Damage Detection and Localization for Condition Assessment

Shaik Althaf Veluthedath Shajihan[1], Aditya Chanodia[2], Viswarup Misra[3]

***Abstract*— Being able to detect the progressive damages well in advance is one of the primary goals of structural health monitoring of bridges and buildings, as it can not only assist in preventing the loss of infrastructure, but can also help to save lives. However, it is pretty challenging to perform real-time processing and robust decision making with acceleration time-history data if we have a limited number of sensors and if high level of environmental noise is present during the data acquisition. In this study, we deal with these challenges using multiple Machine Learning based techniques, namely, Vector Auto-Regressive(VAR) model, Hidden Markov model (HMM) and Recurrent Neural network (RNN). We also illustrate the benefits and drawbacks of each of these methods in terms of test accuracy and computational efficiency. A raw ambient vibration time-history data is used for training, testing, and predicting the state of the structure. If any damage gets detected, then we proceed to locate the damage on the building and eventually quantify the intensity of the damage.**

I. INTRODUCTION

Structural Health Monitoring (SHM) is an approach used for civil, mechanical, and aerospace engineering to reduce the irreparable deterioration, damage and maintenance costs, while, at the same time improving the safety aspects of the system. Damage is by nature a local phenomenon in the initial stages and is not well-defined [1]; Hence, SHM tends to adopt a scheme to distribute a dense-sensor network throughout the structure for acquiring maximal information of the structural dynamics. Despite numerous research works to solve this nature of problem, along with successful results, there are still limitations in the actual real-life application to full-scale infrastructures. Altogether, there is a growing effort to come up with robust techniques that can be applied to wide variety of real-life SHM problems.

In this study, an open-source data set is used which provides the ambient vibration time-history records of the set of experiments conducted on a 4-storey , 1:4 scaled building frame model, illustrated in Figure 1,

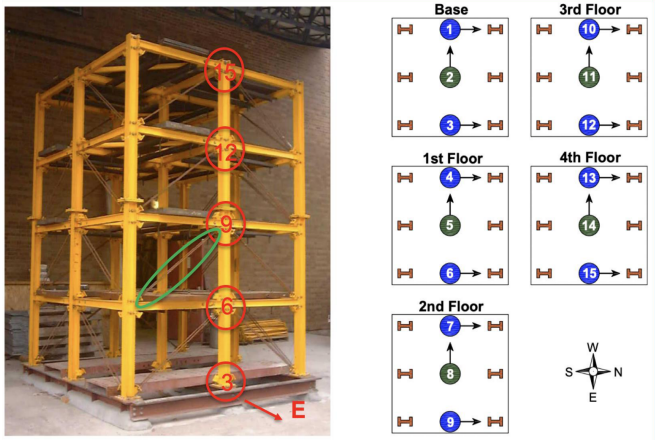

Fig. 1. Scaled building model instrumented with sensors

with 15 accelerometers installed on the structure. The acceleration data was acquired at a sampling rate of 200 Hz for a time frame of 300 secs, giving 60,000 samples for each of the damage conditions considered for the structure. Note that the three sensors mounted at the base of the building model are not considered in this study as it lacked relevant information about the dynamics of the structure. Contrary to the common practise in signal processing to take the time-history signal data to time-frequency based domains for developing ML models, we work with the raw-time signal data in all the models considered in this study. This approach is adopted for a few reasons, the damage scenarios considered for the building are not significant enough to cause notable changes in short-term Fourier transform (STFT) domain which losses time-resolution of data for classification[2]. Even though, few researchers have used Wavelet[3] and Hilbert–Huang transform (HHT)[4] based empirical mode decomposition techniques to tackle this type of problem. Also,this enables the possibility of direct classification of time series features without dimensionality reduction and information loss. The data set consists of ambient vibration reading for different states of the structure which was simulated by inducing structural changes on the frame. The damage scenarios included removal of bracing ties from the east-facing exterior

[1]Ph.D. Student, Department of Civil Engg., sav4@illinois.edu
[2]M.S. Student, Department of Statistics, adityac3@illinois.edu
[3]M.S. Student, Department of Statistics, vmisra3@illinois.edu

Table 1
Damage scenarios of the second phase of the IASC-ASCE structure.

Case no.	Condition	Description
1	Healthy	Fully braced configuration
2	Damaged	Removing the braces of all floors from the east side
3	Damaged	Removing the braces of all floors from one bay on the southeast corner
4	Damaged	Removing the braces of the first and fourth floors from one bay on the southeast corner
5	Damaged	Removing the braces of the first floor from one bay on the southeast corner

Fig. 2. Simulated Healthy and Damage cases for Training and Testing data [5]

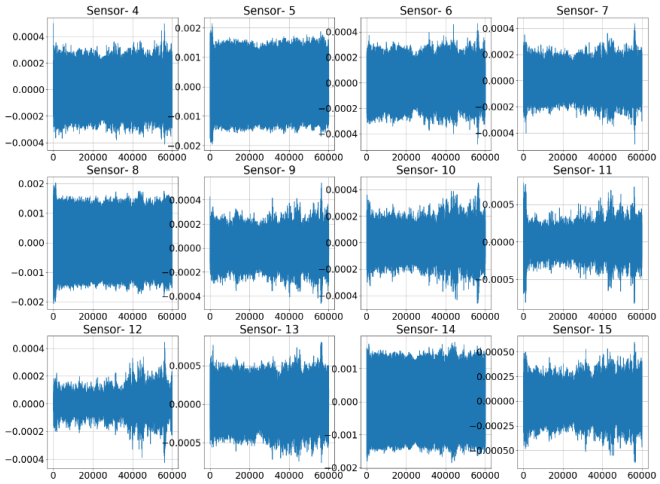

Fig. 3. Healthy case ambient vibration signal data for 12 sensors

side of building and loosening of bolts at different beam-column junctions. Table 1 depicts the healthy and damaged scenarios simulated in the experimental setup conducted by the researchers. The vibration signal of the 12 sensors used as representative of the Healthy condition of the building model is represented in Figure 3.

We adapt three different approaches to classify and localize damage in the model based on any given test sequence of signals. Firstly, a linear VAR model is used to incorporate the spatial distribution of sensors in the structure. The inter-dependency between the data acquired by different sensors is captured in the coefficient matrix fit by the VAR model on the training data. Subsequently, the model is utilized to classify the test signal sequence as belonging to any one of the damage cases and the Mahalanobis distance measure is used to estimate the level of anomaly for localization of feature. Secondly, a Hidden Markov Model is used for classification and locatization. A Hidden Markov Model classifies data by learning the statistical properties of the underlying process. HMM is a stochastic approach which easily accomodates the uncertainities in structural health monitoring problems[3]. Thirdly, a Convolutional Neural Network Long Short-Term Memory (CNN-LSTM) model is used. This is basically an LSTM architecture that utilizes Convolutional Neural Network (CNN) layers for feature extraction on input data, followed with LSTMs to support sequence prediction[6].

The methods are discussed in detail in the subsequent sections, highlighting the pros and cons of the adopted strategies.

II. Vector Autoregressive Model

The VAR model, unlike the standard Auto-Regressive (AR) model, takes into account the inter-dependency between different variables together with the influence of past input, to predict the time-step ahead. In our case, modelling the vibration response at multiple locations of the same system with the spatial information of sensors can be achieved using the coefficient transformation matrix in VAR model. This allows us to potentially extract damage sensitive features local to each sensor to aid the localization task of the problem.

In the pre-processing stage, the raw time-history data is standardized with respect to the mean and variance of individual signals and de-trended using linear regression. The waveform is checked for stationarity with the Dickey–Fuller test (ADF) statistic before proceeding with the VAR model. The data from the sensors was divided into smaller time-signal sequences with a sliding window providing 50% overlap, such that each sequence consists of 800 data points, allowing us to fit multiple VAR models rather than just one VAR model over the entire recorded vibration data.

The order of the VAR model is determined by minimizing the AIC criterion and the p-value statistic of residuals to ensure uncorellation. This led to the use of 10 number of lags in this study for a robust estimation and prediction of time-series, and the coefficients were calculated using ordinary least squares regression. The dataset was split into 80% and 20% for training and testing of VAR model respectively. The VAR model was fit onto the smaller sequences of training data for all the 4 damage cases considered in Table 1 and the healthy case, and evaluated over the test sequence. The least error norm was used as an index for classification of the test data in one of the damage classes, or to predict it as a healthy class. The confusion matrix

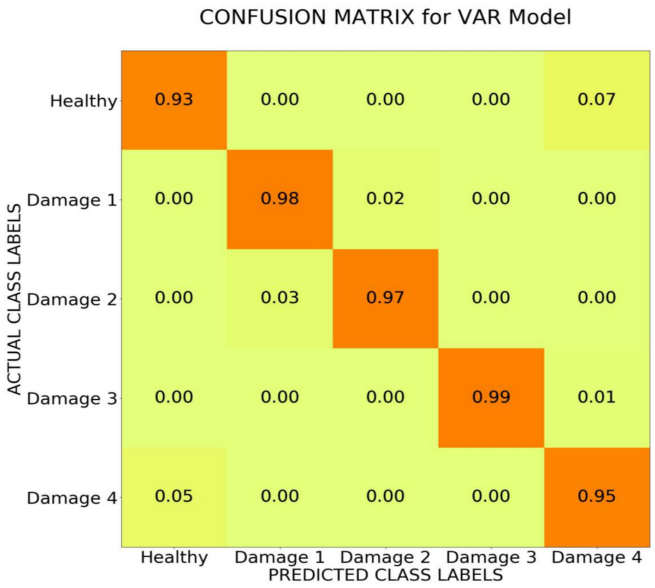

Fig. 4. Confusion matrix for classification using VAR model

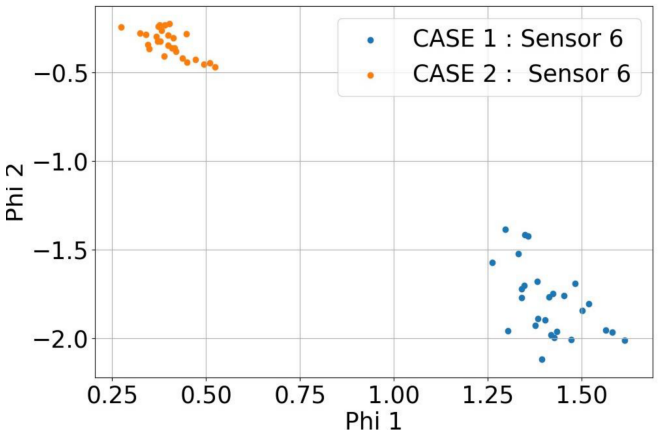

Fig. 5. VAR coefficient of Sensor 6 in Healthy & Damage case 1

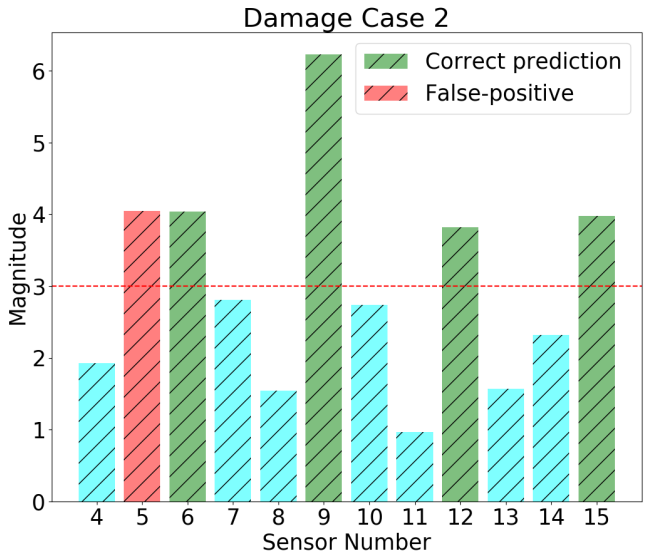

Fig. 6. Localization for Damage Case 2 using VAR Model

obtained for classification using VAR model, depicted in Figure 4, shows a good score with more than 95% for all the cases.

For the localization task a damage sensitive feature corresponding to each sensor was identified as the diagonal components of the coefficient matrix for the lags considered in model. In our case the coefficient matrix was of size 12×12 ,for which the diagonal coefficients corresponding to each sensor, $\phi_{ii,1...n}$,in a damage case were compared with its corresponding value in Healthy case using the Mahalanobis distance metric for anomaly detection. For illustration, using only 2 lags, the coefficient distribution of sensor 6 for Healthy and damage case 1 is shown in Figure 5. This allows us to fit a multivariate Gaussian distribution on the cases compared and the magnitude of distance obtained for all sensors between compared cases is taken as a measure of intensity of damage to spatially locate the sensor number. The damage location identification is performed using Fisher criterion, Equation 1, which normalizes the different case scenarios to get the level of deviation of damage feature from Healthy case.

$$f_{cr} = \frac{(\mu_{damage} - \mu_{healthy})^2}{\sigma^2_{damage} + \sigma^2_{healthy}} \quad (1)$$

Subsequently, it's assumed that the sensors closest to actual damage location in the building would indicate the locations of damage. Performing the localization for test data using VAR model indicate all the sensors closest to damage are identified correctly, but in cases 2 and 3 , a false-positive sensor also exceeds the threshold,$(\mu + 1.\sigma)$ set to indicate damage location. Figure 6 depicts the false positive (sensor 5, in red color) detected in addition to the other expected sensors (green color) using VAR model. Further, to improve the localization results, we implement ML models with probabilistic and non-linear nature in the following sections, but at the cost of computational efficiency.

III. Hidden Markov Model

The Hidden Markov Model or HMM is specified by a set of parameters, namely, π, A, B, where π is an initial probability distribution over states. π_i denotes the probability that the Markov chain will start in state i (π is a N dimension vector). A is an NxN state transition probability distribution matrix, where each

a_{ij} represents the probability of moving from state i to state j. B is a matrix of emission probabilities, each element of B expresses the probability of an observation being generated from a state i. The HMM uses the parameters π, A, B to stochastically model finite-length sequences of observations.

The training data is utilized to learn the parameters of HMM. Given the data, we compute the maximum-likelihood estimate of the parameters using the Baum-Welch algorithm, which iteratively maximizes the likelihood. For classification purpose, we use Equation 2 given below. It classifies a test signal y to a damage class m.

$$m = argmax_{m\epsilon 1,\ldots,M} logp(y|\theta^m) \tag{2}$$

Here θ is the parameter of the model for damage class m, which is obtained after performing the Baum Welch algorithm.

We model the data in two ways. Firstly, we build Gaussian Mixture HMM models for each case, thereby getting five models. Each model consists of three states, with two Gaussians per state and a diagonal covariance matrix. Every model is fit on the raw time series data, where each sensor is considered as a feature for the model. The model is trained using 80% of the data and is then evaluated on the remaining 20% test data, which is further splitted into smaller segments to measure the model's accuracy. Classification is performed according to the maximum log probability. The test data is classified as belonging to the class for which model's log probability is maximum. Figure 7 shows the confusion matrix for the same.

In the second approach, we consider each time step to be a hidden state and for each hidden state we have an observable, which is the acceleration data obtained from the sensors. In this particular problem, the hidden states are the four damage cases and the healthy case.

We use supervised hidden Markov model[7] to estimate the parameters. Our aim to predict the hidden state given the observation data is achieved with the help of Viterbi algorithm. Each sensor is individually modelled using a Gaussian HMM which gives a total of twelve models. For example, for sensor 9, the model is able to correctly classify the three cases (Healthy, Damage Case 1 and Damage Case 2), as illutrated in figure 8. This approach works well for most of the cases, but, in some scenarios, where two classes are very similar, the model fails to classify them accurately. One such example is when the model tries to classify between healthy case and damage case 4. The only difference

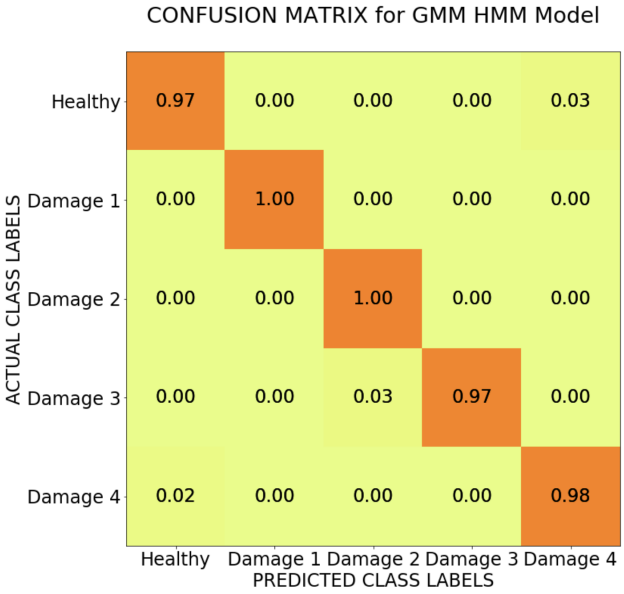

Fig. 7. Confusion matrix for classification using GMM HMM model

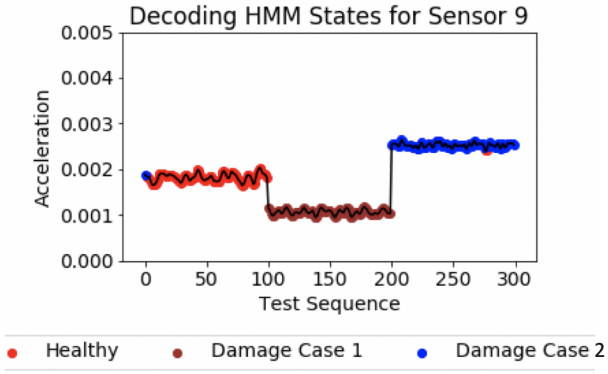

Fig. 8. Decoding hidden states given the observation using HMM model

between the two is that for damage case 4, we are removing only the braces of the first floor from one of the bays.

For the localization purpose, we use GMM HMM with three states and two Gaussians per state, and then calculate the L1 norm between the means of healthy case and a given damage case for each sensor. The sensors with differences more than the threshold (mean + one standard deviation) are supposed to contribute mostly towards characterizing the damage. A plot of Damage Case 2 and Damage Case 3 is shown in Figure 9. The model correctly localizes the Damage Case 2 but gives a false positive for Damage Case 3.

Since HMM is a first order model, the current state of the model depends only on the previous state. This

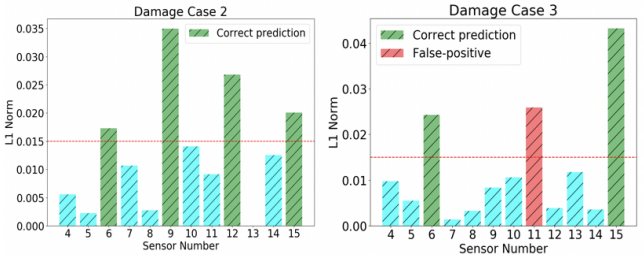

Fig. 9. Localization for Damage Case 2 and Damage Case 3 using HMM model

```
# CNN LSTM model
model = Sequential()
model.add(TimeDistributed(Conv1D(filters=64, kernel_size=3,
activation='relu'),
                          input_shape=(None,subseq_length,num_features)))
model.add(TimeDistributed(Conv1D(filters=64, kernel_size=3,
activation='relu')))
model.add(TimeDistributed(Dropout(0.5)))
model.add(TimeDistributed(MaxPooling1D(pool_size=2)))
model.add(TimeDistributed(Flatten()))
model.add(LSTM(100))
model.add(Dropout(0.5))
model.add(Dense(100, activation='relu'))
model.add(Dense(n_outputs, activation='softmax'))
```

Fig. 10. CNN-LSTM Model Architecture

makes HMM a shallow ML model. Every step in HMM is a linear transformation and the model parameters remains same with time. These limitations make HMM a restrictive model. We, therefore, move to neural networks, which, owing to their complexity and non-linearity, might help to further improve the performance for damage localization.

IV. Recurrent Neural Network

In traditional neural networks, all the inputs and outputs are independent of each other. Recurrent neural networks, also known as RNNs, are a class of neural networks that allow previous outputs to be used as inputs while having hidden states. We can think of the "hidden state" as the memory of the network that captures information about what happened in all the previous time steps.

Unlike a traditional deep neural network, which uses different parameters at each layer, a RNN shares the same parameters across all steps. This reflects the fact that we are performing the same task at each step, just with different inputs and, hence, greatly reduces the total number of parameters we need to learn.

However, a traditional vanilla RNN cannot process very long sequences of data if we use *tanh* or *relu* as an activation function. Therefore, we use Long Short Term Memory Network(LSTM) instead, which is a variation of RNN that is able to learn and remember over long sequences of input data[8]. In concept, an LSTM recurrent unit tries to “remember” all the past knowledge that the network is seen so far and tries to “forget” irrelevant data. LSTMs are intended for use with data that is comprised of long sequences of data, up to 200 to 400 time steps and hence seem to be a good fit for this problem.

A. Network Architecture

For this problem, we train a Convolutional Neural Network Long Short Term Memory or CNN-LSTM model which involves using CNN layers for feature extraction on input data combined with LSTMs to support sequence prediction[9]. The CNN-LSTM model reads in subsequences of the main sequence as blocks, extracts features from each block, and then allows the LSTM to interpret the features extracted from each block.

Our approach to implement this CNN-LSTM model was to split each window of 200 time steps into four subsequences of 50 time steps for the CNN model to process. We then define a CNN model that expects to read in sequences with a length of 50 time steps and 12 features. We use two consecutive CNN layers followed by dropout and a max pooling layer.

The entire CNN model is wrapped in a TimeDistributed layer, which is basically a layer wrapper that allows us to apply a layer to every temporal slice of an input, to allow the same CNN model to read in each of the four subsequences in the window. The extracted features are then flattened and provided to the LSTM model to read and extract its own features, before a final mapping to an activity is made.

B. Classification and Localization

The model was able to classify the test data with more than 98% accuracy for all the five cases, as can be seen from figure 11. For localization, we used *Permutation Importance*[10]. The permutation importance is calculated after a model has been fitted and the technique works on a simple principle: If we randomly shuffle a single feature in the data, leaving the target and all others in place, how would that affect the final classification performance of the model? In this problem, data from every sensor is taken as a feature. In contrast to the simple LSTM model that we implemented earlier for this study, which localized three out of four damages correctly, the CNN-LSTM model was able to excel and localize all four damages

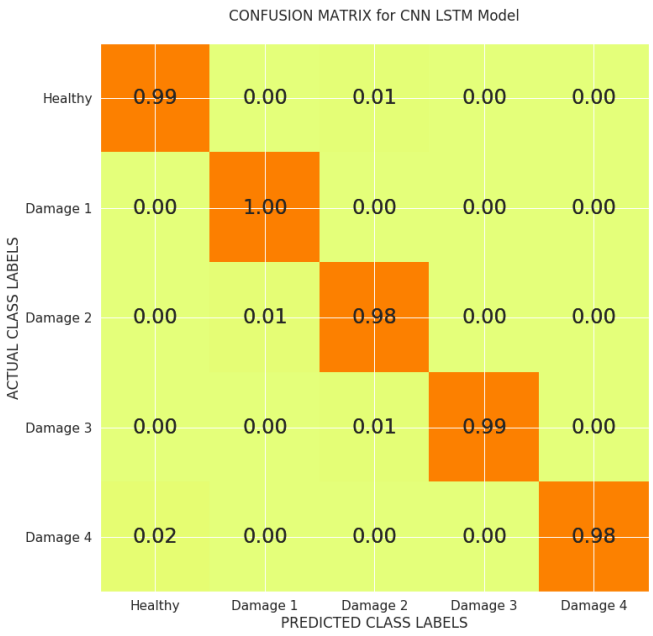

Fig. 11. Confusion matrix for classification using CNN-LSTM model

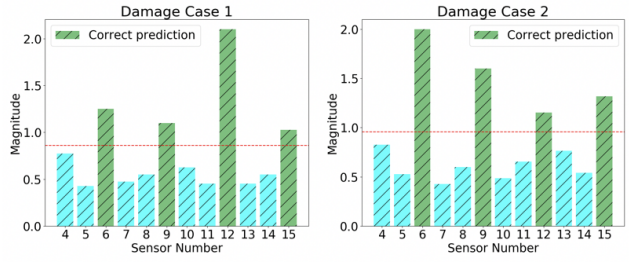

Fig. 12. Localization for Damage Case 1 and Damage Case 2 using CNN-LSTM model

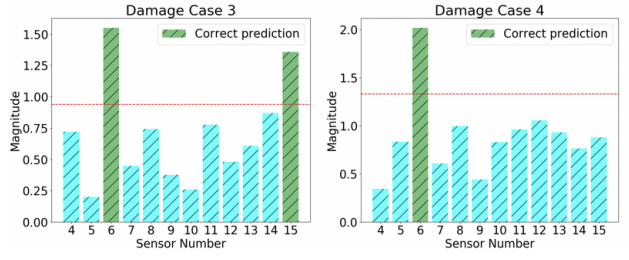

Fig. 13. Localization for Damage Case 3 and Damage Case 4 using CNN-LSTM model

correctly. For instance, in case of damage 1, where all the braces from east side of the building were removed, the model exhibited highest difference in the performance when we shuffled the data for sensors 6, 9, 12, and 15, which matches with the ground truth. Similarly, for damage case 3, where the braces of the first and fourth floor were removed from one bay on the southeast corner of the building, shuffling the data for sensors 6 and 15 resulted in lowering the model's accuracy the most.

Thus, the CNN-LSTM model worked the best out of all the ML models that we implemented for this study, but at an expense of computational cost.

V. CONCLUSIONS

The three ML based algorithms helped to predict the state of a structure using the raw time-history data and are able to localize the damage even with the limited number of sensors instrumented on the building. All the models explored in this study are able to correctly classify the state of the structure as belonging to Healthy or one of the damage cases. The linear VAR model captures the spatial information of sensors and is able to localize damage using the Mahalanobis distance for anomalous data, but with false-positives in some damage cases. HMM, which is extensively used in speech recognition and modelling of other temporal processes, was able to classify the five cases correctly, given its characteristic to capture the underlying hidden process responsible for emitting the dependent observations, but gave some false-positives during damage localization. The CNN-LSTM Neural network combines the time-series modelling's non-linear capabilities with the stochastic nature of HMM, thereby aiding us in not only classifying the damages correctly, but also identifying the right locations for all the damage cases.

ACKNOWLEDGEMENT

We would like to express our gratitude towards Prof. Paris Smaragdis for giving us the much needed bigger picture of Machine Learning for Signal Processing. We also extend our sincere gratitude to Shrikant Venkataramani for helping strengthen our conceptual framework for time-series modelling using Machine Learning methods through his office hours, Piazza comments and discussions with us on our project.

REFERENCES

[1] C. R. Farrar and K. Worden, *Structural Health Monitoring.: A Machine Learning Perspective*. John Wiley & Sons, 2012.

[2] Y. Lei, J. Lin, Z. He, and M. J. Zuo, "A review on empirical mode decomposition in fault diagnosis of rotating machinery," *Mechanical systems and signal processing*, vol. 35, no. 1-2, pp. 108–126, 2013.

[3] R. Rammohan and M. R. Taha, "Exploratory investigations for intelligent damage prognosis using hidden markov models," in *2005 IEEE international conference on systems, man and cybernetics*, vol. 2, pp. 1524–1529, IEEE, 2005.

[4] H. Chen, Y. Yan, and J. Jiang, "Vibration-based damage detection in composite wingbox structures by hht," *Mechanical systems and signal processing*, vol. 21, no. 1, pp. 307–321, 2007.

[5] A. Entezami and H. Shariatmadar, "Structural health monitoring by a new hybrid feature extraction and dynamic time warping methods under ambient vibration and non-stationary signals," *Measurement*, vol. 134, pp. 548–568, 2019.

[6] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4580–4584, IEEE, 2015.

[7] J. Du, J. S. Rozowsky, J. O. Korbel, Z. D. Zhang, T. E. Royce, M. H. Schultz, M. Snyder, and M. Gerstein, "A supervised hidden markov model framework for efficiently segmenting tiling array data in transcriptional and chip-chip experiments: systematically incorporating validated biological knowledge," *Bioinformatics*, vol. 22, no. 24, pp. 3016–3024, 2006.

[8] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "Lstm fully convolutional networks for time series classification," *IEEE Access*, vol. 6, pp. 1662–1669, 2017.

[9] R. Zhang, Y. Liu, and H. Sun, "Physics-guided convolutional neural network (phycnn) for data-driven seismic response modeling," *arXiv preprint arXiv:1909.08118*, 2019.

[10] A. Altmann, L. Toloşi, O. Sander, and T. Lengauer, "Permutation importance: a corrected feature importance measure," *Bioinformatics*, vol. 26, no. 10, pp. 1340–1347, 2010.