Smart Structures and Systems, *Vol. 30, No. 6 (2022) 613-626*
https://doi.org/10.12989/sss.2022.30.6.613

Structural health monitoring data anomaly detection by transformer enhanced densely connected neural networks

Jun Li [2a], Wupeng Chen [1b] and Gao Fan*[1]

[1] *School of Civil Engineering, Guangzhou University, Guangzhou 510006, China*
[2] *Centre for Infrastructure Monitoring and Protection, School of Civil and Mechanical Engineering, Curtin University, Bentley, WA 6102, Australia*

(*Received June 11, 2022, Revised October 15, 2022, Accepted October 30, 2022*)

Abstract. Guaranteeing the quality and integrity of structural health monitoring (SHM) data is very important for an effective assessment of structural condition. However, sensory system may malfunction due to sensor fault or harsh operational environment, resulting in multiple types of data anomaly existing in the measured data. Efficiently and automatically identifying anomalies from the vast amounts of measured data is significant for assessing the structural conditions and early warning for structural failure in SHM. The major challenges of current automated data anomaly detection methods are the imbalance of dataset categories. In terms of the feature of actual anomalous data, this paper proposes a data anomaly detection method based on data-level and deep learning technique for SHM of civil engineering structures. The proposed method consists of a data balancing phase to prepare a comprehensive training dataset based on data-level technique, and an anomaly detection phase based on a sophisticatedly designed network. The advanced densely connected convolutional network (DenseNet) and Transformer encoder are embedded in the specific network to facilitate extraction of both detail and global features of response data, and to establish the mapping between the highest level of abstractive features and data anomaly class. Numerical studies on a steel frame model are conducted to evaluate the performance and noise immunity of using the proposed network for data anomaly detection. The applicability of the proposed method for data anomaly classification is validated with the measured data of a practical supertall structure. The proposed method presents a remarkable performance on data anomaly detection, which reaches a 95.7% overall accuracy with practical engineering structural monitoring data, which demonstrates the effectiveness of data balancing and the robust classification capability of the proposed network.

Keywords: anomalous data; anomaly detection attention mechanism; deep learning; structural health monitoring

1. Introduction

In the past decades, numerous large-scale civil infrastructures with great social value have been equipped with comprehensive SHM systems for assisting the condition evaluation and maintenance, and predicting the remaining life of structures (Ni *et al*. 2009, Chen *et al*. 2011, Cross *et al*. 2013, Xia *et al*. 2013). SHM systems contain a variety of sensors to consciously measure the structural behavior and the working ambient conditions, which produce a large amount of data. Bao *et al*. (2019) stated that engineers implement sense, evaluation, and warning about the structural conditions in real time by analyzing the representative features of the measured data. Under this situation, guaranteeing the quality and integrity of the measured data is the premise for grasping reliable structural condition. However, due to system failures, sensor faults, transmitting interference and environmental effects, data anomaly is an inevitable issue that needs to be carefully addressed before data analysis and mining. As the increase in the data size, that is generally few gigabits a day for a SHM system, the development of an anomalous data detection method that can automatically and timely detect anomaly from large amounts of data is urgently required.

Discriminating the anomalous SHM data from normal measurement is the first but the most important step in exploring data anomaly. Kerschen *et al*. (2004) presented a procedure based on principal component analysis (PCA) theory to detect anomalies using principal angles differences between reference data and objective data. Abdelghani and Friswell (2004) used the residuals generated by the modal filtering and parity space approach to detect data anomalies and demonstrated a good performance. Thiyagarajan *et al*. (2017) used an autoregressive integrated moving average (ARIMA) model to detect anomalies. The data points outside of the 95% confidence interval of the ARIMA forecasted data are considered as anomalies. Kullaa (2013) used the minimum mean square error (MMSE) to perform sensor fault detection, identification and correction. Yuen and Mu (2012) proposed a probabilistic method to detect outliers by quantifying the probability of data points being outliers. The probability of outlier for each data point can be computed based on the posterior probability density

*Corresponding author, Ph.D., Professor,
E-mail: gao.fan@gzhu.edu.cn
[a] Associate Professor, E-mail: junli@curtin.edu.au
[b] Master Student, E-mail: 2112016128@e.gzhu.edu.cn

ISSN: 1738-1584 (Print), 1738-1991 (Online)

function which is determined by the Bayes theorem. Yi *et al*. (2016) used cumulative sum chart for the detection of small but persistent shifts in the high-rate GPS carrier-phase measurements and achieved encouraging experimental results on a long-span bridge. Rabatel *et al*. (2011) detected monitoring data anomalies by comparing new patterns with all patterns extracted from historical data that describe normal behavior. Wan and Ni (2018) proposed a Bayesian modeling approach with Gaussian processes for future responses prediction and data anomaly detection. The existing approaches for structural data anomaly detection has achieved remarkable progress under specific conditions. However, these methods may be inefficient when processing large quantities of data, or detecting multiple types of anomalies within one framework.

To improve the detection of complex anomaly patterns from larger volume of SHM data, machine learning techniques have been widely investigated recently. Chenglin *et al*. (2011) employed chaos particle swarm optimization algorithm and support vector machine (CPSO-SVM) for fault diagnosis of sensors. It was validated that CPSO-SVM recognized four faulted types of wireless sensor with high accuracy. For the case of fewer faulty sensors. Lo *et al*. (2015) introduced a Kalman filter-based group test method for sensor fault detection, which accurately detect various faults. Ibarguengoytia *et al*. (2007) proposed an algorithm based on Bayesian network for intelligent sensor validation real time under working environments. The algorithm utilized a Bayesian network, which represents the dependencies and independencies among all the sensors, for the fault detection by estimating the value of a sensor with respect to the most related variables. Arul and Kareem (2020) employed the shapelet transform to search for the best shape-based feature representation, and then the Euclidean distance between the discovered shapelet and the time series in learning set was used to train a random forest classifier for data anomaly classification. The algorithm achieved a high overall accuracy of 93% on the test data measured from a SHM system installed on a large span bridge, but the performance for some certain anomaly classes needs more improvement. Deep learning, as a new advanced technique in machine learning research, has attracted growing attention in data anomaly detection. Smarsly and Law (2014) embedded a multi-layer pre-trained artificial neural network (ANN) into wireless sensor nodes of the SHM system for autonomous detection and isolation. The validity of the method was verified by identifying drift and bias faults. Fu *et al*. (2019) designed an ANN that trained by the data from faulty sensors containing three categories of anomalies (spikes, drift and bias). The trained ANN was tested with data collected from Jindo Bridge, and a high overall accuracy in identifying faulty sensors is achieved. Bao *et al*. (2018) presented a deep neural network for automatic anomaly detection based on computer vision and deep learning techniques. The one-dimensional original time series signals were converted into two-dimensional grayscale images by sections, and each image was manually labeled. This method achieved an 87% global accuracy rate in classification of anomalies on one year data collected from a long-span bridge. To further enrich the input information and improve the detection capability in an existing study Bao *et al*. (2018), Tang *et al*. (2019) visualized the data in time and frequency domains respectively and fused those two single channel images into a single dual-channel image. The overall accuracy increased to 93.5%, which demonstrates that the frequency information of data is beneficial to anomaly detection. Furthermore, imbalanced training dataset was found to have a negative effect on the accuracy of data anomaly detection.

Despite that imbalanced training dataset is proved harmful, the prepared datasets from practical measured data are not ideally balanced, because sensors operate normally for most of the cases, while data anomaly is a rarely and randomly occurred short-time phenomenon in SHM systems. The numbers of occurrence for each class of anomaly are also unequal. The unsupervised learning-based approach can circumvent the problem of data imbalance by leveraging the normal data only to train a mathematical model that represents the normal data features. For instance, Mao *et al*. (2020) proposed an unsupervised learning approach based on generative adversarial networks (GAN) and autoencoder to distinguish the normal and anomalous data with a high accuracy. However, unsupervised learning-based approach simplifies the data anomaly detection as a binary classification problem, which loses the information of anomalies class, even if a specific class of anomalous data is very informative for structural analysis. The existing research has shown promising results in addressing data anomaly detection problem with certain limitations in practical engineering. An effective approach that can mitigate the effect of imbalanced dataset while accurately classifying the anomaly is still a worth studying topic.

In this article, a novel method for data balancing and anomaly detection is proposed based on data-level and deep learning techniques. A new data balancing method based on data-level technique Krawczyk (2016) concentrating on the training dataset perfection is developed to alleviate the data imbalance effect and satisfy with the standard learning algorithm in supervised learning framework. For multi-classes data anomaly detection, a densely connected neural network enhanced by the advanced Transformer encoder (TDNet) is designed. The proposed network inherits the advantage of DenseNet Fan *et al*. (2021), Huang *et al*. (2017), which is outstanding in extracting detail features of structural responses. Furthermore, with the assistance of Transformer encoder, which is composed of multi-headed attention (MHA), the long-distance dependency and global features extraction is further improved, leading the extracted high-dimensional features more robust to data anomalies detection. It is also remarkable that the proposed TDNet uses time series directly as the network input without any manually feature transformation, which has a high degree of automation and less subjective effect on the performance.

This paper is organized as follows. Section 2 shows the process of data balancing method to simulate anomalous data. Three common anomalies with distinguishable characteristics, namely outlier, trend and missing, are simulated using normal samples. Section 3 describes the

detailed configuration of TDNet. The structure and advantages of DenseNets and Transformer encoder, which are parts of the proposed network, are described in detail in this section. Section 4 introduces the numerical studies on a seven-story steel frame model, which explores the effects of training sample size and noise on the classification performance of the TDNet. Besides, visualization of the output of the internal layers of TDNet is shown to explain the learned knowledge. Section 5 conducts practical engineering studies with Guangzhou New Television Tower (GNTT) to evaluate the applicability of the proposed approach for data anomaly detection. Section 6 concludes the findings of this paper and discusses potential further future research.

2. Simulation of anomalous data

framework of the proposed approach includes two steps: training dataset balancing and data anomaly detection. In the first step, a novel approach based on data-level technique is proposed for generating anomalous data. Three typical classes of data anomaly, including missing, trend and outlier, are artificially analyzed to determine the rough features of each class. Based on those extracted features, a training dataset is generated by processing the measured normal data as anomalous data, where each class of anomalies can be equally produced. In the second step, a specific TDNet that developed based on a modified DenseNet for structural response reconstruction Fan *et al*. (2021a), is trained based on the prepared dataset for anomaly detection. For deep learning-based classification task, effectively extracting robust features from input data is the key to distinguishing the discrepancy of classes in high-dimensional spaces. Therefore, attributed to the outstanding ability of MHA involved in Transformer encoder to learn the long-distance correlation among time series data, a specific Transformer encoder is imbedded into the DenseNet to ameliorate the capability of data anomaly detection. It is worth noting that due to the outstanding capability of TDNet in extracting hidden features of structural response, a training dataset generated based on rough features is sufficient for anomaly detection.

The proposed network for data anomaly detection is trained in a supervised manner. A dataset with three classes of anomalous data and normal data with correct labels needs to be prepared for training the network. It is a common procedure to manually select and label data anomalies from the large amount of SHM data. Although anomalous data only exist in a small fraction of the SHM data, it is still time-consuming to select and label sufficient training samples. Besides, the prepared dataset consisting of selected anomalous data samples from measured data is always imbalanced, which means the numbers of training samples for anomalous classes are unequal. Imbalanced training dataset may cause the network to learn inaccurate features and produce biased predictions. In contrast, simulating anomalous data by utilizing the characteristics of anomalies instead of burdensome manually picking is more available for practical engineering. Anomalies are essentially temporal events, with normal data suddenly transforming into anomalous conditions and rapidly reverting back to normal. Inspired by the process of real anomaly generation, the anomalous data are simulated by adding specific anomalous patterns to normal data. The raw normal time series structural responses are divided into segments in a length of 1024 sampling points. All these segments are subsequently split as four equal groups, where one group of segments remains unchanged, and the remaining three are simulated as three types of anomalous data, that are missing, trend and outlier.

Data loss is a frequent data anomaly in wireless transmission. It may be caused by hardware fault, signal interference, environmental variation, etc. The way to simulation the missing data is to replace the values of a small succession of data points by same length of zeros. The starting time point and the duration of data loss is randomly selected in each sample to imitate the real-world sensor data loss process in SHM system. The starting point of data loss is within 1 to 897 and the length interval of the missing data segment is set to between 128 and 512.

Data trend is often occurred since sensor faults and shock of sensors or cables. For trend anomaly simulation, a normal data segment adds a small segment of consecutive data points in a linear relation of the first order to form the trend anomaly data segment. The location of the trend anomaly, the length of the anomaly data point, and the slope of the linear function are all random. The starting point and length interval of the trend data are the same as the missing data, 1 to 897 and 128 to 512, respectively. The slope intervals are set as union of intervals 5×10^{-6} to 5×10^{-5}, and -5×10^{-5} to -5×10^{-6}, which ensure that the simulated data samples have distinguishable upward or downward feature of trend.

Outlier in structural response is an instantaneous spike caused by hardware fault, transmission error or local shocks on or near the sensor. It is a short period of sampling points that do not coincide with the data distribution and much larger than the adjacent points. Consequently, the outlier anomaly is modeled as adding a short period of abnormal data points with random length into any time step of normal data segments. The number of outliers in a data piece is randomly selected between 1 and 20 and the maximum value in one outlier is limited to between 5 and 9 standard deviations of the corresponding normal data. The procedures of simulating the three anomalous samples are

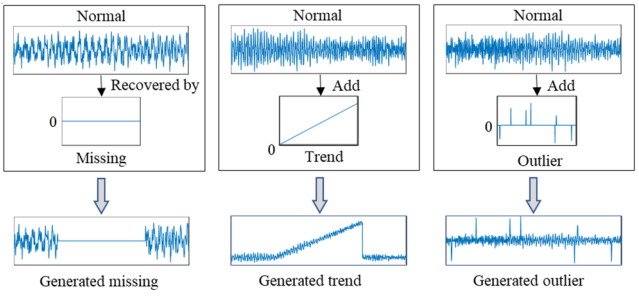


Fig. 1 The processes of 3 types of anomalous samples simulation

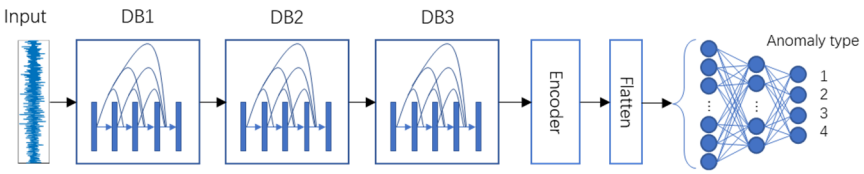


Fig. 2 The architecture of TDNet

schematically shown in Fig. 1. After the data simulation is completed, the normal data and the simulated anomalous data are labeled separately in order to adapt to the classifier trained in a supervised manner.

3. TDNet

The specific TDNet is developed based on the architecture of DenseNet with Transformer encoder embedded for anomaly detection. It is trained in a supervised manner by using prepared training data that is composed of paired input structural response and the corresponding label, indicating the class of input. The generated training dataset contains three common classes of abnormal data and one class of normal data. The time-domain structural responses are directly fed into the proposed network for feature learning without any manually feature transformation. Then the learned features with high level of abstraction are processed automatically by fully connected layers to complete the anomaly classification. The output of the network is a vector in a length of 4, and the value of each element represents the probability of the input segment belongs to the corresponding anomaly class. It is worth noting that the input to the model should carry sufficient structural vibration information, and meanwhile, meet the size requirement of data compression by the downsampling. Therefore, the duration of input segments should be at least longer than the period of the first vibration mode of the target structure. In this paper, the downsampling process half the length of feature maps each time, so the length of input segment is set as 1024 sampling points for all the following studies. Optimized hyperparameters are also critical for training deep learning models to determine the global optimal model parameters. However, hyperparameter optimization is a multiparametric optimization problem, determine the optimal combination need significant computation power to conduct the model training hundreds of times, which is impractical for uncommercial developers. Therefore, in this research, the approximate optimal hyperparameters including the batch size and learning rate range are initialized empirically based on the previous finding in Fan *et al.* (2021a, b) of the manuscript. The batch size is increased from 4 to 8 and finally set as 16 that is mainly limited by the GPU memory size. It is recommended to use as large size of batch as possible for advancing the stabilization of training. The interval of learning rate is initially set between 1×10^{-8} and $\times10^{-3}$. It is then determined by evaluating the recognition accuracy of models with randomly sampled learning rate from this range, to gradually approaching the optimal. The range of learning rate is compressed successively and finally reached the optimal for this classification task as 1×10^{-5}.

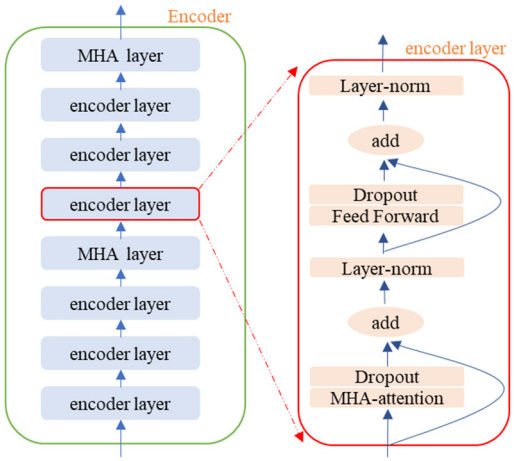


Fig. 3 The architecture of the embedded Transformer encoder

3.1 The architecture of TDNet

The elaborated architecture of the proposed TDNet is shown in Fig. 2. The first layer aims to enrich the shallow features of the original input data by applying the convolution and padding operations with rich convolution kernels. The padding operation allows the length of the produced feature maps to be the same with input length divided by the convolutional stride. After enriching the shallow features from the original input data, three successive dense blocks are employed to gradually extract the higher representative features of the input. In each of the dense blocks, the number of kernels and kernel size are kept the same. After each dense block, the length of feature map is halved to fine higher abstractive level of features. To further extract long-distance dependency features, the modified Transformer encoder is embedded between the feature extraction module and classification module as shown in Fig. 3. The transformer encoder in the proposed network consists of six encoder layers for feature extraction and two MHA layers to resize the feature maps to fit the later classification.

Note that convolution is a linear operation and thus cannot learn the complex nonlinear relationships of the input. To improve the capability of TDNet, the output of convolution operation is sent to an activation function to transfer the connection from linear to nonlinear. An advanced activation function named leaky rectified linear unit (Leaky ReLU) developed from rectified linear unit (ReLU) is used in this study. The Leaky ReLU activation

function inherits the advantage of the ReLU function to mitigate the gradient vanishing while overcoming the defect of ReLU, that is, the gradients stop propagating when most of the neurons return zero. The expression of the Leaky ReLU function is as follows

$$LeakyReLU(x) = \begin{cases} x & if\ x > 0 \\ 0.1x & otherwise \end{cases} \tag{1}$$

The dropout technique, which has been shown to be effective in mitigating overfitting, is applied in the proposed training framework (Srivastava *et al*. 2014). By discarding a portion of randomly selected neurons when training with different batches of samples, dropout enhances the generalization capability of the network. In TDNet, each convolutional layer is followed by a dropout layer and an activation function layer.

The last part of the proposed network framework is the classification module consisting of a flatten layer and three full connected layers. High-level features delivered from feature extraction module are unfolded as a feature vector by the flatten layer. The feature vector is then mapped to the sample marker space for classification through three full connected layers. The mapping formula for one fully connected layer *l* can be expressed as

$$x^{l+1} = \sigma(w^l x^l + b^l) \tag{2}$$

where x^l is the input of the *l*th fully connected layer. w^l and b^l refer to the weight and bias parameter matrixes, that are updated during network training. $\sigma(\cdot)$ denotes the activation function and x^{l+1} denotes the output of the *l*th fully connected layer. The ReLU activation function is used in the first three convolutional layers for establishing mappings between features and output. For outputting the probability of the input sample that belongs to each class, Softmax activation function is employed in the last fully connected layer, where the input data is defined as the class with the maximum probability. Softmax activation transforms the input vector into a probabilistic representation, which is defined as

$$p_k = \frac{e^{z_k}}{\sum_{i=1}^{K} e^{z_i}}, \qquad k \in [1, K] \tag{3}$$

where *K* is the amount of class and p_k denotes the output probability for a certain class *k*. z_k refers to the value of the *k*th element in the input vector. Due to the limitation of computing and memory resources, the training dataset with a massive number of samples is fed into the network in batches to turn model parameters during the training process. For the classification tasks, the target of the network training is to minimize the difference of the discrete probability distributions between real classes and predicted classes. Cross-entropy is used as the objective function to measure the difference of probability distribution between the true and predicted classes, which is defined as

$$Loss = -\frac{1}{N}\left[\sum_{n=1}^{N}\sum_{k=1}^{K} y_k^n \log(p_k^n)\right] \tag{4}$$

where *N*, *K* denote the total number of samples and classes, respectively. p_k^n is the predicted probability of *n*th training sample that belongs to the class *k* data anomaly. y is a one-hot vector denoting the input of cross-entropy and y_k^n refers to the value of the *k*th element of the one-hot vector corresponding to sample *n*. Following the aforementioned procedure, TDNet can be properly trained in the supervised manner. The trained network in test can efficiently extract robust features of subsequent measured responses and classify the anomalous data in almost real time by forward only computation.

3.2 DenseNet

The structure of densely connected network is commonly used to perform computer vision tasks such as image classification, image semantic segmentation, etc. For SHM related tasks that most of the data are one-dimensional structural responses, Fan *et al*. (2021a) modified the structure of original DenseNet to adapt response reconstruction, which is a one-dimensional pixel to pixel task. DenseNet proved to be effective in extracting hidden features of structural responses, which should be also available for anomaly classification, that the input and output are the measured responses and anomaly classes, respectively. Compared with the traditional CNN, DenseNet equipped with dense blocks (as shown in Fig. 4) reuse extracted features with different abstraction, which can mitigate gradient vanishing, facilitate feature propagation, and reduce the number of required model parameters Huang *et al*. (2017).

A key characteristic of dense block is the dense connectivity pattern. Each layer in an individual block connects every other subsequent layer in a feed-forward fashion by skip connection, which preserves and reuses features at different levels. Each of the dense blocks consists of four densely connected convolutional layers with the same kernel size to ensure the consistency of feature maps in the block for feature fusion. The shuttled features maps from the preceding layers are concatenated together as the input of the successive layer. In contrast with the layer-to-layer connected CNN with *n* convolutional layers and *n* connections, dense connection has *n* layers but *n*(*n*+1)/2 connections. This means that each layer receives and fuse a wealth of features with variety levels of abstraction from all the previous layers, thus strengthening the information flow among the network. It should be also

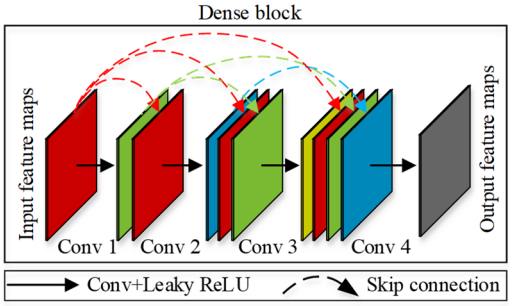


Fig. 4 A dense block with four convolutional layers

noted that when using the backpropagation algorithm for gradient updating, deep layer-to-layer connection is restricted to gradient vanishing issue in the training process. Gradient vanishing significantly decreases the training speed and may cause non-convergence problem. In this case, skip connection directly pass bottom features to the shallow layers of the network allow the delivery of gradient in a much shorter way, which remains the gradient to tune the parameters of shallow layers and thus alleviates the gradient vanishing.

3.3 Transformer encoder

Despite that DenseNet is capable of learning detail features of structural responses, the extraction of global features and long-distance correlations is ordinary. On the contrary, MHA as a core of Transformer encoder has assumed dominant position in natural language processing such as machine translation and text generation, which can be attributed to its outstanding capability in extracting long-distance dependency of sequences. MHA is an integration of multiple parallel self-attention modules that inherit the advantage of self- attention (Vaswani *et al*. 2017). It is therefore necessary to understand the mechanism of self-attention in learning the long-distance dependency and global features. Self-attention, also called intra-attention, compute the representative features of a sequence or matrix to relate all the elements or vectors. In this way, regardless of the distance between any two elements or vectors, the correlation can be computed by auto-correlation or cross-correlation. Those elements or vectors that are highly correlated with others are assigned with larger weights when producing the output feature matrix, which can be considered as paid more attention to those elements or vectors. The highlighted features contribute significant in the feature matrix, while the elements or vectors that are weakly connected with others are dismissed in the subsequent feature extraction. In addition, the self-attentive enables computational parallelism when processing sequences compared to RNN. The computation process of the self-attention mechanism in TDNet can be divided as two steps. The input of feature maps that are matrixes, are multiplied with three trainable weight matrixes (W_Q, W_K, W_V) to obtain Q (query), K (key), and V (value). Then the weight distribution of V is determined by calculating the similarity of Q and K, according to the following formula

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{5}$$

The output of the self-attention operation is computed as a weighted sum of the values, where the weight assigned to each value is computed as the dot product of query and key. After the self-attention processing, the feature vector at a certain position contains the original information and additional information regarding the relation with other vectors. From the aspect of feature engineers, self-attention enriches features that represent the long-distance dependency and global characteristics, leading the network to pay more attention to those elements or vectors that are strongly correlated with others. MHA integrates multiple parallel self-attention to allow the model to jointly attend to the information from different representation subspaces at different positions. The computation of MHA can be expressed as the following equation

$$MHA = Concat(head_1, \dots, head_h)W \tag{6}$$

$$head_i = Attention(QW_i^Q, KW_i^k, VW_i^v) \tag{7}$$

MHA as a core component of Transformer encoder is imbedded in the encoder layers. Transformer encoder is composed of multiple stacked encoder layers. An encoder layer consists of one MHA layer, two dropout layers, two normalization layers and one feedforward layer for features compression. The specific connection of the encoder layer is shown in Fig. 3. The skip connection technique is used in the encoder layer to strengthen the feature propagation and consequently boosts the convergence efficiency. The dropout technique, which can mitigate overfitting is applied in the encoder layer. In addition, layer normalization is also used to maintain the stability of the model during training.

4. Numerical studies

In this section, numerical studies are conducted to validate the effectiveness of the proposed approach for data anomaly detection by using simulated time-domain acceleration responses. A finite element model (FEM) of a seven-story steel frame with a width of 0.5 m, a height of 2.1 m, as shown in Fig. 5, is constructed to generate dataset. The mass density, Young's modulus are 7850 kg/m^3 and 210 GPa, respectively. The cross sections of the column and beam elements are 50 mm × 5 mm and 50 mm × 9 mm, respectively. To simulate the mass of the floor of this steel frame structure model, each floor is equipped with two

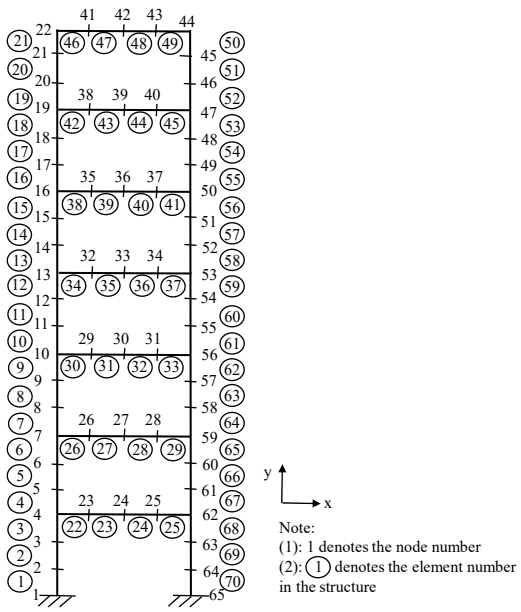


Fig. 5 FEM of the steel frame structure

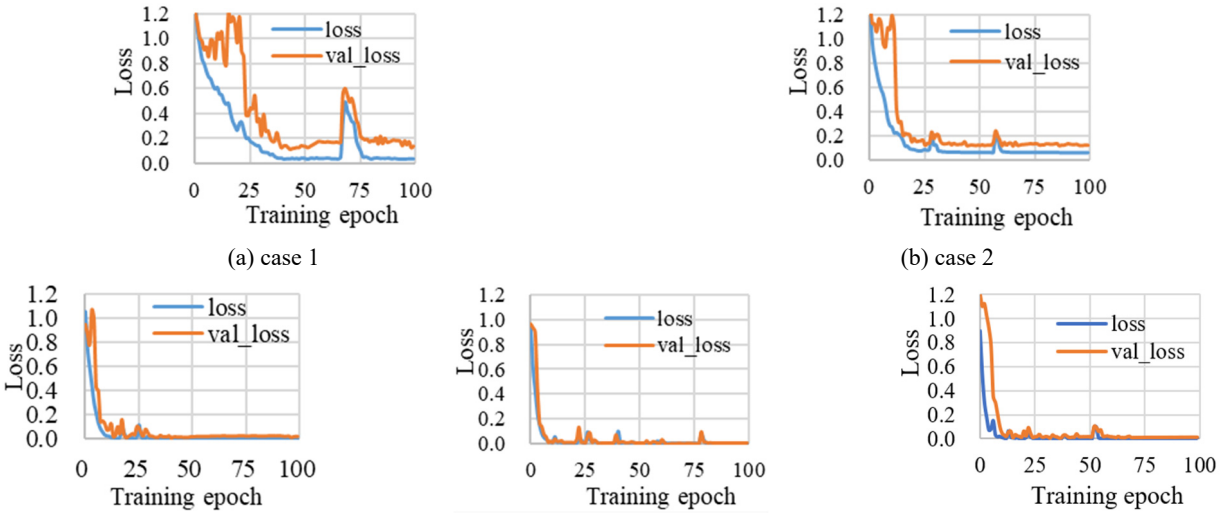
mass blocks weighing 4 kg at a quarter and three-quarter locations, respectively. The length of finite element is defined as 0.1 m, and there are 70 planar elements and 65 nodes in total. Each node in the FEM has three degrees of freedom, the translational displacements in the x and y directions and the rotational displacement in the x-y plane. Fixed supports are modeled at the position of nodes 1 and 65 to restrict the corresponding degrees of freedom. The x direction acceleration responses of node number 4, 7, 10, 13, 16, 19, and 22 corresponding to the responses of each floor are recorded as the raw measured data. The training, validation and testing datasets are generated using the data processing techniques and data balancing method as mentioned in Section 2. To achieve the optimal performance efficiently, the effect of the volume of training dataset on the accuracy of data anomaly classification is comprehensively investigated. Meanwhile, the effect of noise on classification accuracy is also investigated.

4.1 Data preparation

The numerical studies simulate 200-minute acceleration responses of the frame model under ambient excitations along x direction to validate the proposed approach. The Newmark-beta method is applied to compute the structural responses at a 2000 Hz sampling rate. As the first seven natural frequencies of this model are within 30 Hz, the raw simulated responses are processed by a low pass filter with a 30 Hz cutoff frequency, and then a downsampling procedure is implemented to reduce the sampling rate to 100 Hz. The data preprocessing process eliminates redundant information and significantly reduces the data size, thus greatly facilitating TDNet to extract features more efficiently. The training and testing samples is prepared by firstly segmenting the long time series data as small segments using a moving window in a length of 1024 without overlap. There is a total of 8400 segments generated from the 200-minute response and all of them are normal data samples. Here, to investigate the minimum data size required for optimize TDNet, a certain portion of samples are randomly selected. The selected samples are then divided into four equal groups for data balancing processing. One of the groups is regarded as normal samples, and the other three groups are processed as three types of abnormal data, including missing, trend and outlier. The processed three groups of anomalous data samples and one group of normal data samples form a dataset for validating the proposed anomaly detection method. The dataset is split as 70%, 20% and 10% for training, validation and testing.

4.2 Effect of the number of training samples

In this section, the effect of the number of training samples on the accuracy of data anomaly classification is investigated. Five cases with increasing number of samples, that are 840, 1680, 3360, 5040, and 6720 corresponding to 10%, 20%, 40%, 60%, 80% of the total samples, are prepared for training, validation and testing. Training of the proposed network is conducted in batches and the batch size is chosen as 32 considering the GPU memory. Adam, an efficient optimization algorithm, is used to update the parameters of the proposed network and the learning rate in the Adam optimizer is set as 1×10^{-5}. To enhance the generalization capability of the network, the dropout rate is chosen as 0.5 in the training process of the proposed TDNet. The proposed network is trained by the training dataset for 150 epochs. Note that the training setup keeps constant for each subsequent network training. As shown in Fig. 6, the fitted curves of Cases 3 to 5 remain stable after a rapid and steady decline. The fitted curves of Cases 1 to 2 show different degrees of oscillation during descending, which indicates that the fewer number of training samples affect the stability of network training when other conditions are kept consistent. Compared with Case 2, the fitted curve of Case 1 with a smaller number of training samples shows a more severe oscillation, which is the reason for the degraded classification performance. On the



(a) case 1 (b) case 2

(c) case 3 (d) case 4 (e) case 5

Fig. 6 The convergence curve for five cases

contrary, the fitted curves for Cases 3 to 5 do not show large differences with the increasing number of training samples, which indicates that the number of samples in Case 3 is sufficient for stable training of the network. On the other hand, insufficient sample volume can lead to overfitting as shown by Case 1 and 2. The validation loss is always larger than the training loss until reaching stability. When the number of training samples is insufficient but the capability of network is large, it is difficult to learn robust features from training dataset. Hence, the generalization ability of the network is reduced as represented by the low classification accuracy in the test.

Three indexes, including precision, recall, and F_1 score, are employed to measure classification performance. F_1 is a harmonic mean of the precision and recall. These metrics are defined as

$$precision = \frac{tp}{tp + fp}, \quad recall = \frac{tp}{tp + fn} \tag{8}$$

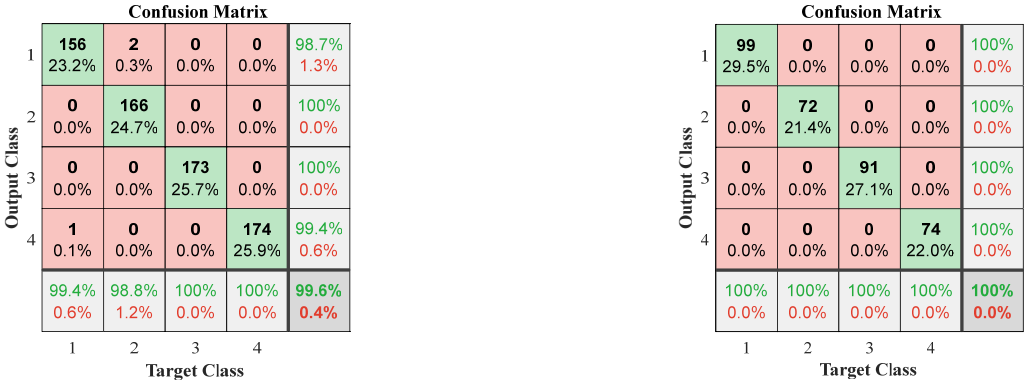
$$accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{9}$$

$$F1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = 2\frac{precision \cdot recall}{precision + recall} \times 100\% \tag{10}$$

where tp (true positive) refers to the number of samples correctly classified by the model as the object class. fp (false positive) denotes the number of samples from other classes misclassified by the model as the object class. fn (false negative) denotes the number of samples that are misclassified into other classes from the object class. tn (true negative) represents the number of samples from other categories that are correctly classified by the model. The precision for reliability evaluation is based on the classification results, while recall is based on the ground truth. The accuracy reflects only the overall classification performance rather than a specific class. As an overall consideration, the F_1 score, which comprehensively takes precision and recall into consideration, is employed as a performance indicator. The validation and test results for all the cases are shown in Fig. 7. Here, the F_1 cores are represented by the bars of histogram and the overall accuracies are marked in the brackets. As can be seen from Fig. 7, the networks trained by dataset of Case 3 to 5 demonstrate excellent performance in anomaly detection. The trained networks achieved over 99% overall accuracies, and in some cases, the accuracies are even close or equal to 100%. The F_1 scores for four classes on both the validation and testing dataset are also more than 0.99, indicating strong capability of determine all kinds of common data anomalies. Here, Case 3 reaches the optimal performance with a smaller training data size. However, when the number of training samples reduced to 1680, the overall accuracies of validation and test data decrease to 96.4% and 94.6%, respectively. The F_1 score of Case 2 is lower than that of Case 3 to 5 but still above 0.91 for all classes on

(a) Validation set (b) Test set

Fig. 7 Classification results of 5 cases



(a) Validation set (b) Test set

Fig. 8 The confusion matrix of Case 3 (Anomaly patterns: 1—normal, 2—outlier, 3—trend, 4—missing)

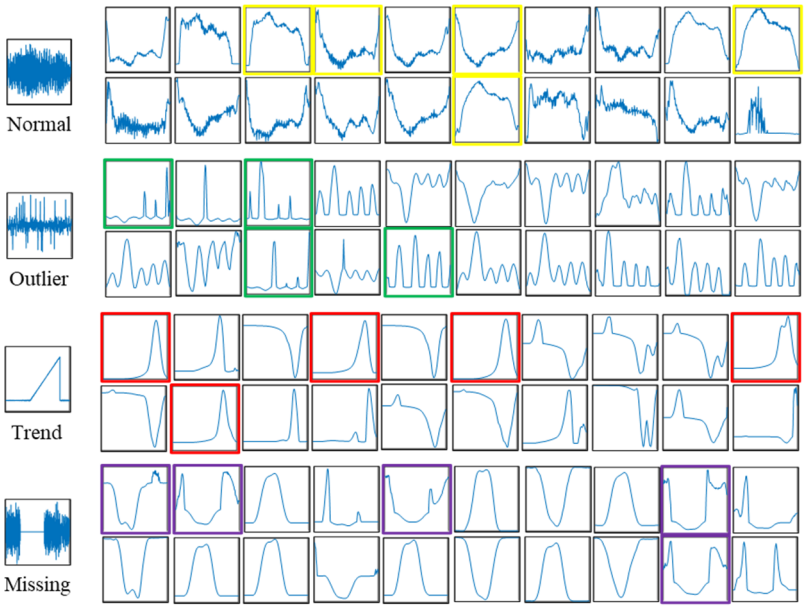


Fig. 9 Visualization of the output of the DenseNet module

both datasets, indicating the classification capability of the corresponding trained model is still acceptable. In contrast, network of Case 1 that are trained using 840 samples only, loss the effectiveness in classifying data anomaly with low F_1 scores for at least two classes, although the overall accuracies for validation and test sets reach 86.9% and 84.5%. Collectively, the classification performance of network growing with the increase of training samples until 3360. It is worth mentioning that leveraging 3360 training samples for deep learning to extract hidden features of data is a tough task. The results demonstrate the remarkable capability of TDNet in learning the pattern of structural responses and classifying the anomalous data. The effectiveness of TDNet is further elaborated by the confusion matrices of Case 3, as shown in Fig. 8. The confusion matrix records the results of the classification, including the number of predictions and their corresponding proportions. Two reliability indicators regarding the classification results, precision and recall, are highlighted in grey alongside the confusion matrix. As shown in Fig. 8, the miss classification rarely occurs in both validation and test datasets. It can be found that only one normal data sample is classified as anomalous and two missing samples are defined as normal data.

To further explore the inner learned feature of TDNet, the output of DenseNet is visualized when using a representative sample of each pattern as input, respectively. The input samples and the corresponding visual output of DenseNet are shown in Fig. 9. In Fig. 9, The colored border refers to the output of the DenseNet layer with outline feature with the same pattern with the corresponding type of input sample. The yellow border refers to the outline of feature map consistent with the normal type, where green corresponds to outlier, red corresponds to trend, and purple corresponds to missing samples. For normal data, the corresponding features contours are similar to the upper or lower contour of the input response, as marked by the yellow border in Fig. 9. The output is processed and compressed by DenseNet and still retains the distinguishable contour features, which proves the strong ability of DenseNet to extract features. The outputs of the outlier pattern marked by the green border have similar characteristics to the outlier, that is, the outlier is much larger than the adjacent points. For trend and missing patterns, the corresponding outputs also have outline information, marked with red and purple borders, respectively. DenseNet compresses the length while still retaining distinguished features of the input samples, such as contour, to distinguish different classes of samples. Overall, the output of each class of data has specific features to allow the part of TDNet behind DenseNet to perform the classification task more accurately.

4.3 Noise effect

The noise effect on classification accuracy is investigated by adding a certain level of white Gaussian noise to the input measurement. The noisy acceleration response A_{noise} is generated as

$$A_{noise} = A + N \times RMS(A) \times l \tag{11}$$

where A is the raw acceleration response. N is a vector that represents the white noise with the same length and sampling rate as the raw response. l refers to the noise level and RMS denotes the root mean square computation. Note that when generating the noisy training samples for the missing anomaly, noise is also considered as lost and the lost sampling points remain 0. A comparison of the normal data without noise and with 0.3 noise level added is shown in Fig. 10 in time domain.

Three datasets with noise levels of 0.1, 0.2, and 0.3 are generated based on the dataset of Case 3 for training, validation, and testing of the network. The classification

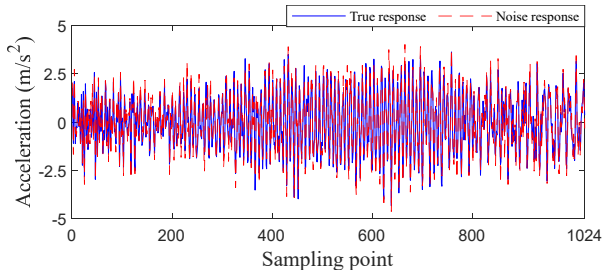


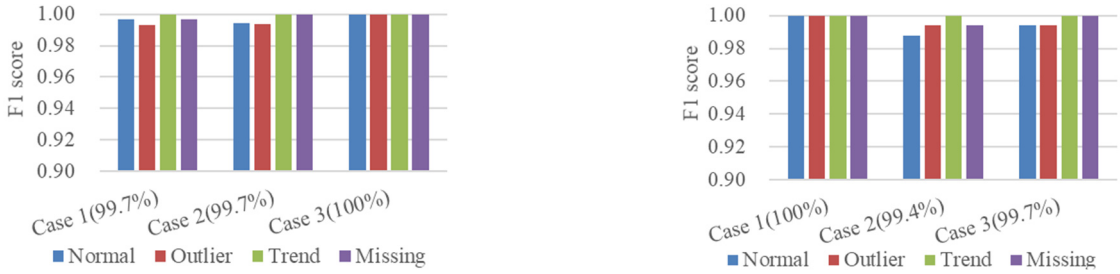
Fig. 10 Comparison between the true response and noisy response with 30% noise

results of the trained model on the validation and test sets are shown in Fig. 11. In cases 1 to 3, the F1 scores of the four patterns reach above 0.98 on both the validation set and the test set, while the overall accuracy achieves above 98%. The classification performance of the trained model does not degrade with increasing noise level either on the validation set or on the test set, which demonstrates the strong robustness of the proposed data anomaly detection method to noise.

5. Practical engineering studies with GNTT

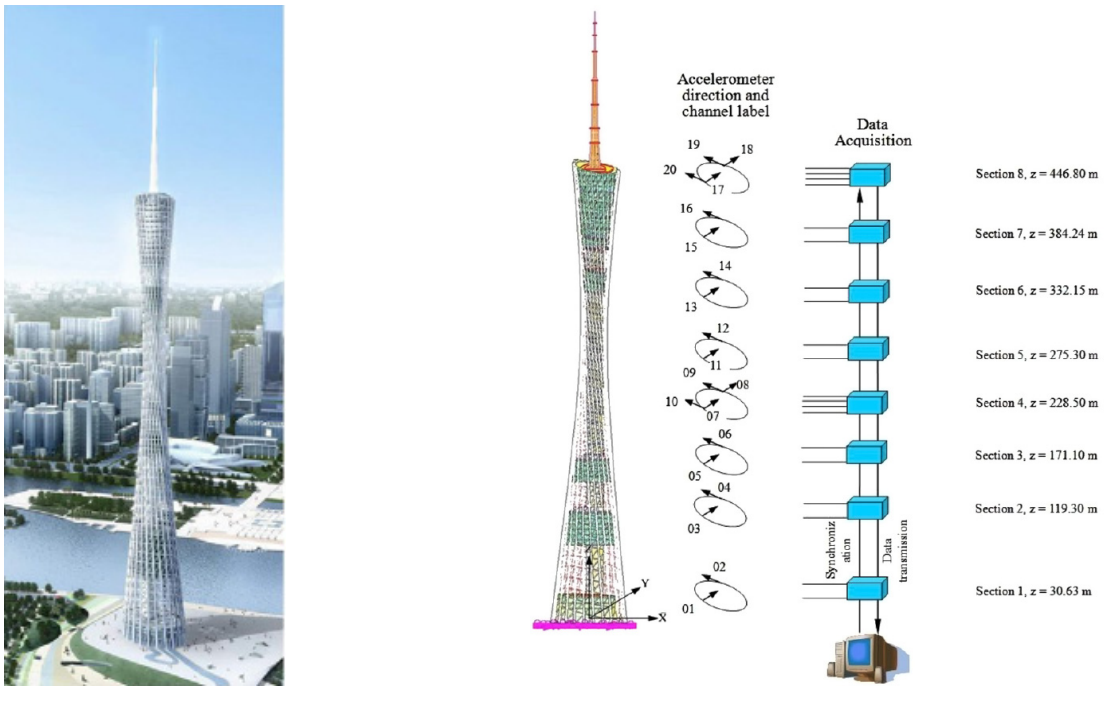
5.1 The SHM system of GNTT

GNTT is a slender tube-in-tube supertall structure with a height of 610 m. It is composed of a reinforced concrete inner tube, a steel outer frame tube and a combined floor connecting the two. The steel outer frame tube consists of 24 concrete-filled tube columns twisted counterclockwise from the bottom to the top. There are 46 steel ring beams and bracings among the entire structure. An oval-shaped gradient network structure was designed to effectively reduce the bulkiness of the tower and wind loads, as shown in Fig. 12(a). Thirty-seven floors between the inner and outer steel tubes are used for offices, entertainment, catering and emission of the television signals. This tower has a sophisticated SHM system including accelerometers, anemometers, thermometers, etc., to continuously monitor



(a) Validation set (b) Test set

Fig. 11 Classification results with noisy data



(a) (b)

Fig. 12 (a) GNTT; (b) The deployment of accelerometers and sensor numbers

the structural vibration behavior and ambient environmental conditions. The acceleration data measured from the SHM system are used to validate the proposed method for practical engineering. The placement of the accelerometers is shown in Fig. 12(b). Twenty uni-axial accelerometers with a sampling rate of 50 Hz are installed at proper height in both the long-axis and short-axis directions.

In this study, 24 hours measured acceleration data from 20 channels are selected for evaluating the proposed approach for anomaly detection. A small part of normal acceleration data is processed as the training and testing datasets to validate the effectiveness of the proposed method for practical engineering. On the other hand, the data anomaly existing in the remaining normal measurement are manually selected and labeled from the rest of measured data to verify the robustness of the proposed method on identifying real anomalous data, that the anomalous pattern may differ from the simulated ones.

5.2 Data pre-processing

The raw measurement is processed by a high-pass filter with a 0.05 Hz cutoff frequency to eliminate the shift at the zero frequency. It has been investigated that the first fifteen vibration modes of the GNTT are within 2 Hz. Therefore, the low-pass filtering with a cut-off frequency of 10 Hz is conducted to eliminate the redundant information contained in the signals. The processed data is then divided as about 78000 segments to generate the training, validation and test datasets. Due to the complex feature and measurement noise contained in real measurement, to guarantee the network fully learn the hidden features, a larger dataset containing totally 15,600 normal samples are randomly chosen to generate the balanced dataset. Then the training, validation and test dataset are split from the created balanced dataset with a ratio of 70%, 20%, and 10%. Another unseen 400 normal data and 345 outlier samples are also selected from the rest data to evaluate the trained network. There are no samples with obvious linear trend characteristics and continuous points loss can be found from the measured period in this study. A comparison of the simulated outlier with the real measured outlier is shown in Fig. 13.

5.3 Network training and testing results

The training is performed until the loss and accuracy curves converged with an increase in iterations, as observed in Fig. 14. It can be observed that both the training and validation losses decrease sharply in the first several epochs because of the involved dense connection and skip connection techniques which improve the convergence speed effectively. These two curves continue to decrease with increased training epoch and tend to converge after 40 epochs. A minor discrepancy between the training and validation errors means no overfitting occurred, that the trained network learns robust representative features from the training data for which also fits the validation data pattern well.

The confusion matrixes for the validation dataset and the testing dataset obtained by training on the proposed network is depicted in Fig. 15. The confusion matrixes show that the classification accuracies are over 99% on both validation and test datasets. In the case that the validation and test datasets contain 3120 and 1560 samples, DenseNet presents outstanding overall prediction capability for the simulated anomaly detection. For both validation and test datasets, precision and recall are observed to be consistently more than 98% for all the classes, indicating that the trained network effectively learned the representative features of

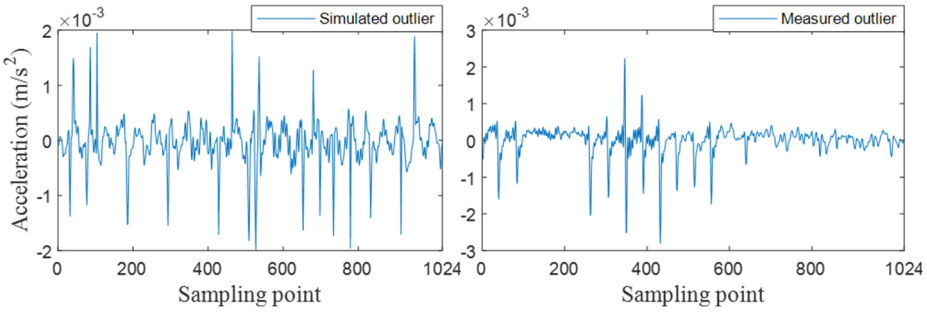


Fig. 13 Comparison of the simulated outlier with the real measured outlier

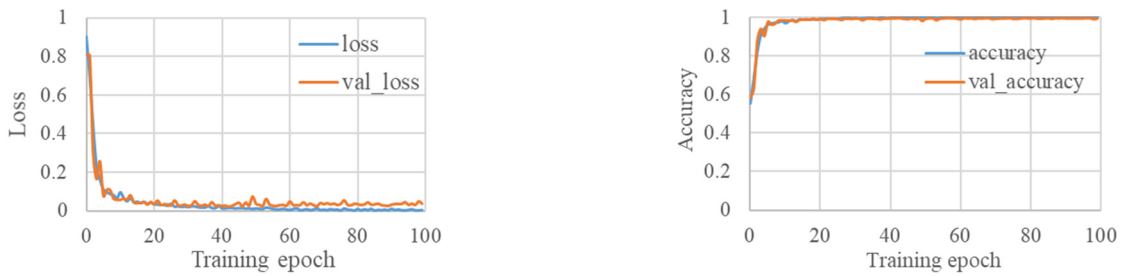
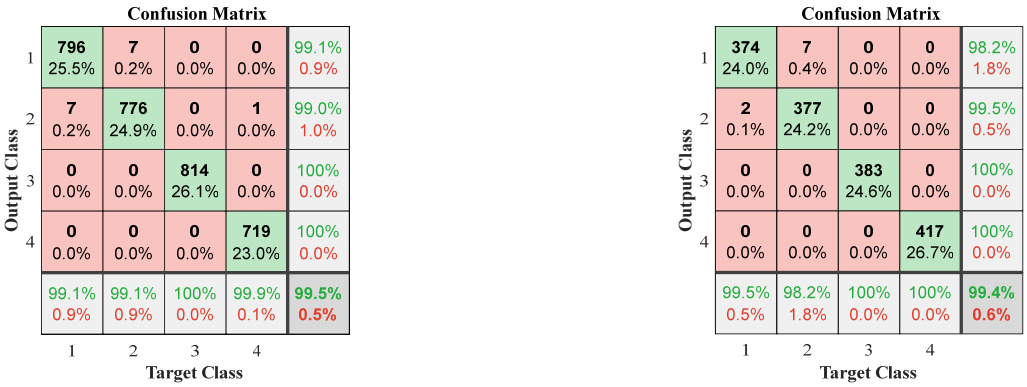


Fig. 14 The training process of TDNet for data of GNTT



(a) Validation set (b) Testing dataset

Fig. 15 Confusion matrix

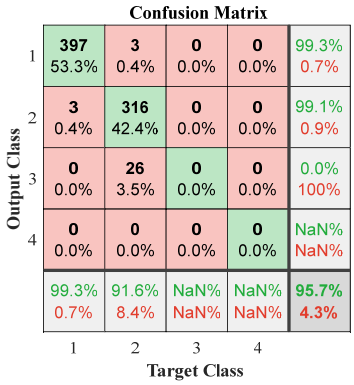


Fig. 16 Confusion matrix for real measured samples

the normal and anomalous data and correctly established the mapping between features and classes, which is capable to classify all types of data.

To further validate the effectiveness and robustness of the proposed method, the 400 real normal samples and 345 real outlier samples is fed into the trained model for classification. The classification result for this test dataset is illustrated by a confusion matrix as shown in Fig. 16. The trained model achieves a remarkable overall classification accuracy of 95.7%, despite that none of the real anomalous data samples are used for training, which shows a competitive adaptability in anomaly detection. It is also worth noting that only three normal data samples that are 0.4% of total samples are misidentified as abnormal, meanwhile, three outlier data samples are identified as normal. The trained model has a strong ability to distinguish normal and anomalous data. Besides, such a low error rate is acceptable for practical SHM of long-term abnormal condition warning. For the outlier data classification, there are another 26 samples misclassified as trend, resulting in a relatively lower recall of 91.6%. The inferior performance for outlier classification can be explained that the misidentified outlier data samples also featured ‘trend’ characteristics, i.e., the monotonous trend of data as shown in Fig. 17. The red ellipses marked the parts that resemble the trend anomaly feature. In fact, when defining the label

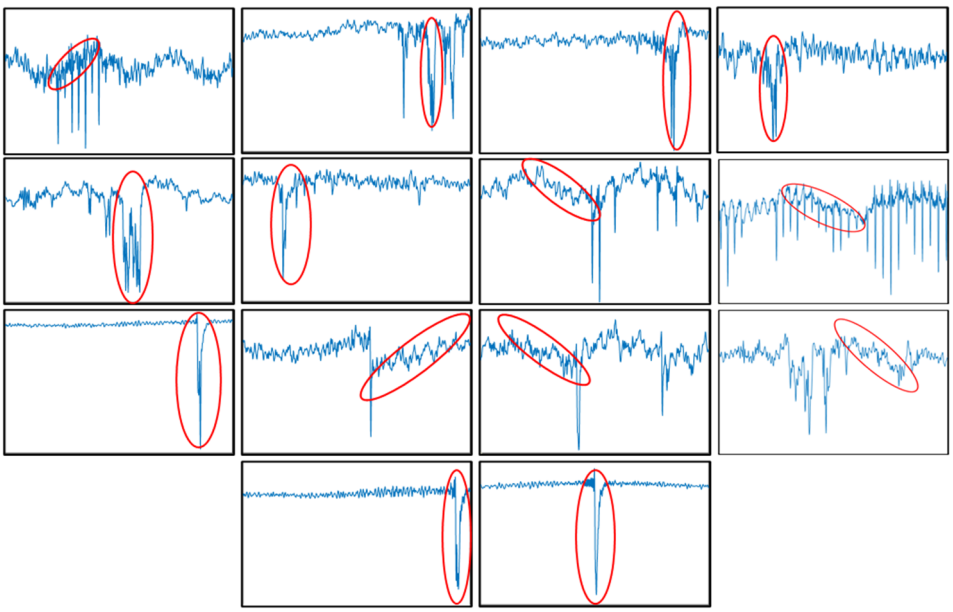


Fig. 17 Examples of outlier with trend characteristics

for those anomalous data, it is even confused for the authors to classify those samples. If the misidentified samples are neglected, the performance of TDNet is almost perfect for anomaly detection of a practical supertall structure, which verifies the robustness and effectiveness of the proposed method combining the data balancing and anomaly detection procedures. In addition, miss identification of any anomalous data may cause failure to warn the abnormal condition timely. The tiny number of miss identification of anomalous data demonstrate that this method can provide a reliable network for SHM.

6. Conclusions

This article proposes a novel method for data anomaly detection. The training data imbalance issue that significantly affect the performance of data anomaly detection is first addressed by using the proposed data-level technique. Then, data anomaly detection is implemented by a specific deep learning model named TDNet that is developed based on DenseNet and enhanced by imbedding attention mechanism to improve the global feature extraction. The numerical studies with a FEM of a seven-story frame and a practical test with GNTT are subsequently conducted to validate the effectiveness and robustness of the proposed method for practical civil engineering. In the numerical studies, it shows that dense connectivity and attention techniques are valid for boosting the feature extraction of structural response and establishing the mapping between features and data anomaly classes. The results demonstrate that TDNet can learn the hidden features from generated data efficiently from a small dataset. The noise test results also show the strong capability of TDNet on noise immunity. For the practical validation, DenseNet trained by simulated anomalous data samples achieved remarkable results in anomaly detection of real anomalous data. The high classification accuracy on the real dataset indicates that the employed data balancing technique and the proposed TDNet is feasible and reliable for engineering practice. However, it is worth mentioning that there are no missing and trend anomalous samples in the practical test. To verify the effectiveness of the method more comprehensively, missing and trend anomalies should also be considered in future research.

Acknowledgments

The support from the National Natural Science Foundation of China Project No. 52178279 and Guangzhou Basic and Applied Basic Research Foundation project, is acknowledged.

References

Abdelghani, M. and Friswell, M.I. (2004), "Sensor validation for structural systems with additive sensor faults", *Struct. Health Monitor.*, **3**(3), 265-275. https://doi.org/10.1177/1475921704045627

Arul, M. and Kareem, A. (2020), "Data anomaly detection for structural health monitoring of bridges using shapelet transform". https://doi.org/10.48550/arXiv.2009.00470

Bao, Y., Tang, Z., Li, H. and Zhang, Y. (2018), "Computer vision and deep learning–based data anomaly detection method for structural health monitoring", *Struct. Health Monitor.*, **18**(2), 401-421. https://doi.org/10.1177/1475921718757405

Bao, Y., Chen, Z., Wei, S., Xu, Y., Tang, Z. and Li, H. (2019), "The state of the art of data science and engineering in structural health monitoring", *Engineering*, **5**(2), 234-242. https://doi.org/10.1016/j.eng.2018.11.027

Chen, W.H., Lu, Z.R., Lin, W., Chen, S.H., Ni, Y.Q., Xia, Y. and Liao, W.Y. (2011), "Theoretical and experimental modal analysis of the Guangzhou New TV Tower", *Eng. Struct.*, **33**(12), 3628-3646. https://doi.org/10.1016/j.engstruct.2011.07.028

Chenglin, Z., Xuebin, S., Songlin, S. and Ting, J. (2011), "Fault diagnosis of sensor by chaos particle swarm optimization algorithm and support vector machine", *Expert Syst. Applicat.*, **38**(8), 9908-9912. https://doi.org/10.1016/j.eswa.2011.02.043

Cross, E.J., Koo, K.Y., Brownjohn, J.M.W. and Worden, K. (2013), "Long-term monitoring and data analysis of the Tamar Bridge", *Mech. Syst. Signal Process.*, **35**(1), 16-34. https://doi.org/10.1016/j.ymssp.2012.08.026

Fan, G., Li, J. and Hao, H. (2019), "Lost data recovery for structural health monitoring based on convolutional neural networks", *Struct. Control Health Monitor.*, **26**(10), p. e2433. https://doi.org/10.1002/stc.2433

Fan, G., Li, J. and Hao, H. (2021a), "Dynamic response reconstruction for structural health monitoring using densely connected convolutional networks", *Struct. Health Monitor.*, **20**(4), 1373-1391. https://doi.org/10.1177/1475921720916881

Fan, G., Li, J., Hao, H. and Xin, Y. (2021b), "Data driven structural dynamic response reconstruction using segment based generative adversarial networks", *Eng. Struct.*, **234**, 111970. https://doi.org/10.1016/j.engstruct.2021.111970

Fu, Y., Peng, C., Gomez, F., Narazaki, Y. and Spencer Jr, B.F. (2019), "Sensor fault management techniques for wireless smart sensor networks in structural health monitoring", *Struct. Control Health Monitor.*, **26**(7), p. e2362. https://doi.org/10.1002/stc.2362

Hou, J., Jiang, H., Wan, C., Yi, L., Gao, S., Ding, Y. and Xue, S. (2022), "Deep learning and data augmentation based data imputation for structural health monitoring system in multi-sensor damaged state", *Measurement*, **196**, 111206. https://doi.org/10.1016/j.measurement.2022.111206

Huang, Y., Beck, J.L., Wu, S. and Li, H. (2016), "Bayesian compressive sensing for approximately sparse signals and application to structural health monitoring signals for data loss recovery", *Probabil. Eng. Mech.*, **46**, 62-79. https://doi.org/10.1016/j.probengmech.2016.08.001

Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K.Q. (2017), "Densely connected convolutional networks", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, July.

Ibarguengoytia, P.H., Sucar, L.E. and Vadera, S. (2007), "Real time intelligent sensor validation", *IEEE Power Eng. Rev.*, **21**(9), 63-64. https://doi.org/10.1109/MPER.2001.4311630

Kerschen, G., De Boe, P., Golinval, J.C. and Worden, K. (2004), "Sensor validation using principal component analysis", *Smart Mater. Struct.*, **14**(1), p. 36. https://doi.org/10.1088/0964-1726/14/1/004

Krawczyk, B. (2016), "Learning from imbalanced data: open challenges and future directions", *Progress Artif. Intell.*, **5**(4), 221-232. https://doi.org/10.1007/s13748-016-0094-0

Kullaa, J. (2013), "Detection, identification, and quantification of sensor fault in a sensor network", *Mech. Syst. Signal Process.*, **40**(1), 208-221. https://doi.org/10.1016/j.ymssp.2013.05.007

Lei, X., Xia, Y., Wang, A., Jian, X., Zhong, H. and Sun, L. (2023), "Mutual information based anomaly detection of monitoring data with attention mechanism and residual learning", *Mech. Syst. Signal Process.*, **182**, 109607. https://doi.org/10.1016/j.ymssp.2022.109607

Lin, Y.Z., Nie, Z.H. and Ma, H.W. (2017), "Structural damage detection with automatic feature-extraction through deep learning", *Comput.-Aided Civil Infrastr. Eng.*, **32**(12), 1025-1046. https://doi.org/10.1111/mice.12313

Lo, C., Bai, Y., Liu, M. and Lynch, J.P. (2015), "Efficient Sensor Fault Detection Using Group Testing", ArXiv, abs/1501.04152. https://doi.org/10.1109/DCOSS.2013.57

Mao, J., Wang, H. and Spencer Jr, B.F. (2020), "Toward data anomaly detection for automated structural health monitoring: Exploiting generative adversarial nets and autoencoders", *Struct. Health Monitor.*, **20**(4), 1609-1626. https://doi.org/10.1177/1475921720924601

Ni, Y.Q., Xia, Y., Liao, W.Y. and Ko, J.M. (2009), "Technology innovation in developing the structural health monitoring system for Guangzhou New TV Tower", *Struct. Control Health Monitor.*, **16**(1), 73-98. https://doi.org/10.1002/stc.303

Rabatel, J., Bringay, S. and Poncelet, P. (2011), "Anomaly detection in monitoring sensor data for preventive maintenance", *Expert Syst. Applicat.*, **38**(6), 7003-7015. https://doi.org/10.1016/j.eswa.2010.12.014

Smarsly, K. and Law, K.H. (2014), "Decentralized fault detection and isolation in wireless structural health monitoring systems using analytical redundancy", *Adv. Eng. Software*, **73**, 1-10. https://doi.org/10.1016/j.advengsoft.2014.02.005

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014), "Dropout: a simple way to prevent neural networks from overfitting", *J. Mach. Learn. Res.*, **15**(1), 1929-1958.

Tang, Z., Chen, Z., Bao, Y. and Li, H. (2019), "Convolutional neural network-based data anomaly detection method using multiple information for structural health monitoring", *Struct. Control Health Monitor.*, **26**(1), p. e2296. https://doi.org/10.1002/stc.2296

Thiyagarajan, K., Kodagoda, S. and Van Nguyen, L. (2017), "Predictive analytics for detecting sensor failure using autoregressive integrated moving average model", *Proceedings of 2017 12th IEEE Conference on Industrial Electronics and Applications* (*ICIEA*), Siem Reap, Cambodia, June.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I. (2017), "Attention is all you need", arXiv. https://doi.org/10.48550/arXiv.1706.03762

Wan, H.P. and Ni, Y.Q. (2018), "Bayesian modeling approach for forecast of structural stress response using structural health monitoring data", *J. Struct. Eng.*, **144**(9). p. 04018130. https://doi.org/10.1061/(asce)st.1943-541x.0002085

Wu, B., Huang, Y. and Li, H. (2015), Sparse reconstruction of flaw signal from noisy ultrasonic data: A bayesian framework.

Xia, Y., Chen, B., Zhou, X.Q. and Xu, Y.L. (2013), "Field monitoring and numerical analysis of Tsing Ma Suspension Bridge temperature behavior", *Struct. Control Health Monitor.*, **20**(4), 560-575. https://doi.org/10.1002/stc.515

Yi, T.H., Li, H.N., Song, G. and Guo, Q. (2016), "Detection of shifts in GPS measurements for a long-span bridge using CUSUM chart", *Int. J. Struct. Stabil. Dyn.*, **16**(04), p. 1640024. https://doi.org/10.1142/s0219455416400241

Yuen, K.V. and Mu, H.Q. (2012), "A novel probabilistic method for robust parametric identification and outlier detection", *Probabil. Eng. Mech.*, **30**, 48-59. https://doi.org/10.1016/j.probengmech.2012.06.002