



Article

Enhancing Recovery of Structural Health Monitoring Data Using CNN Combined with GRU

Nguyen Thi Cam Nhung ¹, Hoang Nguyen Bui ² and Tran Quang Minh ^{3,*}

¹ Faculty of Civil Engineering, University of Transport and Communications, Hanoi 100000, Vietnam; ncnhung@utc.edu.vn

² School of Engineering and Applied Sciences, State University of New York at Buffalo, New York, NY 14360, USA; hbuinguy@buffalo.edu

³ ISISE, ARISE, Department of Civil Engineering, University of Minho, 4800-058 Guimarães, Portugal

* Correspondence: minhqt@civil.uminho.pt

Abstract: Structural health monitoring (SHM) plays a crucial role in ensuring the safety of infrastructure in general, especially critical infrastructure such as bridges. SHM systems allow the real-time monitoring of structural conditions and early detection of abnormalities. This enables managers to make accurate decisions during the operation of the infrastructure. However, for various reasons, data from SHM systems may be interrupted or faulty, leading to serious consequences. This study proposes using a Convolutional Neural Network (CNN) combined with Gated Recurrent Units (GRUs) to recover lost data from accelerometer sensors in SHM systems. CNNs are adept at capturing spatial patterns in data, making them highly effective for recognizing localized features in sensor signals. At the same time, GRUs are designed to model sequential dependencies over time, making the combined architecture particularly suited for time-series data. A dataset collected from a real bridge structure will be used to validate the proposed method. Different cases of data loss are considered to demonstrate the feasibility and potential of the CNN-GRU approach. The results show that the CNN-GRU hybrid network effectively recovers data in both single-channel and multi-channel data loss scenarios.

Keywords: data recovery; convolutional neural network; gated recurrent units; structural health monitoring



Citation: Nhung, N.T.C.; Bui, H.N.; Minh, T.Q. Enhancing Recovery of Structural Health Monitoring Data Using CNN Combined with GRU. *Infrastructures* **2024**, *9*, 205. <https://doi.org/10.3390/infrastructures9110205>

Academic Editor: Alejandro Jiménez Rios

Received: 19 October 2024
Revised: 13 November 2024
Accepted: 15 November 2024
Published: 16 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Mega-projects and infrastructure are being built rapidly due to urbanization and industrialization. This is essential to the socioeconomic development of every country. Bridges are among the most essential and critical infrastructures serving transportation purposes. They are located on all routes, helping to overcome obstacles, reduce distances between areas, facilitate the circulation of goods and services, and enhance the country's economic potential. However, regular maintenance and supervision are essential to ensure that bridges can operate safely. In this context, SHM systems are becoming increasingly important [1]. Early structural damage and deterioration detection by SHM enable prompt maintenance and repair actions to reduce hazards and expensive fallout from unplanned breakdowns [2–4].

The SHM system comprises sensors and a data reception and processing system. These sensors are permanently installed on the bridge structure and collect critical signals such as vibrations, displacements, deformations, or tilting, among others [5–8]. These signals are transmitted through data transmission systems (cables or wireless) to the analysis and data processing system. Here, the raw data are subjected to analytical techniques to produce results. Based on these results, experts and managers can make decisions related to maintaining the operational condition of the bridge [9,10].

The sensor system is designed to be placed at various locations on the bridge structure, allowing for comprehensive data collection regarding the structural performance [11,12]. The sensor system is typically carefully protected but still exposed to external environmental factors. Over time, due to various influences, the sensor system may encounter issues such as reduced sensitivity, signal loss, or even damage [13]. This poses potential risks of data loss. Additionally, the data transmission system may also experience failures. These may include broken cables, wireless transmission interruptions, or system short circuits. Such factors can lead to data disruptions, loss, or even inaccuracies in the transmitted data [14]. When data errors or interruptions occur, the SHM system is significantly affected. Specifically, faulty data can lead to insufficient information for analyzing and assessing the structural condition. Interrupted data cause a loss of information, resulting in inaccurate evaluations by experts. Furthermore, data transmitted to the processing center may be distorted, leading to incorrect analysis results. The conclusions drawn could be misleading, resulting in incorrect decisions regarding the operation and maintenance of the bridge.

To minimize data loss in SHM, several intervention strategies have been proposed. The SHM system is equipped with more durable devices, and the design allows sensors to withstand impacts and weather conditions. However, despite the significant resources invested, the effectiveness remains limited. After an extended period, data loss still occurs. The repair and replacement of damaged components have been suggested in some cases, but this is only a temporary solution. SHM systems are often deeply embedded in the structure, making them difficult to replace. Furthermore, repairs and replacements can create additional complications for homogeneous systems. Not to mention, accessing sensor locations is often extremely difficult if not impossible. As a result, the effectiveness of repair intervention strategies for SHM systems is low.

Many researchers have started focusing on recovering SHM data in cases of loss or error [15]. The proposed solutions have been tested and have shown potential in enhancing SHM systems. Using statistical methods for data reconstruction in SHM is one of the traditional and prevalent approaches, aiding in restoring and completing missing or noisy data. Statistical methods typically involve building mathematical models based on existing data samples to predict and reconstruct missing data. One common technique is interpolation, which estimates missing data points using the values of neighboring data points. Additionally, more complex statistical models, including linear regression, nonlinear regression, and time-series analysis models, are applied for data reconstruction in SHM. The advantages of statistical methods include their simplicity and ease of implementation, requiring relatively few computational resources. They are particularly useful in cases where data loss is minimal and the data exhibit periodic or predictable patterns. However, the limitations of this approach lie in its potential inaccuracy in complex data situations or when there is extensive data loss. Utilizing statistical methods requires a deep understanding of the data and the structure of the monitored system to select the most appropriate model [16].

In recent years, artificial intelligence (AI) and intelligent algorithms have emerged as a trend to solve many problems related to reconstruction data [17]. The application of AI in data reconstruction for SHM is becoming increasingly popular and offers numerous outstanding advantages. AI models can learn from historical data patterns to intelligently predict and restore missing or noisy data [18,19]. Wan and Ni [20] proposed a method to recover SHM data in the context of Bayesian multi-task learning with multidimensional Gaussian programming. The reconstruction performance achieved is highly reliable. Zhang and Luo [21] used data correlation to recover lost stress data. This method results in an interpolation error of about 5–7%. Other studies [22–24] have proposed data recovery methods that also provide highly accurate results. Lei et al. [9] used deep convolutional generative adversarial networks to rebuild lost data in SHM. The proposed deep learning architecture is capable of reconstructing data from accelerometers and strain sensors on models and, in reality, with high performance. Jiang et al. [25] performed the recovery of incomplete data from a long-term health monitoring system using an unsupervised learning method based on a generative adversarial network. Introduced by Fan et al. [26],

the process of using convolutional neural networks to reconstruct SHM data is implemented. This method allows for outstanding lost data recovery even for signals with severe data loss rates of up to 90%. Fan et al. [27] conducted research and achieved good results in dynamic response reconstruction using densely connected convolutional networks. In addition, many other studies [28–31] use deep learning in data reconstruction.

In this study, a hybrid CNN-GRU network is used to reconstruct data from accelerometers for structural health monitoring of the bridge. The proposed method is a promising solution to solve the problems related to data loss of SHM systems. In case the data collected from the sensors show abnormalities or undergo a special event, the method can be applied to restore the lost data with a high level of reliability. It contributes to improving the efficiency of SHM systems of large bridges. The article is organized into four sections: the introduction, methodology, case study, and finally, conclusion. The results of the case study show that CNN-GRU has an excellent ability to reconstruct sensor data.

2. Methodology

2.1. Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) [32,33] are a special deep-learning model developed to process grid data. CNNs work by automatically learning data features through multiple layers. The convolution layer is the foundation of CNNs. It works by sliding a filter across the entire input image and calculating the convolution between the filter and the corresponding sub-data region [34]. This process helps extract local features [35]. The formula for calculating the convolution of a filter W over a portion of input data X is [36–38]

$$S(i, j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X(i + m, j + n) \cdot W(m, n) + b \tag{1}$$

where $S(i, j)$ is the output value at positions (i, j) ; $W(m, n)$ is the value of the filter at position (m, n) ; $X(i + m, j + n)$ is the value of the input data at the corresponding position; b is the bias value; and M and N are the height and width of the filter. The operator between $X(i + m, j + n)$ and $W(m, n)$ denotes element-wise multiplication. This process results in a feature map, which helps the CNNs capture local features from the input.

After the convolutional layers, CNNs continue using activation functions such as ReLU, Sigmoid, Softmax, etc. In this study, the ReLU activation function is used to help remove negative values and preserve positive values [36–38]:

$$f(x) = \max(0, x) \tag{2}$$

Next, the pooling layers are responsible for reducing the data’s dimensionality and the number of parameters while maintaining the important features. Pooling layers include MaxPooling, AveragePooling, etc. This study uses MaxPooling to reduce the dimensionality of the data. The formula of MaxPooling for a sub-region of size $k \times k$ is [36–38]

$$S(i, j) = \max_{m=0}^{k-1} \max_{n=0}^{k-1} X(i + m, j + n) \tag{3}$$

After the convolution and pooling layers, the data are “flattened” into a one-dimensional vector. This vector is then fed into the fully connected layers. Each neuron in these layers connects to all the neurons in the previous layer, which helps generate the final decision of the model. The formula of fully connected class is shown below [36–38]:

$$y = f\left(\sum_{i=1}^n W_i x_i + b\right) \tag{4}$$

Here, y is the output, W is the weight, x_i is the input, b is the bias, and f is the activation function.

The convolution process can be represented as shown in Figure 1.

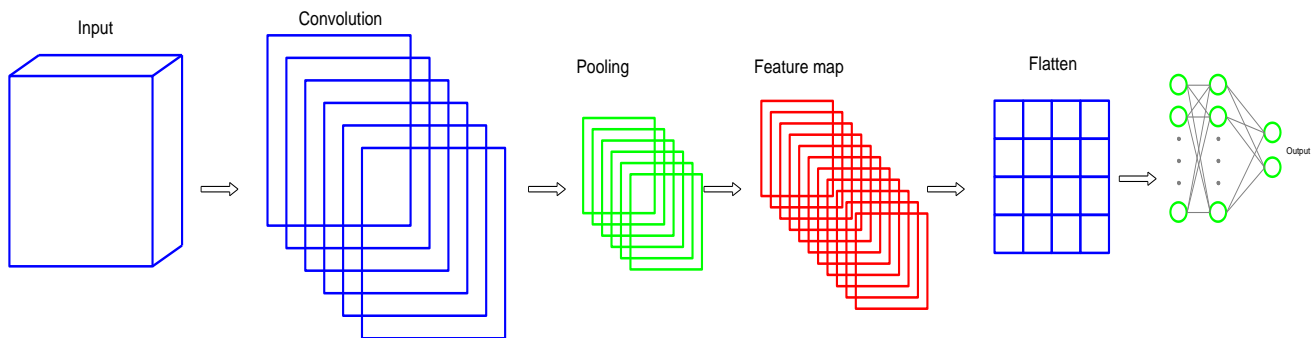


Figure 1. Convolutional Neural Networks.

2.2. Gated Recurrent Units (GRUs)

Recurrent Neural Networks (RNNs) [39] are a type of neural network that can remember and use information from previous steps to predict the output for the next step thanks to the feedback loop mechanism. However, traditional RNNs have difficulty handling long time series due to the “vanishing gradient” phenomenon, when the gradient becomes too small to update the weights during backpropagation [40]. Gated Recurrent Units (GRUs) [40,41] are a variant of RNNs developed to improve its limitations especially in handling long time series and overcoming the vanishing gradient problem.

GRUs have two main gates: update gates and reset gates. These gates help adjust the flow of information over each time step, ensuring that the model can learn important information and discard unnecessary information. Unlike LSTMs, GRUs do not have separate input and output gates, which makes the model simpler.

The update gate regulates the extent to which the old state will be retained and the extent to which the new state will be updated. This allows the GRUs to remember important information in the time series without losing it. The formula for the update gate z_t is [40,42]

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{5}$$

where z_t is the updated gate at time t ; h_{t-1} is the hidden state from the previous step; x_t is the current input; W_z is the learned weight and σ is the sigmoid function

The reset gate determines how much information from the previous state h_{t-1} will be forgotten at the current time step. This allows the GRUs to discard unnecessary information from previous steps to focus on newer information [40,42].

$$z_r = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{6}$$

where z_r is the reset gate at time t ; h_{t-1} is the hidden state from the previous step; x_t is the current input; W_r is the learned weight and σ is a sigmoid function

GRUs is a powerful and efficient architecture for processing time-series data. With a simple yet effective structure, GRUs not only solve the problem of vanishing gradients well but also help the model learn long-term dependencies in the data series [43]. The architecture of GRUs is shown in Figure 2 [44].

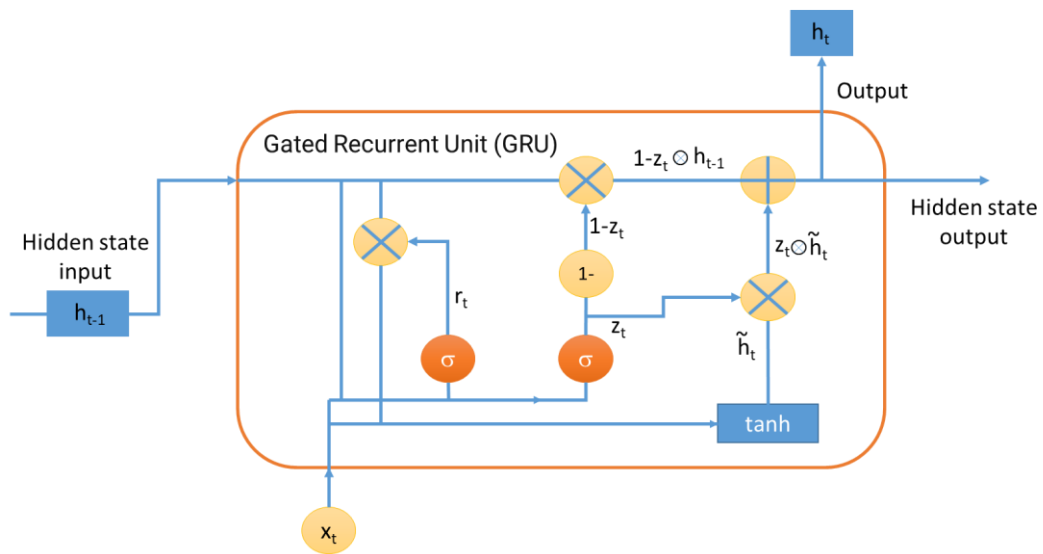


Figure 2. The structure of GRU network [44].

2.3. Using CNN-GRU to Recover Data

When CNNs and GRUs are used together, they become a more potent tool for retrieving sensor data and have many benefits over utilizing each model separately. CNNs extract significant spatial features from sensor data, including structures or hidden patterns within data matrices. Their convolutional layers enable the detection and analysis of local features. These features contain crucial information about the data’s spatial distribution and are the characteristics of ML models. Following the CNN layers of refinement, the data are sent to the GRU layers to process the temporal relationships within the retrieved features. Memory and learning from sequential data are critical components of GRU’s capacity to recognize patterns and trends over time.

The system overcomes the drawbacks of employing each model separately by combining their strengths. It results in a more adaptable and reliable method for recovering and analyzing sensor data—especially in intricate and expansive structural health monitoring systems. Figure 3 depicts the procedure for recovering data.

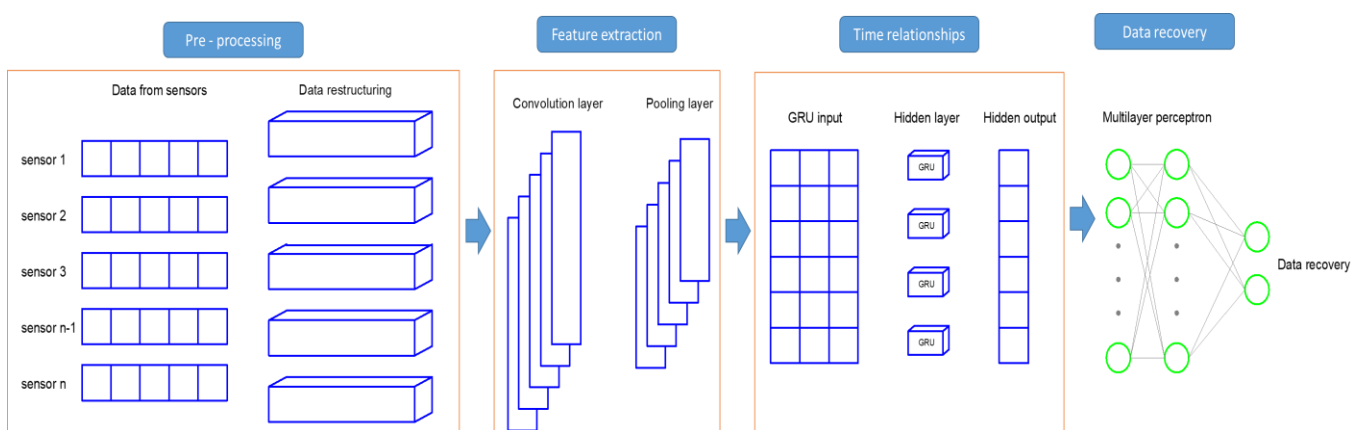


Figure 3. Data recovery process using CNN-GRU.

First, in the preprocessing stage, sensor data are collected and restructured into a suitable format for input into the network. Specifically, one-dimensional time-series data are restructured into two-dimensional data (adding more dimensions to the data, where one dimension is data and the other dimension is 1). Features from the raw data are prepared for training the CNN-GRU model. Next, the data are passed through the convolution layer

in the feature extraction stage to extract essential features. The data are then forwarded to the pooling layer to reduce its size while retaining essential information. After that, in the time relationship stage, the data are passed through the GRU layers, where the network learns temporal relationships from the sensor data through hidden layers. Finally, the data are processed by a multilayer perceptron to predict and recover the data from the faulty sensors.

3. Case Study

3.1. Case Study Description

In this study, the Thang Long Bridge [45,46] (Figure 4)—the largest steel truss bridge in Hanoi—will be considered. Before other bridges were built to span the Red River, the Thang Long Bridge was pivotal in connecting Hanoi with the northern provinces, becoming a vital artery for transportation and economic exchange. For many years, it was the only bridge capable of supporting both railway traffic and heavy vehicles across the Red River, reinforcing its central role in Vietnam’s development. Despite the emergence of new bridges, the dual functionality and historical significance of the Thang Long Bridge continue to make it one of the region’s most strategically important infrastructure projects.



(a)

(b)

Figure 4. Thang Long Bridge: (a) side view; (b) lower floor.

The Thang Long Bridge is designed to exploit road and rail traffic, creating a unique structural feature. The bridge consists of two levels: the upper level is exclusively for road vehicles, while the lower level is designed for trains with rail gauges of 1000 mm and 1435 mm. Cantilevers are installed on both sides of the lower level to allow rudimentary vehicles and motorcycles to cross the bridge.

The Thang Long steel truss bridge comprises 15 spans divided into five modules, each containing three continuous steel truss spans. The bridge is designed according to the 18-79TCN standard of Vietnam [47]. The operational load capacity of the bridge is T24 for trains and H30-XB80 for vehicles. The bridge has undergone two significant repairs and has been upgraded to meet the new design load standard—HL93. The truss members are made from alloy steel and have box-shaped cross-sections. The upper and lower truss members have a height of up to 800 mm, while the vertical and diagonal members are 600 mm. Transversely, cross-stiffening ribs are installed at 3.6 m intervals to enhance the overall rigidity of the bridge. Upper and lower wind bracing systems are also designed with I-shaped steel to resist lateral movement and increase the bridge’s stiffness.

During the assessment campaign for the renovation of the Thang Long Bridge in 2020 [48], a vibration survey of the bridge was conducted. Vibration data were collected at the lower truss nodes of the first span of the Thang Long Bridge. The layout of the vibration

measurement points is shown in Figure 5. The objective of the survey was to identify the mode shapes and their corresponding frequencies to assess the bridge's condition.

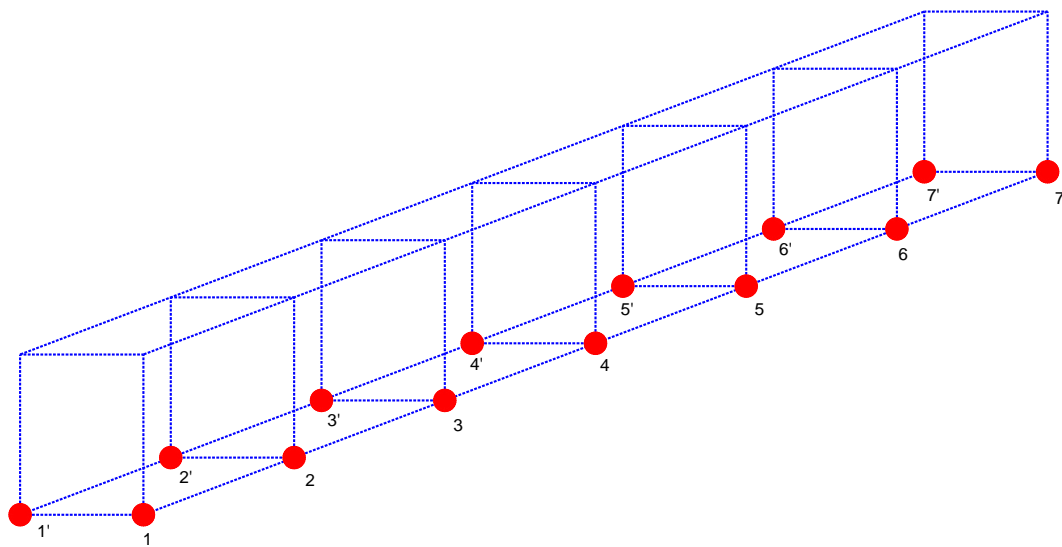


Figure 5. Arrangement of measuring points at Thang Long Bridge.

PCB accelerometer sensors (model 393B12) [49] were deployed on site at the truss nodes where measurement points were designated. Once the data collection station was established, signal transmission cables were connected to a signal converter, and the data were stored on a computer. Each sensor was labeled and stored similarly to prevent loss and ensure effective data management. Due to the limitation of the number of sensors, the overall measurement grid is divided into two smaller measurement grids. Each smaller measurement grid consists of eight measurement points corresponding to eight sensors. For the data to be linked together, reference points will be established. In this study, the reference point is located at location 3. After completing a small measurement grid, the remaining points will be reinstalled at the remaining points to cover the overall measurement grid. For each sub-grid, data collection took approximately 40 min with a sampling frequency of 1651 Hz. Half of the collected data will be used to train and validate the network. The other half will be used as a test set, which will then be used for detailed analysis. Figure 6 shows the data collection station and the locations where the sensors were installed.



Figure 6. Data collection: (a) equipment station; (b) sensors' installation location.

The collected dataset consists of vibration signals from pre-designated measurement locations. A data reconstruction study will be conducted based on this dataset. Specifically, various data loss scenarios will be simulated, ranging from single-channel data loss to more complex multi-channel loss. Once trained on the complete dataset, the CNN-GRU network will be used to recover the missing data in these scenarios. The training process enables the model to learn key features of the vibration data, accurately predicting the missing data based on the remaining information. This is especially important for ensuring the continuity and accuracy of the SHM system. After the data are recovered, a detailed analysis will be performed to evaluate the quality of the reconstructed data. The recovered data will be directly compared with the collected data to assess the model's accuracy.

3.2. Single Channel Data Recovery

In the first scenario, single-channel data loss will be simulated. The overall dataset will include eight columns of data corresponding to signals from eight sensors. One sensor will be assumed to malfunction, and its values will be set to zero. The remaining sensors will keep their values to represent regular operation. The input to the CNN-GRU network will consist of seven columns of data from the functioning sensors, while the output will be the data from the faulty sensor. The network will extract critical features from the input sensors and find correlations with the data from the faulty sensor. Through multiple training iterations, the CNN-GRU network will learn the relationship between the input and output data, achieving a certain level of recovery performance. Once trained, the network will be used to recover the missing data. Data recovery will be conducted for each sensor at every location to ensure the method performs well on all sensors. This means that the inputs and outputs will be adjusted multiple times. In each case, the input will consist of data from seven sensors, while the output will be the data from the faulty sensor. Once training is completed for all locations, the data will be recovered, forming a new dataset (the recovery dataset).

Data preprocessing plays a crucial role in achieving good training performance. Sensor data are preprocessed, organized, and formatted to meet the requirements of the CNN-GRU network. The data are divided into two parts with a 70/30 split, where 70% is used to train the network and 30% is used to evaluate its performance. The input and output data are assigned to two separate matrices in each dataset. The CNN-GRU network is designed with three CNN layers and three GRU layers. The first CNN layer is configured with 512 filters, while the subsequent two layers are reduced to 256 and 128 filters, respectively. The kernel size is uniformly set to 10, and the ReLU activation function is used. MaxPooling layers are added after each CNN layer to reduce the size of the data, allowing better feature extraction. Afterward, the data are flattened using a Flatten layer to create an appropriate input format for the subsequent GRU layers. This setup ensures that the data are appropriately structured before entering the GRU layers, which are designed to capture temporal relationships in the data.

The number of hidden units in the GRU layers is progressively reduced, starting from 256 units in the first layer, down to 128 in the next, and finally 64 units in the last GRU layer. This gradual reduction in hidden units helps the neural network simplify the information as it passes through each layer, allowing it to focus more on extracting key features at different levels of abstraction. Between the GRU layers, Batch Normalization is applied to standardize the output of each layer, ensuring that the values have a stable distribution with a mean of 0 and a standard deviation of 1. This normalization not only reduces the network's reliance on the initial weight initialization but also improves the convergence speed, making the training process faster and more efficient. At the end of the network, a Dense layer is added to predict the output, corresponding to the data from the faulty sensor. This Dense layer acts as a fully connected layer, where each neuron is connected to all the neurons in the previous layer. Its primary function is consolidating and interpreting the high-level features extracted by the preceding layers, enabling the network to make accurate predictions about the missing data from the faulty sensor.

Finally, the model compiler is launched with the Adam optimizer and the MSE (Mean Squared Error) loss function. The data are set to train for a maximum of 1000 epochs with a batch size of 50. To demonstrate the advantage of the proposed network, individual CNN and GRU networks are also implemented. The training results are shown in Figure 7.

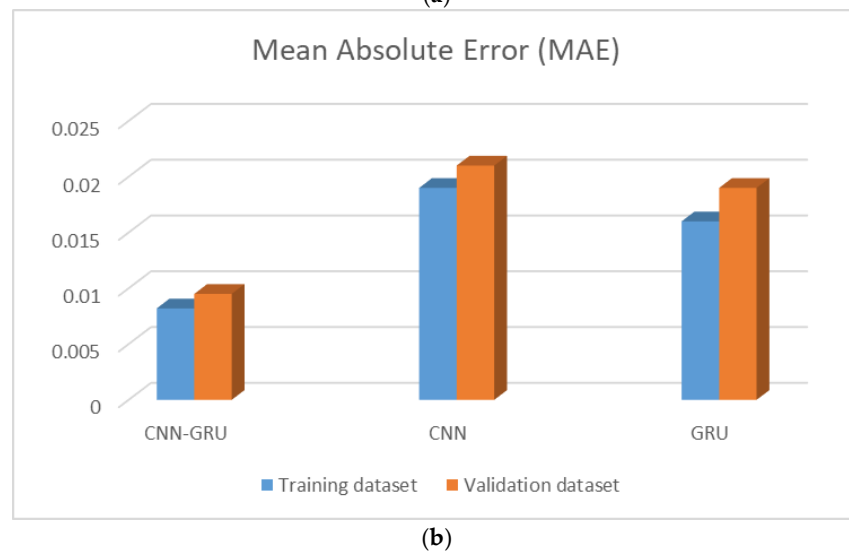
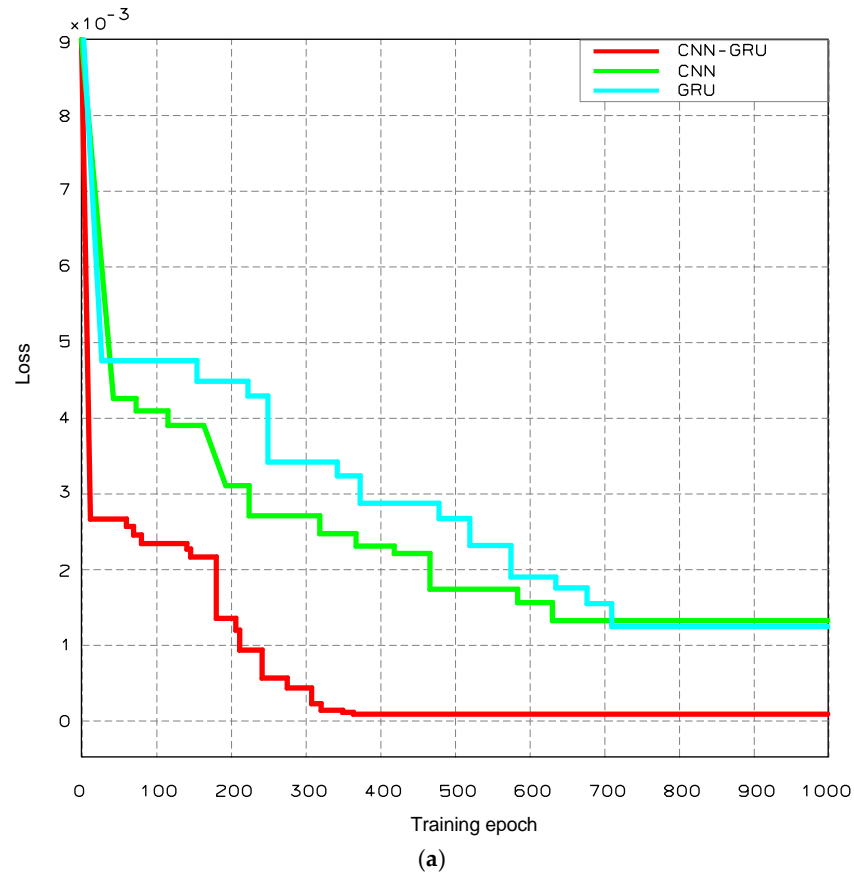


Figure 7. Network training results in single-channel data recovery scenario: (a) training convergence curve; (b) mean absolute error.

Figure 7a illustrates the fluctuation of the loss function during the training of three models: CNN-GRU, CNN, and GRU. In the early stages of training, all three models show a significant reduction in the loss value. However, as they progress into subsequent epochs, the differences between the models become increasingly evident. The CNN-GRU model

stands out with the best performance, achieving a loss value that approaches nearly zero. CNN-GRU maintains the lowest loss value throughout the training process, demonstrating its superior ability to capture data features compared to the individual GRU and CNN models, which exhibit higher loss values. Additionally, CNN-GRU converges faster and reaches a stable state in just about 350 epochs, while the other two models take more than 700 epochs to achieve similar stability. These results indicate that combining CNN and GRU can perform significantly better than independent neural networks.

Figure 7b presents the mean absolute error (MAE) between the recovered and actual values for the three types of models on both the training and testing datasets. The proposed CNN-GRU model exhibits a significantly lower MAE than the other two. The MAE values for the CNN-GRU model on the training and testing datasets are low and show no significant disparity. This indicates that the model performs well and has a high generalization ability. The model successfully learns the critical features from the data without over-relying on the training data, allowing it to make accurate predictions on new data. Figure 8 shows a recovery data segment using the above three models.

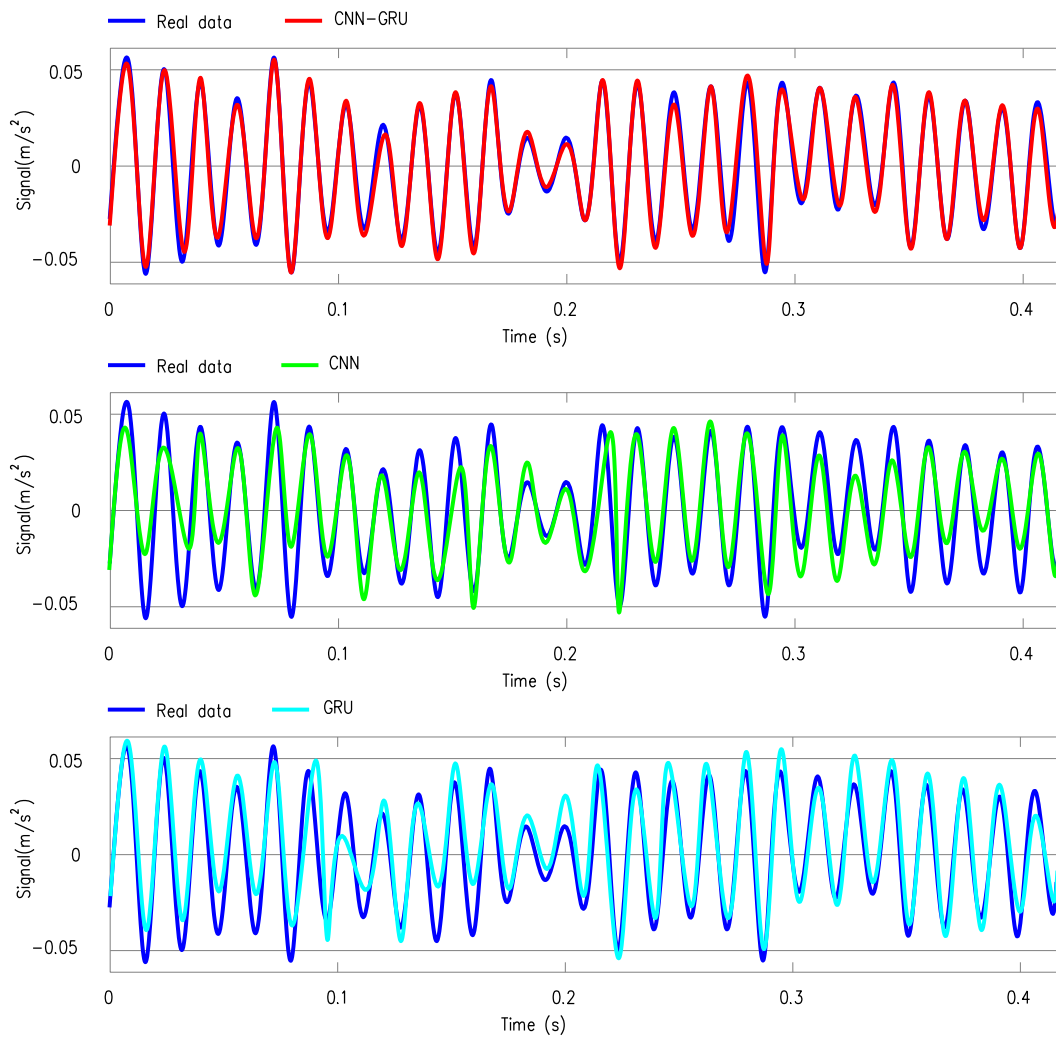


Figure 8. Recovery data segment using CNN-GRU; CNN and GRU.

Figure 8 shows that the data recovered by the CNN-GRU are nearly identical to the actual data. In contrast, the data recovered by the individual CNN or GRU models exhibit specific differences. This further demonstrates the robust recovery capability of the proposed method. To further evaluate the data recovered by the CNN-GRU, modal analysis was conducted on the two datasets—the actual data and the data recovered by the

CNN-GRU. The results of the mode shapes obtained from the two datasets are presented in Figure 9 and Table 1.

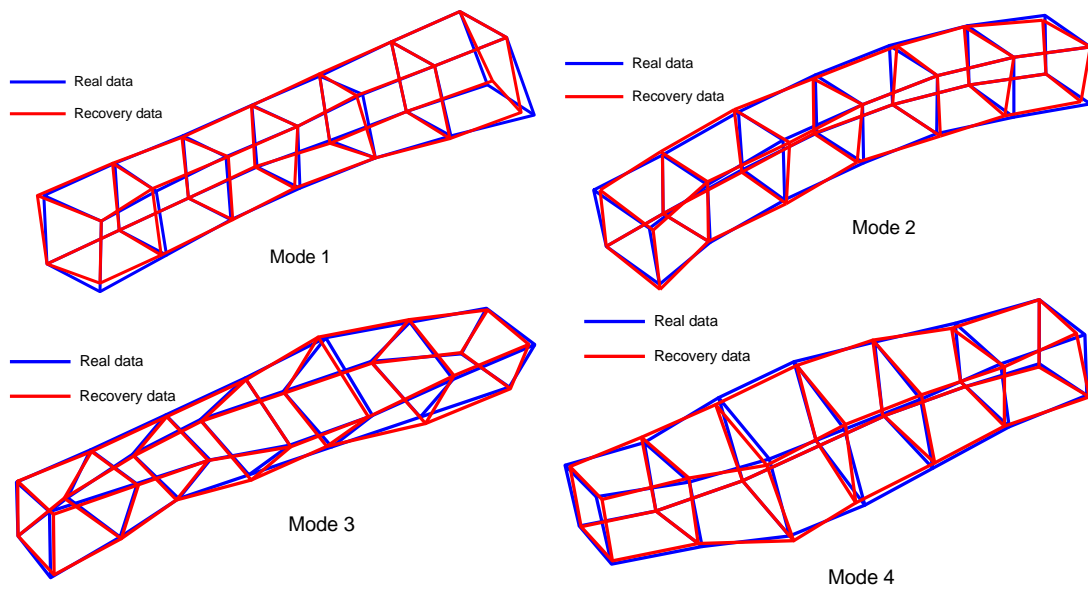


Figure 9. Mode shapes of two datasets.

Table 1. Results of analysis of real and recovery data.

Mode	Frequency (Hz)		Error (%)	MAC
	Real Data	Recovery Data		
1st	1.05	1.08	2.86	0.968
2nd	1.71	1.59	7.02	0.954
3rd	2.04	2.21	8.33	0.951
4th	2.98	2.72	8.72	0.946
Error = $(f_{\text{recovery}} - f_{\text{real}}) / f_{\text{real}} \times 100\%$				

Table 1 compares real and recovery data across four modes, focusing on frequency, error percentage, and MAC (Modal Assurance Criterion). Regarding frequency, the first mode shows minimal deviation between actual data (1.05) and recovery data (1.08) with an error of only 2.86%. However, the discrepancy increases in higher modes with the second mode having an error of 7.02% and the third and fourth modes showing errors of 8.33% and 8.72%, respectively. Despite these rising errors, the MAC values remain consistently high across all modes, indicating a strong similarity in mode shapes. The first mode has the highest MAC at 0.968, and while the MAC decreases slightly in the second, third, and fourth modes (0.954, 0.951, and 0.946), the correlation between real and recovered data remains robust. The analysis results show that the data reconstruction model can accurately reproduce the main features of real data.

3.3. Multi-Channel Data Recovery

In the next research phase, multi-channel data loss scenarios will be explored to evaluate the potential of the CNN-GRU network under more complex conditions, where the number of input sensors decreases and the number of output sensors to be recovered increases. Specifically, cases of data loss will be simulated with two to four sensors failing. In these scenarios, the input data will gradually reduce from six columns to five and then four, while the output data that need to be recovered will increase from two to three and then four columns. The faulty sensors will be assigned a zero value, while

the remaining functioning sensors will retain their original values. These multi-channel data loss scenarios aim to assess the proposed model’s performance and effectiveness in handling more complex situations.

The configuration of the CNN-GRU network is kept the same as in the single-channel data loss scenario. By maintaining the same network architecture, the evaluation ensures consistency, allowing for a fair comparison between the different cases. The data preprocessing and network training processes are carried out in the same manner as before. The only distinction lies in the number of input and output columns, as described earlier. In this way, the network’s ability to adapt to the increased complexity in multi-channel data loss scenarios can be rigorously assessed while keeping the other variables constant. This consistent setup helps in accurately evaluating the robustness and scalability of the proposed model as it handles more challenging recovery tasks with reduced inputs and increased outputs. The result of the multi-channel data recovery process is shown in Figure 10.

The results in Figure 10a illustrate the convergence process of the model across different data recovery scenarios. In the initial phase, all three scenarios show a significant drop in loss with rapid convergence during the first epochs. The model performs best in recovering data from two sensors as the loss value approaches zero. After around 350 epochs, the convergence curve stabilizes and gradually converges to zero, indicating accurate data recovery. For the scenario involving three faulty sensors, the convergence process is slower, taking over 700 epochs to achieve stability. In the case of recovering data from four sensors, the model initially converges very quickly, but the loss value does not approach zero. This suggests the model may have overfitted and is less effective at handling more complex scenarios where multiple sensors are faulty.

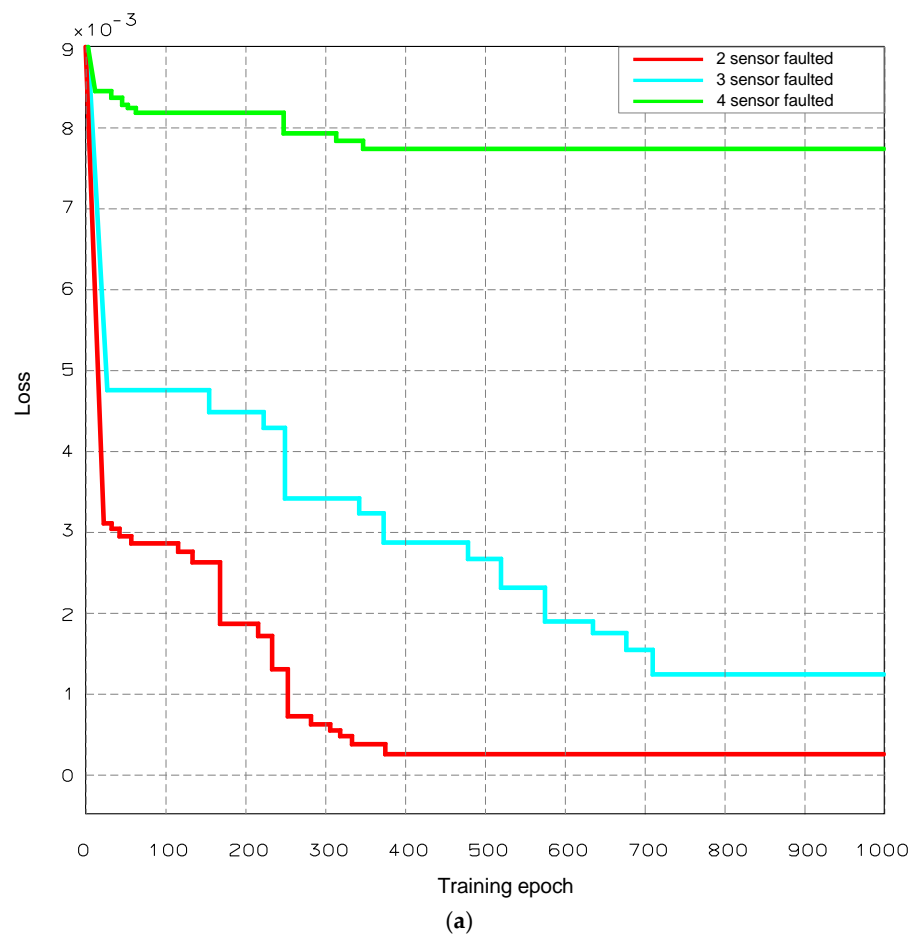
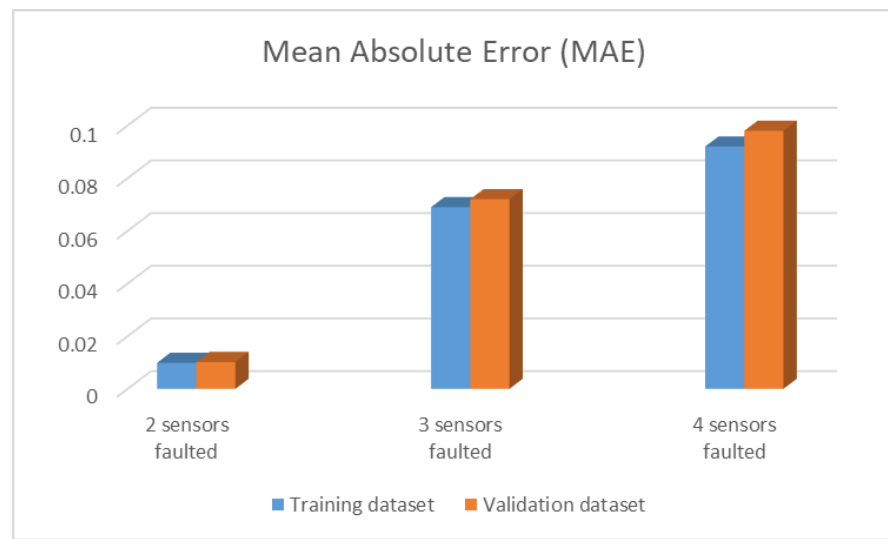


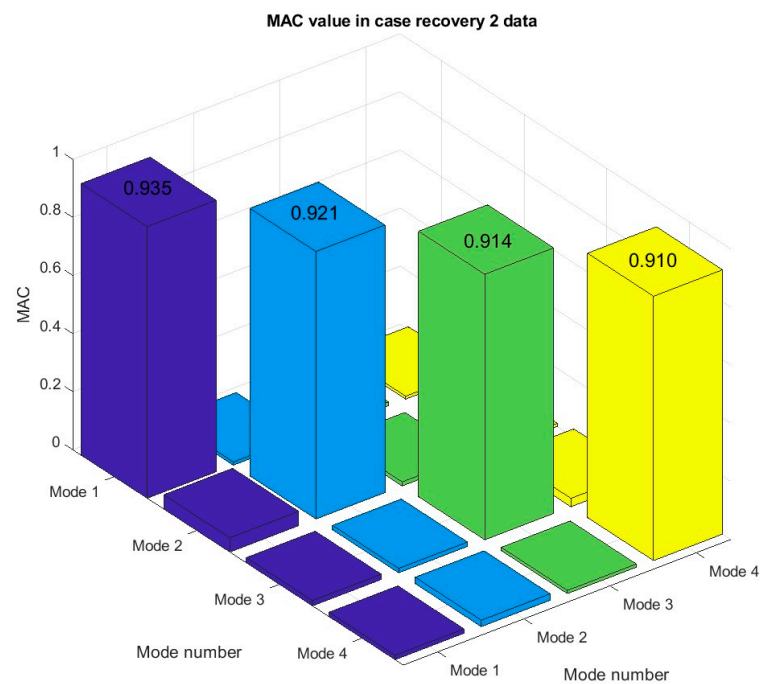
Figure 10. Cont.



(b)

Figure 10. Network training results in multi-channel data recovery scenario: (a) training convergence curve; (b) mean absolute error.

Figure 10b illustrates the MAE values across different scenarios. The initial evaluation indicates that as the number of faulty sensors increases, the MAE value between the recovered data and the actual data also rises accordingly. Specifically, when only two sensors are faulty, the MAE value remains low but experiences a significant increase in the other two scenarios. Notably, the discrepancy between the training and test sets becomes pronounced when three or four sensors are faulty. This suggests that as the number of faulty sensors increases, the effectiveness of the proposed model gradually diminishes. The datasets were also subjected to modal analysis under different data recovery scenarios. After comparing the vibration mode forms of the datasets, the MAC value is displayed in Figure 11.



(a)

Figure 11. Cont.

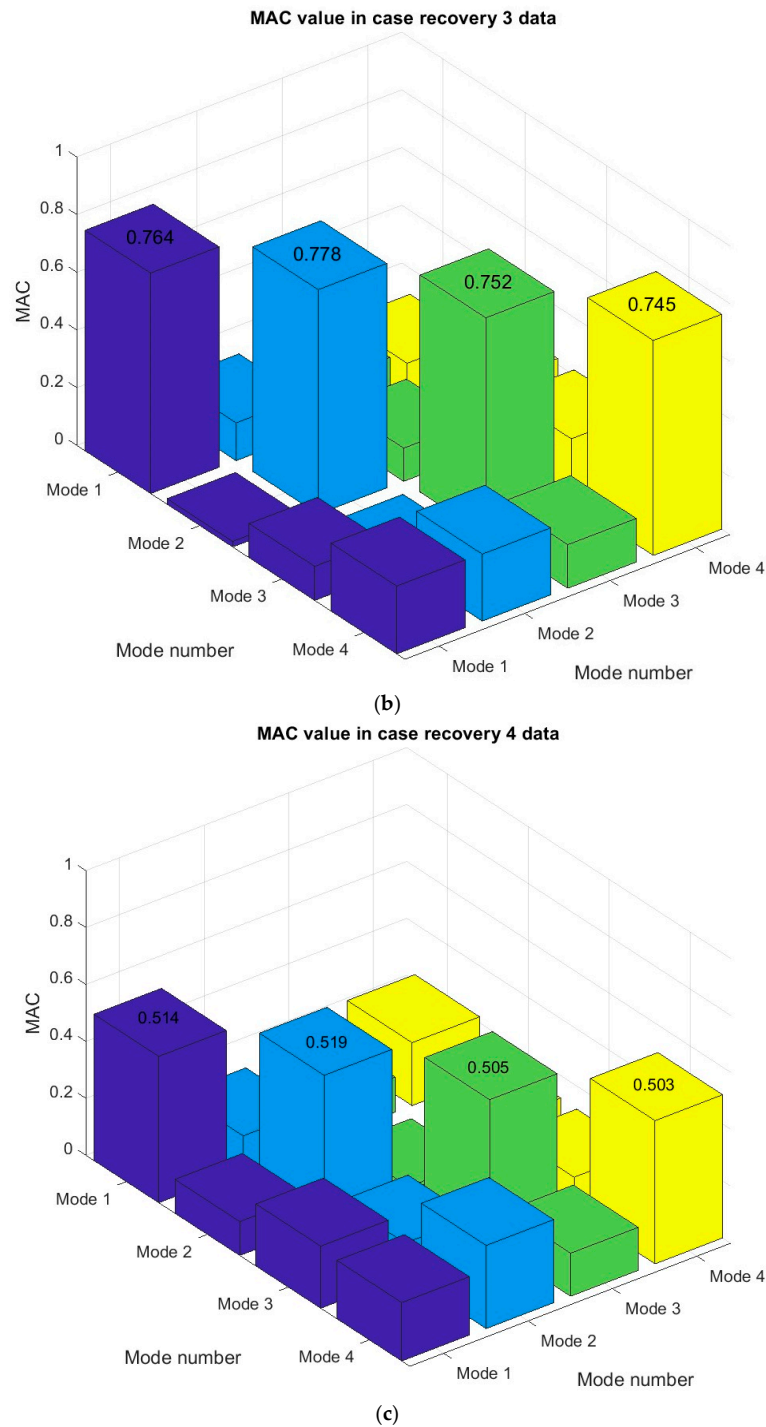


Figure 11. MAC values: (a) two-sensor data recovery; (b) three-sensor data recovery; (c) four-sensor data recovery.

There is a considerable similarity between the recovered and genuine data, particularly in the initial vibration modes, as demonstrated by the relatively high MAC values (0.910 to 0.935) in the event of recovering two defective sensors. Even while there is still a relatively strong association, the MAC values drop to a range between 0.745 and 0.778 when the number of defective sensors rises to three. The MAC values, however, sharply decrease to approximately 0.503 to 0.519 as the number of malfunctioning sensors rises to 4, indicating that the model has more incredible difficulty recreating the initial vibration features. Thus, the proposed model can work effectively for the cases of one to two sensor loss. However, the model’s efficiency will decrease when the number of sensors increases.

4. Conclusions

For the purpose of recovering sensor data in SHM, this study suggests combining CNN-GRU models. The efficacy of the recommended approach is demonstrated by the case study conducted on the Thang Long Bridge dataset. The model proved it could reliably recreate data despite errors or missing data. The approach has demonstrated significant promise in raising the precision of condition assessments for the monitored structure, decreasing mistakes, and increasing the correctness of recovered data. Some of the important findings this study makes are outlined below:

1. The integration of CNN and GRU effectively captures both spatial and temporal sensor data features, delivering superior performance in recovering lost data.
2. Compared to standalone models, the CNN-GRU model achieves higher accuracy in data reconstruction even in complex scenarios involving data loss from multiple sensors.
3. The model performs well when one to two sensors fail, but its accuracy decreases as the number of faulty sensors rises, revealing the model's limitations in handling more complex cases.
4. The CNN-GRU model shows significant potential for improving the effectiveness of structural health monitoring (SHM) systems by ensuring data integrity and accuracy.
5. The CNN-GRU model not only performs well on training data but also generalizes effectively to new data, demonstrating its potential for widespread application in other SHM data.

Author Contributions: Conceptualization, N.T.C.N., T.Q.M. and H.N.B.; methodology, N.T.C.N. and T.Q.M.; software, T.Q.M.; validation, N.T.C.N. and T.Q.M.; formal analysis, H.N.B.; investigation, N.T.C.N., T.Q.M. and H.N.B.; resources, T.Q.M.; data curation, N.T.C.N.; writing—original draft preparation, N.T.C.N. and T.Q.M.; writing—review and editing, N.T.C.N. and H.N.B.; visualization, N.T.C.N. and T.Q.M.; supervision, N.T.C.N.; project administration, N.T.C.N.; funding acquisition, N.T.C.N. All authors have read and agreed to the published version of the manuscript.

Funding: University of Transport and Communications (UTC)—T2024-CT-KDN-003/.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: This research is funded by the University of Transport and Communications (UTC) under grant number T2024-CT-KDN-003/.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Inaudi, D. Structural Health Monitoring of Bridges: General Issues and Applications. In *Structural Health Monitoring of Civil Infrastructure Systems*; Woodhead Publishing: Cambridge, UK, 2009. [CrossRef]
2. Wang, G.; Ke, J. Literature Review on the Structural Health Monitoring (SHM) of Sustainable Civil Infrastructure: An Analysis of Influencing Factors in the Implementation. *Buildings* **2024**, *14*, 402. [CrossRef]
3. Tran, M.Q.; Sousa, H.S.; Ngo, T.V.; Nguyen, B.D.; Nguyen, Q.T.; Nguyen, H.X.; Baron, E.; Matos, J.; Dang, S.N. Structural Assessment Based on Vibration Measurement Test Combined with an Artificial Neural Network for the Steel Truss Bridge. *Appl. Sci.* **2023**, *13*, 7484. [CrossRef]
4. Nguyen, N.T.C.; Tran, M.Q.; Sousa, H.S.; Van Ngo, T.; Matos, J.C. Damage Detection of Structural Based on Indirect Vibration Measurement Results Combined with Artificial Neural Network. *J. Mater. Eng. Struct.* **2022**, *9*, 403–410. Available online: <http://revue.ummto.dz/index.php/JMES/article/view/3286> (accessed on 24 December 2022).
5. Nhung, N.T.C.; Vu, L.V.; Nguyen, H.Q.; Huyen, D.T.; Nguyen, D.B.; Quang, M.T. Development and Application of Linear Variable Differential Transformer (LVDT) Sensors for the Structural Health Monitoring of an Urban Railway Bridge in Vietnam. *Eng. Technol. Appl. Sci. Res* **2023**, *13*, 11622–11627. [CrossRef]
6. Rabi, R.R.; Vailati, M.; Monti, G. Effectiveness of Vibration-Based Techniques for Damage Localization and Lifetime Prediction in Structural Health Monitoring of Bridges: A Comprehensive Review. *Buildings* **2024**, *14*, 1183. [CrossRef]
7. Azhar, A.S.; Kudus, S.A.; Jamadin, A.; Mustaffa, N.K.; Sugiura, K. Recent Vibration-Based Structural Health Monitoring on Steel Bridges: Systematic Literature Review. *Ain Shams Eng. J.* **2024**, *15*, 102501. [CrossRef]
8. Bunce, A.; Hester, D.; Taylor, S.; Brownjohn, J.; Huseynov, F.; Xu, Y. A Robust Approach to Calculating Bridge Displacements from Unfiltered Accelerations for Highway and Railway Bridges. *Mech. Syst. Signal Process.* **2023**, *200*, 110554. [CrossRef]

9. Thedy, J.; Liao, K.-W.; Tseng, C.-C.; Liu, C.-M. Bridge Health Monitoring via Displacement Reconstruction-Based NB-IoT Technology. *Appl. Sci.* **2020**, *10*, 8878. [[CrossRef](#)]
10. Bento, A.M.; Minh, T.Q.; Ferradosa, T.; Sousa, H.S.; Nguyen, H.X.; Campos e Matos, J. Study on the Model to Determine Riverbed Scour and the Influence of Bridge Construction on Riverbed Deformation. In *Analytical and Experimental Methods in Mechanical and Civil Engineering. OES 2023. Structural Integrity*; Springer Nature: Cham, Switzerland, 2024. [[CrossRef](#)]
11. Meo, M.; Zumpano, G. On the Optimal Sensor Placement Techniques for a Bridge Structure. *Eng. Struct.* **2005**, *27*, 1488–1497. [[CrossRef](#)]
12. Quang, M.T.; Sousa, H.S.; Duc, B.N.; Matos, J.C.; Bento, A.M.; Ferradosa, T.; Nguyen, H.X. Effect of Bridge Foundation Stiffness on Dynamic Behavior of Bridge Structure. In *IABSE Congress New Delhi 2023; Engineering for Sustainable Development: New Delhi, India, 2023*. [[CrossRef](#)]
13. Thadikemalla, V.S.G.; Gandhi, A.S. A Data Loss Recovery Technique Using Compressive Sensing for Structural Health Monitoring Applications. *Struct. Eng.* **2018**, *22*, 5084–5093. [[CrossRef](#)]
14. Thadikemalla, V.S.G.; Gandhi, A.S. A Simple and Efficient Data Loss Recovery Technique for SHM Applications. *Smart Struct. Syst.* **2017**, *20*, 35–42. [[CrossRef](#)]
15. Tran, M.Q.; Sousa, H.S.; Matos, J.C. Application of AI Tools in Creating Datasets from A Real Data Component for Structural Health Monitoring. In *Data Driven Methods for Civil Structural Health Monitoring and Resilience*; CRC Press: Boca Raton, FL, USA, 2023. [[CrossRef](#)]
16. Sinharay, S.; Stern, H.S.; Russell, D. The Use of Multiple Imputation for the Analysis of Missing Data. *Psychol Methods* **2001**, *6*, 317. [[CrossRef](#)] [[PubMed](#)]
17. Buzzicotti, M. Data Reconstruction for Complex Flows Using AI: Recent Progress, Obstacles, and Perspectives. *EPL Europhys. Lett.* **2023**, *142*, 23001. [[CrossRef](#)]
18. Singh, P. Systematic review of data-centric approaches in artificial intelligence and machine learning. *J. Inf. Technol. Data Manag.* **2023**, *6*, 144–157. [[CrossRef](#)]
19. Shibu, M.; Kumar, K.P.; Pillai, V.J.; Murthy, H.; Chandra, S. Structural Health Monitoring Using AI and ML Based Multimodal Sensors Data. *Meas. Sens.* **2023**, *27*, 100762. [[CrossRef](#)]
20. Wan, H.P.; Ni, Y.Q. Bayesian Multi-Task Learning Methodology for Reconstruction of Structural Health Monitoring Data. *Struct. Health Monit.* **2019**, *18*, 1282–1309. [[CrossRef](#)]
21. Zhang, Z.; Luo, Y. Restoring Method for Missing Data of Spatial Structural Stress Monitoring Based on Correlation. *Mech. Syst. Signal Process.* **2017**, *91*, 266–277. [[CrossRef](#)]
22. Sun, S.; Jiao, S.; Hu, Q.; Wang, Z.; Xia, Z.; Ding, Y.; Yi, L. Missing Structural Health Monitoring Data Recovery Based on Bayesian Matrix Factorization. *Sustainability* **2023**, *15*, 2951. [[CrossRef](#)]
23. Lu, W.; Teng, J.; Li, C.; Cui, Y. Reconstruction to Sensor Measurements Based on a Correlation Model of Monitoring Data. *Appl. Sci.* **2017**, *7*, 243. [[CrossRef](#)]
24. Zhang, J.; Huang, M.; Wan, N.; Deng, Z.; He, Z.; Luo, J. Missing Measurement Data Recovery Methods in Structural Health Monitoring: The State, Challenges and Case Study. *Measurement* **2024**, *231*, 114528. [[CrossRef](#)]
25. Jiang, H.; Wan, C.; Yang, K.; Ding, Y.; Xue, S. Continuous Missing Data Imputation with Incomplete Dataset by Generative Adversarial Networks–Based Unsupervised Learning for Long-Term Bridge Health Monitoring. *Struct. Health Monit.* **2021**, *21*, 1093–1109. [[CrossRef](#)]
26. Fan, G.; Li, J.; Hao, H. Lost Data Recovery for Structural Health Monitoring Based on Convolutional Neural Networks. *Struct Control Health Monit* **2019**, *26*, e2433. [[CrossRef](#)]
27. Fan, G.; Li, J.; Hao, H. Dynamic Response Reconstruction for Structural Health Monitoring Using Densely Connected Convolutional Networks. *Struct Health Monit* **2020**, *20*, 1373–1391. [[CrossRef](#)]
28. Fan, G.; Li, J.; Hao, H.; Xin, Y. Data-Driven Structural Dynamic Response Reconstruction Using Segment-Based Generative Adversarial Networks. *Eng. Struct.* **2021**, *234*, 111970. [[CrossRef](#)]
29. Jeong, S.; Ferguson, M.; Hou, R.; Lynch, J.P.; Sohn, H.; Law, K.H. Sensor Data Reconstruction Using Bidirectional Recurrent Neural Network with Application to Bridge Monitoring. *Adv. Eng. Inform.* **2019**, *42*, 100991. [[CrossRef](#)]
30. Kim, S.G.; Chae, Y.H.; Seong, P.H. Development of a Generative-Adversarial-Network-Based Signal Reconstruction Method for Nuclear Power Plants. *Ann. Nucl. Energy* **2020**, *142*, 107410. [[CrossRef](#)]
31. Huang, Y.W.; Wu, D.G.; Li, J. Structural Healthy Monitoring Data Recovery Based on Extreme Learning Machine. *Jisuanji Gongcheng/Comput. Eng.* **2011**, *37*, 16. [[CrossRef](#)]
32. Mayank Mishra Convolutional Neural Networks, Explained. 2020. Available online: <https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939> (accessed on 1 November 2024).
33. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)]
34. Yamashita, R.; Nishio, M.; Do, R.K.G.; Togashi, K. Convolutional Neural Networks: An Overview and Application in Radiology. *Insights Imaging* **2018**, *9*, 611–629. [[CrossRef](#)]
35. Abhishek Jain All About Convolutions, Kernels, Features in CNN. Available online: <https://medium.com/@abhishekjainindore24/all-about-convolutions-kernels-features-in-cnn-c656616390a1> (accessed on 1 November 2024).
36. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; The MIT Press: Cambridge, MA, USA, 2016; ISBN 978-0-262-03561-3.

37. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
38. Venkatesan, R.; Li, B. *Convolutional Neural Networks in Visual Computing—A Concise Guide*; CRC Press: Boca Raton, FL, USA, 2017. [[CrossRef](#)]
39. Amidi, A.; Amidi, S. *Recurrent Neural Networks Cheatsheet*; Stanford University: Stanford, CA, USA, 2018.
40. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. *Comput. Lang.* **2014**. [[CrossRef](#)]
41. Oğuz, A.; Ertuğrul, Ö.F. Introduction to Deep Learning and Diagnosis in Medicine. In *Diagnostic Biomedical Signal and Image Processing Applications with Deep Learning Methods*; Academic Press: Cambridge, MA, USA, 2023. [[CrossRef](#)]
42. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *Neural Evol. Comput.* **2014**. [[CrossRef](#)]
43. Cheng, G.; Povey, D.; Huang, L.; Xu, J.; Khudanpur, S.; Yan, Y. Output-Gate Projected Gated Recurrent Unit for Speech Recognition. In Proceedings of the Conference: Interspeech, Hyderabad, India, 2–6 September 2018. [[CrossRef](#)]
44. Liu, D.; Bao, Y.; He, Y.; Zhang, L. A Data Loss Recovery Technique Using EMD-BiGRU Algorithm for Structural Health Monitoring. *Appl. Sci.* **2021**, *11*, 10072. [[CrossRef](#)]
45. Nhung, N.T.C.; Nguyen, H.B.; Minh, T.Q. Reconstructing Health Monitoring Data of Railway Truss Bridges Using One-Dimensional Convolutional Neural Networks. *Eng. Technol. Appl. Sci. Res.* **2024**, *14*, 15510–15514. [[CrossRef](#)]
46. Le Khanh Hà Nội: Chiêm Ngưỡng Cầu Thăng Long Bắc qua Sông Hồng Sau Gần 40 Năm Hoạt Động. Đại Đoàn Kết–Cơ Quan Trung Ương Của MTTQ VN. 2023. Available online: <https://daidoanket.vn/ha-noi-chiem-nguong-cau-thang-long-bac-qua-song-hong-sau-gan-40-nam-hoat-dong-10258684.html> (accessed on 8 November 2023).
47. Bộ Giao thông Vận tải Quy Trình Thiết Kế Cầu Cống Theo Trạng Thái Giới Hạn - 22TCN 18-79. Bộ Giao thông Vận tải - Việt Nam 1979.
48. UCT company Báo Cáo Kiểm Tra Cầu Thăng Long- Dự Án Sửa Chữa Mặt Cầu Thăng Long. 2020.
49. PCB Piezotronics an Amphenol Company Accelerometers for Research & Development.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.