Smart Structures and Systems, *Vol. 35, No. 6 (2025) 337-349*
https://doi.org/10.12989/sss.2025.35.6.337

Data anomaly detection in structural health monitoring using modified transformer encoders with 1D-CNN layers

Sirojiddin Nuriev [1a], Ji-Hye Kwon [1b], Youngsu Kim [1c], Min-Joon Kong [2d] and Jong-Jae Lee*[3]

[1] *Research and Development Center, SISTech, 209, Neungdong-ro , Gwangjin-gu, Seoul, Republic of Korea*
[2] *THESOLT INC., AF002-0007, 202 Dasanjigeum-ro, Namyangju, Gyeonggi-do, Republic of Korea*
[3] *Department of Civil and Environmental Engineering, Sejong University, 209, Neungdong-ro, Gwangjin-gu, Seoul, Republic of Korea*

(*Received June 29, 2024, Revised June 17, 2025, Accepted July 8, 2025*)

Abstract. Structural health monitoring (SHM) has been widely used in civil infrastructure in recent decades. In SHM, vast amounts of data are collected using diverse sensors to monitor the health of civil structures. During this process, various types of anomalies may occur, which hindering an accurate assessment of the structure's condition. Anomalies mainly occur due to the influence of the harsh environment, sensor faults, or actual damage to the monitored structure. Therefore, early detection of anomalies is essential for monitoring the condition of structures. Conventional anomaly detection algorithms used in SHM systems, such as statistical thresholding, distance-based, rule-based, and clustering methods, have become ineffective today with growing data flow. These traditional algorithms face several limitations, including scalability issues, lack of adaptability to changing conditions, sensitivity to noise, and extensive feature engineering requirements. To address these issues, this paper proposes a modified transformer-based multiclass anomaly detection method for SHM systems. In our approach, we replace the feed-forward layers in the transformer encoder with two 1D-CNN layers and opt not to use positional encoding, as the occurrence of anomalies in SHM systems is not strongly related to specific positions within the sequence. Initially, the statistic and frequency domain features are extracted from the labeled time-series raw data. Then the modified transformer-based anomaly detection model is trained with extracted features and validated with acceleration data measured from a long-span cable-stayed bridge. The results confirm that the modified transformer encoders with 1D-CNN layers, without positional encoding, provide improved performance in detecting and classifying multiple types of anomalies with high accuracy. This demonstrates the potential of our method for enhancing the effectiveness of SHM systems.

Keywords: anomaly detection; deep learning; structural health monitoring; time-series classification; vibrational signal

1. Introduction

Structural health monitoring (SHM) in civil engineering is a technique used to monitor and assess the condition and performance of structures such as bridges, buildings, and other infrastructure, to identify any signs of damage or degradation that may occur over time (Farrar and Worden 2006, Li and Ou 2016). SHM aims to ensure the safety and reliability of structures and prevent or mitigate the consequences of structural failures. They can also help to ensure that structures can withstand extreme events, such as earthquakes, storms, and other disasters. SHM systems use a variety of sensors to collect data on the structural behavior of a structure (Sony *et al*. 2019). These sensors can measure various parameters such as displacement, strain, temperature, and vibration, and they can be placed on different parts of the structure to monitor its response to different loads and conditions.

The volume of SHM data is increasing yearly due to the growing use of SHM systems and the increasing availability of sensors and monitoring technologies (Bao *et al*. 2019a). For example, The SHM system installed on China's Sutong Bridge generates 2.5 TB of data annually with 785 sensors (Tang *et al*. 2019). In recent years, many different methods and techniques have been studied for damage detection and identification in SHM, such as vibration-based damage detection (Abdeljaber *et al*. 2017, Zhang *et al*. 2019), image-based damage detection (Dong and Catbas 2020, Dworakowski *et al*. 2016), machine learning techniques (Kurian and Liyanapathirana 2020, Malekloo *et al*. 2019) and others. However, researchers face difficulties because SHM data usually contain several types of anomalies due to environmental influences, system disturbances, sensor failures, and other factors, making it difficult to monitor a structure's behavior accurately. It is important to identify anomalies in SHM data to ensure the data's accuracy and reliability and the SHM system's effectiveness.

Anomaly detection in SHM systems is a challenging problem, as it requires the ability to accurately identify the anomaly patterns in data that may affect the accurately predicted condition of the monitored structure. Various methods have been explored in recent decades, including

*Corresponding author, Ph.D., Professor,
E-mail: jongjae@sejong.ac.kr
[a] Ph.D., E-mail: nuriyev14051989@gmail.com
[b] Ph.D., E-mail: wisegruv@gmail.com
[c] Ph.D., E-mail: yskim4257@gmail.com
[d] Ph.D., E-mail: minjoonkong@gmail.com

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

statistical and signal processing methods, physical modeling, rule-based methods, and machine learning techniques (Chandola *et al*. 2009, Ni *et al*. 2009, Sharma *et al*. 2010). Each approach has its merits, yet conventional algorithms can be expensive to implement and maintain, especially for large or complex structures. In addition, they often struggle with distinguishing multiclass anomalies.

In recent years, deep learning (DL) algorithms have had a significant impact on the field of anomaly detection, particularly in the area of SHM. For example, convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have been used for anomaly detection in SHM data, including data from sensors such as accelerometers and strain gauges. Similarly, computer vision (CV) approaches are becoming increasingly popular because they promise to solve complicated issues. CV is also helpful for anomaly identification, as visual evaluation of acceleration data makes it possible to spot many types of defective data. Therefore, several research papers have proposed CV-and-deep-learning combination approaches for spotting anomalies in data. A deep learning-based method for classifying acceleration data from gray-scale pictures was developed by Bao *et al*. (2019b). Data visualization was used to transform the gathered measurements into a format suitable for later deep neural network classification. The approach can identify outlier, missing, trend, minor, square, and drift abnormalities in addition to the more common trend and drift. The technique was tested using an acceleration dataset collected from a long-span cable-stayed bridge (Bao *et al*. 2021), with successful results indicating an accuracy of 87%. Tang *et al*. (2019) proposed anomaly detection method by providing CNN with input from the gray-scale image of the acceleration data and the matching FFT spectrum. This technique was trained and validated with the acceleration data from a long-span cable-stayed bridge (Bao *et al*. 2021). This method succeeded in obtaining 93.5% mean overall accuracy. Arul and Kareem (2022) proposed combining a time series "shapelet transform" feature extractor with a random forest classifier to detect anomalies in SHM data. This method was trained and tested with first-month acceleration data and achieved 93% overall accuracy. GoogLeNet, CNN, and ensemble methods were employed by Chou *et al*. (2022) on images of acceleration data to detect anomalies. This approach was also verified using the one-month subset of data and achieved an accuracy of 94.4%.

However, a closer examination of these methodologies reveals underlying issues. Specifically, converting time-series data into image representations, particularly across both time and frequency domains, can be a double-edged sword. This transformation, while innovative, can introduce significant variability and inconsistency. For instance, visual cues for anomalies like trends, drifts, or sudden changes might vary considerably depending on their position or the nature of the disruption. Coupled with the inherent challenge of imbalanced datasets in anomaly detection, CNNs face a heightened risk of skewed classification. The potential fallout from this, models might be prone to overfitting, reduced robustness, and predictive biases, favoring the more frequent classes while sidelining the less common anomalies.

Building on the observations of the challenges and inconsistencies in existing methodologies, we identified an avenue for progressive improvement. To bypass the complications arising from the transformation of time-series data into image representations and the associated issues of CNN-based anomaly detection, we ventured into the realm of Transformer architectures. Initially introduced by Vaswani *et al*. (2017) in their groundbreaking work "Attention is All You Need" in 2017, the Transformer has revolutionized the field of natural language processing (NLP). It has been used in various tasks, including machine translation, question answering, text summarization, language modeling, among others (Brown *et al*. 2020, Devlin *et al*. 2018, Raffel *et al*. 2019). One key advantage of the Transformer architecture is that it is fully self-attention based, which allows it to model long-range dependencies in the input data more effectively than traditional RNNs. In addition to its strong performance on NLP tasks, the Transformer architecture has also been applied to other areas such as CV (Dosovitskiy *et al*. 2020, Khan *et al*. 2022) and time series analysis (Li *et al*. 2019, Liu *et al*. 2021, Wen *et al*. 2022, Zhou *et al*. 2021). This strength offers a distinct edge in tasks where intricate and distant relationships in data sequences must be discerned, a self-attention mechanism especially beneficial in tasks like anomaly detection in SHM data.

The Transformer model is widely recognized in NLP, CV, and time-series analysis, yet its potential for anomaly detection in SHM systems remains largely untapped. In generative tasks, positional encoding is essential as it provides context by capturing the order and relative positions of data points, ensuring accurate and coherent data generation. However, in the realm of SHM data and more broadly in time-series classification tasks, anomalies can surface at any point in the sequence, with their importance not rigidly tied to their temporal position. Given the indeterminate locations of SHM anomalies, we chose to omit positional encoding. This choice is informed by characteristic traits of SHM data, where the relevance of an anomaly is less about its time placement and more about its interrelation within the data sequence. Additionally, to better tailor our model to the nuances of SHM data, we incorporated two 1D-CNN layers instead of the feed-forward layers, aiming to minimize model complexity and enhance its ability to effectively capture local patterns.

The proposed method involves applying an encoding method to univariate time-series data, extracting statistical and frequency domain features from the raw data, and then using the modified encoder-based transformer model to classify anomaly patterns from the extracted features. The proposed strategy was validated with a two-month acceleration dataset from a long-span cable stayed bridge. Experimental analysis shows that the proposed method can accurately classify multi-pattern data anomalies.

The remaining sections of this paper are structured as follows. Section 2 presents the time-series feature extraction method and modified transformer-based encoder architecture. Section 3 provides experimental findings and an in-depth discussion of the proposed method. In the end,

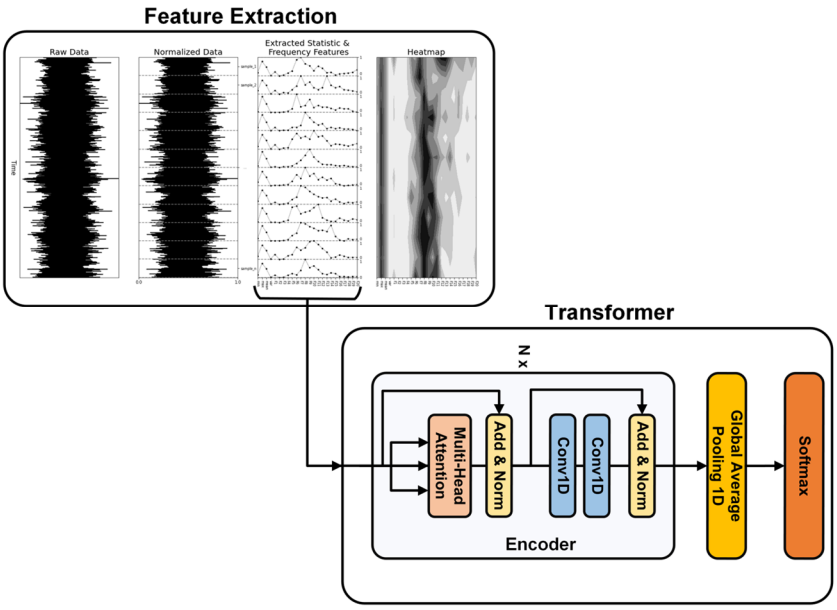


Fig. 1 Proposed transformer-based method layout

the conclusion is presented in Section 4.

2. Methodology

This paper aims to detect and classify multiple types of anomalies in SHM data with high accuracy. Fig. 1 depicts the overall layout of the proposed transformer-based data anomaly detection method.

It can be seen in Fig. 1 that the proposed transformer-based method layout consists of the following two main stages:

1) feature extraction from raw data, and
2) transformer training for data anomaly classification.

In the feature extraction stage, statistical and frequency features are extracted from the raw data measured by the sensors on the civil structures. A transformer-based data anomaly classification model is designed and trained in the next stage with extracted features and their corresponding labels. The trained transformer-based model can detect and classify the type of anomalies in large quantities of SHM data. The further mathematical interpretation of both parts is presented below.

2.1 Feature extraction

In general, sensors measure a lot of noise data, which is not helpful for pattern recognition. Generally, SHM systems use various feature extraction algorithms to remove unwanted noisy features (Amezquita-Sanchez and Adeli 2019, Zhang *et al*. 2022). The proposed technique employs statistical and frequency domain feature extraction algorithms. The detailed stepwise procedure is explained as follows:

In the first step, univariate time-series raw data is normalized using the following Eq. (1)

$$norm(x) = \frac{x - min\,(x)}{max(x) - min(x)} \tag{1}$$

Where x ($x \in R^{1\times ts}$) represents sequential data that is measured by the sensor during a certain period, and ts is the length of the data. In the second step, the normalized data is divided into n equal-length subsequences without overlapping by using the window slicing principle as given by Eq. (2)

$$x \Longrightarrow norm(x) \Longrightarrow \begin{bmatrix} x^1 \\ x^2 \\ \vdots \\ x^n \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_m^1 \\ x_1^2 & x_2^2 & \cdots & x_m^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^n & x_2^n & \cdots & x_m^n \end{bmatrix}_{n\times m} \tag{2}$$

Where n represents the number of subsequences, m is the length of each subsequence, and $n \times m$ is the length of x equal to ts. In the third step, four basic statistical features (*min*, *max*, *mean*, *variance*) are extracted from each split subsequences using Eq. (3)

$$min^k = \min x^k, \quad max^k = \max x^k,$$
$$mean^k = \frac{\sum_{i=1}^{m} x_i^k}{m}, \quad var^k = \frac{\sum_{i=1}^{m}(x_i^k - mean^k)^2}{m} \tag{3}$$

Where k ($1 \le k \le n$) is the index of each subsample. Afterward, absolute frequency amplitudes are calculated from normalized subsequences using the fast Fourier transform (FFT) algorithm using the Eq. (4)

$$f_j^k = \left| \sum_{r=1}^{m} x_r^k \cdot e^{-\frac{2\pi i}{m} j(r-1)} \right|, \qquad 0 \le j \le m-1 \tag{4}$$

All frequency results of each subsequence using FFT are given by Eq. (5)

$$Extracted\ Features = \left[\begin{array}{cccc|cccc} min^1 & max^1 & mean^1 & var^1 & \hat{f}_1^1 & \hat{f}_2^1 & \cdots & \hat{f}_w^1 \\ min^2 & max^2 & mean^2 & var^2 & \hat{f}_1^2 & \hat{f}_2^2 & \cdots & \hat{f}_w^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ min^n & max^n & mean^n & var^n & \hat{f}_1^n & \hat{f}_2^n & \cdots & \hat{f}_w^n \end{array}\right]_{n\times(w+4)} \quad (8)$$

$$f = \begin{bmatrix} f^1 \\ f^2 \\ \vdots \\ f^n \end{bmatrix} = \begin{bmatrix} f_0^1 & f_1^1 & \cdots & f_{m-1}^1 \\ f_0^2 & f_1^2 & \cdots & f_{m-1}^2 \\ \vdots & \vdots & \vdots & \vdots \\ f_0^n & f_1^n & \cdots & f_{m-1}^n \end{bmatrix}_{n\times m} \quad (5)$$

The zero-frequency component contains the sum of the time series data. Except for zero-frequency, the first half of the frequency components represents the positive frequency, and the second half represents the negative frequency terms. Maximum amplitude frequencies indicate important features of civil structures. Therefore, maximum frequency amplitudes are obtained from each frequency window using Eq. (6)

$$f_l^k = max(f_{w+1}^k, f_{w+2}^k, \ldots, f_{w+fs}^k), 1 \le l \le w \quad (6)$$

Where w $(1 \le w \le \frac{m}{2 \times fs})$ represents each frequency in the first half of the FFT and fs represents the frequency window size. Frequency features may fall in different ranges with statistical features that might dominate other features while training the model. Therefore, extracted maximum absolute frequency amplitudes are normalized using Eq. (7)

$$\hat{f}_l^k = \frac{f_l^k - min(f_{[1:w]}^k)}{max(f_{[1:w]}^k) - min(f_{[1:w]}^k)} \quad (7)$$

The final extracted statistic and frequency domain features for each normalized subsequence from univariate time-series raw data are given by Eq. (8)

2.2 Transformer-based model architecture

In the proposed transformer-based encoder architecture, extracted statistical and frequency domain features are used as input without positional encoding. Then feed-forward networks are replaced with two one-dimensional convolutional neural networks (1D-CNN) compared to the encoder part of the original paper. Next, the output of the encoder is passed to the global average pooling layer. The output of the global average pooling layer passes into the final SoftMax layer, and it predicts one of the multi-class anomalies (see Fig. 2).

In Fig. 2, X is the input of the transformer model, represents features extracted from the raw data, Z_i matrices represent the output of each layer, and n_p represents the total number of patterns. The various components of the actual transformer-based model are implemented as follows in this paper.

Positional Encoding: Positional encoding is the first step in the transformer model. Generally, anomalies appear based on damage or sensor faults in SHM systems of civil structures. In the main situations, there are no positional relationships appearing as anomalies with other features in the sequence. Therefore, this positional encoding is not considered in the proposed model.

Encoder: The main component in the transformer model is the encoder. In this, the N identical encoder layers are connected in series. The first encoder takes features extracted from the raw data as its input. The output of each encoder layer is passed to the next encoder as its

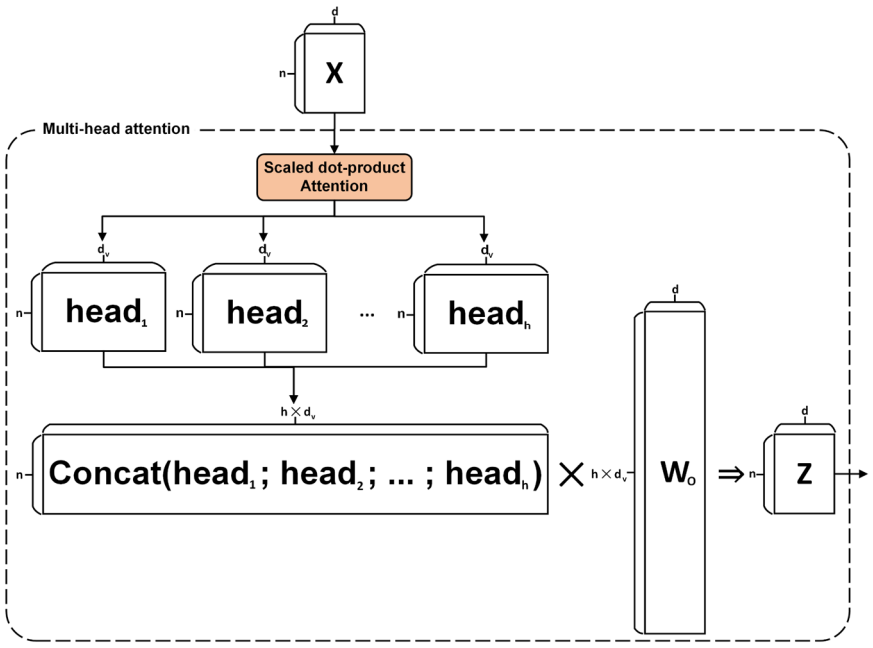


Fig. 3 Multi-head attention mechanism

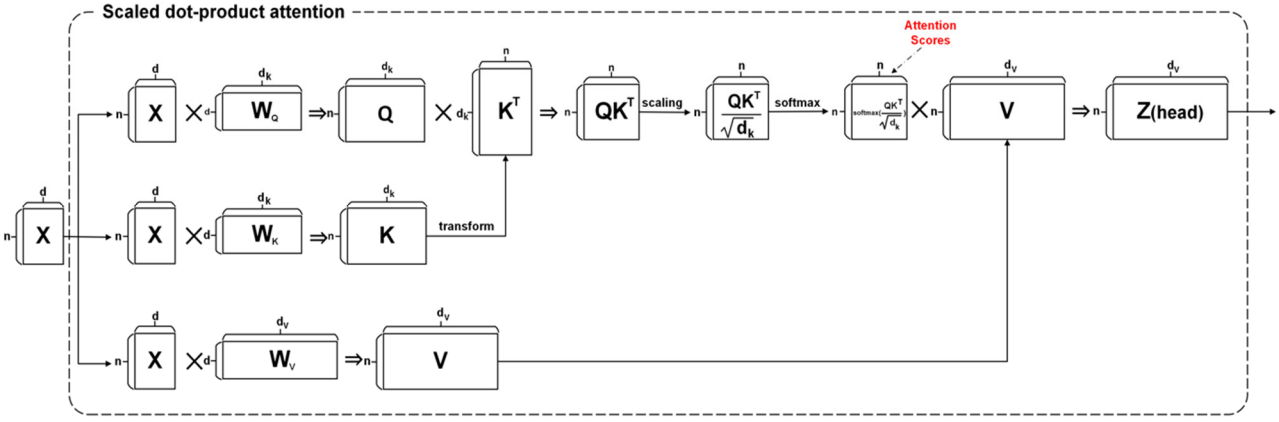


Fig. 4 Scaled dot-product attention

input. Each encoder consists of two major components: a multi-head attention mechanism and two 1D-CNN A residual connections used for each component. Both mechanisms are explained as follows:

Multi-head Attention: It is the first main component of the encoder with multiple attention heads. Although each attention head learns the relevance of given sequence values to each other, multiple attention heads can learn different definitions of "relevance." Each attention head size is equal to $n \times d_v$. Here, n is equal to the input sequence length, and d_v is equal to the value's (V) dimension, which will be explained later in a scaled dot-product subsection. The output of the concatenated attention heads passes into the feed-forward neural network layer. The feed-forward layer's learnable parameters are defined by $W^o \in R^{hd_v \times d}$ matrix. Where h is the number of heads. The output shape of multi-head attention is equal to the input shape. The multi-head attention mechanism is illustrated in Fig. 3.

It can be seen in Fig. 3 that the multi-head attention mechanism mainly comprises the scaled dot-product attention and concatenation of each head. Details about the scaled dot-product attention are explained below:

Scaled dot-product attention: It is applied for each head in the multi-head attention mechanism. Fig. 4 depicts its calculation process.

Each scaled-dot product attention unit learns three weight matrices: the query weights ($W_Q \in R^{d \times d_k}$), the key weights ($W_K \in R^{d \times d_k}$), and the value weights ($W_Q \in R^{d \times d_v}$). The dot-product of input ($X \in R^{n \times d}$) matrix with these three weights produces query ($Q \in R^{n \times d_k}$), key ($K \in R^{n \times d_k}$), and value ($V \in R^{n \times d_v}$) matrices. Initially, the attention scores are computed by taking the dot-product of the query and transposing key matrices (weights). Therefore, the dimensions of W_Q and W_K should be equal to each other. Next, the QK^T result is scaled by dividing the square root of d_k. Then, attention scores of the given input are calculated by applying softmax as shown in Eq. (9)

$$Attention\ Scores = softmax(\frac{QK^T}{\sqrt{d_k}}) \tag{9}$$

Afterward, the output of the scaled dot-product attention unit is obtained by the dot-product between attention scores and value matrix as given in Eq. (10)

$$Z = softmax(\frac{QK^T}{\sqrt{d_k}}) \cdot V \tag{10}$$

Here, the value matrix's d_v dimension can be different from d_k. Before concatenating with other heads, the output of the scaled-dot product attention is subjected to the residual connection and normalizing (see Fig. 2).

One-Dimensional Convolutional Neural Networks: CNN is the deep learning subset that looks for various patterns in data through kernels. CNNs are well known as two-dimensional CNN (2D-CNN) and have become the standard in computer vision applications (He *et al.* 2016, Simonyan and Zisserman 2014). 1D-CNN are similar to 2D-CNN, and they are mainly used on 1D signals and text (Abdeljaber *et al.* 2018, Kiranyaz *et al.* 2021). In the development of our transformer encoder architecture, standard feed-forward networks are replaced with two 1D-CNN layers, employing a 1×1 kernel size, to enhance the model's capability to identify local patterns in SHM data. This modification is motivated by the proficiency of 1D-CNNs in recognizing localized features within time-series data, crucial for detecting anomalies in SHM data, and their ability to introduce additional nonlinearity into the model, facilitating the recognition of complex and non-linear patterns in anomaly detection. The parameter and computational efficiency of 1D-CNNs, involving generally fewer parameters than their feed-forward counterparts, yield a model that is not only computationally efficient but also resilient to overfitting, particularly beneficial in instances where training data are limited. Following the multi-head attention mechanism, data is conveyed through the 1D-CNN layers, with the initial layer adopting a number of filters equal to n_{fil}. Subsequently, the second 1D-CNN layer's output is subjected to residual connection and layer normalization, thereby stabilizing the learning process and ensuring efficient backpropagation through the network layers.

Global Average Pooling: In classical CNN architectures, the CNN layers pass the output to fully connected layers, which tends to overfit. Global average pooling is a pooling operation without learnable parameters,

Table 1 Confusion matrix for binary classification

Total data = P + N	Predicted Positive (PP)	Predicted Negative (PN)
Actual Positive (P)	True Positive (TP)	False Negative (FN)
Actual Negative (N)	False Positive (FP)	True Negative (TN)

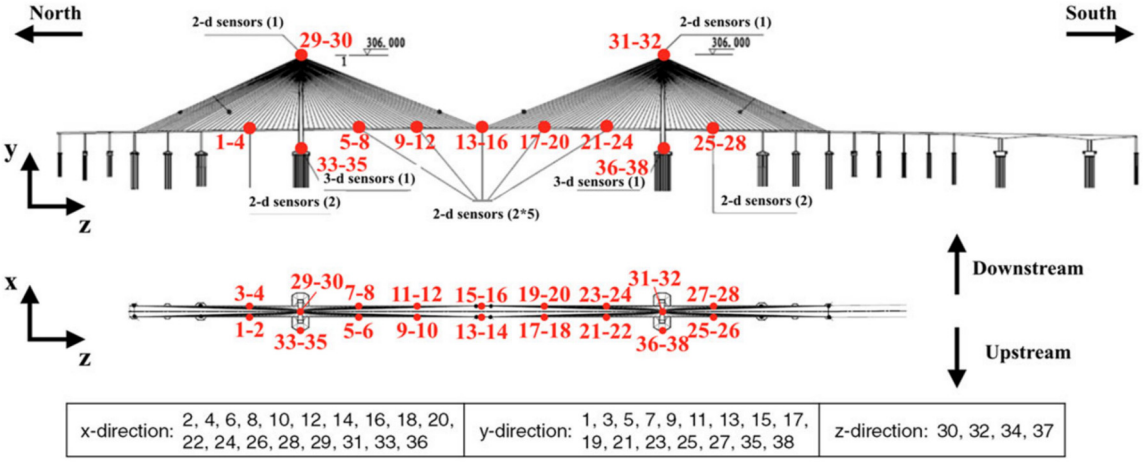


Fig. 5 Location of accelerometers on the bridge denoted by channel (Bao *et al*. 2021)

and it is designed to replace fully connected layers (Lin *et al*. 2013). The proposed model's last encoder layer is connected to the one-dimensional global average pooling operation to avoid overfitting. The output vector from this process is then used as input into the SoftMax layer, which averages the feature maps.

Evaluation metrics are used to measure how well the given model works. Details about evaluation metrics are explained as follows.

2.3 Evaluation metrics

This paper employs a confusion matrix to evaluate the performance of the proposed method. The confusion matrix is a special kind of table that lets you see how well an algorithm, usually a classification algorithm, works. As shown in Table 1, each row of the matrix is an instance in a real class, while each column is an instance in a predicted class.

The precision of a classifier can be determined by how many correct identifications it produces or how many instances it successfully recognizes an actual class. The rate at which a model makes accurate predictions compared to the total number of ground truths is referred to as its recall. This rate determines how well a model can distinguish different classes. Recall is the ability to properly identify the majority of ground-truth classes, and good models have a high recall, while correctly classifying only the actual classes shows high precision. A model is considered flawless if its recall and precision are equal to one, at which point the false-negative number drops to zero. The accuracy and recall rates were computed by contrasting the outcomes of the suggested method with the data that served as the basis for the comparison. The anomaly detection model's accuracy, precision and recall were determined using the following formulas

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

The $F_1\ score$ metric considers both precision and recall and is given by the following Eq. (14)

$$F_1\ score = 2 \times \frac{Precision\ \times\ Recall}{Precision\ +\ Recall} \tag{14}$$

This metric is also called the harmonic mean of the precision and recall.

3. Experimental analysis

3.1 Dataset

The efficiency of the suggested method was evaluated using data gathered from acceleration sensors mounted on a cable-stayed bridge with a main span of 1088 m, two side spans of 300 m each, and a height of 306 m with two towers. Multiple types of sensors are installed on the bridge, such as strain gauges, thermometers, anemometers, accelerometers, and a global positioning system (GPS). This paper only focused on acceleration data. Sixteen 2-channel and two 3-channel accelerometers are mounted on the bridge deck and its towers. The sampling frequency of each sensor is 20 Hz. Fig. 5 depicts the placement of the

sensors. Acceleration data from a total of 38 channels were collected over two months (2012-01-01 to 02-29). The data from each channel is split hourly without overlap, and each window length is equal to 72000 ($20 \times 20 \times 60$) measured points.

The hourly split data samples from the 38 accelerometer channels in two months equal 57,720 ($38 \times 24 \times 60$) samples. All samples were labeled with one of seven patterns by experts. The details of each pattern can be found in Bao *et al*. (2021). A time domain and frequency domain visualization of each pattern is shown in Fig. 6. The number of samples from each pattern is given in Table 2 (see Model training subsection 3.3).

Anomalies in data often manifest in various distinct patterns, as illustrated in Fig. 6. "Normal" patterns display a symmetrical time response, characterized by a peak-like frequency response. "Missing" patterns show a substantial absence in the time response, rendering both time and frequency responses non-existent. The. "Minor" pattern can be recognized by its significantly reduced amplitude in the time domain compared to normal data. "Outlier" patterns contain one or more conspicuous outliers in their time response. A "Square" patterns' time response resembles that of a square wave, while "Trend" patterns display an evident trend in the time domain accompanied by a prominent peak in the frequency domain. Lastly, "Drift" patterns represent nonstationary vibration responses marked by random drifts. Each of these patterns has its own unique characteristics and quantity distribution, shedding light on the varied nature of data anomalies (Tang *et al*. 2019).

3.2 Results of feature extraction

Fig. 7 illustrates feature extraction results from one of the normally labeled data by 3D visualization. In this figure, the x-axis indicates features extracted from each subsequence, the y-axis indicates the orders of subsequences, and the z-axis normalized values of extracted features as defined in Section 2.1. The graph's plot in the yz-plane shows 1 hour of acceleration data. This raw data is labeled as normal by experts. The distance between the vertical black lines on the plot indicates the length of a sliced window. It is equal to 1 minute in this graph. Thus, 1 hour of data is divided into 60 sequences.

Since the frequency sampling of the accelerometers is 20 Hz, the proposed technique extracted 20 frequency features and four statistical features from each sliced window. The plot in the xz plane shows the features

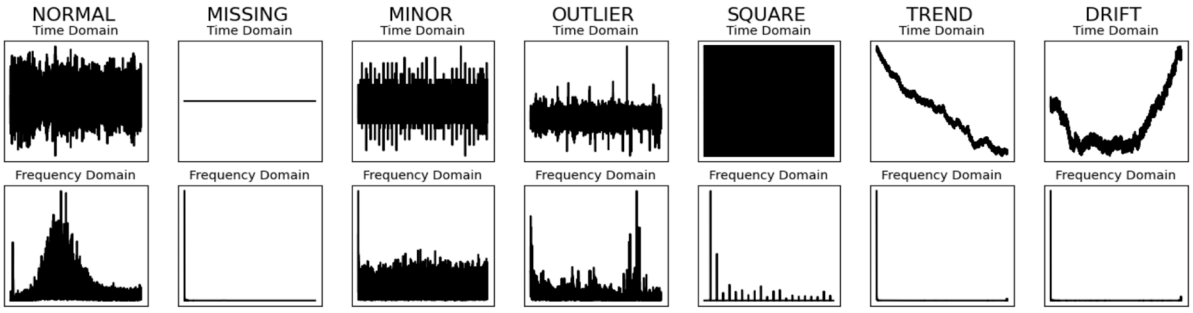


Fig. 6 Data anomaly patterns

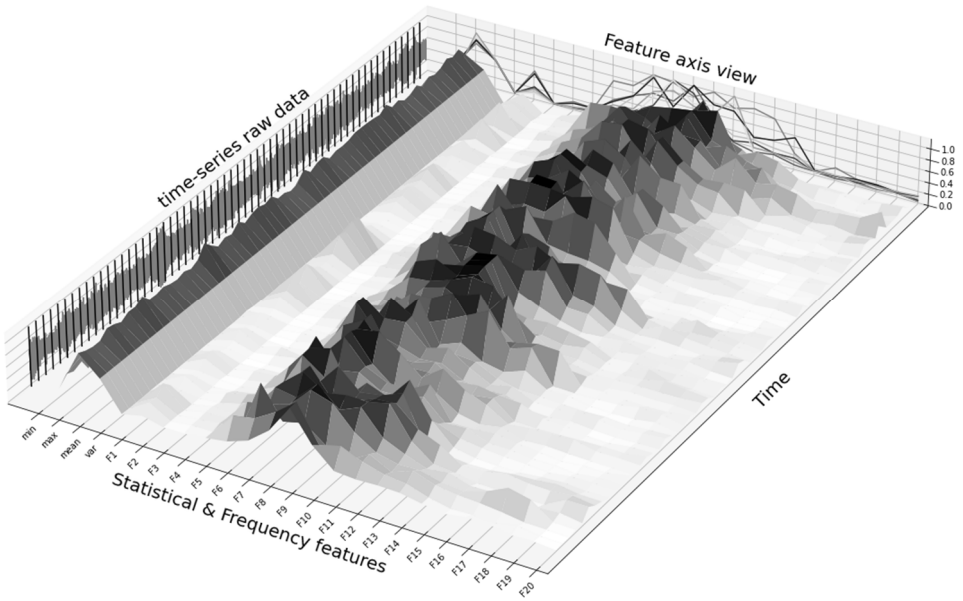


Fig. 7 3D visualization of features extracted from the raw data

Table 2 Train-validation-test data splitting

	Train	Validation	Test
Duration	**01/01/2012-24/01/2012**	**25/01/2012-31/01/2012**	**01/02/2012-29/02/2012**
Normal	10471	3104	13575
Missing	2419	523	2942
Minor	1378	397	1775
Outlier	356	171	527
Square	2300	696	2996
Trend	4426	1352	3778
Drift	538	141	679
Total number of data	21888	6384	28272

Table 3 Extracted features shape from 1 hour labeled raw data

Subsequence length	Extracted features shape
30 seconds	(120, 24)
1 minute	(60, 24)
5 minutes	(12, 24)
10 minutes	(6, 24)

extracted from each subsequence. The shape of the features extracted from the labeled raw data equals (60, 24) (number of sliced subsequences, number of extracted features). The training of this data is explained as follows.

3.3 Model training

The statistical information regarding the acceleration data for the past two months is summarized in Table 2.

It can be seen in Table 2 that the training data consists of the first 24 days of January 2012, while the remaining data of January is used for the validation. The whole data of February 2012 is used to test the proposed technique's performance. The number of samples in each section is as follows: 21,888 (24 days × 38 sensors × 24 hours) samples in training, 6,384 (7 days × 38 sensors × 24 hours) samples in validation, and 28,272 (29 days × 38 sensors × 24 hours) samples in testing sections.

Furthermore, it is evident from Table 2 that the given data is highly imbalanced. Most of the existing models encounter problems in handling highly imbalanced data. However, transformer-based models are more robust in handling imbalanced classification problems compared to other deep learning models. Therefore, the proposed technique does not suffer from this problem of imbalanced classification data. The length of the subsequence windows can be used differently based on the problem. In this dataset, the proposed technique has used four lengths of subsequence windows, as shown in Table 3.

It is evident from Table 3 that overall, 24 statistical and frequency features are extracted from each subsequence. However, the number of subsequences depends upon the length of the subsequences. Therefore, the proposed technique has considered four different subsequence window lengths of 30 seconds, 1 minute, 5 minutes, and 10 minutes, respectively, out of 1 hour of labeled raw data as shown in Table 3. To find the best hyperparameters for the transformer-based model, some range values for each hyperparameter are used, as shown in Table 4.

In general, to find the best hyperparameter, all possible combinations of the above ranges are computed, which results in 6000 possible combinations. However, 6000 combinations of the hyperparameters are huge and make it computationally very expensive to train all models. This issue can be solved efficiently by employing a certain range of important learnable parameters. The total number of learnable parameters of the model is calculated using the following Eq. (15)

$$N_p = N \cdot [h \cdot (d+1) \cdot (2 \cdot d_k + d_v) + d \cdot (h \cdot d_v + 1) + n_{fil} \cdot (d+1) + d \cdot (n_{fil} + 1)] \quad (15)$$

Therefore, out of 6000 combinations, the proposed method has selected a total of 137 combinations of the hyperparameters. Among 137 combinations, 60 combinations

Table 4 Hyperparameter values

Hyperparameter	Definition	Values range
N	Number of encoder layers	[1, 2, 3, 4, 6, 8]
h	Number of attention heads	[1, 2, 4, 8, 10, 12, 14, 16]
d_k	Key dimension	[16, 24, 32, 64, 128]
d_v	Value dimension	[16, 24, 32, 64, 128]
n_{fil}	Number of convolution filters first 1D-CNN layer in encoder	[8, 16, 32, 64, 128]

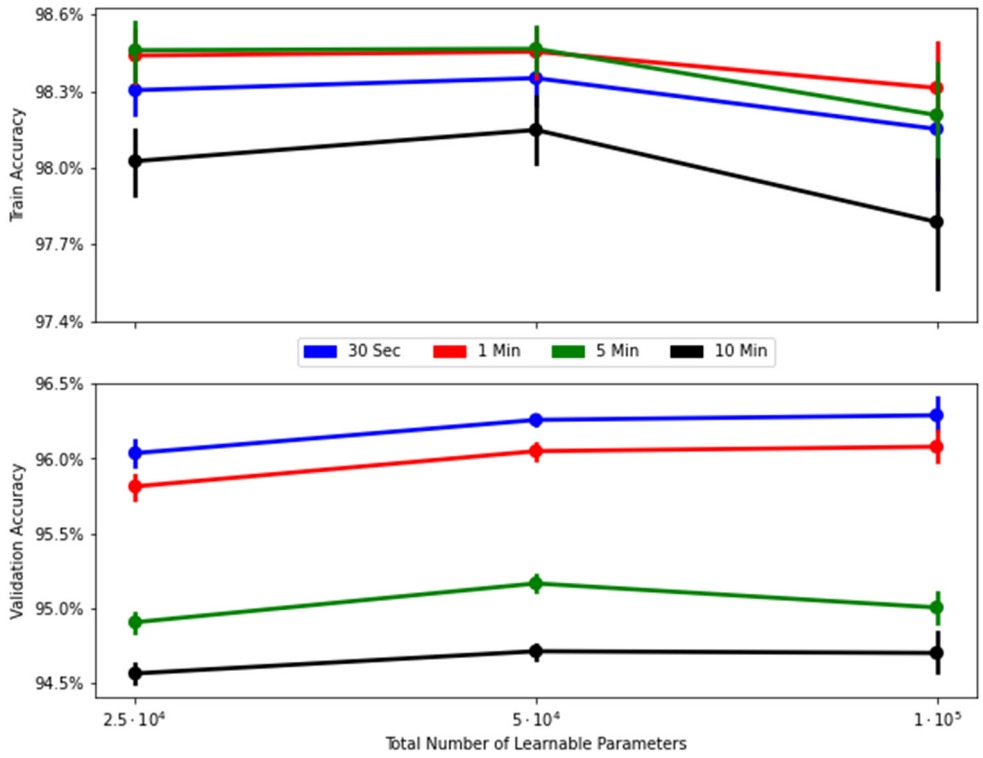


Fig. 8 Train and validation accuracy of the models with different hyperparameters

are selected with a total number of learnable parameters of around 25000 ± 1000, 60 combinations with 50000 ± 1000 learnable parameters, and 17 combinations with 100000 ± 1000 learnable parameters.

Handling imbalanced data remains a perennial challenge in many machine learning tasks. In many cases, specialized techniques such as data augmentation, resampling strategies, or employing custom loss functions are necessitated. However, proposed Transformer architecture didn't require aforementioned techniques for imbalanced classification. The inherent self-attention mechanism of the Transformer offers a natural capability to amplify the significance of rare patterns. This is adaptively assigning weights, allowing the model to concentrate more intently on less frequent, yet informative patterns without being dominated by the majority class.

Table 5 Models' average accuracies for different-length subsequence

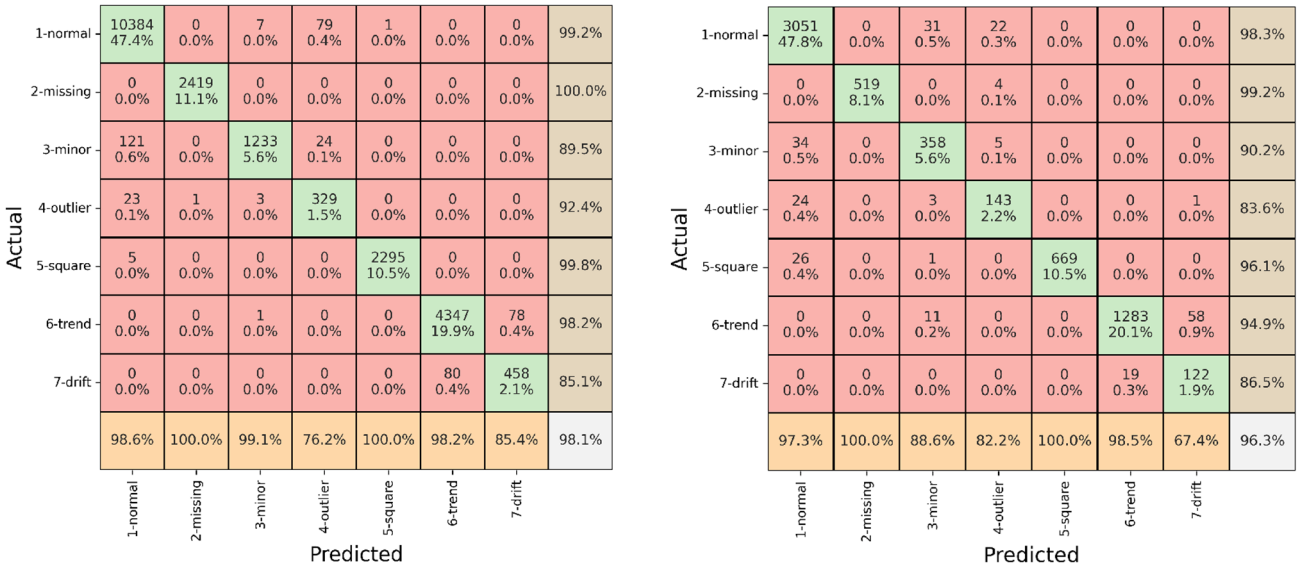
Metrics\Window length			30 seconds	1 minute	5 minutes	10 minutes
Train accuracy			98.90% ± 0.37	99.04% ± 0.41	99.06% ± 0.47	98.63% ± 0.51
Validation accuracy			95.54% ± 0.34	95.31% ± 0.31	94.41% ± 0.27	94.06% ± 0.24
Validation data	Normal	Precision	97.08% ± 0.46	96.76% ± 0.41	95.62% ± 0.56	95.19% ± 0.55
		Recall	97.48% ± 0.58	97.34% ± 0.58	96.63% ± 0.72	96.33% ± 0.75
	Missing	Precision	99.82% ± 0.35	99.82% ± 0.43	99.91% ± 0.18	99.98% ± 0.08
		Recall	99.24% ± 0.00	99.24% ± 0.00	99.24% ± 0.00	99.24% ± 0.02
	Minor	Precision	81.50% ± 3.46	80.24% ± 3.27	76.25% ± 3.68	73.92% ± 3.59
		Recall	93.47% ± 1.75	92.16% ± 1.89	89.16% ± 2.08	86.27% ± 2.28
	Outlier	Precision	83.89% ± 3.86	81.60% ± 3.94	82.62% ± 4.31	80.43% ± 4.46
		Recall	67.35% ± 6.58	63.80% ± 6.50	53.74% ± 6.47	51.23% ± 6.92
	Square	Precision	99.85% ± 0.28	99.74% ± 0.50	99.93% ± 0.20	99.95% ± 0.12
		Recall	96.18% ± 0.21	96.16% ± 0.14	96.17% ± 0.17	96.12% ± 0.00
	Trend	Precision	97.10% ± 0.88	97.36% ± 1.00	95.30% ± 0.72	96.62% ± 1.12
		Recall	95.73% ± 1.05	95.62% ± 1.05	97.40% ± 0.92	96.26% ± 1.04
	Drift	Precision	70.41% ± 5.64	71.73% ± 6.90	78.41% ± 7.52	72.59% ± 5.86
		Recall	74.08% ± 8.43	76.31% ± 8.92	54.10% ± 7.20	67.54% ± 11.20

3.4 Results and discussion

Fig. 8 illustrates the training and validation accuracy for different subsequence lengths and selected hyperparameters using the proposed transformer-based model. In Fig. 8, circles show the average training and validation accuracies, and the vertical bar shows the minimum and maximum accuracies of each time subsequence length and total learnable parameters.

It may be noted from Fig. 8 that among four subsequence lengths, the proposed transformer-based models achieved higher validation accuracies at a feature extraction sliding window length of 30 seconds. Furthermore, it is also seen that increasing the number of learnable parameters does not affect the validation accuracy. The variation of train and validation accuracies from 25,000-100,000 learnable parameters showed that $5 \cdot 10^4$ learnable parameters are more robust compared to others. Therefore, Table 5 illustrates the training and validation average accuracies with variations and precision and recall of each type of pattern for $5 \cdot 10^4$ parameters.

Table 4 shows that the proposed model achieved the best precision and recall for the most anomaly patterns (Normal, Square, Missing, Trend, and Minor) by extracting statistical and frequency features from every 30-second subsequence. On the contrary, the lowest precision and recall accuracies



(a) (b)

Fig. 9 Confusion matrix of classification results: (a) training set; (b) validation set

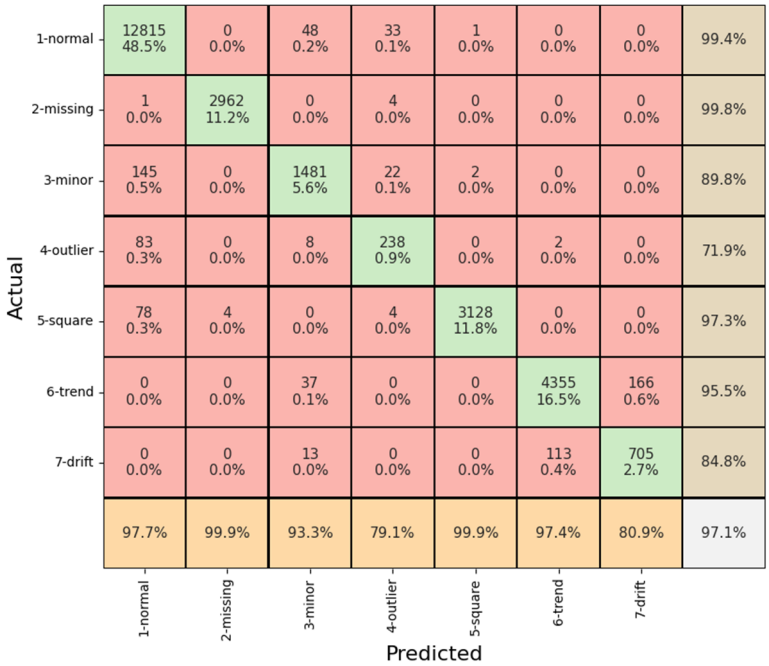


Fig. 10 Confusion matrix of classification results of test dataset

of 70.41% and 67.35% were obtained for the "Drift" and "Outlier" patterns, respectively. Moreover, it was also observed that increasing the subsequent window length is negatively correlated to validation accuracy. In other words, the model becomes overfitted. Therefore, the best hyperparameters were selected according to validation accuracy. The transformer-based model achieved 96.3% validation accuracy (see Fig. 9) with the best hyperparameters $N = 4,\ h = 4,\ d_k = 32,\ d_v = 24,\ n_{fil} = 32$, and 30-second subsequence length. Furthermore, the internal interaction among the detection results patterns for the training and validation dataset was investigated and can be effectively presented using a confusion matrix.

From Fig. 9, the following important points can be observed:

1. The proposed model achieved overall 98.1% training accuracy and 96.3% validation accuracy.
2. In the training dataset, the precision and recall are over 98% for "normal," "missing," "minor," "square," and "trend" patterns except recall for the "minor" pattern; it is 89.5%, which is acceptable; precision and recall for the "outlier" pattern are 76.2% and 92.4%, respectively; the "drift" pattern has moderate recall and precision, which is 85.4% and 85.1%, respectively.
3. In the validation dataset, the proposed model achieved over 97% precision and over 94% recall for "normal," "missing," "square," and "trend" patterns; the "minor" pattern has over 88% precision and recall; the "outlier" pattern has over 82% recall and precision; and the "drift" pattern has 86.5% recall and 67.4% precision.

Similarly, the internal interaction among the detection results for the testing dataset was also investigated and is presented using the confusion matrix (see Fig. 10) and corresponding examples (Fig. 9).

In Fig. 10, the following important points can be observed:

1. The proposed transformer-based model with the best-selected hyperparameters achieved 97.1% accuracy for the test dataset.
2. The "normal," "missing," "square," and "trend" patterns are predicted with more than 97% precision; the precision of "minor," "outlier," and "drift" patterns are 93.3%, 79.1% and 80.9% respectively.
3. The recall of "normal," "missing," "square," and "trend" patterns are over 95%; the recall of "minor," "outlier," and "drift" patterns are 89.8%, 71.9% and 84.8% respectively.

To further evaluate the performance of proposed modified transformer encoder model, it was compared with other variations of the transformer model. Table 5 presents a comparison of these models in terms of accuracy for training, validation, and testing datasets.

In Table 6, it is evident that the transformer encoder with 1D-CNN Layers and without positional encoding achieved the highest test accuracy compared to the other variations. This model also has the smallest number of parameters (51K), which indicates that the proposed modifications lead to a more efficient and compact model. The removal of positional encoding contributes to improved performance in the validation and test datasets for the models with feed-forward layers and 1D-CNN layers. This suggests that positional encoding may not be as relevant for SHM anomaly detection, where anomalies can occur at different positions in the data sequence without a strong positional relationship. Replacing the feed-forward layers with 1D-CNN layers results in a significant reduction in the number of model parameters while maintaining or improving the model's performance on the validation and test datasets. This indicates that the 1D-CNN layers are better suited for capturing local patterns in the data and detecting various types of anomalies. It is worth noting that the validation accuracies are lower than the test accuracies for all model variations. This discrepancy can be attributed to the misclassification of some common patterns, as elaborated in more detail later in this section.

Furthermore, the F_1 scores for all seven anomaly patterns obtained by the proposed transformer-based model are shown in Table 7.

It is noticeable from Table 7 that the proposed transformer-based model has the highest F1 score of 99.8% for the "Missing" pattern, while the minimum F1 score obtained is 75.3% for the "Outlier" pattern.

From the standpoint of the ground truth, as indicated in the rightmost column of Fig. 10, all seven data patterns

Table 6 Performance comparison of transformer-encoder model variations

Model	Params	Accuracy		
		Train	Validation	Test
Transformer Encoder (with Feed-Forward Layers and with Positional Encoding)	446K	99.4%	94.2%	95.3%
Transformer Encoder (with Feed-Forward Layers and without Positional Encoding)	446K	98.9%	94.5%	95.8%
Transformer Encoder (with 1D-CNN Layers and with Positional Encoding)	51K	98.5%	95.1%	96.2%
Transformer Encoder (with 1D-CNN Layers and without Positional Encoding)	51K	98.1%	96.3%	97.1%

Table 7 F1 scores of the anomaly patterns for test dataset

	Normal	Missing	Minor	Outlier	Square	Trend	Drift
F_1 score	98.5%	99.8%	91.5%	75.3%	98.6%	96.4%	82.8%

have been correctly identified. The recall accuracy of these seven sets of information varies from 71.9% to 99.8%, while the precision accuracy of all seven patterns varies from 79.1% to 99.9%. However, a few low precision and recall rates of "Outlier" and "Drift" patterns are due to five categories of misclassified patterns: "Normal" misclassified as "Outlier," "Outlier" misclassified as "Normal," "Trend" misclassified as "Drift," and "Drift" misclassified as "Trend," and "Outlier" misclassified as "Minor" pattern as shown in Fig. 10. The typical images of all seven patterns are shown in Fig. 6. These images serve as a basis for deducing the likely causes of these incorrect classifications. It can be seen that the only difference between the "Normal," "Minor" and "Outlier" patterns is the presence of spikes in the "Outlier" pattern. Therefore, if the spikes are significantly higher above the typical range, this indicates that the relevant sample is an "Outlier," as opposed to a "Normal" pattern or "Minor" pattern. In a similar vein, there is ambiguity in the labeling of "Trend" and "Drift" because numerous data are "trending with drifting." This coexisting-pattern feature can be found in a significant number of the samples that were incorrectly classified.

To put it another way, these examples are available in more than one pattern. During the stage that involves manually labeling the samples, however, each sample is only assigned a single pattern for labeling. As a result, when applied to those samples, the suggested model has a low detection performance.

4. Conclusions

This paper has proposed the novel application of a transformer-based data anomaly detection method for SHM systems. The proposed technique employs the univariate time-series data encoding method. The important statistical and frequency features are extracted from the raw data to enable the transformer-based model to easily differentiate anomalies from the normal data. In the transformer-based encoder model, two 1D-CNN layers are used instead of a feed-forward neural network to look at nonlinear patterns in the data. Moreover, the global average pooling 1D operation is applied before the final classification layer to avoid overfitting problems. The proposed transformer-based model is trained and validated with extracted features in statistical and frequency domains from the acceleration data of a long-span cable-stayed bridge in China. The experimental results confirm that the proposed method has the ability to detect and classify multiple types of anomalies from SHM data with 97.1% overall accuracy. Furthermore, the proposed transformer-based model has an F_1 score for all seven patterns as 98.5%, 99.8%, 91.5%, 75.3%, 98.6%, 96.4%, and 82.8%, respectively. Finally, the results demonstrate that the proposed method is robust to distinguish multiple types of anomalies from the sequential data, and it can be used for other time-series data.

Despite the promising results, it's essential to acknowledge the limitations in this study. Primarily, the statistical and frequency features were manually extracted, which may introduce bias or overlook potentially informative data patterns. Furthermore, the model was validated using a specific type of structure, which might limit the generalizability of the findings across different structural forms or other contextual anomalies are not present in the used dataset.

In light of the aforementioned limitations and the potential that deep learning models present, future work will explore developing an end-to-end architecture that can process raw data directly, eliminating the need for manual feature extraction. This direction aims to enhance the model's ability to autonomously learn and decipher useful and potentially novel features from the data. Furthermore, employing adaptive learning strategies will be investigated to bolster the model's generalizability and responsiveness to varied, and particularly, infrequent, or novel anomalies across diverse SHM scenarios.

Acknowledgments

The authors are thankful to the organizers of the International Project Competition for SHM (IPC-SHM 2020): ANCRiSST, Harbin Institute of Technology (China), and the University of Illinois at Urbana-Champaign (USA) for providing the valuable data of actual structures. The authors also thank IPC-SHM 2020 chairmen Professor Hui Li and Professor Billie F. Spencer Jr. for their leadership in the competition. The authors would also like to thank Professor Yuequan Bao for providing access to the dataset.

This work was supported by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT). (RS-2021-NR059168)

References

Abdeljaber, O., Avci, O., Kiranyaz, S., Gabbouj, M. and Inman, D.J. (2017), "Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks", *J. Sound Vib.*, **388**, 154-170. https://doi.org/10.1016/j.jsv.2016.10.043

Abdeljaber, O., Avci, O., Kiranyaz, M.S., Boashash, B., Sodano, H. and Inman, D.J. (2018), "1-D CNNs for structural damage detection: Verification on a structural health monitoring benchmark data", *Neurocomputing*, **275**, 1308-1317. https://doi.org/10.1016/j.neucom.2017.09.069

Amezquita-Sanchez, J.P. and Adeli, H. (2019), "Nonlinear measurements for feature extraction in structural health monitoring", *Scientia Iranica*, **26**(6), 3050-3059. https://doi.org/10.24200/sci.2019.21669

Arul, M. and Kareem, A. (2022), "Data anomaly detection for structural health monitoring of bridges using shapelet transform", *Smart Struct. Syst., Int. J.*, **29**(1), 93-103. https://doi.org/10.12989/sss.2022.29.1.093

Bao, Y., Chen, Z., Wei, S., Xu, Y., Tang, Z. and Li, H. (2019a), "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

Bao, Y., Tang, Z., Li, H. and Zhang, Y. (2019b), "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., Li, J., Nagayama, T., Xu, Y., Spencer Jr, B.F. and Li, H. (2021), "The 1st international project competition for structural health monitoring (IPC-SHM, 2020): a summary and

benchmark problem", *Struct. Health Monitor.*, **20**(4), 2229-2239. https://doi.org/10.1177/14759217211006485

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A. and Agarwal, S. (2020), "Language models are few-shot learners", *Adv. Neural Inform. Process. Syst.*, **33**, 1877-1901. http://arxiv.org/abs/2005.14165

Chandola, V., Banerjee, A. and Kumar, V. (2009), "Anomaly detection: A survey", *ACM computing surveys* (*CSUR*), **41**(3), pp. 1-58.

Chou, J.Y., Fu, Y., Huang, S.K. and Chang, C.M. (2022), "SHM data anomaly classification using machine learning strategies: A comparative study", *Smart Struct. Syst., Int. J.*, **29**(1), 77-91. https://doi.org/10.12989/sss.2022.29.1.077

Devlin, J., Chang, M.W., Lee, K. and Toutanova, K. (2018), "Bert: Pre-training of deep bidirectional transformers for language understanding", *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*: *Human Language Technologies*, pp. 4171-4186. https://doi.org/10.18653/v1/N19-1423

Dong, C.Z. and Catbas, F.N. (2020), "A review of computer vision–based structural health monitoring at local and global levels", *Struct. Health Monitor.*, **20**(2), 692-743. https://doi.org/10.1177/1475921720935585

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. and Uszkoreit, J. (2020), "An image is worth 16x16 words: Transformers for image recognition at scale". http://arxiv.org/abs/2010.11929

Dworakowski, Z., Kohut, P., Gallina, A., Holak, K. and Uhl, T. (2016), "Vision-based algorithms for damage detection and localization in structural health monitoring", *Struct. Control Health Monitor.*, **23**(1), 35-50. https://doi.org/10.1002/stc.1755

Farrar, C.R. and Worden, K. (2006), "An introduction to structural health monitoring", *Philosoph. Transact. Royal Soc. A: Mathe. Phys. Eng. Sci.*, **365**(1851), 303-315. https://doi.org/10.1098/rsta.2006.1928

He, K., Zhang, X., Ren, S. and Sun, J. (2016), "Deep residual learning for image recognition", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778. http://image-net.org/challenges/LSVRC/2015

Khan, S., Naseer, M., Hayat, M., Zamir, S.W., Khan, F.S. and Shah, M. (2022), "Transformers in vision: A survey", *ACM Computing Surveys* (*CSUR*), **54**(10s), 1-41. http://arxiv.org/abs/2101.01169

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M. and Inman, D.J. (2021), "1D convolutional neural networks and applications: A survey", *Mech. Syst. Signal Process.*, **151**, p. 107398. https://doi.org/10.1016/j.ymssp.2020.107398

Kurian, B. and Liyanapathirana, R. (2020), "Machine learning techniques for structural health monitoring", *Proceedings of the 13th International Conference on Damage Assessment of Structures*, pp. 3-24. https://doi.org/10.1007/978-981-13-8331-1_1

Li, H. and Ou, J. (2016), "The state of the art in structural health monitoring of cable-stayed bridges", *J. Civil Struct. Health Monitor.*, **6**(1), 43-67. https://doi.org/10.1007/s13349-015-0115-x

Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.X. and Yan, X (2019), "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting", In: *Advances in Neural Information Processing Systems*.

Lin, M., Chen, Q. and Yan, S. (2013), "Network in network." https://doi.org/10.48550/arXiv.1312.4400

Liu, M., Ren, S., Ma, S., Jiao, J., Chen, Y., Wang, Z. and Song, W. (2021), "Gated transformer networks for multivariate time series classification." https://doi.org/10.48550/arXiv.2103.14438

Malekloo, A., Ozer, E., AlHamaydeh, M. and Girolami, M. (2019), "Machine learning and structural health monitoring overview with emerging technology and high-dimensional data source highlights", *Struct. Health Monitor.*, **21**(4), 1906-1955. https://doi.org/10.1177/14759217211036880

Ni, K., Ramanathan, N., Chehade, M.N.H., Balzano, L., Nair, S., Zahedi, S., Kohler, E., Pottie, G., Hansen, M. and Srivastava, M. (2009), "Sensor network data fault types", *ACM Transact. Sensor Networks*, **5**(3), 1-29. https://doi.org/10.1145/1525856.152586

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P.J. (2019), "Exploring the limits of transfer learning with a unified text-to-text transformer", *J. Mach. Learn. Res.*, **21**(140), 1-67. https://doi.org/10.48550/arXiv.1910.10683

Sharma, A.B., Golubchik, L. and Govindan, R. (2010), "Sensor faults: Detection methods and prevalence in real-world datasets", *ACM Transact. Sensor Networks*, **6**(3), 1-39. https://doi.org/10.1145/1754414.1754419

Simonyan, K. and Zisserman, A. (2014), "Very deep convolutional networks for large-scale image recognition", *Proceedings of the 3rd International Conference on Learning Representations*, *ICLR 2015*. https://arxiv.org/abs/1409.1556v6

Sony, S., Laventure, S. and Sadhu, A. (2019), "A literature review of next-generation smart sensing technology in structural health monitoring", *Struct. Control Health Monitor.*, **26**(3), p. e2321. https://doi.org/10.1002/stc.2321

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

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I. (2017), "Attention is all you need", *Adv. Neural Inform. Process.*, **30**. http://arxiv.org/abs/1706.03762

Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J. and Sun, L. (2022), "Transformers in time series: A survey." http://arxiv.org/abs/2202.07125

Zhang, Y., Miyamori, Y., Mikami, S. and Saito, T. (2019), "Vibration-based structural state identification by a 1-dimensional convolutional neural network", *Comput.-Aided Civil Infrastr. Eng.*, **34**(9), 822-839. https://doi.org/10.1111/mice.12447

Zhang, C., Mousavi, A.A., Masri, S.F., Gholipour, G., Yan, K. and Li, X. (2022), "Vibration Feature Extraction Using Signal Processing Techniques for Structural Health Monitoring: A Review", *Mech. Syst. Signal Process.*, **177**, p. 109175. https://doi.org/10.1016/j.ymssp.2022.109175

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H. and Zhang, W. (2021), "Informer: Beyond efficient transformer for long sequence time-series forecasting", *Proceedings of the AAAI Conference on Artificial Intelligence*, **35**(12), 11106-11115. https://doi.org/10.1609/aaai.v35i12.17325