Article:
Champneys, M.D., Green, A., Morales, J. et al. (2021) On the vulnerability of data-driven structural health monitoring models to adversarial attack. Structural Health Monitoring, 20 (4). pp. 1476-1493. ISSN: 1475-9217

https://doi.org/10.1177/1475921720920233

Special Issue Article

SHM Structural Health Monitoring

Structural Health Monitoring
2021, Vol. 20(4) 1476–1493

DOI: 10.1177/1475921720920233
journals.sagepub.com/home/shm

On the vulnerability of data-driven structural health monitoring models to adversarial attack

Max David Champneys[1,2], Andre Green[3], John Morales[3], Moisés Silva[3] and David Mascarenas[3]

Abstract

Many approaches at the forefront of structural health monitoring rely on cutting-edge techniques from the field of machine learning. Recently, much interest has been directed towards the study of so-called adversarial examples; deliberate input perturbations that deceive machine learning models while remaining semantically identical. This article demonstrates that data-driven approaches to structural health monitoring are vulnerable to attacks of this kind. In the perfect information or 'white-box' scenario, a transformation is found that maps every example in the Los Alamos National Laboratory three-storey structure dataset to an adversarial example. Also presented is an adversarial threat model specific to structural health monitoring. The threat model is proposed with a view to motivate discussion into ways in which structural health monitoring approaches might be made more robust to the threat of adversarial attack.

Keywords

Structural health monitoring, adversarial attack, threat model

Background

Data-driven modelling in structural health monitoring

Monitoring the health of engineering structures is of critical importance to countless engineering disciplines and applications. Since it is very difficult for most sensors to measure damage,[1] the practising engineer must instead rely on a range of indirect tools and techniques in order to identify, locate and manage damage in structures.

Much interest in the field of structural health monitoring (SHM) has been in data-driven approaches that leverage techniques from machine learning and statistical pattern recognition.[2] Several authors have extolled the virtues of casting SHM problems in this way.[3,4] The modern engineer is fortunate to have at their disposal a litany of tools that are able to perform pattern recognition, and the literature reflects this. To date, authors have employed outlier analysis,[5] neural networks,[6] support vector machines[7] and decision trees[8] among many other approaches. The reader is directed to Farrar and Worden[9] for a comprehensive reference text.

Despite the wealth of progress made in the last two decades, the large-scale deployment of SHM methodologies is still in its infancy. There are many well cited reasons for this, with the scarce availability of damaged training data and issues surrounding environmental variation emerging as key themes. However, innovative solutions to these problems are steadily being found.

In the face of these challenges, there are several large-scale systems that are currently in deployment. Perhaps most famous is the OnStar navigation and diagnostics system available in some commercial vehicles. There is also the integrated condition assessment system (ICAS)[10] deployed by the US navy for the monitoring of hardware, as well as a number of techniques

[1]Industrial Doctorate Centre in Machining Science, The University of Sheffield – Advanced Manufacturing Research Centre with Boeing (AMRC), Rotherham, UK
[2]Dynamics Research Group, The University of Sheffield, Sheffield, UK
[3]Los Alamos National Laboratory, Los Alamos, NM, USA

Corresponding author:
Max David Champneys, Industrial Doctorate Centre in Machining Science, The University of Sheffield – Advanced Manufacturing Research Centre with Boeing (AMRC), Rotherham S60 5TZ, UK.
Email: mdchampneys1@sheffield.ac.uk

developed for monitoring rotor-craft under the umbrella term of health and usage monitoring systems (HUMS).[11] These systems are all implemented in potential life-safety applications and are therefore potential targets for malicious attack.

It is the opinion of the authors that if SHM frameworks are to be adopted in life-safety or economically critical projects, then it is of utmost importance that the security and robustness of the underlying models are rigorously examined.

Adversarial attacks on pattern recognition models

It is important here to distinguish between the similar but distinct topics of adversarial attack and adversarial machine learning. The latter refers to machine learning methods whereby two learning models are pitted against each other in order to produce generative models[12] that are able to sample from the underlying input distributions.

Adversarial attack, however, refers to the construction of adversarial examples for classification models. These are deliberately perturbed inputs for which the classifier assigns an incorrect label despite small or imperceptible semantic alteration from the true state. Mathematically, for a classification model $\mathcal{M}$ that assigns a class label y to input vector $\{x\}$

$$\mathcal{M}(\{x\}) = y \tag{1}$$

An adversarial example $\{x'\}$ will result in an adversarial classification (erroneous label)

$$\mathcal{M}(\{x'\}) = y' \tag{2}$$

The adverse label is assigned despite semantic similarity to a human observer

$$\{x\} \approx \{x'\} \tag{3}$$

The vulnerability of neural-network models to adversarial attack was first presented by Szegedy et al.[13] It was later shown that the vulnerability of such models was not limited to the area of neural networks.[14] In fact, a wide array of classification algorithms are susceptible to this type of attack. Subsequently, Carlini and Wagner[15] produced a general approach for the construction of adversarial examples that were difficult to detect and were able to bypass several of the recently proposed adversarial defence strategies.

Since this revelation, the literature has grown rich with contributions exploring adversarial attacks and the generation of adversarial examples. A recent review can be found in Yuan et al.[16] By far, the majority of the case studies and the application papers thus far published on adversarial attack have been concerned with image classification and machine vision. It is in these areas that the majority of the taxonomy has been developed. In this article, the authors hope to demonstrate that adversarial attack is also a real threat to SHM models.

Data-driven approaches to SHM rely at their core on machine learning models that are susceptible to adversarial attack. While some of these rely directly on neural networks,[6,17–19] Papernot et al.[14] demonstrated extensively the vulnerability of techniques beyond neural networks, including support vector machines, decision trees, logistic regressors and others in their highly cited paper. With this in mind, it is clear that the adversarial attack is a threat to a great deal of data-driven SHM.

While no direct consideration of adversarial vulnerability for SHM has yet been presented in the literature, the susceptibility of data-driven approaches has already been demonstrated for the related field of process monitoring.[20] In their paper, the authors demonstrate the adversarial fragility of a deep neural network trained to detect system failures and offer an adversarial training method similar to Madry et al.[21] for hardening the classifier.

With the rapid pace of adversarial attack research in mind, it is the aim of this article to motivate serious discussion into the vulnerability of the learning models proposed for SHM. Presented here are two contributions. The following section envisages an adversarial attack threat model for SHM. The threat model is accompanied by a taxonomy specific to threats arising in SHM frameworks. The third and the fourth sections provide demonstration of adversarial attack on a damage detection model trained on an SHM-benchmarking dataset. It is shown that simple transformations can be constructed that map every true input to an adversarial example, even when the inputs have not been used to train the classification model. The final section outlines the directions for further investigation into ways in which SHM might be made more robust to adversarial attack.

An adversarial attack threat model for data-driven SHM

The foundation for this threat model will be the vibration-based damage identification framework presented in Farrar et al.[22] The approach can be summarised by four steps:

1. Operational evaluation
2. Data acquisition
3. Feature extraction or pre-processing
4. Label discrimination

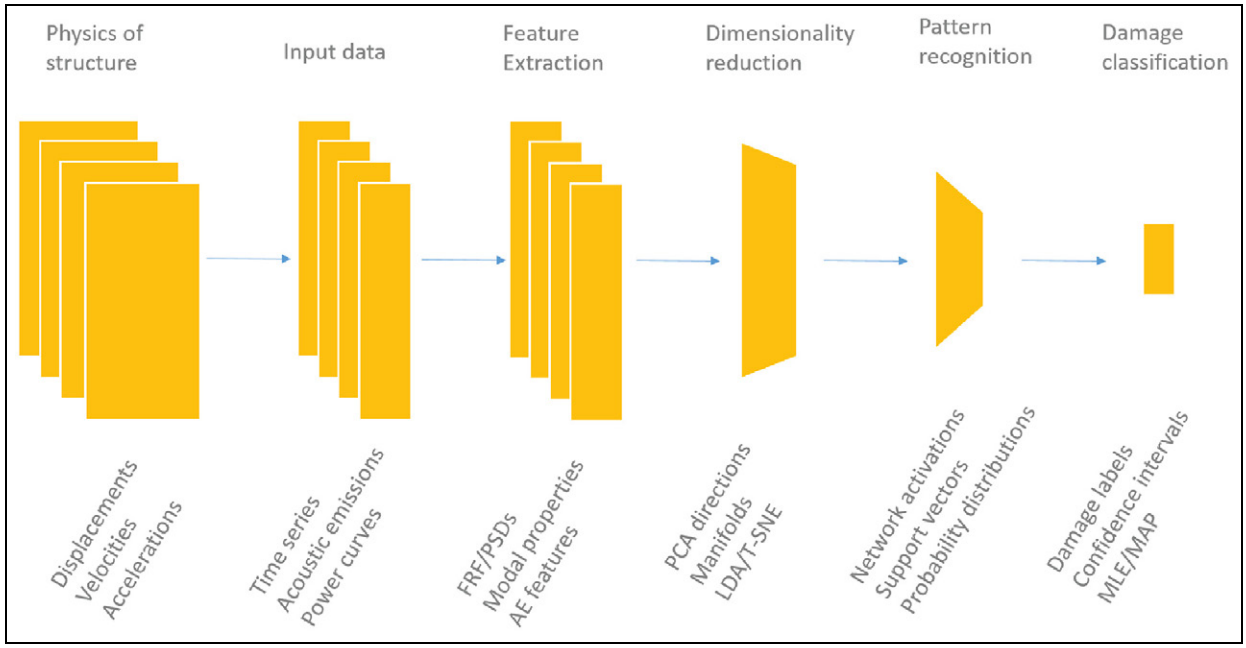

Figure 1. An example overview of a data-driven SHM methodology.

This framework is expanded in Figure 1 to produce a physics-to-label representation of the SHM framework. For convenience, the following definitions are presented:

- A SHM framework (the system, Figure 1) that utilises a statistical pattern recognition model (the model, $\mathcal{M}(\{x\})$) is deployed for the purpose of identifying damage or otherwise measuring the health state of some structure of interest (y).
- A malicious entity (the attacker) is attempting to influence or compromise the accuracy of the model by leveraging adversarial examples ($\{x'\}$) to induce false or misleading results.

The principal threats in this scenario are thus defined as follows:

- The false labelling of the damaged state as undamaged (false negative, $y_+ \rightarrow y'_-$);
- The false labelling of the undamaged state as damaged (false positive, $y_- \rightarrow y'_+$);
- Semantic similarity between real and adverse inputs to the model ($\{x\} \approx \{x'\}$).

Thus, we may define a general objective function for the production of semantically convincing adversarial examples as

$$J(\{x'\}|\{x\}, \mathcal{M}) = \mathcal{F}(\{x\}, \{x'\}) + \lambda \mathcal{G}(y, y') \qquad (4)$$

where $y = \mathcal{M}(\{x\})$ and $y' = \mathcal{M}(\{x'\})$ are as defined earlier. The objective function has two terms and a weighting parameter. The first term penalises examples that are not semantically similar to true inputs such as the L1 or L2 norm. The second term is a metric, which ensures that the adversarial example produces adversarial miss-classifications such as cross-entropy loss metrics. The nefarious labelling of damaged structures as healthy is of serious concern. If left undetected, attacks of this type have the potential to result in worsening of structural health or critical failure.

The consequences of a false-positive classification may seem minor compared to that of the false-negative classification and in the extreme this is certainly the case. However, unnecessary maintenance and inspection may bear a financial toll and repeated miss-classifications may erode confidence in the monitoring system. Unaddressed, either of these eventualities are likely to render the system completely useless in the long term. Furthermore, if the adversarial examples have high semantic similarity to measured healthy data, it would be very difficult for a human to recognise the occurrence of the adversarial examples and trouble-shooting the problem may be difficult.

The SHM analogue of semantic similarity from image recognition is not immediately intuitive. It is unlikely that a human would be able to identify damage by observing measured signals from a structure alone. However, there are domains that do present semantic information to the trained eye. For intuition, consider the frequency response functions (FRFs) of a

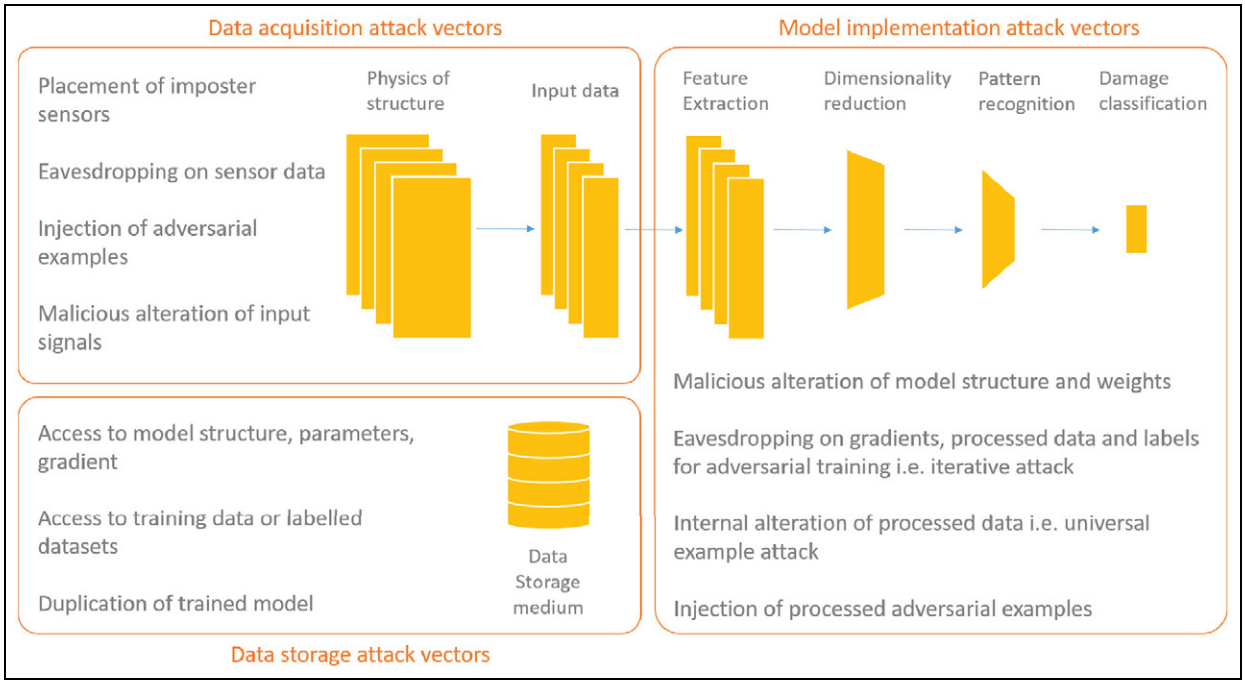

Figure 2. Categorisation of attack vectors.

vibrating structure. FRFs are frequently used as damage and condition-sensitive features in SHM as they encode physical dynamic properties and can be efficiently computed. To a trained engineer, the natural frequencies and damping properties of a structure are approximately tractable visually and it would certainly be possible to identify significant deviation from the expected structure of the signal. FRFs, as well as other features such as coherence and acoustic emission signals, contain semantic structure and are all potential targets for adversarial attack.

Threat model

In order to arrive at a coherent adversarial attack threat model for SHM, it is important to first consider the vectors for attack that are present in the system. Figure 2 presents a breakdown of the principal attack vectors into three domains.

The first of these is the data-acquisition domain and is largely concerned with the physical security of sensing equipment. Given access, the attacker is able to adversely influence or gain access to structural measurement data. This includes both online data collection and training data collection. Vectors for attack include the placement of impostor sensing equipment, 'spoofing' sensors with adversarial inputs and eavesdropping on sensor data.

The second domain is related to data storage security. Threat models for generic data storage security have been previously explored in Hasan et al.[23] In the context of SHM, the principal threat is that training and model data might be maliciously accessed with the intent to construct adversarial examples. A data breach could arise as the result of either a physical or cyber intrusion and so hardening the system to these vectors is especially difficult.

The third domain is concerned with the implementation of the model. The principal threat is that knowledge of the structure and parameterisation of the model might become available to the attacker. Risks in this domain are largely concerned with insider threats, whereby a person with trusted access to the implementation and acquisition components of the system is able to act maliciously. However, other means of access, such as social engineering and espionage, are also vectors by which access to model implementation might be acquired. With total access to the system, an attacker would be trivially able to bypass safeguards and affect their machinations.

Threat taxonomy

In light of the earlier discussion, the principal distinction between adversarial attack types is the level of access that is afforded to the attacker. This is certainly the case in the machine learning literature where attacks are characterised as either white- or black-box.

In a *white-box* attack, it is assumed that the attacker has complete access to the system. This could include

model structure, parameters and gradient information, as well as access to the training data, inputs and outputs. This corresponds to access in all three domains of Figure 2. The white-box attack is primarily a simulation of the insider threat. The scenario can also apply to systems whereby the implementation has been made publicly available, for example, if the model had been published in the academic literature.

In a *black-box* attack, it is assumed that the attacker has query access to the model only, with no knowledge of the model structure, training procedure or access to a training dataset. There is some variation in exactly what is available to the attacker in the literature. Papernot et al.[24] assumed that a very small set (less than 10 per class) of examples from the input domain (but not necessarily the training data) are available. In an SHM context, we include scenarios whereby the attacker is able to query the model and has access to incoming data as black-box attacks.

The black-box attack is indicative of an outsider threat whereby an attacker is able to query the classifier either remotely via a cyber attack or by gaining access to models and data during a physical attack. In order for the black-box attack to be realistic, the number of queries to the classifier must be kept to a minimum. Guo et al.[25] argued that any black-box attack that makes use of many thousands of queries can easily be defeated by query limitation.

Consideration of both the white- and black-box scenarios is important for SHM applications. One question the SHM community is going to have to deal with is how to certify SHM systems for monitoring publicly owned critical infrastructure for life-safety applications. The importance of this issue will increase as SHM research has begun to mature and SHM systems begin to be sold commercially. The complexity of SHM systems is high enough that the community itself will need to provide public authorities guidance to guard against commercial entities selling malicious or sub-standard SHM systems. One way to guard against the use of sub-standard SHM systems is to mandate that the design of these systems be transparent. However, transparency is often not in the interest of commercial entities, and it does make white-box attacks more easy to execute. This tradeoff between security and ensuring the performance of SHM systems must be considered. Furthermore, as 5G networks become more prevalent, they will increasingly be used to implement SHM systems. A number of security risks have been identified with 5G networks (e.g. supply chain, interdependencies and increased overall attack surface) that make white-box attacks more plausible.[26] As a result, both white- and black-box scenarios should be considered by the SHM community.

Based on the original work on adversarial examples,[13] a great number of approaches have been developed for the construction of adversarial examples. The recent review by Yuan et al.[16] does an excellent job in recording and categorising the approaches that have thus far been proposed. A brief outline of the key definitions is included here.

Attack scope. An *example search* is a style of attack that performs optimisation in the input space of the SHM model to specify single or multiple adversarial examples ($\{x'\}$) that are optimised independently. This type of attack is most likely to be enacted as a one-time attack.

A *training attack* (also referred to as *data poisoning*) is conducted during the training phase of the SHM framework, before the system has been fully implemented. During a training attack, the training data are augmented or appended with adversarial examples with a view to maximise erroneous classification within the model itself.

A *universal example search* is an altogether different type of attack. Instead of optimising individual examples, the attack is a search for a function ($\mathcal{A}(\{x\})$) that transforms every input to an adversarial example

$$\mathcal{M}(\mathcal{A}\{x\}) = \mathcal{M}(\{x'\}) = y' \tag{5}$$

$$\mathcal{A} : y \rightarrow y' \tag{6}$$

This is often a more challenging task (as the mapping must span the input space) and is more dangerous to the operation of an SHM system as the adversarial examples can be delivered continuously in an online fashion.

Adversarial specificity. In a *targeted attack*, the adverse class labels are specifically chosen such that, for example, a damaged structure is selectively labelled as undamaged

$$\mathcal{A}_{\text{targeted}} : \quad \begin{matrix} y_+ \rightarrow y_- \\ y_- \rightarrow y_+ \end{matrix} \tag{7}$$

In an *untargeted attack*, there is no specific attention paid to which label is assigned to the adverse example as long as it is not the true class. In the binary classification case, this is equivalent to the targeted attack

$$\mathcal{A}_{\text{untargeted}} : y_i \rightarrow \{y_j\}, \quad i \neq j \tag{8}$$

Attack frequency. A *one-time attack* involves the specification and injection of adversarial examples without querying the classifier. Such attacks might be appropriate for attack motives that require only temporary falsification of the SHM system.

An *iterative attack* makes multiple queries to the classifier to assess the effectiveness of the adverse

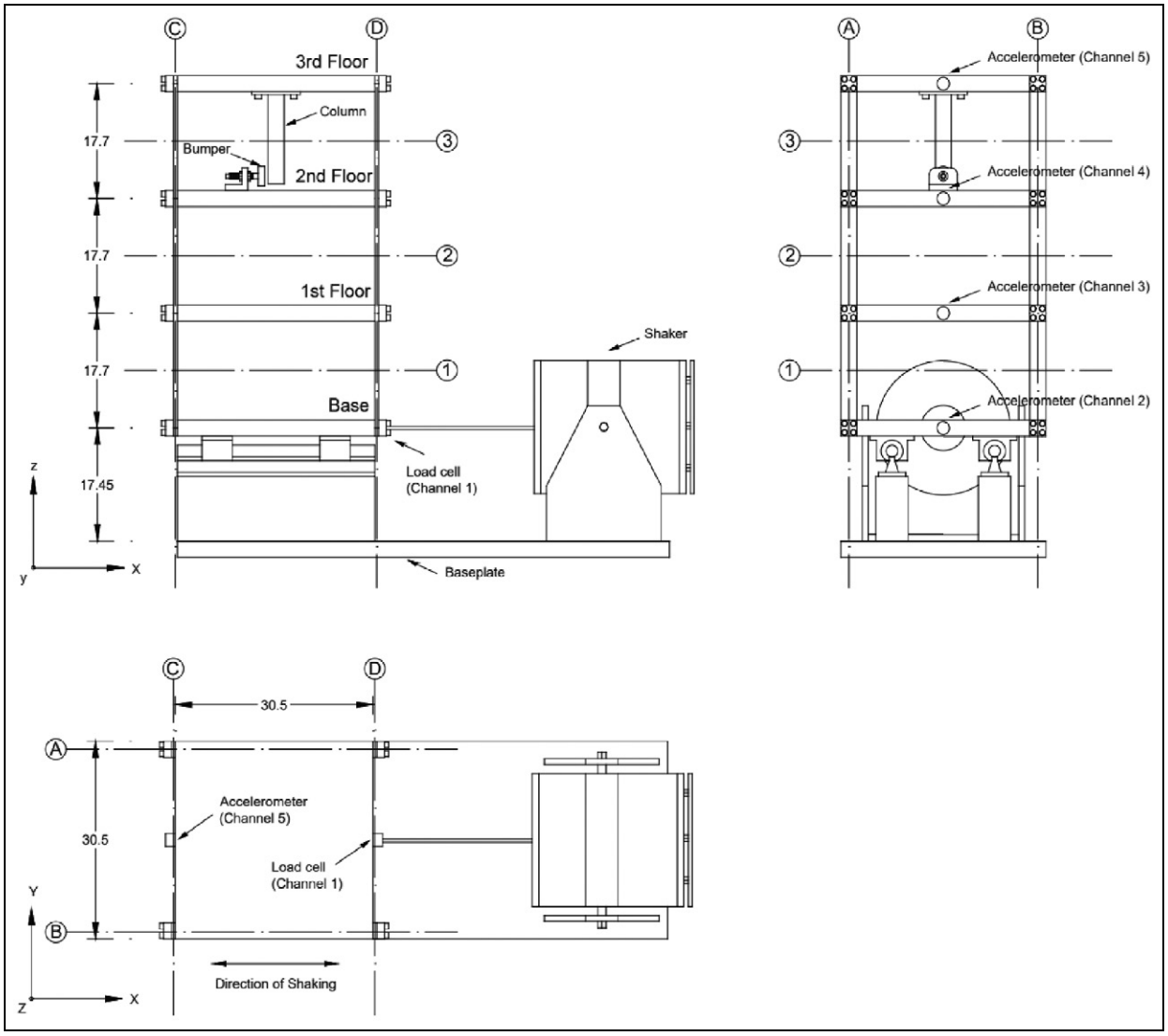

Figure 3. Schematic view of the three-storey structure (all dimensions are in cm).
Reproduced with permission from Figueiredo et al.[27]

examples, improving their potency iteratively. This attack style tends to result in more semantically convincing examples but comes at the cost of increased computational effort and risk of exposure.

Dataset and classification results

This section details the specification and training of a classification algorithm for the Los Alamos National Laboratory (LANL) three-storey structure dataset (Figure 3).[27] The primary objective of the classifier will be to accurately predict damage labels despite the presence of simulated environmental variation.

There are essentially two classes of algorithm available for performing this task. *Generative* models aim to construct the full joint probability of the labels and the data. The advantage is that the algorithm is able to return predictive probability distributions and so uncertainty in the predictions is handled graciously.

By comparison, *discriminative* models aim to reconstruct only the class conditional probabilities and so only the labels themselves can be returned. For this work, only discriminative models are considered.

LANL three-storey structure dataset

The three-storey structure was conceived as a test-bed for SHM algorithms. Many published approaches to SHM demonstrate their effectiveness on this dataset. This prominence in the literature makes the three-storey dataset ideal for demonstrating the vulnerability to adversarial attack. Another objective of the original

Table 1. The 17 structural states recorded in the LANL three-storey dataset.

Label	Damage condition	Description
0	Undamaged	Normal condition
1	Undamaged	Mass = 1.2 kg at the base
2	Undamaged	Mass = 1.2 kg on the 1st floor
3	Undamaged	87.5% stiffness reduction in column 1BD
4	Undamaged	87.5% stiffness reduction in column 1AD and 1BD
5	Undamaged	87.5% stiffness reduction in column 2BD
6	Undamaged	87.5% stiffness reduction in column 2AD and 2BD
7	Undamaged	87.5% stiffness reduction in column 3BD
8	Undamaged	87.5% stiffness reduction in column 3AD and 3BD
9	Damaged	Gap = 0.20 mm
10	Damaged	Gap = 0.15 mm
11	Damaged	Gap = 0.13 mm
12	Damaged	Gap = 0.10 mm
13	Damaged	Gap = 0.05 mm
14	Damaged	Gap = 0.20 mm and mass = 1.2 kg at the base
15	Damaged	Gap = 0.20 mm and mass = 1.2 kg on the 1st floor
16	Damaged	Gap = 0.10 mm and mass = 1.2 kg on the 1st floor

Reproduced with permission form Figueiredo et al.[27] Column references refer to the indices described in the original report and in Figure 3.

report on the dataset is to assess robustness in the face of environmental and operational variation. As such, there are 17 structure configurations (detailed in Table 1) representing either a damaged or undamaged state.

Damage is simulated in the structure by the inclusion of a bumper that acts between the second and third storeys of the structure. The impacting bumper adds significant nonlinearity to the dynamics of the structure. The gap, measured from the equilibrium point to the bumper, is varied to simulate the progression of damage. The bumper engages once the inter-storey displacement exceeds the gap distance. This means that smaller gap distances simulate increased levels of damage. A full account of the configuration of the structure in each of the 17 states is recorded in Table 1.

The dataset consists of the input force and acceleration responses measured at each storey of the structure. The structure is excited with a band-limited (20–150 Hz) Gaussian forcing signal by an electrodynamic shaker attached to the base. The data are recorded with 50 tests per state where each test consists of 8192 points recorded at a sampling frequency of 320 Hz. In order to increase the number of examples available for training the classifier, each of the tests are divided in half, resulting in 100 tests per state of 4096 points (1700 examples in total).

Classification approach

As state labels are available for the dataset, the classifier will be trained in a supervised manner. The extracted features will be the accelerance FRFs estimated by the Welch method (resulting in a real-valued spectral density) with no overlap, the Hanning window and five-fold averages. FRFs are computed for the base of the structure and each for of the three-storeys.

The FRF for the ith storey (x_i) is calculated from the accelerations (a_i) as

$$\{x_i\} = \frac{\mathcal{W}(\{a_i\})}{\mathcal{W}(\{a_0\})} \tag{9}$$

where $\mathcal{W}$ indicates the Welch operation as described earlier. The x_i are then concatenated and normalised (zero mean, unit variance) to form training vectors with 1640 points. This reduction in sampling points is due to the five-fold averaging in the Welch operation.

The use of FRFs as features is motivated by the observation that variations in the normal condition (masses, stiffnesses) exhibit more variance in the natural frequencies of the spectra, whereas the damage progression is most evident in the variance of the higher frequencies. The FRF is therefore a suitable feature as it is independently expressive in both of these directions.

For further motivation and in order to aid visualisation, Figure 4(a) and (b) depict the first two principal components of the dataset for the time series and the FRFs, respectively. The individual classes are clearly more separable in the FRF basis. In addition, variations in normal condition seems to be approximately orthogonal to the progression of damage. Figure 5 shows the first two principal components of the FRFs recoloured to depict damage labels only.

For the task of damage identification, a multi-layer perceptron (MLP) is trained with a single hidden layer. The structure of the classification model is detailed in

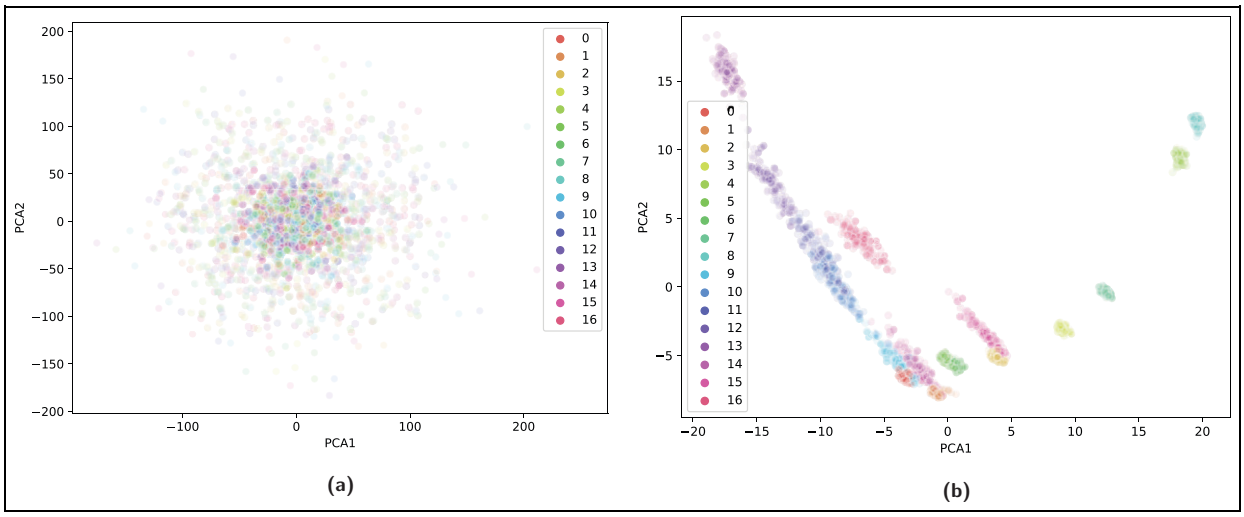

Figure 4. Comparison of label-sensitive features: the first two principal component directions plotted and coloured by label. Label clusters are visually far more separable in the FRF domain: (a) time series data and (b) FRF data.

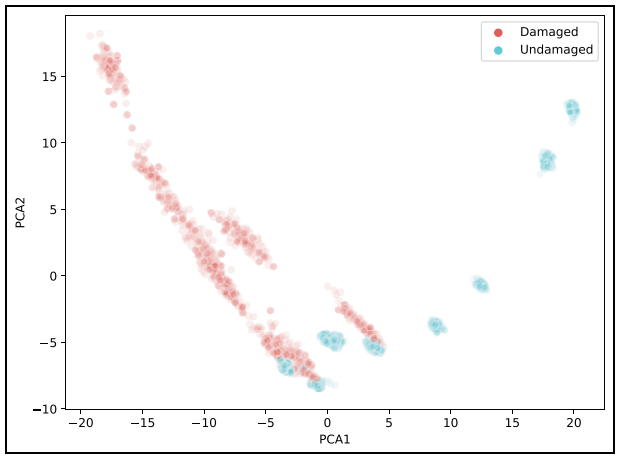

Figure 5. First two PCA components of the FRF data, recoloured to reflect damage labels.

Table 2. The dataset is divided into training and validation sets with 500 examples separately reserved for the evaluation of performance on unseen examples.

The network is initialised with random weights uniformly distributed on the interval [–1, 1] and then trained for up to 100 epochs using a cross-entropy loss function and the Adam optimiser. The hyperparameters for the optimiser are set to the default values provided in the original study.[28]

The training process is repeated for hidden node numbers in the range [1, 100]. Figure 6 depicts the training curves for training and validation sets. The figure clearly depicts a stable training regime with excellent validation performance on networks with greater than around 15 hidden nodes. Although many network

Table 2. Structure of the MLP used for damage identification.

Layer	Nodes	Activation
Input	1640	Linear
Hidden	16	Hyperbolic tan
Output	17	Normalised exponential

structures give optimal performance, in order to ensure our demonstration is as realistic as possible, the simplest model with the best validation accuracy is selected. This is a network that achieves a validation accuracy of 99.58% with 16 hidden nodes trained over 79 epochs. The structure of the classification model is detailed in Table 2.

Classification results

Figure 7 depicts the classification confusion matrix on the 500 unseen testing examples that were not used during training. The classifier achieves a multi-class prediction accuracy of 98.60% on the testing data, with only two (0.40%) false-positive and zero false-negative classifications. The remaining miss-classifications (1.00%) are between the three damage classes that have the smallest geometric differences in gap size.

Conventional analysis of the capacity of classification models such as the Vapnik–Chervonenkis (VC) dimension[29] places lower bounds on the number of training examples required to ensure generalisation. However, empirical evidence and recent studies[30] show that neural networks and deep learning models routinely outperform the theoretical limits placed on them.

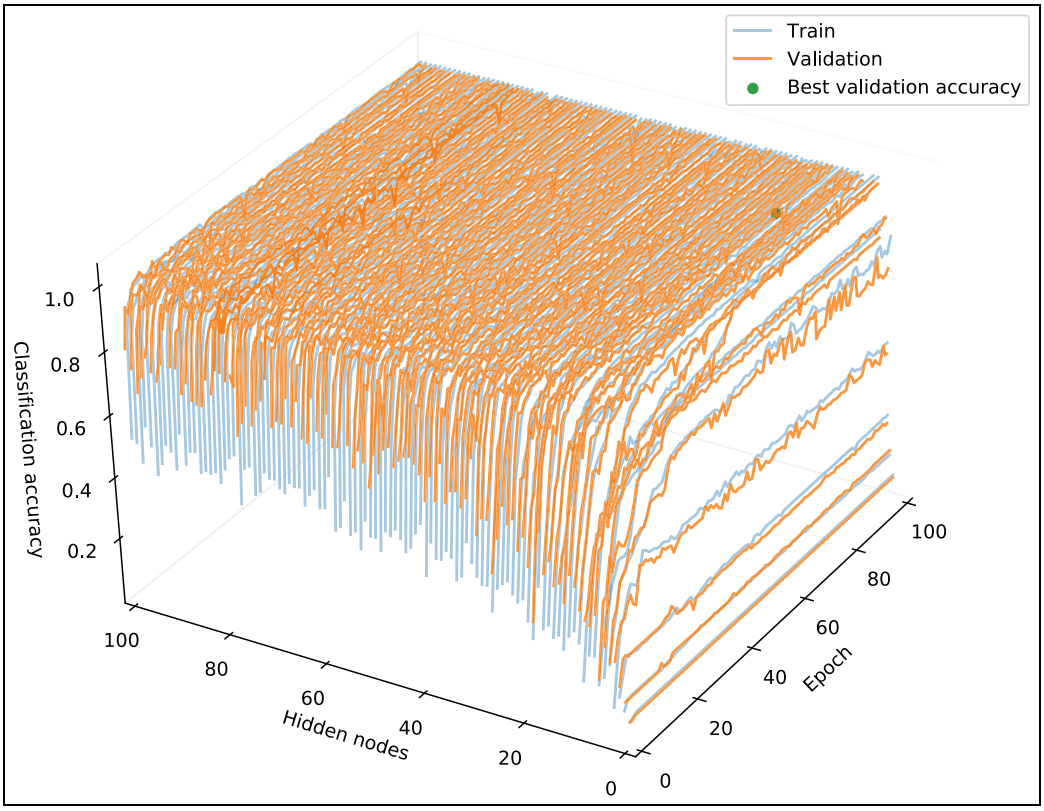

Figure 6. Training curves for the classifier, varying the number of hidden nodes.

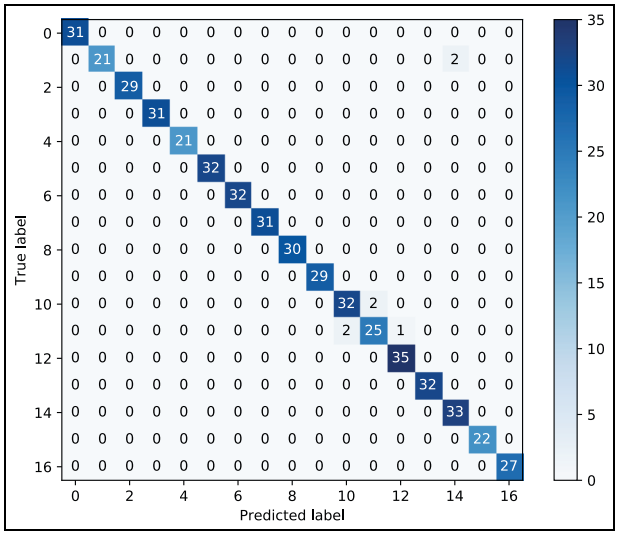

Figure 7. Confusion matrix of the classifier on unseen testing data.

Such models are often shown to achieve strong empirical generalisation despite enormous VC-dimensions and limited training data.

The dataset used here is small (1700 examples) compared to the number of examples in modern deep learning benchmark problems (typically on the order of 100,000 examples) and indeed the VC-dimension of the model (a simplified estimate would be $2^{n_h} = 65,536$ examples required, where $n_h = 16$, the number of hidden nodes). Despite this, the classifier is able to achieve high accuracy and good generalisation on unseen data. The authors therefore argue that this strong performance on unseen data is ample justification that the training data are sufficient to learn a general mapping from the data manifold to the labels.

Demonstration of adversarial vulnerability

The objective of this article is to motivate serious discussion into the vulnerability of data-driven SHM models by demonstrating the most threatening attacks in both the *white-box* and *black-box* threat scenarios. For brevity, we consider only the more challenging task of performing a universal example search. The goal is to identify an adversarial transformation $\mathcal{A}$ that produces an adversarial output for every true input, while maintaining semantic similarity between true and adversarial examples

$$\mathcal{A}(\{x\}, \theta_{\mathcal{A}}) = \{x'\} \tag{10}$$

Table 3. Parameters of the MLP used for adversarial perturbation.

Layer	Nodes	Activation
Input	1640	Linear
Hidden	200	Rectilinear unit
Output	1640	Linear

where $\theta_{\mathcal{A}}$ are the parameters of the adversarial transformation. Semantic similarity is judged by eye, in the FRF domain. This is conducted by comparing the adverse example $\{x'\}$ to examples from both its true class $\{x\}$ and the adverse class $\{x_{l'}\}$. An adverse example will be judged to have semantic similarity if it more closely resembles the true class while still inducing an adversarial classification.

For maximum impact, the demonstration is conducted as a *targeted attack*. The approach here is the more challenging task of aiming for perfect misclassification with every label representing either a specific false-positive or false-negative result. Adverse training labels are constructed by the following targeted scheme

$$\{l'\} = \begin{cases} l'_i = 0 & l_i \leqslant 8 \\ l'_i = 13 & l_i > 8 \end{cases} \qquad i \in [1, N] \tag{11}$$

where N is the number of training examples. A class label of zero corresponds to the normal condition, whereas a class label of 13 relates to the maximally damaged state. Constructing the adversarial target labels in this way ensures that $\mathcal{A}$ represents a transformation that maximises the rates of false positives and false negatives. The demonstrations here are also iterative attacks in that the parameters of the transformation are optimised over successive queries to the classifier.

Demonstration of white-box attack

The first demonstration of adversarial attack on an SHM implementation is conducted in a *white-box* manner. Since gradient information is available in this context, the parameters of the adversarial transformation $\theta_{\mathcal{A}}$ can be optimised via gradient descent. Several authors have presented efficient methods for conducting attacks in this context.[31,32] However, in order to emphasise vulnerability to naive attackers, the method used here is simplistic and consists of two distinct phases.

For the white-box demonstration, the adversarial transformation $\mathcal{A}$ is specified as an MLP with randomly initialised weights (the adversarial perturbation network (APN)). The first phase in the attack is the *listening phase* whereby $\mathcal{A}(\{x\}, \theta_{\mathcal{A}})$ is trained to reproduce the inputs from the training data.The purpose of this listening phase is to reduce the total number of queries that must be made to the target classifier. By pretraining the adversarial transformation to reproduce the inputs, an internal model of the input is constructed in the same manner as that of an auto-encoder. This internal model can then be optimised to produce adversarial examples during the second phase.

The second phase is the *learning phase* during which the network learns a perturbing transformation that maps true inputs to adversarial examples. The exact procedure is as follows. During the listening phase, an MLP (Table 3) with parameter set $\theta_{\mathcal{A}}$ is first trained to reproduce the inputs under a mean-squared error loss function

$$J'_{\text{listening}}(\theta) = \frac{1}{NL}\sum_{i}^{N}\sum_{k}^{L}(x_{ik} - x'_{ik})^2 \tag{12}$$

where $J'(\theta_{\mathcal{A}})$ is the adversarial loss. Next, the network is appended to the input of the classifier and trained on the adversarial labels to produce adversarial examples.

During the learning phase, training is conducted as a multi-objective problem with constraints placed on both the cross-entropy loss between the predicted and adversarial labels and the mean-squared error between the adversarial example and the true output

$$\begin{aligned} J'_{\text{learning}}(\theta_{\mathcal{A}}) = &-\frac{1}{N}\sum_{i}^{N}\sum_{j}^{M} y'_{ij}\log(\hat{y}'_{ij}) \\ &+ \frac{\lambda}{NL}\sum_{i}^{N}\sum_{k}^{L}(x_{ik} - x'_{ik})^2 \end{aligned} \tag{13}$$

where M is the number of classes in the dataset and L is the dimension of the input. The parameter λ is a hyperparameter that controls the weighting of the misclassification and mean-squared error reconstruction terms. Several approaches to this multi-objective optimisation problem are possible, including placing a minimum threshold on either of the terms. For these demonstrations, it is set to unity in order to give equal weighting to the classification and similarity terms. The parameters of $\theta_{\mathcal{A}}$ are then optimised via the Adam stochastic gradient descent algorithm using the classification training data.

In order to verify that the attack has been successful, the testing set of examples not used during adversarial training is fed through the perturbing network and classified. Figure 8 depicts the confusion matrix of the classifier on the adversarially perturbed validation examples. The confusion matrix represents 99.58% and 100% as false-negative and false-positive classification rates, respectively. In fact, only a single example was

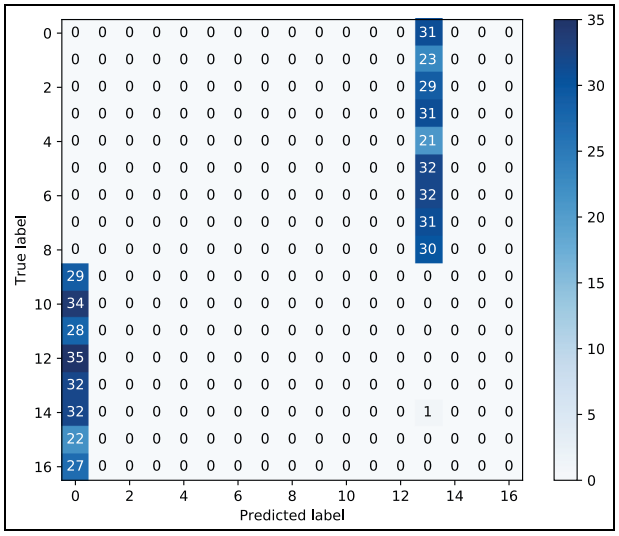

Figure 8. Confusion matrix of the classifier during a white-box attack.

correctly assigned a damage label by the classifier. The adversarial transformation has rendered the SHM classifier useless.

Figure 9 plots the resulting adversarial examples from the false-positive and false-negative cases, respectively. Looking at the adversarial examples, it can be seen that there is a high level of semantic similarity between the adversarial and true signals. This is especially pronounced in the false-positive examples which are almost indistinguishable from the unperturbed input. It seems that the largest semantic differences are present in the variance of the false-negative adversarial examples.

For intuition as to how the semantic structure of the signals remains intact during the perturbation, the first two principal component analysis (PCA) directions of the FRF data are plotted in Figure 10. Overlayed on the figure are 10 of the transformations from true inputs (dots) to adverse examples (crosses) coloured by damage label. In the figure, it can be seen that the perturbation is small in the PCA basis suggesting that the majority of the structure has been maintained.

Demonstration of black-box attack

For the black-box attack demonstration, it is assumed that the attacker only has access to the classifier on an input–output basis and has no knowledge of the inner workings of the algorithm. We permit the attacker access to a set of training examples and corresponding target labels, but crucially not the gradient information. While many black-box approaches to adversarial attack rely on the construction of a surrogate model in order to generate a synthetic classifier that can be attacked as a white-box, the approach utilised here is deliberately more naive. It is the reasoning of the authors that vulnerability to such naive attacks lowers the bar for adversarial attack and further motivates investigation into the ways in which SHM algorithms might be made more robust.

The aforementioned approach consists of two phases. During the listening phase a 1024-512-*n* deep auto-encoder (DAE)[33] is trained to learn a reduced-order representation of the inputs. While other more sophisticated reduced-order models (such as hierarchical models or modal analysis–based approaches) are clearly appropriate, the objective here is to demonstrate

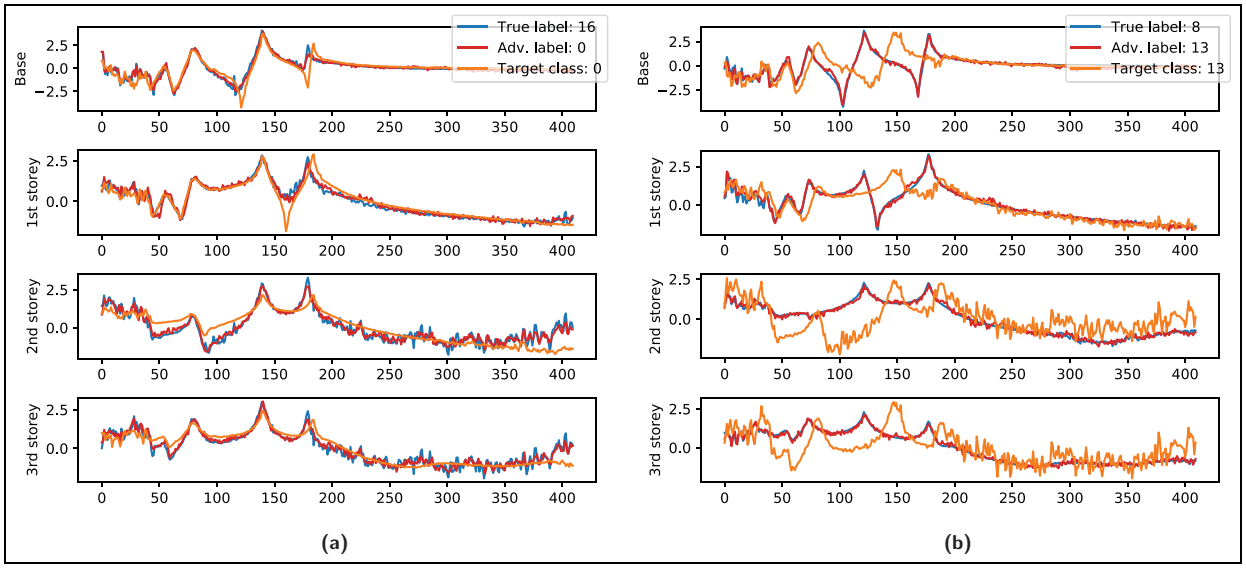

Figure 9. Adversarial examples arising from the white-box attack. The adversarial input is semantically similar and follows the overall structure of the true inputs more closely than the adversarial target signal. There is however a notable difference in the variance of the signal in the false-negative case: (a) false-negative example and (b) false-positive example.

a naive approach that makes little to no assumptions about the structure of the input. The DAE consists of two components, the encoding transformation ϕ and decoding transformation ψ

$$\begin{aligned}\phi(\{x\}) &= \{z\}\\ \psi(\{z\}) &= \{\hat{x}\}\end{aligned} \tag{14}$$

where $\{\hat{x}\}$ is the reconstructed signal. The parameters of the DAE are first trained under a mean-squared error loss using the same objective function as the white-box attack in equation (12). This step is critical for the black-box attack as it greatly reduces the number of parameters that must be optimised without gradient information.

Several authors have studied adversarial attacks on DAEs and other latent space models.[34,35] The approach shown here is most similar to that of Creswell et al.[36] in that the latent space is perturbed directly. However, while the authors specify an additive transformation for $\mathcal{A}(z)$, the approach here is a more general n-dimensional affine transformation, where n is the size of the latent encoding. In the latent encoding, the transformation is

$$\{z'\} = A\{z\} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & a_{1n+1} \\ a_{21} & a_{22} & \dots & a_{2n} & a_{2n+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} & a_{nn+1} \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \\ 1 \end{bmatrix} \tag{15}$$

During the learning phase, the DAE is prepended to the classifier and the perturbing transformation is inserted between the encoder and the decoder. The forward adversarial transformation is now given by

$$\{\hat{x}'\} = \psi(A\phi(\{x\})) \tag{16}$$

As before, this is a multi-objective optimisation problem that seeks to maximise both semantic similarity and miss-classification. The learning phase objective function is now given by

$$\begin{aligned} J'_{\text{learning}}(\theta_A) = &-\frac{1}{N}\sum_{i}^{N}\sum_{j}^{M} y'_{ij}\log(\hat{y}'_{ij}) \\ &+ \frac{\lambda}{NL}\sum_{i}^{N}\sum_{k}^{L}(x_{ik} - \hat{x}'_{ik})^2 \end{aligned} \tag{17}$$

where $\{\hat{x}'\} = \psi(A\phi(\{x\}))$ and N, M and L are the number of examples, classes and input dimensions, respectively. However, there are now two hyperparameters that must be specified. These are the size of the DAE latent encoding n and weighting term λ which is set to unity.

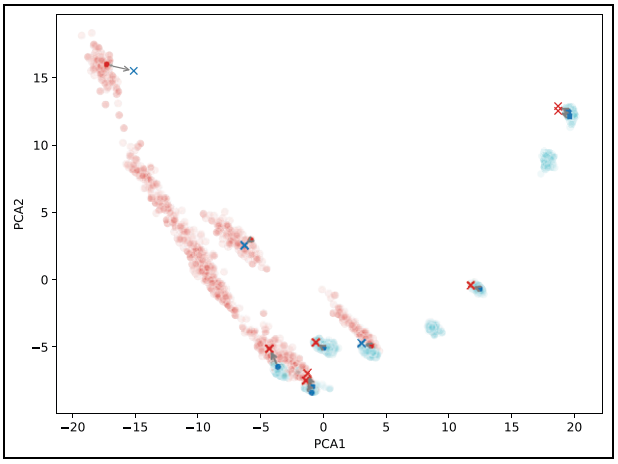

Figure 10. First two PCA directions of training and validation data (coloured by damage label) as well as latent representation of adversarial perturbation. Arrows show mapping from true examples (dots) to adversarial examples (crosses).

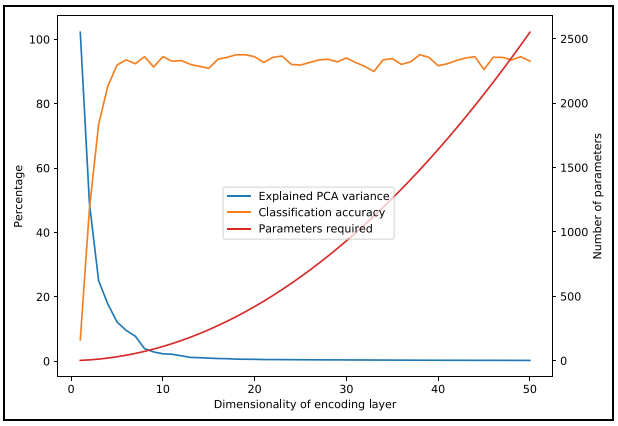

Figure 11. Explained variance of PCA representation, plotted alongside the reconstructed classification accuracy and parameters in the affine transformation for differing values of n (based on the figure, the value $n = 10$ is selected).

The size of the latent encoding represents a trade-off between accurate reconstruction of the inputs and the complexity (number of parameters) of the affine transformation. More parameters in the latent encoding will result in higher fidelity representations of the inputs, but will require more parameters to be optimised in order to find the adversarial transformation. The number of parameters in $\mathcal{A}$ grows as $n^2 + n$ and so choosing a lower value of n is preferable.

In order to select a value for n, the explained variance of the first 50 PCA components are plotted alongside the classification accuracy of the reconstructed signal in Figure 11. Although not directly related, the PCA explained variance plot affords intuition into the number of parameters that are required to provide an

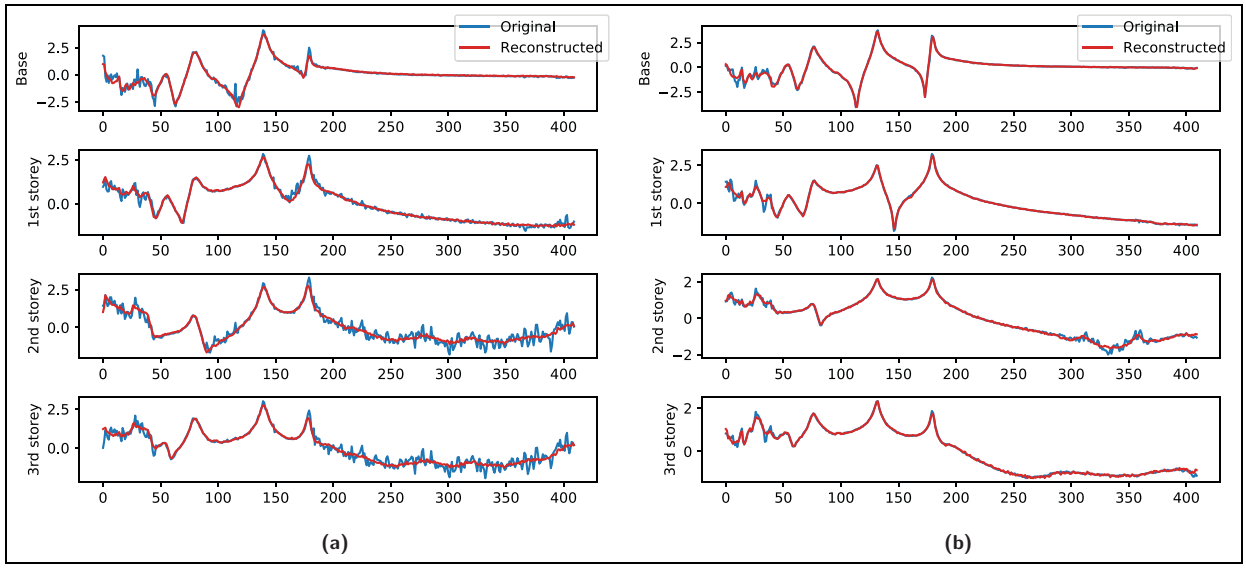

Figure 12. Original inputs and DAE reconstructions after the learning phase. The DAE has learned to represent the structure of the data well but is struggling to reproduce the increased variance in the higher frequencies found in the damaged signals: (a) damaged case and (b) undamaged case.

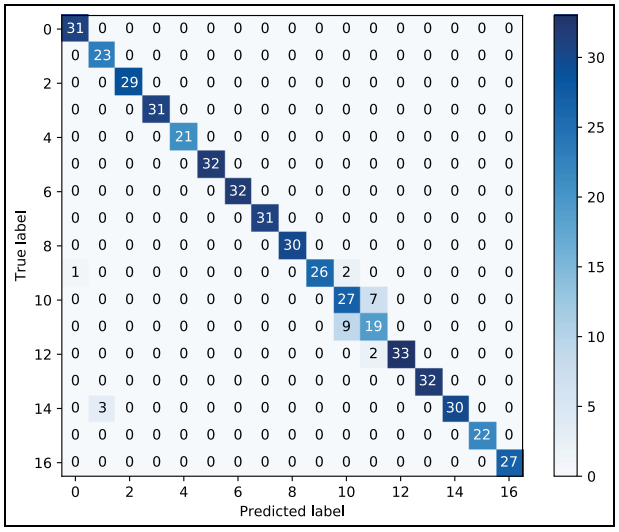

Figure 13. Confusion matrix of the classifier predictions on the DAE reconstructed signal before adversarial perturbation.

accurate representation of the input. It can be seen from the figure that only a small number of parameters are responsible for the majority of the variance in the inputs (this is expected from highly structured data such as FRFs). The classification accuracy does not seem to be affected above values of 5. With this in mind, a conservative value of $n = 10$ is selected.

Figure 12 depicts the input and reconstruction for the trained DAE. It can be seen in the figure that the DAE has done a good job of representing the overall structure of the data but struggles to emulate the variance seen in the higher frequencies of the damaged examples. Nevertheless, this representation is clearly able to capture the damage-sensitive nature of the FRFs as the classification confusion matrix in Figure 13 is largely unchanged.

Table 4. Hyperparameters of SADE optimiser.

Parameter	Notation	Value
Mutation parameter mean	*Fm*	0.5
Crossover parameter mean	*Cr*	0.5
Learning period	*Lp*	10

The parameters of the adversarial transformation $\theta_{\mathcal{A}}$ are then optimised to produce adversarial examples. As gradient information is not available, a heuristic algorithm must be used. In this study, optimisation is achieved via a self-adapting differential evolution (SADE) scheme, with a loss function defined in equation (17). The hyperparameters of the SADE optimiser are presented in Table 4, and these parameters have been selected based on the authors' experience with the method. Other than these parameters, the implementation of the SADE algorithm is as described in the original paper.[37] As SADE is an adaptive optimiser, it is already largely insensitive to the selection of the initial values of the learning, mutation and crossover parameters. Presentation of the full iterative scheme is beyond the scope of this current work and the interested reader is directed to the original paper for details.

The parameters of 100 trial transformations are optimised over 10 runs from an initial uniform distribution

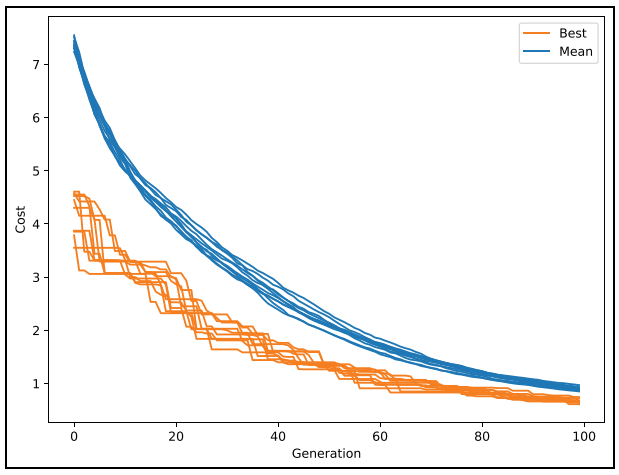

Figure 14. Convergence history of the SADE optimiser over 10 runs.

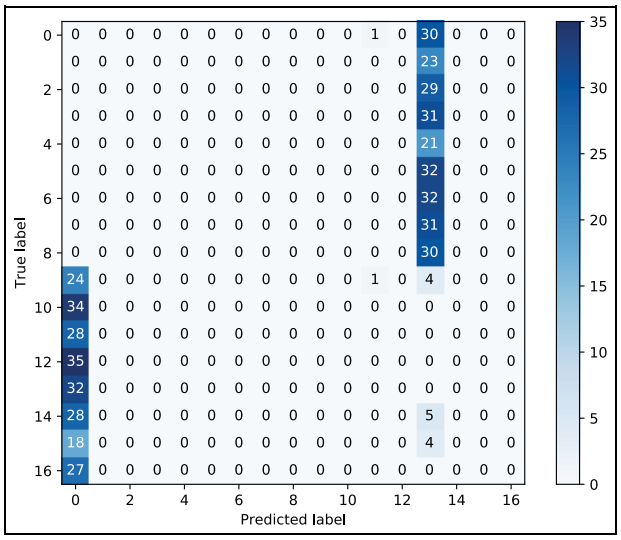

Figure 15. Classification confusion matrix resulting from the black-box attack.

on the interval $[-1,\ 1]$. The SADE algorithm is allowed to proceed for 100 iterations per run. After each run, the convergence history is plotted to verify that the optimiser has been successful in finding a global minima. Figure 14 shows the best and the mean cost function values plotted for each run. The lowest cost transformation over the 10 runs is then selected and evaluated on the unseen examples as before.

Figure 15 depicts the classification confusion matrix on the unseen data. Although not as successful as the white-box attack, the latent space perturbation has still managed to induce false-positive and false-negative classifications in almost every case. Figure 16 depicts the adversarial examples generated in the black-box attack from the same unperturbed inputs used to generate the examples in Figure 9. However, it is immediately obvious that the black-box attack has failed to produce adversarial examples that maintain semantic similarity to the true inputs. The overall structure of the input has been conserved during the attack and the signals still clearly resemble FRFs. However, the adversarial examples more closely resemble members of the target class, meaning that the chances of such examples fooling a human observer is low.

Figure 17 depicts the PCA directions of the dataset for the first 10 adversarial transformations as in

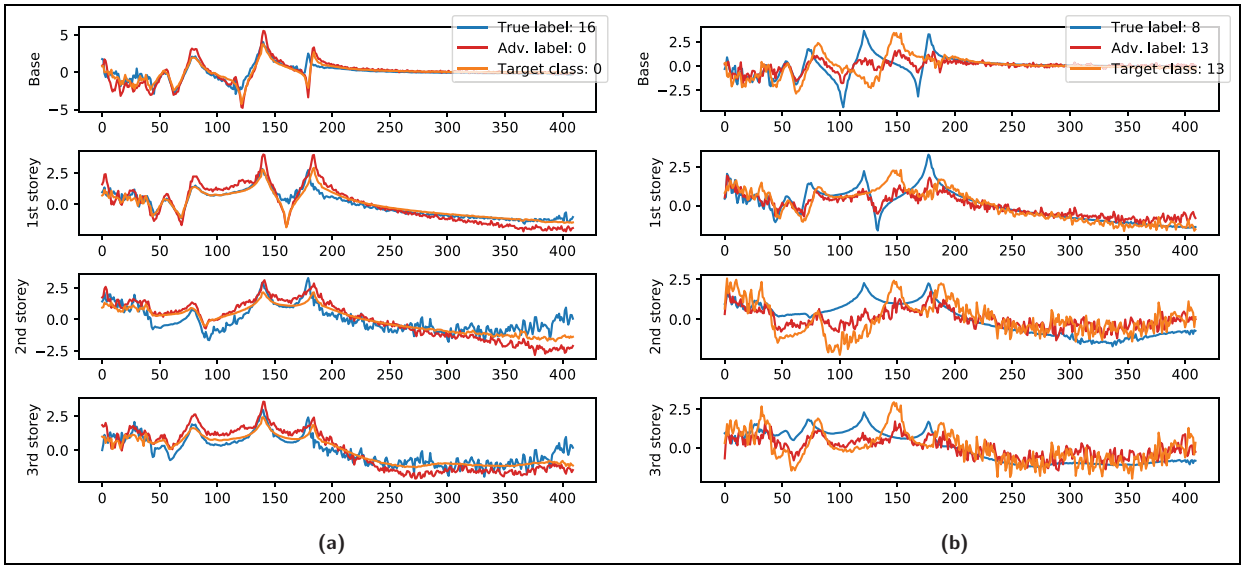

Figure 16. Adversarial examples results arising from a black-box attack. The adversarial examples (red) are semantically more similar to the target class (orange) than the true input (blue): (a) false-negative example and (b) false-positive example.

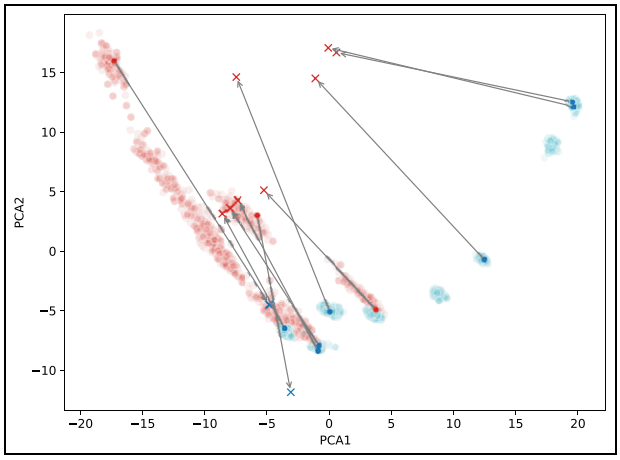

Figure 17. First two PCA directions of training and validation data (coloured by damage label) as well as latent representation of adversarial perturbation (arrow from true to adversarial).

Figure 10. In this figure, the reason for the loss of semantic similarity might be explained. The perturbations in the latent space have translated to large shifts in the PCA projection, suggesting that the adversarial examples are structurally quite dissimilar to the true inputs from which they were constructed. Another possible explanation is a poor correlation between latent and classification spaces. The nonlinearity introduced by the decoding layer of the DAE has the effect of exaggerating small changes made in the latent space into large changes in the adversarial examples. Such an effect would make optimisation on the parameters of $\mathcal{A}$, extremely difficult as the resulting objective surface would be highly non-smooth. A future approach to manage this phenomenon could be to use a linear transformation (such as PCA) into a reduced-order space.

Despite these results, some semantically convincing examples can still be constructed by a variant of the latent attack method described earlier. By optimising the parameters of $\mathcal{A}$ to transform a single example at a time (example search attack) and tuning the λ parameter in the objective function, the adversarial examples in Figure 18 are generated. While not the result of a universal example search, these examples maintain better semantic similarity to the inputs than those shown earlier. Visually the examples do not show significant change in natural frequency or damping information and it is conceivable that a human observer would be fooled by these examples.

Although these examples were cherry-picked from a large group of potential examples as visually more threatening, the authors would point out that an attacker would also be free to make such a selection. In fact, query limitation would be the only thing preventing an attacker from generating many thousands of trial examples and selecting only the most potent for malicious deployment. In an SHM context, whereby even a small number of false positives (and an even smaller number of false negatives) are enough to render a framework useless, an approach of this type represents a serious threat.

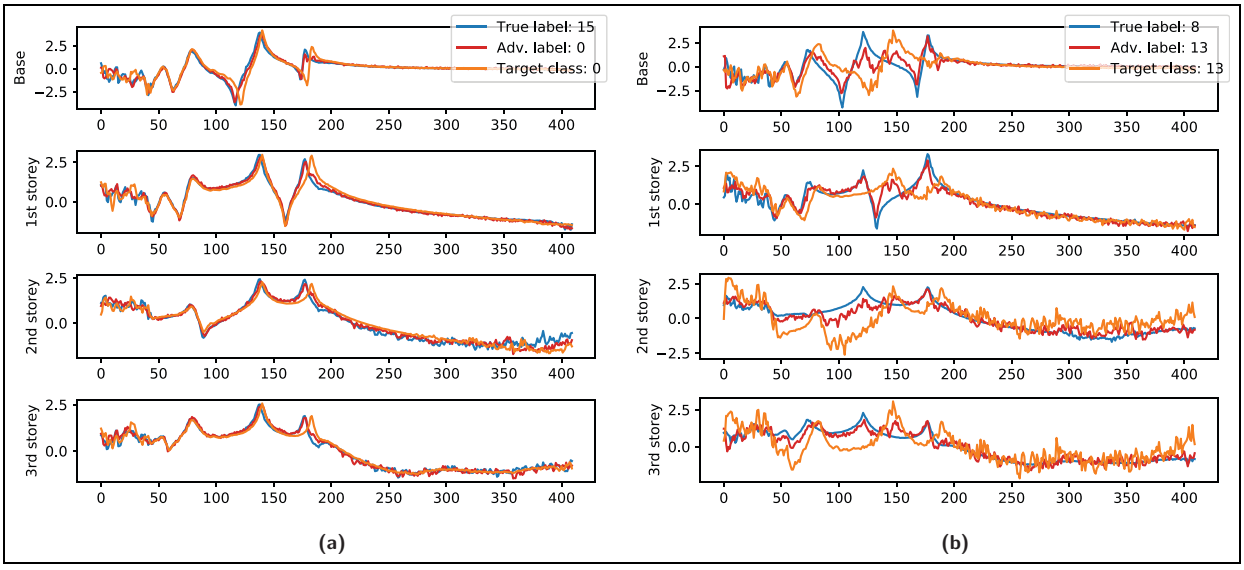

Figure 18. Adversarial examples results arising from the second black-box attack, selected for presentation based on a visual assessment of semantic similarity. The adversarial input is semantically similar and follows the overall structure of the true inputs more closely than the adversarial target signal: (a) false-negative example with $\lambda = 80$ and (b) false-positive example with $\lambda = 40$.

Adversarial defence strategies for SHM

Several adversarial defence strategies have already been presented in the machine learning literature and the development of new approaches is a very active area of research. The development of the techniques has thus far been centred largely around image classification tasks and have often been presented in an ad hoc manner, whereby any one strategy is only effective against a subset of attack types and vise versa.

Of the defences thus far presented, two prominent themes are distillation[38] and adversarial training[21,39] type defences. While distillation has been shown to be effective against some common attacks, it has also been shown to be insecure against attacks deliberately designed to circumvent the method.[40] Adversarial training defences generate adversarial examples during the model training process, with the hope that the trained model will develop robustly. While promising progress has been made, the application of this technique in an SHM context is likely to be limited by the availability of damaged training data.

Recently, Ilyas et al.[41] have presented a new approach related to adversarial training based on the theory that adversarial examples arise due to 'non-robust features' that are present in the dataset. Their approach utilises a penalty term in the objective function that punishes models that rely on the non-robust features.

Another promising direction is the investigation into generative models. Li et al.[42] have recently suggested that generative models may be more robust to adversarial attacks. They present adversarial defence strategies that are able to make use of the full joint distribution in order to detect adverse examples. This is a promising result for SHM, as many modern approaches utilise models of this type, for example, Bayesian networks.[43] However, Gilmer et al.[44] have shown that the presence of adversarial examples in high-dimensional datasets is (at least in a simplified spherical case) independent of the model used for classification.

Conclusion

In this work, it has been possible to demonstrate the serious vulnerability of an SHM classifier to adversarial attack. A universal example search, has been shown to construct semantically convincing adversarial examples that are able to fool an SHM classifier with almost perfect testing accuracy. This was achieved in the white-box threat scenario, which is the more realistic for SHM applications. Although failing to replicate this feat, a naive black-box attack has also been able to construct individual adversarial examples.

Clearly, robustness to adversarial attack is an open problem that presents a real challenge to the robustness of machine learning applications. This work demonstrates that data-driven SHM methods are not exempted. In order to facilitate further investigation, an adversarial attack threat model for SHM has been proposed. It is hoped that this will serve as a platform for future discussion surrounding the security implications of data-driven SHM approaches. The authors believe that adversarial attack robustness will emerge as a key challenge in the widespread deployment of SHM frameworks.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship and/or publication of this article: M.D.C. would like to recognise support from EPSRC (grant no. EP/L016257/1). The research presented in this paper was partially supported by the Laboratory Directed Research and Development Programme of Los Alamos National Laboratory under the Information Science and Technology Institute as part of the 2019 Adversarial Machine Learning Challenge.This work was supported in part by the US Department of Energy through the Los Alamos National Laboratory. Los Alamos National Laboratory is operated by Triad National Security, LLC, for the National Nuclear Security Administration of US Department of Energy (contract no. 89233218CNA000001). This study was also financed in part by the Coordenacao de Aperfeicoamento de Pessoal de Nivel Superior – Brasil (CAPES; Finance code 88881. 190499/2018-01).

ORCID iDs

Max David Champneys https://orcid.org/0000-0002-3037-7584
Moisés Silva https://orcid.org/0000-0001-7897-3978
David Mascarenas https://orcid.org/0000-0002-8548-9401

References

1. Worden K, Farrar CR, Manson G, et al. The fundamental axioms of structural health monitoring. *Proc Roy Soc A Math Phys Eng Sci* 2007; 463(2082): 1639–1664.
2. Worden K and Manson G. The application of machine learning to structural health monitoring. *Philos Trans Roy Soc A Math Phys Eng Sci* 2006; 365(1851): 515–537.
3. Deraemaeker A and Worden K. *New trends in vibration based structural health monitoring*, vol. 520. Berlin: Springer Science + Business Media, 2012.
4. Martinez-Luengo M, Kolios A and Wang L. Structural health monitoring of offshore wind turbines: a review through the statistical pattern recognition paradigm. *Renew Sustain Energ Rev* 2016; 64: 91–105.

5. Park G, Rutherford AC, Sohn H, et al. An outlier analysis framework for impedance-based structural health monitoring. *J Sound Vib* 2005; 286(1–2): 229–250.
6. Manson G, Worden K and Allman D. Experimental validation of a structural health monitoring methodology: part III. Damage location on an aircraft wing. *J Sound Vib* 2003; 259(2): 365–385.
7. Bornn L, Farrar CR, Park G, et al. Structural health monitoring with autoregressive support vector machines. *J Vib Acoust* 2009; 131(2): 021004.
8. Neves AC, Leander J, González I, et al. An approach to decision-making analysis for implementation of structural health monitoring in bridges. *Struct Control Health Monit* 2019; 26(6): e2352.
9. Farrar CR and Worden K. *Structural health monitoring: a machine learning perspective*. Hoboken, NJ: John Wiley & Sons, 2012.
10. Finley B and Schneider EA. ICAS: the center of diagnostics and prognostics for the United States Navy. In: *Proceedings of the component and systems diagnostics, prognosis, and health management*, Orlando, FL, 20 July 2001, vol. 4389, pp. 186–193. Bellingham, WA: International Society for Optics and Photonics.
11. Fraser K. An overview of health and usage monitoring systems (HUMS) for military helicopters. Technical Report, Defence Science and Technology Organisation, Melbourne, VIC, Australia, 1994.
12. Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets. In: *Proceedings of the advances in neural information processing systems*, 2014, pp. 2672–2680, https://arxiv.org/pdf/1406.2661v1.pdf
13. Szegedy C, Zaremba W, Sutskever I, et al. Intriguing properties of neural networks, 2013, https://arxiv.org/abs/1312.6199
14. Papernot N, McDaniel P and Goodfellow I. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples, 2016, https://arxiv.org/pdf/1605.07277.pdf
15. Carlini N and Wagner D. Adversarial examples are not easily detected: bypassing ten detection methods. In: *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 2017, pp. 3–14. ACM, https://arxiv.org/pdf/1705.07263.pdf
16. Yuan X, He P, Zhu Q, et al. Adversarial examples: attacks and defenses for deep learning, 2019, https://arxiv. org/pdf/1712.07107.pdf
17. Sarkar S, Reddy KK, Giering M, et al. Deep learning for structural health monitoring: a damage characterization application. In: *Proceedings of the annual conference of the prognostics and health management society*, 2016, pp. 176–182, https://www.researchgate.net/publication/308971489_Deep_learning_for_structural_health_monitoring_A_damage_characterization_ application
18. Cha YJ, Choi W and Büyüköztürk O. Deep learning-based crack damage detection using convolutional neural networks. *Comput Aid Civ Infrastruct Eng* 2017; 32(5): 361–378.
19. Zapico J, Gonzales M and Worden K. Damage assessment using neural networks. *Mech Syst Signal Pr* 2003; 17(1): 119–125.
20. Specht F, Otto J, Niggemann O, et al. Generation of adversarial examples to prevent misclassification of deep neural network based condition monitoring systems for cyber-physical production systems. In: *Proceedings of the 2018 IEEE 16th international conference on industrial informatics (INDIN)*, Porto, 18–20 July 2018, pp. 760–765. New York: IEEE.
21. Madry A, Makelov A, Schmidt L, et al. Towards deep learning models resistant to adversarial attacks, 2017, https://arxiv.org/abs/1706.06083
22. Farrar CR, Doebling SW and Nix DA. Vibration–based structural damage identification. *Philos Trans Roy Soc A Math Phys Eng Sci* 2001; 359(1778): 131–149.
23. Hasan R, Myagmar S, Lee AJ, et al. Toward a threat model for storage systems. In: *Proceedings of the 2005 ACM workshop on storage security and survivability*, 2005, pp. 94–102. ACM, https://www.researchgate.net/publication/221103837_Toward_a_threat_model_for_storage_systems
24. Papernot N, McDaniel P, Goodfellow I, et al. Practical black-box attacks against machine learning. In: *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 2017, pp. 506–519. ACM, https://arxiv.org/pdf/1602.02697.pdf
25. Guo C, Gardner JR, You Y, et al. Simple black-box adversarial attacks, 2019, https://arxiv.org/abs/1905.07121
26. NIS Cooperation Group, European Union. EU coordinated risk assessment of the cybersecurity of 5G networks, https://www.politico.eu/wp-content/uploads/2019/10/Report-EU-risk-assessment-final-October-9.pdf
27. Figueiredo E, Park G, Figueiras J, et al. Structural health monitoring algorithm comparisons using standard data sets. Technical Report, Los Alamos National Laboratory (LANL), Los Alamos, NM, 1 March 2009.
28. Kingma DP and Ba J. Adam: a method for stochastic optimization, 2014, https://arxiv.org/abs/1412.6980
29. Vapnik VN and Chervonenkis AY. On the uniform convergence of relative frequencies of events to their probabilities. In: Vovk V, Papadopoulos H and Gammerman A (eds) *Measures of complexity*. Cham: Springer, 2015, pp. 11–30.
30. Zhang C, Bengio S, Hardt M, et al. Understanding deep learning requires rethinking generalization, 2016, https://arxiv.org/pdf/1611.03530.pdf
31. Goodfellow I, Shlens J and Szegedy C. Explaining and harnessing adversarial examples. In: *Proceedings of the international conference on learning representations*, 2015, https://www.researchgate.net/publication/269935591_Explaining_and_Harnessing_Adversarial_Examples
32. Papernot N, McDaniel P, Jha S, et al. The limitations of deep learning in adversarial settings. In: *Proceedings of the 2016 IEEE European symposium on security and privacy (EuroS&P)*, Saarbrucken, 21–24 March 2016, pp. 372–387. New York: IEEE.

33. Hinton GE and Salakhutdinov RR. Reducing the dimensionality of data with neural networks. *Science* 2006; 313(5786): 504–507.
34. Tabacof P, Tavares J and Valle E. Adversarial images for variational autoencoders, 2016, https://arxiv.org/abs/1612.00155
35. Kos J, Fischer I and Song D. Adversarial examples for generative models. In: *Proceedings of the 2018 IEEE security and privacy workshops (SPW)*, San Francisco, CA, 24 May 2018, pp. 36–42. New York: IEEE.
36. Creswell A, Bharath AA and Sengupta B. Latent Poison–adversarial attacks on the latent space, 2017, https://www.researchgate.net/publication/320944629_LatentPoison_-_Adversarial_Attacks_On_The_Latent_Space
37. Qin AK and Suganthan PN. Self-adaptive differential evolution algorithm for numerical optimization. In: *Proceedings of the 2005 IEEE congress on evolutionary computation*, Edinburgh, 2–5 September 2005, vol. 2, pp. 1785–1791. New York: IEEE.
38. Papernot N, McDaniel P, Wu X, et al. Distillation as a defense to adversarial perturbations against deep neural networks. In: *Proceedings of the 2016 IEEE symposium on security and privacy (SP)*, San Jose, CA, 22–26 May 2016, pp. 582–597. New York: IEEE.
39. Shafahi A, Najibi M, Ghiasi A, et al. Adversarial training for free! 2019, https://arxiv.org/abs/1904.12843
40. Carlini N and Wagner D. Defensive distillation is not robust to adversarial examples, 2016, https://nicholas.carlini.com/papers/2016_defensivedistillation.pdf
41. Ilyas A, Santurkar S, Tsipras D, et al. Adversarial examples are not bugs, they are features, 2019, https://arxiv.org/pdf/1905.02175.pdf
42. Li Y, Bradshaw J and Sharma Y. Are generative classifiers more robust to adversarial attacks? 2018, https://arxiv.org/pdf/1802.06552.pdf
43. Fuentes R. *On Bayesian networks for structural health and condition monitoring*. PhD Dissertation, The University of Sheffield, Sheffield, 2017.
44. Gilmer J, Metz L, Faghri F, et al. Adversarial spheres, 2018, https://arxiv.org/abs/1801.02774