.*, 2009, **28**, 9–25, doi: 10.1007/s10921-009-0043-y
- [4] F. Inam, B. R. Bhat, T. Vo and W. M. Daoush, *Ceram. Int.*, 2014, **40**, 3793–3798, doi: 10.1016/j.ceramint.2013.09.039
- [5] A. D’Alessandro, M. Rallini, F. Ubertini, A. L. Materazzi and J. M. Kenny, *Cement Concrete Comp.*, 2016, **65**, 200–213, doi: 10.1016/j.cemconcomp.2015.11.001.
- [6] T. Tallman, S. Gungor, K. Wang and C. E. Bakis, *Carbon*, 2015, **95**, 485–493, doi: 10.1177/1045389X17692053.
- [7] A. Thomas, J. Kim, T. Tallman and C. Bakis, *Compos. Part B*, 2019, **177**, 107276, doi: 10.1016/j.compositesb.2019.107276.
- [8] B. R. Loyola, V. La Saponara, K. J. Loh, T. M. Briggs, G. O’Bryan and J. L. Skinner, *IEEE Sens. J.*, 2013, **13**, 2357–2367, doi: 10.1109/JSEN.2013.2253456.
- [9] S. Gupta, J. G. Gonzalez and K. J. Loh, *Struct. Control. Health*, 2017, **16**, 309–323, doi: 10.1177/1475921716643867.
- [10] D. Smyl, M. Pour-Ghaz and A. Seppänen, *NDT E Int.*, 2018, **99**, 123–133, doi: 10.1016/j.conbuildmat.2020.118299.
- [11] S. Nayak and S. Das, *Mater. Design*, 2019, **175**, 107817, doi: 10.1016/j.matdes.2019.107817.
- [12] J. Cagán and L. Michalcová, *J. Nondestruct. Eval.*, 2020, **39**, 1–12, doi: 10.1007/s10921-020-00677-2.
- [13] D. Liu, V. Kolehmainen, S. Siltanen and A. Seppänen, *Inverse Probl.*, 2015, **31**, 035012, doi: 10.1088/0266-5611/31/3/035012.
- [14] D. Hu, K. Lu and Y. Yang, 2019 IEEE International Conference on Imaging Systems and Techniques (IST), 2019, pp. 1–6, doi: 10.1109/IST48021.2019.9010151.
- [15] S. J. Hamilton and A. Hauptmann, *IEEE T. Med. imaging*, 2018, **37**, 2367–2377, doi: 10.1109/TMI.2018.2828303.
- [16] S.W. Huang, H. M. Cheng and S. F. Lin, 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2019, pp. 1551–1554, doi: 10.1109/EMBC.2019.8856781.
- [17] T. Tallman and H. Hassan, *Compos. Sci. Technol.*, 2019, **181**, 107669, doi: 10.1016/j.compscitech.2019.05.026.
- [18] T. Tallman and J. Hernandez, *NDT E Int.*, 2017, **91**, 156–163, doi: 10.30919/esmm5f919.

- [19] D. Smyl, S. Bossuyt, W. Ahmad, A. Vavilov and D. Liu, *Struct. Health Monit.*, 2020, **19**, 215–239, doi: 10.1177/1475921719841012.
- [20] H. Hassan and T. N. Tallman, *Struct. Health Monit.*, 2019, 1475921719863062, doi: 10.1177/1475921719863062.
- [21] T. Tallman, S. Gungor, K. Wang and C. Bakis, *Smart Mater. Struct.*, 2014, **23**, 045034, doi: 10.1088/0964-1726/23/4/045034
- [22] Y.-J. Cha, W. Choi, G. Suh, S. Mahmoudkhani and O. Büyüköztürk, *Comput-Aided Civ. Inf.*, 2018, **33**, 731–747, doi: 10.1111/mice.12334.
- [23] S.-J. Lee, T. Chen, L. Yu and C.-H. Lai, *IEEE Access*, 2018, **6**, 12755–12768, doi: 10.1109/ACCESS.2018.2796722.
- [24] J. Schmidhuber, *Neural networks*, 2015, **61**, 85–117, doi: 10.1016/j.neunet.2014.09.003.
- [25] Y. Kim, arXiv preprint arXiv:1408.5882, 2014, doi: 10.3115/v1/D14-1181.
- [26] K. Krishna and M. N. Murty, *IEEE T. Syst. Man, Cy. B*, 1999, **29**, 433–439, doi: 10.1109/3477.764879.

Publisher's Note Engineered Science Publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

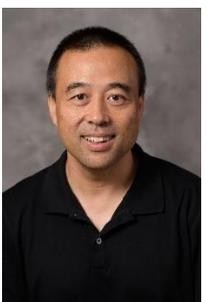
Author Information



Lang Zhao received a B.Eng. degree from Beihang University, and M.S. degree in mechanical engineering from Purdue University in 2020. He is currently a software engineer at TuSimple. His work fields include full-stack software development and machine learning.



Tyler N. Tallman received a B.S. degree in physics from the University of Wisconsin–Eau Claire and a B.S. degree in engineering mechanics from the University of Wisconsin–Madison in 2010, and M.S. and Ph.D. degrees in mechanical engineering from the University of Michigan in 2012 and 2015, respectively. He is currently an Assistant Professor with the School of Aeronautics and Astronautics at Purdue University. His research interests include self-sensing materials, nanocomposites, and inverse problems.



Guang Lin received a B.S. degree in mechanics from Zhejiang University in 1997, a M.S. degree in mechanics and engineering science from Peking University, and a Ph.D. degree from the Division of Applied Mathematics, Brown University in 2007. Currently he is an associate professor at both School of Mechanical Engineering and Department of Mathematics at Purdue University. Guang Lin received NSF faculty early career development award. His research interests include machine learning, uncertainty quantification, optimization, Bayesian algorithms and big data analysis.