Transactions of the Korean Nuclear Society Spring Meeting
Jeju, Korea, May 9-10, 2024

Near-real-time Seismic Damage Identification Using Graph Attention Network

Minkyu Kim
Korea Atomic Energy Research Institute
mkkim92@kaeri.re.kr

Junho Song
Seoul National University
junhosong@snu.ac.kr

Chul-Woo Kim
Kyoto University
kim.chulwoo.5u@kyoto-u.ac.jp

***Keywords: Graph attention network, Damage identification, Near-real-time, Seismic loads**

1. Introduction

Civil infrastructure is a critical foundation of modern societies, essential to economic growth and societal welfare. However, they are frequently exposed to natural disasters and man-made hazards, which may result in significant economic losses and potentially tragic loss of life. Therefore, it is critical to assess the post-disaster integrity of the structural system and detect potential damage through a proper inspection process.

To achieve this, structural health monitoring (SHM) technologies have been developed. This approach enables real-time identification of potential structural problems to guide maintenance efforts. Vibration-based damage identification is an important technique within SHM, as it can detect structural damage early after catastrophic events by observing changes in vibration characteristics.

The integration of vibration analysis techniques with advanced pattern recognition methods, such as deep neural networks (DNNs; LeCun et al., 2015), has expanded the capabilities of vibration-based damage identification. DNNs are algorithms designed to recognize patterns in large datasets. Changes in vibration characteristics can be identified in real-time by efficiently analyzing and interpreting the collected data using DNNs that leverage features derived from vibration signals.

Although DNN-based damage identification methods have significant advantages and potential, they cannot fully exploit the spatial information in vibration signals. Therefore, this paper focuses on a graph neural network (GNN) that can effectively utilize the spatial information by learning the graph-structured data. By modeling the structural system as a graph, GNNs can provide a more detailed representation of structural dependencies, thereby improving the accuracy of damage identification.

However, identifying damage through vibration data poses a challenge for traditional GNNs due to their use of a static graph representation, which does not account for the dynamic nature of structural responses. To address this issue, this paper proposes to use a graph attention network (GAT) that processes graph-structured data by focusing on the relationships between nodes. The GAT utilizes an attention mechanism to dynamically allocate weights to the importance of information gathered from neighboring nodes. This allows us to obtain the topology of the structural system directly from the data. The introduction of the GAT ensures a more accurate representation of the spatial correlations between sensors, considering the dynamic nature of the structural response.

The proposed GAT architecture for damage identification consists of the following three main parts:

1. An “encoder” consists of multiple GAT layers to learn latent variables of the graph-structured input data considering the spatiotemporal information. These layers assimilate data nonlinearity and complex proximity information, and allow the model to dynamically adjust attention weights based on the relationships between nodes.
2. A “graph structure decoder” captures the spatial characteristics of the structural system from latent features by reconstructing the adjacency matrix. This process uses the error between the input and reconstructed adjacency matrix to identify potential structural anomalies based on spatial data.
3. A “node feature decoder” focuses on temporal aspects by reconstructing the structural response from latent features using one-dimensional (1D) convolutional layers. This part identifies anomalies by analyzing the reconstruction error of the structural response.

The loss function for the GAT model is defined based on reconstruction errors and serves as the structural damage index (SDI). The model is trained using the structural responses and adjacency matrix of the target structure in an intact state. Following the training process, the damage of each member is identified in near-real time by calculating the SDI at each time step. The proposed framework is demonstrated through a numerical example of a 3D frame structure under seismic load.

2. Theoretical background for graph neural networks

2.1. Graph representation and graph neural network

A graph $\mathbf{G}$ is denoted by $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ is the set of nodes in which element v_i represents node i, and $\mathbf{E}$ is the set of edges in which element e_{ij} denotes the edge connecting v_i and v_j. The state of $\mathbf{V}$ is described by a node feature matrix $\mathbf{X} \in \mathbb{R}^{N\times F}$, which is composed of the node feature vectors of a length of F associated with each of N nodes. The structural information of the graph is represented by an adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ in which an element $A_{ij} = 1$ if v_i and v_j are connected, otherwise $A_{ij} = 0$. The degree matrix $\mathbf{D} \in \mathbb{R}^{N \times N}$ of a graph is a diagonal matrix where each diagonal element $D_{ii} = \sum_j A_{ij}$ represents the degree of node i.

GNNs (Scarselli et al., 2008) are a class of DNN models designed to perform inference on graph-structured data, as opposed to traditional DNNs that assume the data is in Euclidean space. The development of GNNs was motivated by the complexity of graph-structured data, the need for relational learning, and the desire for flexibility in handling graphs of different sizes and topologies. By integrating the principles of neural networks with graph theory, GNNs provide a neural architecture that can process graphs natively.

GNNs operate based on the fundamental principle of message passing. In this mechanism, nodes in a graph exchange information with their neighbors. The core idea behind message passing is to update the feature representation of each node based on its own features and those of its neighbors. This process is typically performed iteratively over a number of layers or "hops."

Given a graph $\mathbf{G}$, the message passing process can be divided into two main steps: aggregation and update.

1. **Aggregation**: In this step, each node aggregates the features of its neighbors. The aggregation step for a node i at layer $l+1$ can be represented as

$$\boldsymbol{m}_i^{(l+1)} = \sum_{j \in N(i)} \boldsymbol{M}^{(l)}\left(\mathbf{X}_i^{(l)}, \mathbf{X}_j^{(l)}, e_{ij}\right) \quad (1)$$

where $\boldsymbol{m}_i^{(l+1)}$ is the aggregated feature (message) of node i at layer $l+1$; $N(i)$ represents the neighbors of node i; $\boldsymbol{M}^{(l)}$ is the aggregation function that combines the features of node i, node j, and edge e_{ij} at layer l; and $\mathbf{X}_i^{(l)}$ is the feature vector of node i. In general, the aggregation function is defined as a function that aggregates information from neighboring nodes before updating the target node.

2. **Update**: After aggregation, the node features are updated through the update function $\boldsymbol{U}^{(l)}$. This is typically done using a neural network layer followed by a nonlinear activation function. The update step can be represented as

$$\begin{aligned} \boldsymbol{X}_i^{(l+1)} &= \boldsymbol{U}^{(l)}\left(\mathbf{X}_i^{(l)}, \boldsymbol{m}_i^{(l+1)}\right) \\ &= \sigma\left(\mathbf{X}_i^{(l)} + \mathbf{W}^{(l)}\boldsymbol{m}_i^{(l+1)}\right) \end{aligned} \quad (2)$$

where $\mathbf{X}_i^{(l+1)}$ is the updated feature representation for the next layer $l+1$; $\mathbf{W}^{(l)}$ is a trainable weight matrix for layer l; and σ is the nonlinear activation function.

2.2. Graph attention network

GAT (Veličković et al., 2017) introduces an attention mechanism into the GNN to focus on the most relevant parts of the graph. The key idea behind GAT is to allow nodes to weigh the importance of their neighbors' features based on the task at hand, leading to more efficient and effective feature learning.

The attention mechanism in GAT assigns different weights to different nodes in a neighborhood, allowing for a more sophisticated aggregation of neighboring features. For a pair of nodes i and j, the attention coefficient c_{ij} measures the importance of the features of node j to node i. This coefficient is computed as follows:

$$c_{ij} = \mathrm{LeakyReLU}\left(\mathbf{a}^T\left[\mathbf{W}^{(l)}\mathbf{X}_i^{(l)} \| \mathbf{W}^{(l)}\mathbf{X}_j^{(l)}\right]\right) \quad (3)$$

where $\mathbf{a}$ is a weight vector learned by the attention mechanism; $\|$ denotes concatenation; and LeakyReLU is a variant of the rectified linear unit (ReLU) activation function that allows for a small, non-zero gradient when the input is negative.

To make the coefficients easily comparable across different nodes, the normalization over all choices of j can be applied as follows:

$$\alpha_{ij} = \frac{\exp(c_{ij})}{\sum_{k \in N(i)} \exp(c_{ik})} \quad (4)$$

where α_{ij} is the normalized attention coefficient that indicates the weight of the features of node j in updating of the features of node i.

The features of each node are updated by linearly combining the features of its neighbors as follows, weighted by the attention coefficients:

$$\mathbf{X}_i^{(l+1)} = \sigma\left(\sum_{j \in N(i)} \alpha_{ij}\mathbf{W}^{(l)}\mathbf{X}_j^{(l)}\right). \quad (5)$$

To stabilize the learning process, GAT employs a multi-head attention mechanism, where K independent attention mechanisms perform the above process in parallel. For example, if each feature vector is combined by averaging for the final layer, the updated features of node i can be computed as follows:

$$\mathbf{X}_i^{(l+1)} = \sigma\left(\frac{1}{K}\sum_{k=1}^{k=K}\sum_{j \in N(i)} \alpha_{ij,k}\mathbf{W}_k^{(l)}\mathbf{X}_j^{(l)}\right). \quad (6)$$

This multi-head mechanism improves the model's ability to focus on different parts of the graph, and also promotes feature diversity and robustness in the learned representations.

3. Proposed GAT-based damage identification method

3.1. Graph representation of structural system

To represent the structural system as a graph-structured data, the structural response data from each sensor is used as the node feature $\mathbf{X}$, while the adjacency matrix $\mathbf{A}$ is constructed according to the geometrical characteristic of the structure (Dang et al. 2022). As for the traditional adjacency matrix, one can assign values to each element according to the geometrical characteristic of the structural system, i.e., $A_{ij} = 1$ if node i and j are connected, otherwise $A_{ij} = 0$. As a simple example, Fig. 1 shows how the traditional adjacency matrix is constructed for a structure. In the case of v_1, for instance, only A_{12} and A_{13} have a value of 1 since v_2 and v_3 are connected to v_1.

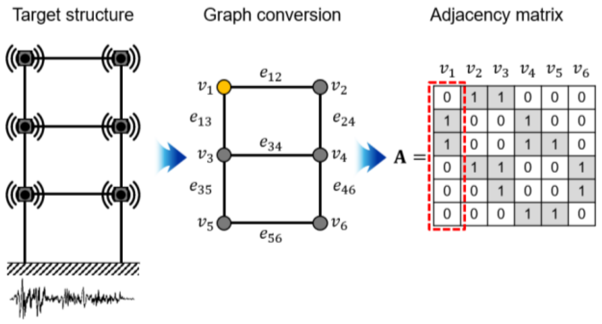

Fig. 1. Example of constructing traditional adjacency matrix for a structural system.

However, the traditional adjacency matrix cannot effectively provide information about the spatial correlation between sensors because it considers each connectivity between sensors with the same weight. In fact, vibration signals are significantly influenced by their structural characteristics such as topology and material properties. In addition, the traditional adjacency matrix is static, whereas the real-time responses of sensors have a dynamic nature due to external excitations or environmental factors. As a result, the temporal variations of the structural response cannot be fully captured by the static graph representation.

To address this issue, this paper introduces GAT to represent the correlation between neighboring nodes as the attention coefficient obtained from the node features, i.e., the structural responses. This approach allows GAT to dynamically adjust the weights between nodes based on the importance of their features for each time step. In this way, the GAT model can consider the varying degrees of influence that different parts of the structural system have on each other. Thus, this dynamic weighting mechanism captures the complex and time-varying spatial correlations between sensors. Additionally, the GAT model utilizes the attention coefficient to prioritize significant connections over weaker ones, enhancing its ability to detect subtle changes in structural condition.

3.2. Proposed architecture of GAT model

Incorporating with the proposed method to represent the structural system as a graph, an architecture of GAT model for structural damage identification is proposed, as illustrated in Fig. 2. The GAT model receives two types of input data: (1) the structural response data as the node feature $\mathbf{X}^{(k)}$ at time step k; and (2) the adjacency matrix $\mathbf{A}$ based on the geometric configuration of the structural system. The model is then trained to learn important latent features of the input data in an intact state and to reconstruct both $\mathbf{X}^{(k)}$ and $\mathbf{A}$ separately utilizing their spatiotemporal information through the following three main parts:

1. **Encoder** is first constructed by stacking multiple GAT layers. The GAT layers with nonlinear activation functions learn the latent features $\mathbf{Z}^{(k)}$ considering the spatiotemporal information and the multi-order proximity. In addition, the attention coefficient α_{ij} allows $\mathbf{Z}^{(k)}$ to capture the temporal variations in the structural response. Thereby, $\mathbf{Z}^{(k)}$ can provide a more comprehensive representation of the structural behavior.
2. **Graph structure decoder** learns spatial characteristics of the sensor network by reconstructing the adjacency matrix $\mathbf{A}$ from the latent features $\mathbf{Z}^{(k)}$. The graph structure decoder consists of a multi-layer perceptron (MLP), which is composed of multiple layers of interconnected neurons. The MLP is followed by a sigmoid activation function, since the adjacency matrix has a value of 0 or 1. The graph structure reconstruction error $\mathbf{R}_S = \left\|\mathbf{A} - \widehat{\mathbf{A}}\right\|_2$ is utilized to determine graph-structural anomalies, where $\widehat{\mathbf{A}}$ denotes the reconstructed adjacency matrix. A large value of $\mathbf{R}_{S,i} = \left\|\mathbf{A}_i - \widehat{\mathbf{A}}_i\right\|_2$ indicates that the i^{th} node has a higher probability of structural damage in terms of the spatial information, where $\mathbf{A}_i$ and $\widehat{\mathbf{A}}_i$ denote the i^{th} row of $\mathbf{A}$ and $\widehat{\mathbf{A}}$, respectively.
3. **Node feature decoder** is constructed to capture the temporal characteristics of the sensor network by reconstructing the node feature matrix $\mathbf{X}^{(k)}$. The node feature decoder consists of 1D convolutional layers to learn the temporal information. Anomalies in the structural responses can be captured with the feature reconstruction error $\mathbf{R}_F^{(k)} = \left\|\mathbf{X}^{(k)} - \widehat{\mathbf{X}}^{(k)}\right\|_2$, where $\widehat{\mathbf{X}}^{(k)}$ denotes the reconstructed node feature matrix, and a large value of $\mathbf{R}_{F,i}^{(k)} = \left\|\mathbf{X}_i^{(k)} - \widehat{\mathbf{X}}_i^{(k)}\right\|_2$ indicates that the i^{th} node has a higher probability of damage from a temporal perspective where $\mathbf{X}_i^{(k)}$ and $\widehat{\mathbf{X}}_i^{(k)}$ denote the i^{th} row of $\mathbf{X}^{(k)}$ and $\widehat{\mathbf{X}}^{(k)}$, respectively.

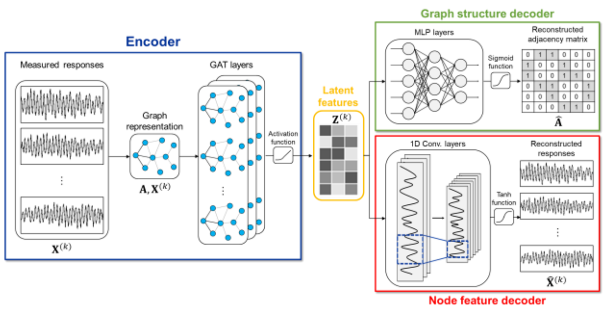

Fig. 2. Proposed architecture of GAT model for structural damage identification.

3.3. Near-real-time damage identification framework

A near-real-time damage identification framework is developed based on the proposed GAT model, which can be divided into two main parts: (1) offline; and (2) online processes.

Offline process: Data preprocessing and network training should be performed prior to performing near-real-time damage identification using the online process. The offline process consists of the following two steps:

1. Field experiment or structural analysis is conducted to obtain structural responses, followed by data preprocessing such as noise filtering. It is assumed that the structural response of the damaged structure is not obtained in this step considering the actual operational environment. The dataset is then divided into the train and validation datasets.
2. After the dataset is generated, the structural responses in the datasets are divided into the samples of $\mathbf{X}^{(k)}$ using a sliding time window. Herein, the time window should be set long enough to capture the vibration characteristics effectively. Let us assume that the sensor network has N sensors and $\mathbf{D} \in \mathbb{R}^{N\times T}$ denotes the structural response dataset with a length T. Using the time window of a length F, the node feature matrix at time step k, $\mathbf{X}^{(k)} \in \mathbb{R}^{N\times F}$, can be obtained by the windowing process. The node feature matrices can be obtained by repeating this process for all time steps. The adjacency matrix $\mathbf{A}$ is also obtained based on the geometric configuration of the structural system. The GAT model is then trained using the following loss function $\mathcal{L}(\mathbf{X}^{(k)}, \mathbf{A}, \alpha)$ based on the reconstruction error:

$$\begin{aligned}\mathcal{L}(\mathbf{X}^{(k)}, \mathbf{A}, \alpha) &= \alpha \mathbf{R}_S + (1-\alpha)\mathbf{R}_F^{(k)} \\ &= \|\mathbf{A} - \widehat{\mathbf{A}}\|_2 + \\ &\quad (1-\alpha)\|\mathbf{X}^{(k)} - \widehat{\mathbf{X}}^{(k)}\|_2\end{aligned} \tag{7}$$

where α denotes a hyperparameter that controls the relative contributions of the graph structure and the node feature reconstruction errors.

Online process: The online process for near-real-time damage identification is performed using the GAT model trained in the offline process. In this process, structural responses are obtained from sensors in real time while the actual state of the structural system is unknown, i.e., whether it is intact or damaged. The node feature matrix $\mathbf{X}^{(k)}$ in the unknown state is then obtained in near-real time using the sliding time window. The SDI of the i^{th} sensor at time step k, $SDI_i^{(k)}$, is defined as the value of the loss function calculated with $\mathbf{X}_i^{(k)}$ and $\mathbf{A}_i$ as follows:

$$\begin{aligned}SDI_i^{(k)} &= \alpha \mathbf{R}_{S,i} + (1-\alpha)\mathbf{R}_{F,i}^{(k)} \\ &= \|\mathbf{A}_i - \widehat{\mathbf{A}}_i\|_2 + \\ &\quad (1-\alpha)\left\|\mathbf{X}_i^{(k)} - \widehat{\mathbf{X}}_i^{(k)}\right\|_2.\end{aligned} \tag{8}$$

Near-real-time damage identification can be performed by repeating the online process at each time step.

To discriminate between the intact and damaged states, a threshold for the SDI is introduced to improve the accuracy of damage identification by reducing false-positive errors and focusing on significant structural changes. The threshold τ_i is defined as follows by introducing the three sigma rule, which is commonly used as an empirical threshold for Gaussian distributions:

$$\tau_i = \mu_i + 3\sigma_i \tag{9}$$

where μ_i is the mean value of the loss function $\mathcal{L}(\cdot)$ over the entire time for the response data from the intact structural system at the i^{th} sensor; and its standard deviation is denoted by σ_i.

Furthermore, the normalized SDI is introduced to consider the variation in the amplitude of the loss function according to the sensor location. The normalization process involves dividing the SDI values by their respective mean values for each sensor, μ_i. The normalized SDI of the i^{th} sensor at time step k, $\overline{SDI}_i^{(k)}$, is defined as follows:

$$\overline{SDI}_i^{(k)} = \frac{SDI_i^{(k)}}{\mu_i}. \tag{10}$$

The normalized threshold $\bar{\tau}_i$ is also defined by the same process as the normalization of the SDI, as follows:

$$\bar{\tau}_i = \frac{\tau_i}{\mu_i} = 1 + \frac{3\sigma_i}{\mu_i}. \tag{11}$$

This process allows the amplitude of the loss function to be normalized across all locations, ensuring uniformity in the assessment of structural damage.

4. Numerical investigations

4.1. Structural properties of target structure

As a target structure, the 3D steel frame structure in Fig. 3 is considered. The finite element (FE) model of the target structure is constructed using OpenSeesPy. The elastic section is used for each element. The target structure consists of uniform column heights of 14 ft (4.2672 m) and columns with section properties of a wide-flanged beam of W27×114 a36 carbon steel. The beams have uniform lengths of 24 ft (7.3152 m), and their section properties are of a W24×94 a36 carbon steel wide-flanged beam. The modal damping ratio ζ is also set to 5%, and the natural frequencies of the first six modes in a healthy state are 0.282, 0.682, 0.788, 0.805, 1.256, and 1.278 Hz, respectively.

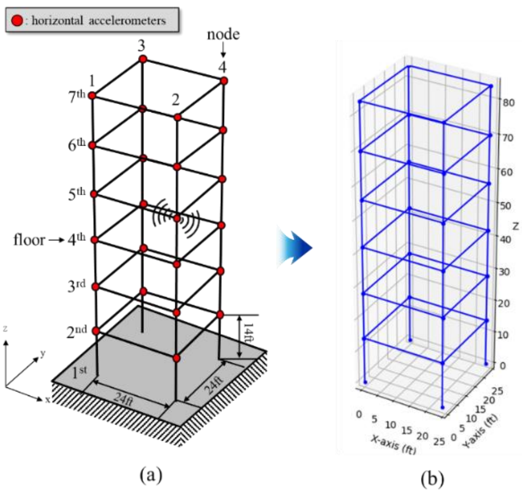

Fig. 3. 7-story steel frame as target structure: (a) conceptual illustration; and (b) FE model.

4.2. Data generation and pre-processing

Structural analysis is also performed by OpenSeesPy using seismic ground motions from the PEER-NGA strong motion database. A total of 499 ground motions with a sampling frequency of 100 Hz are used and are divided into the train, validation, and test datasets in a ratio of 8:1:1. It is assumed that the x- and y-axis accelerations of all nodes except the ground level are measured. The time window length is set to 10.24 s for all cases, and the time window interval is set to 1 second. The datasets are scaled to the range of $[-1, 1]$, normalized by the maximum absolute scaling. The adjacency matrix $\mathbf{A}$ of the target structure is constructed based on the construction method of the traditional adjacency matrix. Since the target structure has 24 sensors, $\mathbf{A}$ has the dimension of 48×48.

4.3. Network training

The GAT model is constructed using the Python deep learning library Tensorflow and trained on a server with two NVIDIA TITAN RTX graphics cards, and 128GB RAM. The hyperparameter α of the loss function is set to 0.5. The numbers of epochs and batch size are set to 1,000 and 32, respectively. The Adam optimizer with a learning rate of 0.001 is used to minimize the loss function. The loss function converges fast and stably without any issues of gradient vanishing or exploding.

4.4. Near-real-time damage identification

To evaluate the performance of the pre-trained network, real-time test simulations are performed with a randomly selected test ground motion. Fig. 4 shows the records of the x- and y-axis components of the test ground motion. The acceleration signals are measured at each time step, and the SDIs are calculated simultaneously through the pre-trained network. In all cases, structural damage is simulated by 25% or 50% degradation in Young's modulus, occurring at peak ground acceleration (PGA).

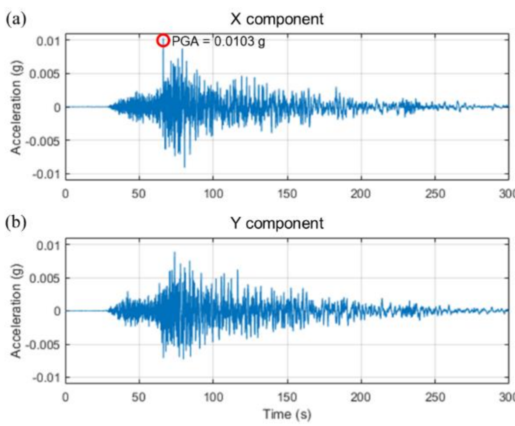

Fig. 4. Records of (a) x-axis; and (b) y-axis components of the test ground motion.

In the figures showing the results of damage identification, the results for nodes on the same floor are shown in order in the corresponding row of figures, while the results from the lowest floor to the highest floor are shown from the bottom row of figures to match the geometry of the target structure. The results for the x- and y-axis responses of each node are shown in order. The lines in each subplot indicate the normalized SDI of each node. The normalized threshold $\bar{\tau}_i$ represented by the black horizontal dashed line, and it can be considered that damage occurs when the SDI crosses above this line. The gray-shaded subplots correspond to the identification results of nodes adjacent to the damaged elements, and the red vertical dashed line indicates the time of damage occurrence. In each shaded subplot, a sharp increase in the SDI exceeding the threshold after the dashed line indicates the damage occurrence, while a higher SDI corresponds to a greater degree of damage.

Each damage level is subdivided into the following three damage cases: (1) damage on the 1st floor; (2) damage on the 4th floor; and (3) damage on the 1st and 4th floors. As shown in Fig. 5, the GAT model provides a higher level of accuracy in identifying the occurrence and location of damage in all damage cases. Note that the damage identification performance for the 4th floor damage case is inferior to that for the 1st floor damage case since the columns on the 4th floor have less influence on the structural behavior than those on the 1st floor. Nevertheless, the GAT model successfully identifies all damage locations without any false-negative errors. Thereby, it is possible to locate damage that occurred near sensors. Furthermore, it takes only 0.45 s on average to obtain the SDI values, which is much shorter than the pre-set time interval, ensuring the implementation of near-real-time damage identification.

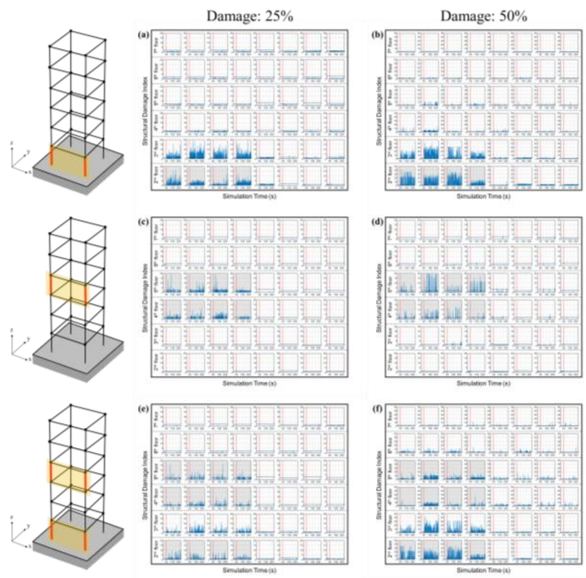

Fig. 5. Near-real-time damage identification results: (a) 25% damage on 1st floor; (b) 50% damage on 1st floor; (c) 25% damage on 4th floor; (d) 50% damage on 4th floor; (e) 25% damage on 1st and 4th floors; and (f) 50% damage on 1st and 4th floors.

5. Conclusions

This paper proposed a new GAT-based framework for near-real-time structural damage identification. A novel GAT architecture was proposed to capture changes in the spatial correlation as well as the dynamic vibration characteristics. The GAT model was trained to reconstruct the vibration signals and the adjacency matrix of the target structure to leverage the spatiotemporal information. This paper also introduced the SDI, which quantifies the extent of damage based on the reconstruction error of input data. Numerical investigations of the 3D steel frames demonstrated that the proposed framework successfully identified damage under seismic load conditions in near-real time. The robust performance of the proposed method under seismic load conditions is expected to reduce the time required for the post-disaster decision-making process. Eventually, the proposed framework will be utilized to develop effective post-disaster operational and maintenance strategies.

Acknowledgement: This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korean government (MSIT) (RS-2022-00144434), and Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (P0017304, Human Resource Development Program for Industrial Innovation).

REFERENCES

[1] V. H. Dang, T. C. Vu, B. D. Nguyen, Q. H. Nguyen, and T. D. Nguyen, Structural damage detection framework based on graph convolutional network directly using vibration data, *Structures*, Vol. 38, pp. 40-51, 2022.
[2] M. Kim, and J. Song, Seismic damage identification by graph convolutional autoencoder using adjacency matrix based on structural modes, *Earthquake Engineering & Structural Dynamics*, Vol.53, No. 2, pp. 815-837, 2024.
[3] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, *nature*, Vol. 521, No. 7553, pp. 436-444, 2015.
[4] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, The graph neural network model, *IEEE transactions on neural networks*, Vol. 20, No. 1, pp. 61-80, 2008.
[5] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, Graph attention networks, *arXiv preprint arXiv:1710.10903*, 2017.