Automated acoustic event-based monitoring of prestressing tendons breakage in concrete bridges

Received: 22 April 2024 | Accepted: 10 July 2024

DOI: 10.1111/mice.13321

INDUSTRIAL APPLICATION

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

Automated acoustic event-based monitoring of prestressing tendons breakage in concrete bridges

Sasan Farhadi | **Mauro Corrado** | **Giulio Ventura**

Department of Structural, Geotechnical and Building Engineering, Politecnico di Torino, Torino, Italy

Correspondence
Giulio Ventura, Department of Structural, Geotechnical and Building Engineering, Politecnico di Torino, Torino, Italy.
Email: giulio.ventura@polito.it

Funding information
Ministero dell'Università e della Ricerca; Politecnico di Torino

Abstract

Prestressing wire breakage induced by corrosion is hazardous, especially for concrete structures subjected to severe aging factors, such as bridges. Developing an automated monitoring system for such a damage event is therefore essential for ensuring structural integrity and preventing catastrophic failures. In line with this target, a supervised deep learning–based approach is proposed to detect and classify acoustic emissions released by prestressing wire breakage. The application of advanced signal processing techniques is central to this study to determine optimal model performance and accurately detect patterns of various events. Diverse pretrained convolutional neural network (CNN) architectures are explored and further enhanced by incorporating Bottleneck Attention Mechanisms to refine their performance capabilities. Additionally, a novel hybrid model, AcousticNet, tailored for acoustic event classification in the context of structural health monitoring, is developed. The models are trained and validated using an extensive data set collected from controlled laboratory experiments and in situ bridge monitoring scenarios, ensuring comprehensive adaptability and generalizability. The comprehensive analysis highlights that the Xception model, enhanced with a bottleneck module, and AcousticNet significantly outperform other models in capturing intricate patterns within acoustic signals. Integrating advanced CNN architectures with signal processing methods marks a substantial advancement in the automated monitoring of prestressed concrete bridges.

1 | INTRODUCTION

Bridges are substantial for having a seamless transportation network and significantly impact economic prosperity and social development. In the post–World War II economic recovery and construction boom, road networks were expanded, and various bridges were built in developed countries. Many of these bridges are reaching or surpassing their originally planned lifespans, sometimes exhibiting structural deterioration, that can be insidiously hidden. Notable collapses, such as that of the internationally renowned Polcevera Viaduct in Genova (Italy), known as Ponte Morandi, which tragically claimed 43 lives (Fox et al., 2023), underscore the importance of addressing aging of bridge infrastructures. A major concern is the difficulty in detecting deterioration of prestressed concrete structures, especially post-tensioned ones, caused by corrosion of prestressing tendons

(Li et al., 2011). This corrosion gets worsened by construction defects, poor-quality materials, exposure to environmental elements, and de-icing agents (Bassuoni & Rahman, 2016; Zhutovsky & Douglas Hooton, 2017). The inaccessibility of cables and the localized nature of degradation make it a formidable challenge to detect. Even though the collapse of a bridge because of the breakage of corroded prestressing tendons involves the progressive breakage of several wires, the inability to detect such warning signals makes the corrosion of prestressing cables highly hazardous. Therefore, the need for comprehensive monitoring and maintenance of these structures to ensure their safety cannot be overstated.

Despite the fact that conventional damage detection methods, such as visual inspection (Saleem et al., 2021), acceleration-based modal identification (Avci et al., 2021), and strain sensing (Zhu et al., 2022) are widely employed for monitoring damages in concrete structures, they face limitations in detecting interior defects, exhibiting minimal sensitivity to them. Visual inspection is a quick and cost-effective nondestructive method used extensively for detecting surface-level damages such as cracks and spalling in concrete structures (Alani et al., 2014). Its effectiveness, however, depends heavily on the expertise of the operators, and it struggles to identify internal defects. Although acceleration-based is commonly used to detect damage by analyzing natural frequencies and mode shapes, its sensitivity is limited in large concrete structures, where even significant damage causes minimal changes in natural frequencies (Cawley, 2018). Additionally, noise can reduce the reliability of mode shape analysis, impacting its effectiveness in extensive structural health monitoring (SHM) applications. More advanced techniques such as fiber optic sensing (Hampshire & Adeli, 2000) present a promising alternative; however, its integration into existing bridges poses challenges and requires careful consideration, structural modification, and technical and financial complexities. Therefore, the urgent need for an innovative and automated approach to detect wire breakage in prestressed concrete bridges is evident.

This study aims to explore the potential of acoustic event detection and classification (AE/DC) using ultrasonic acoustic emission signals, which can provide a noninvasive and efficient solution specifically in the context of prestressed concrete bridges. AE/DC involves precisely detecting and categorizing specific acoustic signals based on their intricate patterns. While AE/DC offers a robust framework for event classification with proven applications in various fields, including surveillance systems (Foggia et al., 2016), scene recognition (Chu et al., 2006), speech recognition (McLoughlin et al., 2015), and acoustic scene segmentation (Madhu & Kumaraswamy, 2023), its potential within SHM has remained overlooked. Building on prior work proposing a sound event detection approach and artificial neural networks for wire breakage detection (Farhadi, Corrado, et al., 2024), this study emphasizes the need to overcome the ongoing challenge of understanding AE signal characteristics. To achieve this, key features must be extracted from AE signals using various signal representations, forming a crucial step for accurate analysis and meaningful event classification. Despite the extensive exploration of signal representations across various fields, such as heart anomaly detection (Wang et al., 2023), zone detection in electrical grids (Ardito et al., 2022), and engine fault diagnosis classification (Ramteke et al., 2022), their potential within the SHM domain using AE signals remains underexplored.

Understanding the propagation characteristics of AE signals poses a key challenge, particularly in distinguishing meaningful data from unwanted signals. Features extracted from 2D spectrograms have shown enhanced acoustic scene classification compared to representations derived from 1D signals, such as energy, time, and frequency-based features (Mesaros et al., 2021). Exploring AE signals across different classes through AE/DC can enhance signal comprehension, event classification, and automation processes in SHM. Although AE monitoring is well established in various fields of structural engineering for early damage detection (Dubuc et al., 2021; Ma & Wu, 2023), its application to detect prestressing wire breakage is substantially challenging (Yuyama et al., 2007). Wire breakage signals possess intricate and variable characteristics influenced by material properties, leading to complexities in consistent detection. Moreover, AE signals face attenuation, reflection, and scattering, especially within concrete structures at frequencies above 20 kHz. Therefore, in this study, diverse signal representations were explored, including short-time Fourier transform (STFT) spectrogram, log-STFT, Mel-frequency cepstral coefficients (MFCC), persistence spectrogram (PS), and Hilbert–Huang transform (HHT), to determine the most effective visual representation of AE signals for the specific task of event classification in prestressed concrete bridges.

An effective approach to discern intricate patterns within signals involves advanced machine learning (ML) techniques, specifically employing deep learning (DL) models. These models can discern highly complex patterns within the data, precisely classify the data into distinct categories, and even predict future occurrences (Jordan & Mitchell, 2015). In recent years, substantial advancements in DL algorithms have been propelled by hardware advancements and the availability of extensive training data. These advancements have significantly expanded the exploration and application of ML and DL models across various scientific and engineering fields, including lithological classification (Farhadi, Tatullo, et al., 2024), acoustic scene classification (Zhang et al., 2020), and health assessment (Giglioni et al., 2023). This list can be extended to a more specific domain of multipoint deflection of large-span bridges (Yin et al., 2023), large-scale SHM (Eltouny & Liang, 2023), vision-based monitoring of structures (Gao et al., 2023), and health condition assessment of structures (Rafiei & Adeli, 2018). Transfer learning (TL) (Gao & Mosalam, 2018) is an advanced approach enabling DL models to leverage knowledge gained from solving one problem and apply it to a related but distinct problem. Utilizing pretrained models on the large data set and fine-tuning them for the specific task of event detection in prestressed concrete beams can enhance the robustness and predictive capabilities of the models, ultimately accelerating the development of the event classification model.

The main original contributions of this study in the field of wire breakage detection are: (1) collecting data through laboratory tests and real-case scenarios, (2) introducing an innovative approach for automated event detection utilizing acoustic signals, (3) assessing diverse signal representations, (4) utilization of different pretrained models including VGG19, ResNet50, Inception, and Xception to efficiently discern complex patterns within the signals, (5) conducting an extensive comparative analysis of both individual pretrained models and a customized hybrid model. This analysis aims to provide valuable insights into their performance across different signal representations, aiding to identify the most effective model and representation for event classification, (6) demonstrating exceptional event detection performance without directly relying on conventional parametric analysis. This provides a significant step toward a more profound comprehension of wire breakage event detection in structures.

2 | METHODOLOGY

This section outlines the methodological framework for the event detection task, which is structured to comprehensively explore signal representation and leverage deep convolutional neural network (DCNN). Subsequent subsections will provide a detailed explanation of each component, offering a comprehensive insight into the innovative approach proposed in this study.

2.1 | Signal representations

In the domain of event classification, signal representations significantly influence model performance. The accurate detection and classification of events rely heavily on the model's capacity to discern specific features and patterns within these signals. This study comprehensively explores various signal-to-image transformation methods, namely, STFT, log-STFT, MFCC, PS, and HHT.

2.1.1 | Short-time Fourier transform

The STFT spectrogram is a fundamental and widely used signal representation, offering valuable insight into the time-frequency characteristics of acoustic emission signals. The frequency domain of a signal can be represented mathematically through the discrete Fourier transform (DFT):

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{\frac{-i2\pi nk}{N}} \qquad 0 \leq k \leq N-1 \tag{1}$$

where $e^{\frac{-i2\pi nk}{N}}$ is the twiddle factor that introduces phase shifts for each term in the summation, N is the number of points used to compute the DFT, $X[k]$ is the spectrum, $x[n]$ is the discrete signal, and n represents the discrete time index. The spectrum $X[k]$ is f_s-periodic, with f_s being the sampling frequency. In practice, the STFT involves segmenting the signal into frames of a fixed length N and applying the DFT to each frame after applying a window function $w[n]$ to attenuate discontinuities at the frame boundaries:

$$X[t,k] = \sum_{n=0}^{N-1} w[n] x[tH+n] e^{\frac{-i2\pi k}{N}} \tag{2}$$

Here, $X[t,k]$ represents the STFT of the signal at time frame t and frequency k, $w[n]$ is the window function (e.g., Hamming, Hanning, or Blackman), $x[tH+n]$ is the signal in the t-th frame and k-th sample within the frame, and H is the hope size. In general, the frames overlap, which is achieved by choosing a hop size smaller than the frame length. This overlap introduces statistical dependencies between adjacent frames, resulting in a smoother STFT representation. The STFT allows for defining the linear-frequency spectrogram, which is a 2D representation of the signal where the energy in each frequency band is given as a function of time (Virtanen et al., 2018).

2.1.2 | Mel-frequency cepstrum coefficients

MFCC is one of the most prevalent representations specifically used in speech recognition, which can capture the signal's spectral envelope by extracting cepstral coefficients in a mel-frequency scale. The MFCC computation involves several steps, including framing, windowing, Fourier transform, mel-filterbank computation, and discrete cosine transform (DCT) (Rao & Manjunath, 2017).

The Mel spectrum $S[m]$ is obtained by passing the DFT magnitude spectrum $X[k]$ of Equation (1) through a set of triangular Mel weighting filters:

$$S[m] = \sum_{k=0}^{N-1} [| X[t,k]|^2 H_m[k]] 0 \leq m \leq M-1 \tag{3}$$

where N is the number of points used to compute the DFT, $H_m[k]$ is the weight given to the k-th energy spectrum bin contributing to the m-th Mel filter, computed as follows:

$$H_m[k] = \begin{cases} 0 & k < f(m-1) \\ \dfrac{2(k - f(m-1))}{f(m) - f(m-1)} & f(m-1) \leq k \leq f(m) \\ \dfrac{2(f(m+1) - k)}{f(m+1) - f(m)} & f(m) \leq k \leq f(m+1) \\ 0 & k > f(m+1) \end{cases} \tag{4}$$

The parameter $f(m)$ can be expressed as:

$$f(m) = \left(\frac{N}{f}\right) f_{mel}^{-1} \left(f_{mel}(f_l) + m \frac{f_{mel}(f_h) - f_{mel}(f_l)}{M+1} \right) \tag{5}$$

where f_l and f_h indicate the lowest and highest frequencies, respectively. The transformation from the linear frequency scale (in Hz) to the Mel scale can be approximated using the following formula:

$$f_{mel} = \frac{1000}{\log(2)} \log\left(1 + \frac{f}{1000}\right) \tag{6}$$

The Mel spectrum $S[m]$ represents the energy distribution in the Mel-frequency scale. To compactly represent this information and extract the essential features, the inverse DCT or DCT-III should be applied as presented in Equation (7), which is known as MFCC.

$$C[n] = \sum_{m=0}^{M-1} \log(S[m]) \cos\left(\frac{\pi n(m-0.5)}{M}\right) \tag{7}$$

$$n = 0, 1, 2, \ldots, M-1$$

here, $C[n]$ represents the n-th MFCC, and M is the total number of Mel bands, which correspond to the number of Mel coefficients (Farhadi, Corrado, et al., 2024).

2.1.3 | Persistence spectrum

The PS provides a clear view of frequency component development and a profound understanding of the signal's power-frequency domain, also known as the spectrum histogram. It is particularly effective for observing short events and low-power signals. To extract the PS, first compute the STFT of the signal (Equation (2)), then compute the power spectrum of each segment using the following equation:

$$P[t,k] = |X[t,k]|^2 \tag{8}$$

where $X[t,k]$ represents the STFT coefficients. Then, the bivariate histogram is made using $\log(P[t,k])$, providing valuable information into power distribution across time and frequency. The persistent information can be obtained by summing the histogram for each time value:

$$H[i,j] = \sum_{t=0}^{t_{max}} \log(P[t,k], i, j) \tag{9}$$

where $H[i,j]$ is the accumulated histogram for each bin (i,j), the index i corresponds to specific frequency bands, and j relates to ranges of the logarithm of power values. Each bin (i,j) accumulates the logarithmic power values from all time frames t, thus summarizing how frequently certain power levels occur within specific frequency ranges throughout the signal (Kumbasar et al., 2022; Lee & Le, 2021).

2.1.4 | Hilbert–Huang transform

The HHT, introduced by Huang et al. (1998), stands out as a robust signal processing technique for analyzing nonlinear and nonstationary signals. This method unfolds in two principal stages: empirical mode decomposition (EMD) and Hilbert transform (HT).

Empirical mode decomposition

The EMD or variational mode decomposition (VMD) decomposes an original signal $x(t)$ into a finite number of intrinsic mode functions (IMFs), each representing different frequency components. The decomposition is expressed as:

$$x(t) = \sum_{k=1}^{n} h_k(t) + r_n(t) \qquad n = 0, 1, 2, \ldots, N \tag{10}$$

where n represents the total number of IMFs, $h_k(t)$ denotes the k-th IMF, and $r_n(t)$ indicates the residue of the signal reconstruction after all IMFs have been extracted. Each IMF, $h_k(t)$, is derived through an iterative process called sifting, which involves the following:

1. Identify the maxima and minima of the original signal and fit smooth cubic spline curves, $x_u(t)$ for the upper envelope and $x_d(t)$ for the lower envelope.

2. Calculate the mean of the upper and lower envelope, representing the local mean of the signal and subtracting it from the original signal to obtain the difference $x_1(t)$ as follows:

$$x_1(t) = x(t) - \frac{x_u(t) + x_d(t)}{2}$$

3. Check if $x_1(t)$ satisfies the two following conditions: The number of extrema and zero crossing must be either the same or differ by at most one. The mean value of the local maxima and local minima should be zero. If $x_1(t)$ satisfies the two conditions, then the 1-st IMF $h_1(t)$ is found and a residual $r_1(t)$ is calculated:

$$r_1(t) = x(t) - h_1(t)$$

   Otherwise, the steps should be repeated using $x_1(t)$ as the original signal to find the subsequent IMFs. The sifting process stops when the residual becomes smaller than a set threshold or becomes a nonoscillatory signal.

Hilbert transform

Each IMF is then subjected to the HT to determine its instantaneous frequency and amplitude, essential for constructing the time-frequency-energy of the signal. For any IMF, $h_k(t)$, its HT is defined as follows:

$$H[h_k(t)] = \frac{1}{\pi} P \int_{-\infty}^{\infty} \frac{h_k(\tau)}{t-\tau} d\tau \qquad (11)$$

where P represents the Cauchy principal value, a mathematical concept that highlights the local properties of the signal. For each IMF, $h_k(t)$, it computes the corresponding analytic signal $h_a(t)$ as follows:

$$h_a(t) = h_k(t) + iH[h_k(t)] = A(t)e^{i\theta(t)} \qquad (12)$$

where i is the imaginary unit, A and θ are instantaneous amplitude and instantaneous phase. Subsequently, the instantaneous frequency ω can be computed as follows:

$$\omega(t) = \frac{d\theta(t)}{dt} \qquad (13)$$

With amplitude and frequency varying over time, the Hilbert spectrum $H(\omega, t)$ can be represented as follows:

$$H(\omega, t) = \mathrm{Re} \sum_{j=1}^{n} a_j(t) e^{i \int \omega_j(t) dt} \qquad (14)$$

2.2 | Deep neural networks

In this subsection, a comprehensive description of the utilized models is provided, focusing on their development and integration. Initially, the concept of TL is introduced, which forms the foundational benchmark of this study. The pretrained model leveraging a feature-extraction approach was used, which was subsequently enhanced with fine-tuning using the Bottleneck Attention Module (BAM) to refine their performance. At the core of the study's methodology is the deployment of advanced DL models, including dilated convolutional networks, gated recurrent units, and multihead attention mechanisms. These elements are combined to develop the custom-designed AcousticNet model, which leverages the strengths of each component to effectively address the complexities of acoustic signal classification in SHM.

2.2.1 | TL–based models

TL involves applying knowledge and information from one domain to solve problems in another related domain. Originated from educational psychology, Bray (1928), TL plays a significant role in artificial intelligence, particularly in computer vision domain. In TL, a model is initially trained on a source data set. The unified definition of TL can be stated as follows: Given a source domain D_s with its associated learning task T_s, a target domain D_t associated with its learning task T_t, the main objective of TL is to utilize the knowledge derived from the source domain data X to learn a prediction function $f_t(.)$ within D_t to minimize the prediction risk, ensuring robust performance on the target domain data (Pan & Yang, 2010). There are two main approaches to employ a pretrained model, namely, feature extraction and fine-tuning (Chollet, 2018).

Feature extraction

Feature extraction involves using the knowledge from pretrained models to extract valuable information and features from new instances. These extracted features are then processed by a different classifier trained from scratch. In general, CNN models consist of two main parts: a convolutional base consisting of convolutional, activation, and pooling layers, and a densely connected classifier at the end. During the feature extraction process, the convolutional base is utilized to process the new data, followed by training a new classifier on the output. The features acquired through the convolutional base are generic, making this part reusable. On the other hand, top layers (fully connected layers) can learn a particular set of classes that the model was initially trained on and usually contain only the information about the presence probability of the specific class, which makes them less adaptable for new tasks particularly when the object location matters. Densely connected layers in the classifier lack details about object location and spatial characteristics and eliminate spatial details. The benefit of using feature extraction in TL is that it drastically reduces the epochs and training time due to the reuse of already extracted features.

Fine-tuning

This approach is complementary to feature extraction. It unfreezes a few of the top layers that were previously frozen on the convolutional base part (used for feature extraction) and trains both top layers and newly added classifiers (fully connected layers). In the tuning approach, more abstract representations of the models become adjusted to align them better with the specific task being addressed. The reason for selecting unfreezing layers lies in the nature of the features encoded by these layers. As mentioned earlier, the lower layers extract generic features, making them suitable for reuse. Conversely, the top layers encode more specific features, making them valuable for fine-tuning the new target domains.

2.2.2 | Bottleneck attention module

The BAM is a novel attention mechanism proposed by Park et al. (2018). BAM is a simple and effective attention mechanism that enhances feature representation in DL models through channel and spatial attention.

Channel attention

The channel attention mechanism exploits informative feature responses within a feature map (F). This process begins by applying global average pooling (GAP) on the F and producing a channel vector, which encodes global information in each channel. The channel vector can be represented as follows:

$$CH_v = \text{GAP}(F) \tag{15}$$

Here, $CH_v \in \mathbb{R}^{c\times1\times1}$, and c represents the number of channels. To compute channel-wise attention from CH_v, a multilayer perceptron (MLP) with one hidden layer is needed to be utilized. The dimension of this hidden activation within the MLP is constrained to $\mathbb{R}^{\frac{c}{r}\times1\times1}$, where r is the reduction ratio. Mathematically, the MLP operation for channel attention can be represented as:

$$\text{MLP} = w_1(\text{ReLU}(w_0) + b_0) + b_1 \tag{16}$$

where $w_0 \in \mathbb{R}^{\frac{c}{r}\times c}$, $b_0 \in \mathbb{R}^{\frac{c}{r}}$, $w_1 \in \mathbb{R}^{c\times\frac{c}{r}}$, and $b_1 \in \mathbb{R}^{c}$. The output of MLP represents the attention weights for each channel. The final channel attention can be represented as follows:

$$M_c(F) = \text{BatchNormalization}(\text{MLP}) \tag{17}$$

Spatial attention

The spatial attention module generates attention maps emphasizing specific informative features in various spatial locations. By employing dilated convolutions, the module effectively enlarges the receptive field, leveraging contextual information to understand the input comprehensively. In the initial step of this module, a 1 × 1 convolution is applied to integrate and compress the F across the channel dimension ($F^{1\times1}_{\text{integrated}}$). Then, two 3 × 3 dilated convolutions are applied to obtain contextual information ($F^{3\times3}_{\text{contextual}}$). Subsequently, the feature maps are reduced to $\mathbb{R}^{1\times H\times W}$ using another 1 × 1 convolutions ($F^{1\times1}_{\text{compress}}$). Like channel attention, the feature maps are further adjusted in scale using batch normalization:

$$M_s(F) = \text{BatchNormalization}(F^{1\times1}_{\text{compress}}) \tag{18}$$

Combining attention outputs

After computing the individual attention maps from the channel attention module $M_c(F)$ and spatial attention module $M_s(F)$, they are integrated through element-wise summation to create the final 3D attention map $M(F)$. To refine and normalize the attention distribution, a sigmoid function is applied to the element-wise summation, resulting in the final feature map $M(F)$, which is expressed as:

$$M(F) = \text{sigmoid}(M_c(F) \oplus M_s(F)) \tag{19}$$

To enhance the original input features, $M(F)$ is then element-wise multiplied with the original input feature map and added to it to obtain the final feature map F_{refined}, which can be expressed mathematically as:

$$F_{\text{refined}} = F + (F \otimes M(F)) \tag{20}$$

2.2.3 | Dilated convolutional neural network (dilated CNN)

CNNs have been involved extensively in advancing DL within different domains of computer vision. They have proven highly effective in various tasks, such as image classification, object detection, and image processing. However, conventional CNNs cannot capture large receptive fields without significantly increasing the network's size and parameters. Although increasing the kernel size and receptive field can lead to extracting more informative features, it also increases computational complexity and resource requirements. To address this issue, dilated CNNs introduced by Yu and Koltun (2015) provide a solution that increases the receptive field without enlarging the kernel size. Dilated CNN is a beneficial technique for capturing multiscale contextual information while maintaining the integrity of high-resolution details. This method can build expanded feature maps with informative spatial data. The operation of a 2D dilated convolution can be expressed as follows (Zhang and Chen, 2018):

$$y = \sum_{i=1}^{M} \sum_{j=1}^{N} x(m + r \times i, n + r \times j)\omega(i, j) \tag{21}$$

where x and y are input and output, respectively, r is the dilated rate, which determines the spacing between sampled values and can be adjusted to control the degree of expansion of the receptive field, and ω is the filter. When $r = 1$, it corresponds to the standard (nondilated) convolution.

2.2.4 | Gated recurrent unit (GRU)

GRU is a recent successful variant of recurrent neural network (RNN) architecture that was initially proposed by Cho et al. (2014). GRU facilitates sequential data modeling by effectively mitigating the vanishing gradient problem and capturing long-term dependencies of different time scales. It is a simplified version of the long short-term memory (LSTM) network by utilizing a single reset gate (Chung et al., 2014).

At each time step t, the activation (hidden state) h_t is determined through a linear interpolation process. It is computed as a weighted average of the previous activation h_{t-1} and the candidate activation g_t:

$$h_t = (1 - z_t) \otimes h_{t-1} + z_t \otimes g_t \tag{22}$$

where $\otimes$ refers to element-wise multiplication, z_t denotes the update gate, which decides how much the unit should update its activation or content. The z_t is computed as follows:

$$z_t = \sigma\left(w_{xz}^{\mathsf{T}} \cdot x_t + w_{hz}^{\mathsf{T}} \cdot h_{t-1} + b_z\right) \tag{23}$$

here, w_{xz} and w_{hz} are the weight matrices for the input and hidden states, respectively, x_t is the input, b_z is the bias vector. The update gate in GRU simultaneously performs the role of the input and output gates in LSTM. The sigmoid function (σ) scales z_t to a range of values between 0 and 1, determining how much of h_{t-1} should be retained or how much of g_t can be adopted. The update gate controls the flow of information from the previous hidden state to the current hidden state.

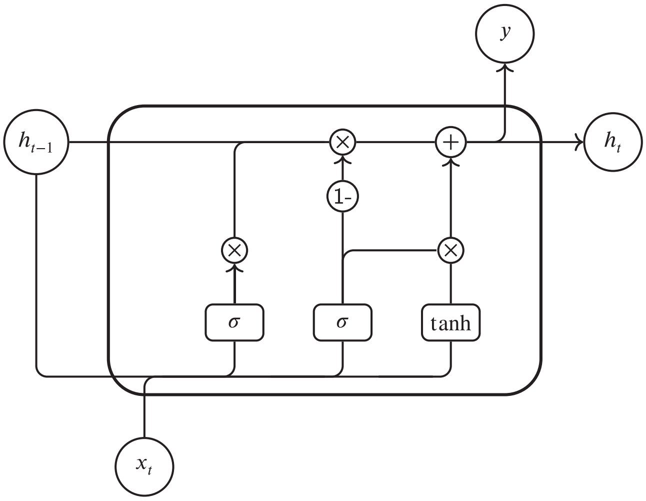


FIGURE 1 Architecture of a gated recurrent unit (GRU).

The candidate activation g_t can be computed similarly to the hidden state in conventional RNN as follows:

$$g_t = \tanh\left(w_{xg}^{\mathsf{T}} \cdot x_t + w_{hg}^{\mathsf{T}} \cdot (r_t \otimes h_{t-1}) + b_g\right) \tag{24}$$

here r_t is the *reset gate* that can be computed akin to the update gate as:

$$r_t = \sigma\left(w_{xr}^{\mathsf{T}} \cdot x_t + w_{hr}^{\mathsf{T}} \cdot h_{t-1} + b_r\right) \tag{25}$$

r_t decides how much of the hidden state to retain from the previous time step for a matrix-based update. Figure 1 illustrates the GRU architecture.

2.2.5 | Multihead attention mechanism

The attention function is a promising mechanism that maps a query, a set of key-value pairs, to produce an output. All the elements—query (Q), keys (K), value (V), outputs—are represented as vectors. The output is computed through a weighted sum of values, and each value's weight is computed based on a compatibility function between the query and its corresponding key. In the context of a single-head attention mechanism, an attention score is computed for each element in the input sequence, indicating their relevance to the query. The two most common attention functions are additive attention (Bahdanau et al., 2014) and multiplicative (dot-product) attention (Luong et al., 2015). While both functions are similar in terms of theoretical complexity, the multiplicative attention mechanism indicates significantly faster and becomes more space-efficient. The attention score can be computed as follows:

$$\text{Attention}(Q, V, K) = \text{Softmax}\left(\frac{Q \cdot K^{\intercal}}{\sqrt{d_k}}\right)V \tag{26}$$

Here, d_k represents the key vector dimension, and the Softmax operation ensures that the attention scores sum up to 1, providing a probability distribution over the elements in the sequence. The scaling factor $\frac{1}{\sqrt{d_k}}$ is employed to normalize the dot product, mitigating the potential magnification effect and excessively small gradients encountered when applying the Softmax function.

The multihead attention mechanism, first introduced in the paper "Attention is all you need" (Vaswani et al., 2017) in the proposed transformer model, employs linear transformation of the query (Q), key (K), and value (V) with d_q, d_k, and d_v dimensions, respectively. This is performed h times instead of relying on a single attention function. The multihead attention mechanism computes the attention for each head and then concatenates the results. This allows the model to focus on distinct aspects of the input sequence. The mathematical expression of this mechanism can be represented as follows:

$$MultiHead(Q, K, V) = Concat(Head_1, \dots, Head_h).W^O \tag{27}$$

$$Head_i = Attention\left(QW_i^{\,Q}, KW_i^{\,K}, VW_i^{\,V}\right) \tag{28}$$

$Head_i$ denotes the output of the i-th attention head, h signifies the number of attention heads, and W^O represents the learnable output weight matrix. Each attention head functions independently, capturing different aspects of the input sequence. The final output is achieved by concatenating the outputs of all attention heads and linearly transforming them using the output weight matrix W^O. The multihead attention mechanism is a powerful way to learn about complex dependencies and relationships between sequences, which makes it an essential part of many DL tasks. This mechanism is integrated into the proposed customized model following the GRU layers, which ultimately enhances the model's robustness in understanding data complexity and improves the model's generalization ability.

3 | DATA ACQUISITION AND PROBLEM CONTEXT

This section highlights two distinct data acquisition pathways, including controlled laboratory experiments and field tests from a few bridges in Italy.

3.1 | Data collection from laboratory tests

Laboratory tests were carried out on prestressed concrete poles, such as those often used in agricultural fields. Having a length of 2.8 m, a cross-section of 11×11 cm, and being made of high-strength concrete, they can be scaled-down models of prestressed concrete beams. Each specimen features four twisted prestressed tendons containing three wires, 12 wires in total. The diameter of a single wire is 2.25 mm. The alignment between laboratory and real-world conditions is crucial to secure robust and reliable experimental tests (see Figure 2). The AE signals were acquired using a MISTRAS Sensor Highway III system, which is considered cutting-edge instrumentation. The system was equipped with two resonant sensors from the PK family, designed for a medium frequency range, that is, between 20 and 500 kHz. The sensors feature an integrated, ultra-low noise, low power, and filtered preamplifier that offers a 26 dB amplification. The AE signals were captured with an acquisition rate of 500 kHz, while the duration of each recorded signal was set to 0.01433 s. In order to mitigate unwanted and irrelevant signals during data acquisition, an optimal triggering threshold, variable between 60 and 75 dB depending on the sensor-to-source distance, was set on amplitudes. One sensor was positioned at the center of the specimen, capturing the signals originating close to the event's occurrence, while the other was installed farther from the event. This setup not only enables the acquisition of a large set of breakage events (impossible in real bridges) and a comprehensive understanding of signal characteristics but also facilitates the exploration of the attenuation behavior of the events, contributing to highly reliable and variegated data, which directly impact the generalization ability of the models.

Four classes of events were collected, including wire breakage, ambient noise, drilling, and hammering. To collect the wire breakage events, the cross-section of wires was cut with an electric trimmer until generating a spontaneous tensile breaking. This test aims to simulate the critical event of wire breakage, which commonly happens in prestressed concrete beams due to factors such as corrosion. The ambient noise signals were collected because of operational noise that could occur before or during the tests. The drilling signals were collected to mimic potentially destructive activities on or close to the bridge structures. Ultimately, hammering tests were performed to replicate the impacts originating from maintenance activities or on bridge joints between spans. These varied test categories provide a comprehensive data set and lead to the understanding of the acoustic characteristics of various events, which is substantial for developing a robust monitoring system for early event detection.

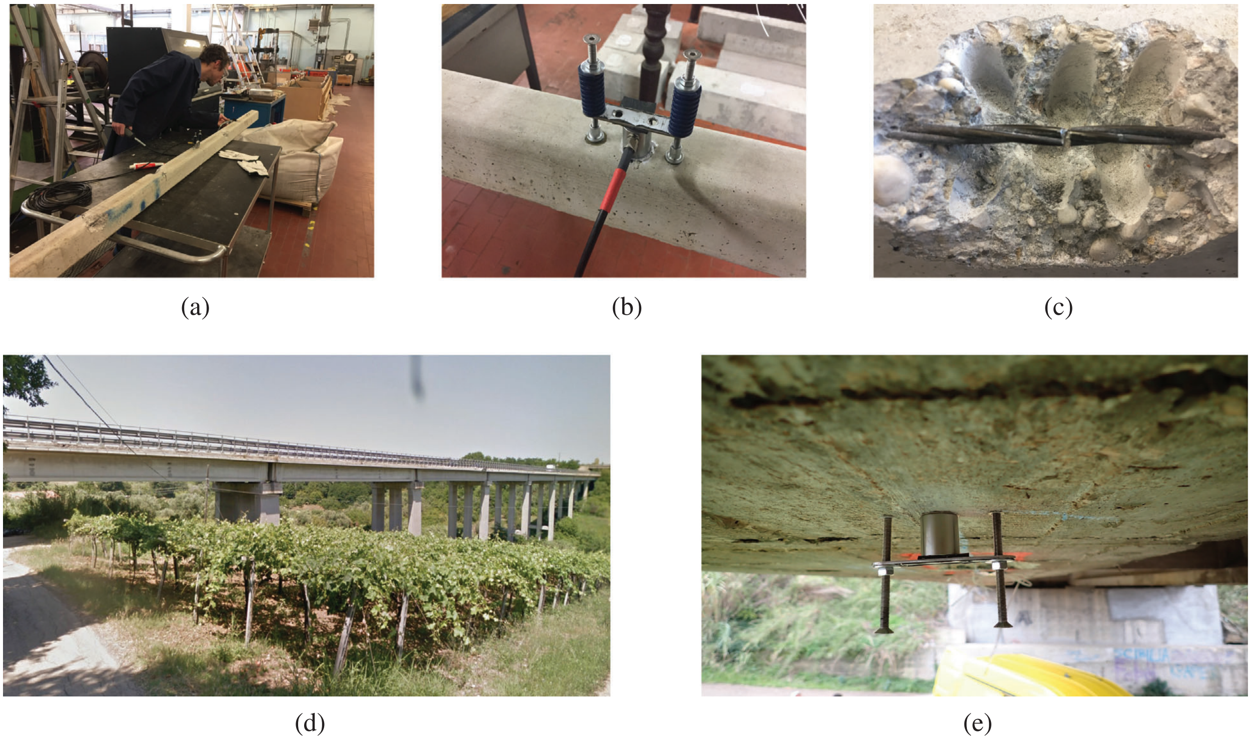

(a) (b) (c) (d) (e)

FIGURE 2 (a) Laboratory test setup featuring a prestressed concrete beam; (b) sensor placement on the surface of the concrete beam during laboratory tests; (c) close-up perspective capturing the wire breakage; (d) panoramic view of the Peticcio bridge; (e) positioning of sensors on the Peticcio bridge.

3.2 | Data collection from Italian bridges

A real-world data set was collected from different Italian bridges: Alveo Vecchio, located on the A16 Napoli–Canosa highway; Ansa del Tevere on the A91 Roma–Fiumicino highway; two viaducts on the SS4 national road; Pesco di Faggio on the SS212, and Peticcio on the SS16. Although the six bridges have different geometric features and material properties, they have a common structural arrangement. They consist of longitudinal prestressed reinforced concrete I-girders with a cast-inplace deck slab; the longitudinal girders are connected by cross beams. I-girders are prestressed by post-tensioned cables placed in grouted corrugated metallic sheaths.

Wire breakage in real bridges was triggered the same way as in laboratory tests, and acoustic emissions were recorded with the same system and configuration parameters to maintain methodological consistency. The acquisition sensors were installed at different distances from the point of event occurrence to include the effect of the signal attenuation in the study. The recorded signals were categorized into two distinct classes of events: wire breakage and ambient noise. In the case of Alveo Vecchio and Ansa del Tevere bridge, eight sensors were installed, whereas two sensors were installed in the remaining bridges. The methodological consistency guarantees that the signal characteristics observed in the laboratory can be directly compared with those from real-world scenarios, although a few differences emerge. First, signals from laboratory tests have more echo because of a smaller sample dimension. Anyway, echo is not an issue for event detection and classification because, even if it has lower amplitudes, it mainly replicates the pattern of the original signal. Another common difference is that the in situ data collection provides a more heterogeneous set of signals, especially concerning the amplitudes. However, even this feature does not affect much the performance of the models, which are trained to recognize the pattern within images rather than the intensity of the image's pixels.

The data set collected through on-site experimental tests was used to test and evaluate the robustness of the trained models rigorously. This evaluation can enhance understanding of the model's generalization capabilities under various conditions and contribute to developing a more reliable SHM approach.

4 | EXPERIMENT AND ANALYSIS

The experiments and analyses were conducted on two workstations equipped with an NVIDIA Tesla P40 (24 GB memory) and RTX A6000 (48 GB memory). These GPUs accelerate CNN training extensively, ensuring an efficient and reasonable time frame. For the customized model using GPU, each epoch requires 5–15 s based on the applied model. Additionally, the hardware system features two Intel Xenon CPUs with 16 cores each and 128 GB RAM, providing appropriate computational power for the research objectives. The DL models were developed using the Keras library high-level neural network API on the TensorFlow 2.0 framework. Python 3.9 was used as a programming language, taking advantage of its rich library ecosystem for scientific computing. The workstations operated on Windows Server 2019, leveraging CUDA Toolkit 11.2 and cuDNN8.0 for GPU acceleration.

4.1 | Data preparation and augmentation

Acquiring large amounts of data for wire breakage from existing bridges is challenging without decreasing their safety and load-bearing capacity. Such an activity is, therefore, limited to structures undergoing dismantling. To overcome this limitation, this study incorporated data from both controlled laboratory tests and real-world bridge monitoring. The laboratory tests were carried out on six prestressed concrete beams, providing 115 samples for wire breakage, 82 for ambient noise, 120 for drilling, and 120 for hammering. Additionally, the data set from real bridges includes 286 acoustic signals, consisting of 145 samples for wire breakage and 141 for ambient noise. The collection of laboratory and real-world data enriches the available data set, enhancing the robustness and applicability of the research. It is important to emphasize that the real-world data were utilized solely as an independent test set. This approach allowed us to rigorously evaluate the trained models' robustness and their generalization ability in real-case scenarios, thereby enhancing the reliability and applicability of the research outcomes. Figure 3 shows the signal time-domain for the four considered classes of events.

The parameters involved in training DL models are notably extensive; therefore, a substantial amount of data for each event is required to build a model that can generalize effectively across various real-world scenarios (Takahashi et al., 2016). Therefore, data augmentation (DA) approaches must be applied. A broad range of available DA methods includes jittering, time-stretching, time-warping (Iwana & Uchida, 2021), and mixup for addressing complex scenarios. In this study, considering the fundamental physics of the acoustic signal, the mixup technique (Zhang et al., 2017) was used. It is important to emphasize that the mixup augmentation technique was applied to the training data set after splitting the original laboratory data set. This strategy is considered to prevent data leakage during the augmentation and subsequent model training phases. This technique involves generating new training data by linearly interpolating existing data. Following the outlined procedure, the final training data set emerges as an extensive collection of acoustic signals containing 992 wire breakage, 995 drillings, 991 hammering, and 942 ambient noise for each representation. To build this extensive study, a data set of substantial size (40,000 images in total) was generated and utilized for both training and testing the models. This involved the development of DL models, each tailored for five unique spectrograms, with consideration for two different window sizes (128 and 256) in each case.

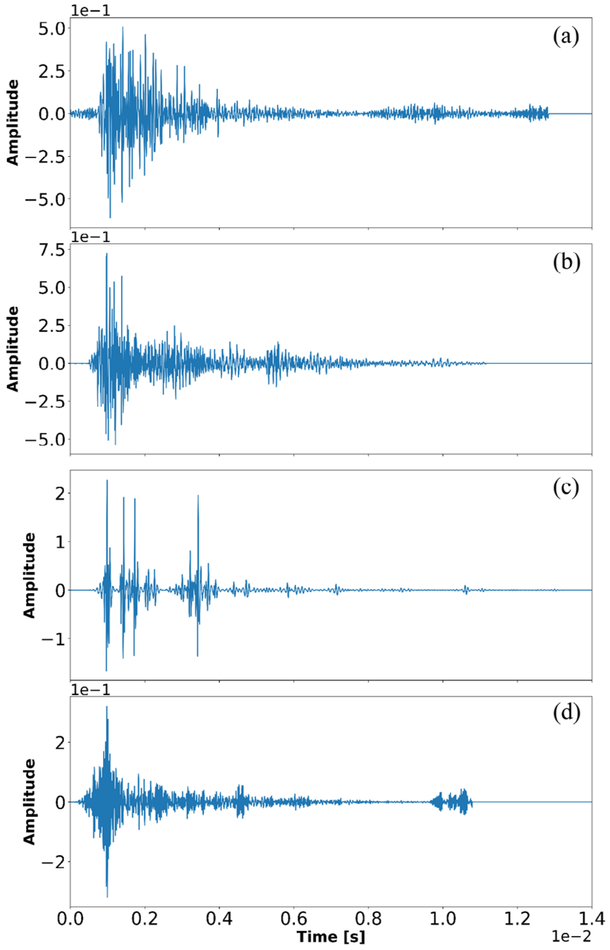
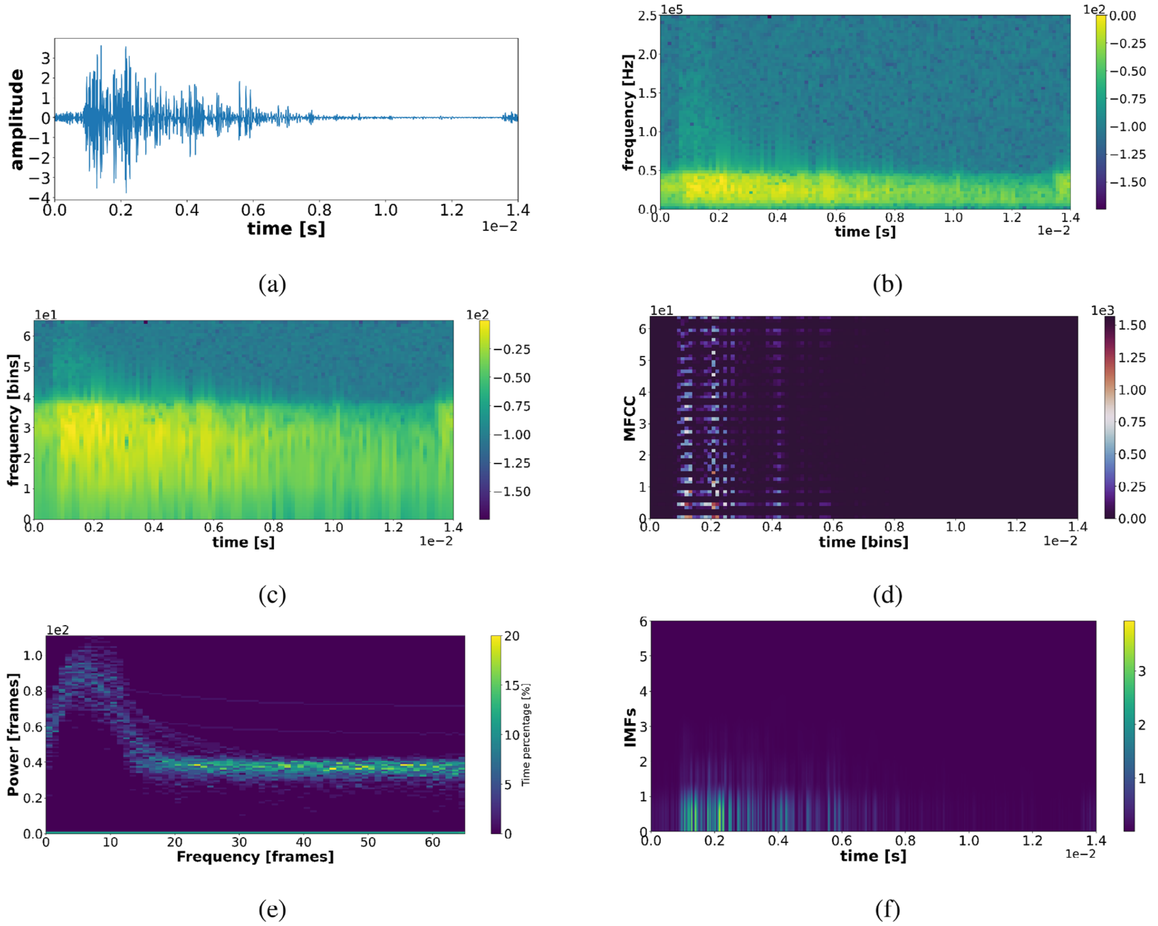


FIGURE 3 Time-domain signal representation for different events: (a) wire breakage, (b) drilling, (c) hammering, (d) environmental noise.

4.2 | Signal processing

Five signal representations–STFT, log-STFT, MFCC, PS, and HHT—were selected for their distinct abilities to capture different aspects of acoustic events. These methods



(a)

(b)

(c)

(d)

(e)

(f)

FIGURE 4 Wire breakage signal representations with window size of 128: (a) time-domain waveform, (b) short-time Fourier transform (STFT) spectrogram, (c) log-STFT spectrogram, (d) Mel-frequency cepstral coefficients (MFCC), (e) persistence spectrogram, (f) Hilbert–Huang transform.

enhance the understanding of acoustic events in prestressed concrete beams and enrich the feature space for robust event classification. Specific details of the signal analysis are discussed below.

The STFT application effectively captures the frequency content of event signals across time. This process enables a detailed examination of how different frequencies contribute to the dynamic characteristics of the signal. Utilizing the Hann window function with two distinct sizes (128 and 256 samples) facilitates a comprehensive analysis of the diverse frequency contributions to the signal's overall behavior. The deliberate selection of two window sizes covers a dual purpose: It provides versatility in capturing broader temporal features, and it allows for a robust comparison of the influence of this parameter on model learning. Additionally, a 50% overlap (hop size) was incorporated, ensuring continuity in the analysis, preventing information loss between adjacent frames, and enhancing the robust representation of signal dynamics. As a result of these considerations, the STFT analysis yields signal shapes of (111, 128) and (55, 256) for 128 and 256 frame sizes, respectively. In Figure 4b, the STFT spectrogram transformed from a wire breakage event is illustrated.

The log-frequency spectrogram is utilized to redefine the frequency axis to correspond to the logarithmically spaced frequency distribution of the equal-tempered scale. The log-spectrogram retains the advantages of the spectrogram while offering improved spectral clarity. This representation is particularly valuable for detecting subtle changes in the frequency components of acoustic emission signals, which can be beneficial for wire breakage detection. Figure 4c showcases the log-STFT spectrogram derived from a wire breakage event.

The MFCC extracted discriminative features from the events utilizing a triangular filter bank configured with 64 filters for the 128 window size and 128 filters for the 256 window size. The implication of the triangular filter bank leads to capturing frequency contents considering the human auditory perception. One notable strength of MFCC is its robustness to noise and environmental variability, making it a reliable choice for discerning variations and patterns in complex acoustic signals. This feature extraction aimed to represent the signals more compactly and informatively, focusing on assessing their impact on the model's performance. Visualizing the MFCC enhances the comprehension of dominant frequency components and their temporal variations. Figure 4d illustrates the MFCC representation of the wire breakage event.

During the data acquisition, there is always the possibility of existing random and abnormal frequencies, which pose a challenge to the event classification task, mainly when these frequencies exhibit low amplitude. To address this, the PS approach was utilized for 128 and 256 window sizes, resulting in dimensions of (111, 65) and (129, 55), respectively. The application of PS offers a unique advantage by representing multiple overlapping spectra rather than a singular line. The overlapping features provide a richer representation, revealing patterns often hidden when using other methods. The spectrum with more frequent spectral passages in these figures represents higher spectral density. This specific representation provides a detailed understanding of the temporal evolution of the topological features and addresses the intricate pattern arising from the coexistence of random and abnormal frequencies. Figure 4e shows the PS representation of the wire breakage event.

HHT-specific strengths lie in capturing nonstationary and nonlinear signal behaviors, shedding light on how signal energy evolves. Applying EMD within HHT without relying on predefined assumptions enables the extraction of localized features within the signal—an essential aspect in event detection. HHT adopts a data-driven and nonparametric approach, preserving signal energy without assuming prior knowledge, in contrast to methods like Fourier analysis, which assumes a periodic and stationary signal. The adoption of two different window sizes in HHT transforms their effect on the kernel and smooths the signal through the determination of the moving average filter size. Each IMF, derived from the EMD algorithm, highlights distinct frequency components, enabling the analysis of how these frequencies contribute to the signal over time. This approach provides valuable insights into the time-frequency distribution of the signal. Figure 4f shows the HHT representation of the wire breakage event.

The representations discussed are foundational elements and are employed as essential input features for the proposed DL models.

4.3 | Model development and training

4.3.1 | Model architecture

The pretrained models were trained on ImageNet, one of the largest data sets for image classification and recognition tasks, offering a vast collection of diverse images. The extensive diversity of ImageNet, especially textured images, aligns with the intricate patterns of signal representations.

TABLE 1 Metrics overview of pretrained convolutional neural network models.

Models	Size (MB)	Top-1 accuracy (%)	Top-5 accuracy (%)	Parameters (M)
VGG19	549	71.30	90.00	143.7
Inception	92	77.90	93.70	23.9
ResNet50	98	76.00	93.00	25.6
Xception	88	79.00	94.50	22.9

In this study, an extensive pretrained approach was employed using some of the most successful pretrained CNN architectures by Keras, including VGG19 (2014), Inception (2015), ResNet50 (2015), and Xception (2016). The selection of these architectures was precisely based on their unique design, ensuring a diverse approach for a comprehensive exploration of their efficacy in the specific task of event classification. Moreover, these models have shown remarkable performance on various image classification tasks and proved their reliability in TL scenarios. Each model has unique features that might benefit the event classification task differently. VGG19 is known for its simplicity, and it has a straightforward architecture that serves as a valuable baseline, offering insights for comparison with more complex models. Inception is characterized by modules employing various kernel sizes within the same layer; it enables capturing features at different scales, which helps to discern intricate patterns. ResNet50 is recognized for implementing skip or shortcut connections and facilitating effective training of deeper networks. Xception, a variant of GoogleNet, stands out with its depthwise separable convolutional layer, reducing computational complexity. Table 1 represents an overview of the selected pretrained model. In this table, "Top-1 accuracy" represents the percentage of instances where the model correctly predicted the correct label with the highest probability and "Top-5 accuracy" indicates the percentage of instances where the true label was among the top five predictions made by the model.

In this study, the feature extraction approach was utilized during the TL process. The lower layers, which have acquired generic features from pretraining, were frozen to retain their learned knowledge, including weights and biases. In contrast, the top layer was fine-tuned to better adapt the model to the specific requirements of the event classification tasks by adding two fully connected layers with 128 units. Furthermore, a dropout regularization layer with a 0.2 dropout rate was incorporated to enhance the model performance. This approach was consistently applied to all pretrained models, ensuring a fair

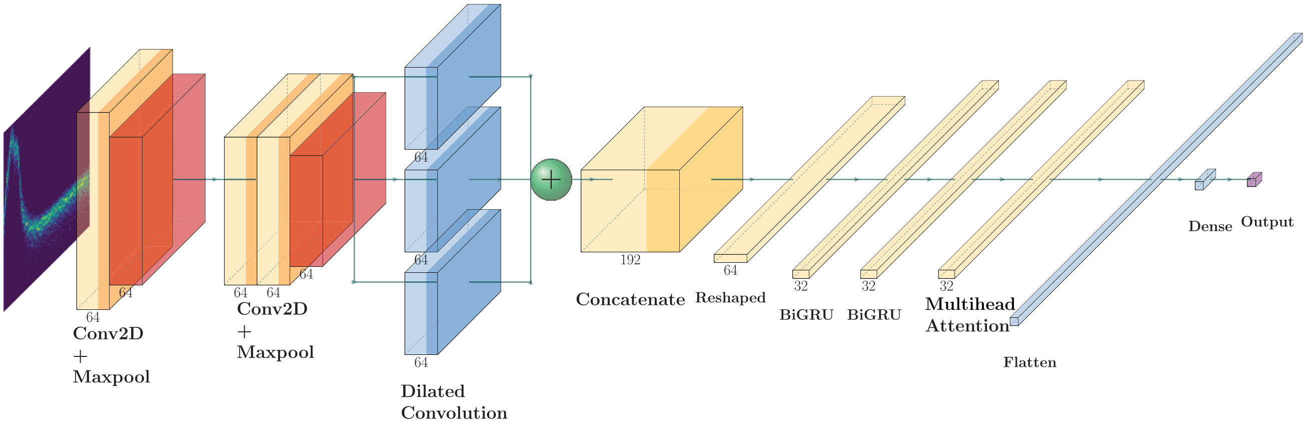


FIGURE 5 Architecture of the proposed customized model (AcousticNet).

and consistent comparison across the board. Sparse categorical crossentropy was chosen as the loss function for its suitability in classification tasks where each instance belongs to a single class.

The BAM was integrated into pretrained models to enhance their performance. This integration stems from the capability of the attention mechanism to enhance the model's focus on critical regions within the input data. BAM introduces a selective attention mechanism, allowing the model to dynamically weigh the importance of different spatial locations in the feature maps. This can play a key role in capturing intricate patterns and features within the signal representations, as it causes the model to prioritize information that is most relevant to the classification task. As recommended in the literature, the BAM block should be placed after the bottleneck blocks. These blocks are composed of a series of convolutional layers that refine the feature maps by reducing their spatial resolution while increasing depth. In this study, BAM was strategically placed following the last building block in the proposed pretrained models before the CNN output. The processed feature map from the BAM layer is then efficiently passed on to a subsequent MaxPooling layer. This setup allows for a comparative analysis of the impact of the BAM layer on model performance. The integration of BAM is tailored to avoid unnecessary memory or computational overhead, ensuring the efficiency of the architecture. Two hyperparameters are considered to optimize the BAM performance: reduction ratio and dilation rate, which are set to 16 and 4, respectively, as introduced in the original BAM research paper.

To further tailor the proposed approach for AE/DC, a customized hybrid model (AcousticNet) was designed to capture intricate patterns present in signal representations. This model integrates a combination of architectural elements to ensure the effective extraction of both spatial and temporal features. As depicted in Figure 5, the initial layer comprises a convolutional layer coupled with max-pooling layers, leveraging CNNs' well-known efficiency in extracting features from spatial data such as signal spectrograms. These layers excel at recognizing complex patterns, including edges and shapes, crucial for precise feature extraction. The subsequent layer introduces a second convolutional layer without pooling, followed by a third convolutional layer that incorporates max-pooling, boosting the model's capacity to discern textural features. Notably, all convolutional layers employ a kernel size of 3, optimizing their receptive fields for hierarchical representation of sequences, which, unlike RNNs, do not depend on previous step computations. This allows for the parallelization of processing, significantly enhancing computational efficiency. To further make the model robust in capturing detailed spatial patterns in the acoustic signals, three parallel dilated convolutional layers with a kernel size of 5 are used. Following this, two bidirectional gated recurrent units (BiGRUs) are strategically positioned to learn temporal dependencies present in the acoustic signals. A multihead attention mechanism with eight parallel attention layers is then incorporated into the model's architecture to focus on specific patterns within the signal, enhancing the model's ability to handle long-range dependencies with high parallelizability, similar to transformers. The subsequent layers include dense layers, utilized for feature aggregation, ending in the final classification layer. Careful consideration of hyperparameters, including the number of filters, kernel sizes, and attention mechanism parameters, such as the reduction ratio and dilation rate, was undertaken to balance computational efficiency and the capture of relevant information. The model output employs a softmax activation function, facilitating multiclass classification and providing probabilities for each event category.

TABLE 2 Summary of training parameters and configurations.

Parameter	Optimal value	Range
Data set split	Training: 65%, Validation: 15%, Test: 20%	–
Cross-validator	Stratified ShuffleSplit, 10-fold	–
Learning rate (initial)	5e-5	1e-3 to 1e-8
Learning rate adjustments	Factor: 0.15, Patience: 5	Factor: 0.1 to 0.5, Patience: 3 to 10
Batch size	32	16 to 64
Epochs	Max 100	50 to 250
Optimizer	Nadam (Adam with Nesterov momentum)	Adam, Nadam, RMSprop
Loss monitoring	Early stopping after 5 epochs with no improvement	–
Regularization	Batch normalization, Dropout rate: 0.2	Dropout: 0.1 to 0.4
Activation function	Leaky ReLU	ReLU, Leaky ReLU, ELU

4.3.2 | Training procedure

The data set was split into training, validation, and test sets to build a robust evaluation framework. The test size was set to 20% of the entire data set, providing a separate set of instances for final model evaluation. A stratified ShuffleSplit cross-validator was employed to ensure a balanced representation of classes in both training and validation sets. This method shuffles the data set before splitting, providing randomness to the sample selection process and maintaining the proportional distribution of classes observed in the original data set. This causes an unbiased performance during the testing phases. Tenfold was selected for this procedure with a 15% size for the validation set.

It is noteworthy that to have a fair comparison, the training procedure is set to be the same across all the models. Selecting an optimal learning rate is substantial for efficient model training. To address this, a learning rate schedule strategy with step-based adjustments was employed. The initial learning rate was set to 5×10^{-5}, and dynamic adjustments with a factor of 0.15 and patience of 5 were considered based on preliminary experiments. This adaptive strategy causes efficient convergence, especially in the later stages of training when the optimization landscape becomes narrower. Specifically, the validation loss was monitored to determine whether there was an improvement or not. All the models in this study were set to train for a maximum of 100 epochs, and a batch size of 32 was selected to balance the computational efficiency and model convergence. Moreover, early stopping was implemented to monitor the validation loss, and the training was halted if no improvement was observed after five consecutive epochs. This regularization approach also helps avoid overfitting and ensures that the model generalizes well on the unseen data.

Nadam optimization, which is Adam optimization plus the Nestrov momentum, was selected as it has shown a slightly faster convergence than Adam. The Nesterov momentum accelerates convergence by considering the future gradient direction, contributing to more effective optimization. This is particularly advantageous for handling large-scale data sets and high-dimensional parameter spaces, making it a reasonable choice for TL scenarios. The “leaky-relu” activation function was employed due to its ability to handle vanishing gradient problems by leaking some gradient backward. To further enhance model robustness and mitigate overfitting, a dropout regularization layer with a rate of 0.2 was incorporated. Table 2 represents the summary of the training parameters.

4.4 | Results

In this section, the results of the comprehensive evaluation of different pretrained models and the proposed AcousticNet model are presented. The main goal is to evaluate the performance of pretrained models, investigate the impact of integrating BAM into the selected pretrained models, and evaluate the proposed customized model. Key metrics, including F1-score, Cohens Kappa, Fowlkes–Mallows index (FMI), and Matthews Correlation Coefficient (MCC), will be employed to provide a detailed assessment.

4.4.1 | Pretrained models

Table 3 represents the metrics performance of various pretrained models. The model evaluation was assessed across different signal representations with two selected window sizes of 128 and 256.

The VGG19 model, chosen as the baseline for its simplicity and effectiveness, demonstrates commendable performance across diverse representations. The best performance is observed in the log-STFT representation using

TABLE 3 Performance metrics of pretrained models across diverse spectrogram representations.

		Spectrograms									
		STFT		Log-STFT		MFCC		PS		HHT	
Models	**Metrics**	**128**	**256**	**128**	**256**	**128**	**256**	**128**	**256**	**128**	**256**
VGG19	F-1 score	82.78	84.30	85.14	85.76	70.66	75.57	81.47	85.47	74.13	74.78
	Cohens Kappa	77.70	79.90	81.08	81.88	61.26	67.40	74.95	81.00	66.01	67.30
	FMI	70.71	74.74	75.08	76.31	57.34	60.15	67.00	74.01	58.01	59.58
	MCC	77.92	79.99	81.48	82.07	61.34	67.60	75.47	81.38	66.67	67.70
ResNet50	F-1 score	90.81	87.19	93.54	89.41	74.96	72.63	82.20	83.92	80.98	81.97
	Cohens Kappa	88.41	83.46	91.31	86.42	67.03	63.73	76.16	78.50	75.93	77.01
	FMI	84.32	78.11	87.15	81.41	60.34	56.32	68.40	71.00	70.31	71.18
	MCC	88.54	83.69	91.45	86.74	67.12	63.88	76.55	78.66	76.08	77.16
Inception	F-1 score	86.67	85.97	87.90	87.86	71.65	76.75	76.63	82.84	77.90	78.47
	Cohens Kappa	82.94	82.04	84.82	84.34	62.56	69.55	68.91	77.04	71.21	71.96
	FMI	77.34	76.58	80.73	78.75	54.91	62.25	61.26	69.37	63.98	64.75
	MCC	83.19	82.14	85.09	84.76	62.61	69.65	69.21	77.12	71.25	72.00
Xception	F-1 score	91.91	87.52	93.54	90.29	72.67	74.63	78.52	79.50	76.74	76.63
	Cohens Kappa	89.97	84.07	91.76	87.51	64.37	66.76	71.35	72.70	70.40	70.25
	FMI	86.94	79.21	88.42	82.63	58.03	59.24	63.01	64.10	64.25	64.08
	MCC	90.03	88.40	91.81	87.59	64.55	66.95	71.89	73.02	70.25	70.40

Abbreviations: STFT, short-time Fourier transform; MFCC, Mel-frequency cepstral coefficients; PS, persistence spectrogram; HHT, Hilbert–Huang transform; FMI, Fowlkes-Mallows index; MCC, Matthews Correlation Coefficient.

a window size of 256, achieving an FMI of 76.31% and an MCC of 82.07%. VGG19 faces challenges in accurately identifying the wire breakage, misclassifying it as drilling and noise.

ResNet50 showcases a consistent and competitive performance across the utilized signal representations. The best performance is achieved with the log-STFT representation with a window size of 128, where FMI and MCC are equal to 87.15% and 91.45%, respectively. The second-best performance was achieved by using STFT representation with a window size of 128. Similar to ResNet50, Inception achieved its best performance using log-STFT with a window size of 128, obtaining 80.73% and 85.09% for FMI and MCC, respectively. The model performance and efficacy varied significantly across different representations, which addresses the existing challenge of optimizing the model.

The Xception model, known for its depth and high performance, indicates remarkable performance across signal representations, particularly in STFT and log-STFT. Its best result was achieved using log-STFT with a window size of 128, obtaining 88.42% and 91.81% for FMI and MCC, respectively. However, a notable contrast occurs when using MFCC, PS, and HHT representations. In particular, using MFCC with a window size of 128 reveals a significant performance drop.

All models' performances highlight sensitivity to different representations and window sizes, emphasizing the need to explore various representations for optimal detection. The Xception model using log-STFT with a window size of 128 achieved the best performance and highest metrics over all the other models.

4.4.2 | BAM-integrated models

As indicated in Table 4, the incorporation of BAM consistently elevates the performance of pretrained models across various signal representations. Notably, the models with BAM outperform their counterparts, showcasing significant improvements, particularly in the PS representation. The utilization of the BAM to the VGG19 model results in consistent performance improvements across all representations. The most notable enhancement is achieved in log-STFT with a window size of 128, where the FMI and MCC experience substantial improvements of over 10% resulting in 83.31% and 88.46%, respectively. However, its performance over MFCC, PS, and HHT highlights the potential for improvements.

ResNet50 integrated with BAM demonstrates substantial and consistent performance enhancements across various signal representations. Notably, it achieved a remarkable boost in MFCC representation with a window size of 256, showcasing an impressive improvement of over 30% in FMI. Overall, the results over different represen-

TABLE 4 Performance metrics of pretrained BAM-integrated models across diverse spectrogram representations.

		Spectrograms									
		STFT		Log-STFT		MFCC		PS		HHT	
Models	Metrics	128	256	128	256	128	256	128	256	128	256
VGG19+BAM	F-1 score	88.33	87.12	91.05	90.05	71.10	73.41	86.15	81.24	72.70	76.29
	Cohens Kappa	84.42	83.07	88.26	87.05	61.84	66.53	81.13	75.23	65.09	70.88
	FMI	78.24	76.92	83.31	82.04	55.49	62.42	73.50	66.98	59.29	65.80
	MCC	84.49	83.27	88.46	87.13	61.90	66.82	81.26	75.60	65.70	71.47
ResNet50+BAM	F-1 score	92.30	92.81	91.75	93.90	83.50	83.93	88.39	89.44	81.23	81.89
	Cohens Kappa	89.93	90.57	89.03	91.92	78.78	79.85	84.39	85.90	75.85	77.10
	FMI	85.97	86.66	84.43	88.25	72.30	74.32	77.95	80.00	68.82	71.01
	MCC	90.01	90.61	89.09	91.98	78.93	79.94	84.50	85.97	76.49	77.42
Inception+BAM	F-1 score	92.87	93.40	92.63	93.33	86.39	82.87	87.42	87.06	87.00	84.15
	Cohens Kappa	90.88	91.32	90.25	91.32	82.76	78.20	82.96	82.65	83.50	80.14
	FMI	87.19	87.51	86.00	87.68	77.42	72.31	76.17	75.65	78.00	74.67
	MCC	90.94	91.37	90.37	91.39	82.81	78.26	83.08	82.73	83.63	80.38
Xception+BAM	F-1 score	91.88	92.65	94.05	94.05	81.65	83.14	87.07	87.14	85.95	84.04
	Cohens Kappa	89.18	90.58	92.38	92.55	76.61	78.75	82.68	82.80	82.13	79.37
	FMI	84.57	87.03	89.23	89.76	70.13	72.97	75.95	75.95	76.31	72.72
	MCC	89.32	90.73	92.44	92.64	76.81	78.90	82.86	82.84	82.23	79.65

Abbreviations: BAM, Bottleneck Attention Module; STFT, short-time Fourier transform; MFCC, Mel-frequency cepstral coefficients; PS, persistence spectrogram; HHT, Hilbert–Huang transform; FMI, Fowlkes-Mallows index; MCC, Matthews Correlation Coefficient.

tations have less variance. The best model performance was obtained by utilizing log-STFT with a window size of 256, where it achieves an FMI of 88.25% and an MCC of 91.98%. Additionally, it could achieve the best result using PS representation among all the other models for both window sizes.

Inception with BAM similarly could experience substantial improvements across different signal representations, underscoring its adaptability. The highest enhancements are achieved in the MFCC with a window size of 128. The model achieves a remarkable increase of over 40% in FMI and a substantial 30% improvement in MCC, reaching 77.42% and 82.81%, respectively. While the most substantial improvements occur in MFCC, it is noteworthy that its overall best performance is in log-STFT representation with a window size of 256. The evaluation indicates that, in both STFT and log-STFT representations, this model achieved superior performance when utilizing a window size of 256. This suggests that the model benefits from a larger temporal context when processing these specific signal representations. Among all the proposed pretrained models, Inception with BAM could achieve the best results using HHT representation.

Xception with BAM not only enhances model performance relative to its counterparts but also appears as the top performer among all other pre-trained models. Utilizing log-STFT with a window size of 256, this model achieves outstanding metrics: FMI of 89.76%, MCC of 92.64%. The model robustness is evident by just four misclassified outputs, proving its reliability in accurately distinguishing various patterns within signals. While log-STFT performs best with remarkable results, other representations such as MFCC, PS, and HHT also experience substantial enhancements. These improvements underscore the model adaptability and its ability to extract intricate features from a diverse range of spectrogram representations.

4.4.3 | Evaluation of AcousticNet

The performance metrics of the AcousticNet model across selected signal representations are presented in Table 5, showcasing its proficiency in event classification across various scenarios. Remarkably, this model exhibits reduced prediction variability compared to pretrained models. AcousticNet not only outperforms all pretrained models, indicating significant performance enhancements for MFCC, PS, and HHT representations, but it also achieves comparative results to pretrained models with BAM. The best performance for this model is observed using log-STFT with a window size of 128, achieving MCC results of 89.82% and FMI of 85.02%. Compared to the other models, AcousticNet excels in PS representations with a window size of 128, where it obtains 85.35% and 89.79% for FMI and MCC, respectively.

TABLE 5 Performance metrics of pretrained models across diverse spectrogram representations.

		Spectrograms									
		STFT		Log-STFT		MFCC		PS		HHT	
Models	Metrics	128	256	128	256	128	256	128	256	128	256
AcousticNet	F-1 score	91.97	90.76	92.49	90.29	80.87	82.85	92.09	87.08	82.06	84.50
	Cohens Kappa	89.32	87.65	89.78	86.59	75.20	78.48	89.65	82.81	77.22	80.13
	FMI	84.61	82.30	85.02	80.60	68.47	73.07	85.35	75.88	70.74	73.64
	MCC	89.37	87.69	89.82	86.64	75.30	78.60	89.79	83.05	77.52	80.37

Abbreviations: STFT, short-time Fourier transform; MFCC, Mel-frequency cepstral coefficients; PS, persistence spectrogram; HHT, Hilbert–Huang transform; FMI, Fowlkes-Mallows index; MCC, Matthews Correlation Coefficient.

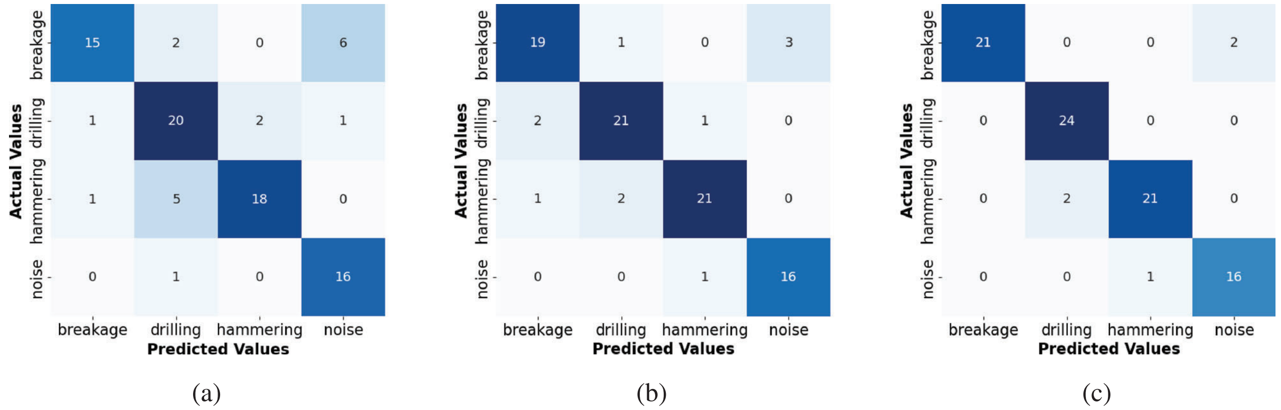


FIGURE 6 Confusion matrices comparing model performance across different architectures using a 128-window size persistence spectrum: (a) Xception, (b) Xception+BAM, (c) AcousticNet.

In addition, AcousticNet ranks as the third-best model for both window sizes in the HHT representation and MFCC, with a window size of 256. AcousticNet stands out as a robust model for event classification across diverse spectrogram representations, exhibiting superior performance and promising potential for applications in acoustic event detection. Further comparative analysis with other pretrained models will enhance our understanding of AcousticNet's strengths in various signal processing scenarios.

For the sake of example, Figure 6 illustrates the confusion matrices across different models that are useful to understand what classes are misclassified. Considering all models performance, wire breakage was mostly misclassified with drilling events.

In the context of this study, the tolerance for false negatives (FNs) can be relatively higher. For instance, in the Ansa del Tevere bridge, each beam contains 378 wires so, missing a single breakage does not immediately risk structural integrity. An FN rate of 5% to 10% is therefore acceptable. Conversely, the impact of false positives (FPs) is more significant due to the operational implications. Even a low FP rate (e.g., 1%) could lead to frequent and unnecessary interventions that are not practical or cost-effective. Therefore, the primary target is minimizing FPs.

4.4.4 | Real-world application

The evaluation of AcousticNet and Xception+BAM models on real-world data from various Italian bridges, as shown in Table 6, highlights the challenges of uncontrolled environments. Performance metrics across different spectrogram inputs (STFT, log-STFT, MFCC, PS, HHT) and resolutions (128, 256) demonstrate the adaptability and efficiency of these models under real-world conditions.

The Xception+BAM model exhibited high performance in high-resolution STFT (256) settings, achieving a remarkable F-1 score of 97.71%, alongside Cohen's Kappa, FMI and MCC scores surpassing 95%. This performance underscores its remarkable ability to distinguish event differences, which is important for applications demanding high-frequency resolution. Conversely, AcousticNet's performance in the same domain was notably less impressive, indicating a potential limitation in handling STFT data.

AcousticNet, on the other hand, showcased its strength in processing log-STFT 128 spectrograms, where it achieved an F-1 score of 83.18%. This indicates a high potential to handle logarithmically transformed signals, suggesting its suitability for scenarios where signal dynamic range compression is advantageous. The Xception+BAM model demonstrated a reduced capacity

TABLE 6 Performance metrics of best models on real-world data.

		Spectrograms									
		STFT		Log-STFT		MFCC		PS		HHT	
Models	**Metrics**	**128**	**256**	**128**	**256**	**128**	**256**	**128**	**256**	**128**	**256**
Xception+BAM	F-1 score	87.47	97.71	55.85	40.61	34.51	34.81	44.93	11.40	39.58	34.02
	Cohens Kappa	78.65	95.43	28.84	8.75	2.13	2.48	2.07	0.00	5.56	8.81
	FMI	87.06	95.57	68.93	66.89	66.66	66.20	56.23	80.10	64.24	66.99
	MCC	79.79	95.52	36.62	17.07	4.82	5.29	2.76	0.00	10.04	16.79
AcousticNet	F-1 score	48.86	38.09	83.18	55.27	58.83	55.28	33.17	25.90	42.65	45.73
	Cohens Kappa	16.30	4.93	68.71	24.53	27.78	23.34	1.75	0.50	9.61	11.65
	FMI	63.39	67.10	77.20	62.51	61.40	61.42	69.79	62.07	63.42	61.30
	MCC	27.27	12.11	72.36	37.27	34.82	31.87	8.67	4.23	17.25	19.48

Abbreviations: STFT, short-time Fourier transform; MFCC, Mel-frequency cepstral coefficients; PS, persistence spectrogram; HHT, Hilbert–Huang transform; BAM, Bottleneck Attention Module; FMI, Fowlkes-Mallows index; MCC, Matthews Correlation Coefficient.

to effectively process logarithmic frequency representations, as evidenced by its lower performance in both resolutions of log-STFT.

Even though Xception+BAM reaches its high performance in specific scenarios like high-resolution STFT, its efficiency across other spectrograms and resolutions varied significantly, indicating a specialization that may limit its applicability in diverse conditions. In contrast, AcousticNet exhibited consistency across broad spectrogram types, including log-STFT and MFCC, indicating a more adaptable application. The distinct performance patterns of these models highlight the importance of selecting the right spectrogram input and resolution based on the specific characteristics of the acoustic emission signals and the environmental context.

4.5 | Constraints and future directions

Although this research significantly advances automated event detection, there are some limitations. Complex architectures and optimized parameters can be hard to interpret, complicating decision making. Hyperparameter optimization requires substantial computational power, limiting scalability. Despite its high efficacy, the proposed hybrid model may struggle with data outside the training set. Data scarcity and relevant events, although partially mitigated by DA, remain significant constraints. The negative impact of the signal attenuation on the model performance shall also be considered when properly designing the position of the acquisition sensors. Additionally, the variety of potential events that can occur in structures is not limited to the four event types defined in this study. Real-world applications may encounter diverse events occurring in different locations and scenarios, which makes detection and classification more demanding and complex. This diversity requires a model capable of generalizing well across a broad range of acoustic signatures while maintaining high reliability. Moreover, practical deployment challenges, including the high costs of extensive sensor networks, vulnerability to extreme weather conditions, and difficulties accessing structural locations, are critical to address. Advancements in cost-effective sensor technologies, enhancing sensor durability for weather protection, and strategic sensor placement are essential for the broad adoption of this monitoring technology.

Despite the achievements of the proposed model, it could be beneficial to explore new approaches that can automatically configure the model based on the data. Techniques such as Natural Architectural Search (NAS) (Mellor et al., 2021; Zoph & Le, 2016) or advanced classification algorithms (Ning & Xie, 2024; Rafiei & Adeli, 2017) might offer more effective and adaptive solutions to this problem. Additionally, the application of advanced learning frameworks such as the finite element machine (FEMa) for fast learning (Pereira et al., 2020) could enhance the learning efficiency of the learning model. Similarly, employing a Dynamic Ensemble Learning Algorithm (Alam et al., 2020) and quantum CNN (Bhatta and Dang, 2024) can improve the robustness and accuracy by effectively managing diverse learning models in a unified framework. Moreover, incorporating a deterministic algorithm for nonlinear, fatigue-based SHM (Pavlou, 2022) can enhance the accuracy of fatigue damage estimation and life prediction under complex loading conditions. The integration of self-supervised learning techniques, as explored by Rafiei et al. (2024), might also allow for leveraging unlabeled data, potentially abundant in SHM scenarios. Moreover, incorporating explainable AI techniques (Cheng et al., 2021) could enhance the transparency of the predictions, enabling a more in-depth understanding and trust in the decision-making process of the models. These advancements could lead to a more nuanced and powerful tool for early detection and monitoring in civil engineering.

5 | CONCLUSION

This study highlights the effectiveness of novel DL-based approaches, specifically the pretrained+BAM and AcousticNet models, for AE/DC tailored for the early monitoring of structural damage in prestressed concrete bridges. Comprehensive evaluations, detailed in Section 4, demonstrate that AcousticNet and Xception+BAM models are proficient in handling complex acoustic signal patterns. AcousticNet achieved an F1 score of 92.49%, while Xception+BAM reached 94.05%, as illustrated in Tables 4 and 5. Considering the information in Table 6, AcousticNet demonstrates more robust and consistent performance across diverse data sets, whereas the Xception+BAM model excels particularly in certain representations. Among the tested signal representations, the log-STFT was identified as the most effective dynamic signal representation, enhancing the models' ability to detect and classify structural damage events due to its robust handling of dynamic patterns. The application of the proposed approaches was validated through experiments on both laboratory and real-world bridges, highlighting the potential of the AE/DC approach in improving monitoring strategies and enhancing infrastructure safety and integrity.

ACKNOWLEDGMENTS

Research Grant "XFAST-SIMS: Extra fast and accurate simulation of complex structural systems" by the Italian Ministry of Education, University, and Research and research support by Politecnico di Torino with the diffuse research grant initiative are gratefully acknowledged. Gratitude is extended to ANAS S.p.A. and Ing. Alberto Gennari Santori of Cesano Research Center for their availability and support in collecting real-world bridge data. Thanks are also due to Autostrade per l'Italia S.p.A. and the University of Trento for their support in collecting Alveo Vecchio bridge data.

Open access publishing facilitated by Politecnico di Torino, as part of the Wiley - CRUI-CARE agreement.

CONFLICT OF INTEREST STATEMENT

The authors declare no potential conflict of interest.

REFERENCES

Alam, K. M. R., Siddique, N., & Adeli, H. (2020). A dynamic ensemble learning algorithm for neural networks. *Neural Computing and Applications*, *32*(12), 8675–8690.

Alani, A. M., Aboutalebi, M., & Kilic, G. (2014). Integrated health assessment strategy using NDT for reinforced concrete bridges. *NDT & E International*, *61*, 80–94.

Ardito, C., Deldjoo, Y., Noia, T. D., Sciascio, E. D., & Nazary, F. (2022). Visual inspection of fault type and zone prediction in electrical grids using interpretable spectrogram-based CNN modeling. *Expert Systems with Applications*, *210*, 118368.

Avci, O., Abdeljaber, O., Kiranyaz, S., Hussein, M., Gabbouj, M., & Inman, D. J. (2021). A review of vibration-based damage detection in civil structures: From traditional methods to machine learning and deep learning applications. *Mechanical Systems and Signal Processing*, *147*, 107077.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXive:1409.0473*.

Bassuoni, M., & Rahman, M. (2016). Response of concrete to accelerated physical salt attack exposure. *Cement and Concrete Research*, *79*, 395–408.

Bhatta, S., & Dang, J. (2024). Multiclass seismic damage detection of buildings using quantum convolutional neural network. *Computer-Aided Civil and Infrastructure Engineering*, *39*(3), 406–423.

Bray, C. W. (1928). Transfer of learning. *Journal of Experimental Psychology*, *11*(6), 443–467.

Cawley, P. (2018). Structural health monitoring: Closing the gap between research and industrial deployment. *Structural Health Monitoring*, *17*(5), 1225–1244.

Cheng, C., Behzadan, A. H., & Noshadravan, A. (2021). Deep learning for post-hurricane aerial damage assessment of buildings. *Computer-Aided Civil and Infrastructure Engineering*, *36*(6), 695–710.

Cho, K., van Merrienboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXive:1409.1259*.

Chollet, F. (2018). *Deep learning with Python*. Manning Publications Co.

Chu, S., Narayanan, S., Kuo, C. C. J., & Mataric, M. J. (2006). Where am I? Scene recognition for mobile robots using audio features. In *2006 IEEE international conference on multimedia and expo* (pp. 885–888). IEEE.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Dubuc, B., Sitaropoulos, K., Ebrahimkhanlou, A., & Salamone, S. (2021). Acoustic emission diagnostics of corrosion monitoring in prestressed concrete using hidden Markov and semi-Markov models. *Structural Health Monitoring*, *20*(6), 2899–2916.

Eltouny, K. A., & Liang, X. (2023). Large-scale structural health monitoring using composite recurrent neural networks and grid environments. *Computer-Aided Civil and Infrastructure Engineering*, *38*(3), 271–287.

Farhadi, S., Corrado, M., Borla, O., & Ventura, G. (2024). Prestressing wire breakage monitoring using sound event detection. *Computer-Aided Civil and Infrastructure Engineering*, *39*(2), 186–202.

Farhadi, S., Tatullo, S., Boveiri Konari, M., & Afzal, P. (2024). Evaluating StackingC and ensemble models for enhanced lithological classification in geological mapping. *Journal of Geochemical Exploration*, *260*, 107441.

Foggia, P., Petkov, N., Saggese, A., Strisciuglio, N., & Vento, M. (2016). Audio surveillance of roads: A system for detecting anomalous sounds. *IEEE Transactions on Intelligent Transportation Systems*, *17*(1), 279–288.

Fox, M. J., Furinghetti, M., & Pavese, A. (2023). Application of the new Italian assessment guidelines to a 1960s prestressed concrete road bridge. *Structural Concrete*, *24*(1), 583–598.

Gao, Y., & Mosalam, K. M. (2018). Deep transfer learning for image-based structural damage recognition. *Computer-Aided Civil and Infrastructure Engineering*, *33*(9), 748–768.

Gao, Y., Yang, J., Qian, H., & Mosalam, K. M. (2023). Multiattribute multitask transformer framework for vision-based structural health monitoring. *Computer-Aided Civil and Infrastructure Engineering*, *38*(17), 2358–2377.

Giglioni, V., Venanzi, I., Poggioni, V., Milani, A., & Ubertini, F. (2023). Autoencoders for unsupervised real-time bridge health assessment. *Computer-Aided Civil and Infrastructure Engineering*, *38*(8), 959–974.

Hampshire, T. A., & Adeli, H. (2000). Monitoring the behavior of steel structures using distributed optical fiber sensors. *Journal of Constructional Steel Research*, *53*(3), 267–281.

Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., Yen, N.-C., Tung, C. C., & Liu, H. H. (1998). The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, *454*(1971), 903–995.

Iwana, B. K., & Uchida, S. (2021). An empirical survey of data augmentation for time series classification with neural networks. *PLOS ONE*, *16*(7), e0254841.

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255–260.

Kumbasar, N., Kılıç, R., Oral, E. A., & Ozbek, I. Y. (2022). Comparison of spectrogram, persistence spectrum and percentile spectrum based image representation performances in drone detection and classification using novel HMFFNet: Hybrid Model with Feature Fusion Network. *Expert Systems with Applications*, *206*, 117654.

Lee, C.-Y., & Le, T.-A. (2021). Identifying faults of rolling element based on persistence spectrum and convolutional neural network with ResNet structure. *IEEE Access*, *9*, 78241–78252.

Li, F., Yuan, Y., & Li, C.-Q. (2011). Corrosion propagation of prestressing steel strands in concrete subject to chloride attack. *Construction and Building Materials*, *25*(10), 3878–3885.

Li, Y., Zhang, X., & Chen, D. (2018). CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/cvpr.2018.00120

Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Ma, G., & Wu, C. (2023). Crack type analysis and damage evaluation of BFRP-repaired pre-damaged concrete cylinders using acoustic emission technique. *Construction and Building Materials*, *362*, 129674.

Madhu, A., & Surech, K. (2023). cRQNet: Residual quaternion CNN for performance enhancement in low complexity and device robust acoustic scene classification. *IEEE Transactions on Multimedia*, *25*, 8780–8792.

McLoughlin, I., Zhang, H., Xie, Z., Song, Y., & Xiao, W. (2015). Robust sound event classification using deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *23*(3), 540–552.

Mellor, J., Turner, J., Storkey, A., & Crowley, E. J. (2021). Neural architecture search without training. In *International conference on machine learning*. (pp. 7588–7598). PMLR.

Mesaros, A., Heittola, T., Virtanen, T., & Plumbley, M. D. (2021). Sound event detection: A tutorial. *IEEE Signal Processing Magazine*, *38*(5), 67–83.

Ning, C., & Xie, Y. (2024). Convolutional variational autoencoder for ground motion classification and generation toward efficient seismic fragility assessment. *Computer-Aided Civil and Infrastructure Engineering*, *39*(2), 165–185.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359.

Park, J., Woo, S., Lee, J.-Y., & Kweon, I. S. (2018). Bam: Bottleneck Attention Module. *arXiv preprint arXiv:1807.06514*.

Pavlou, D. (2022). A deterministic algorithm for nonlinear, fatigue-based structural health monitoring. *Computer-Aided Civil and Infrastructure Engineering*, *37*(7), 809–831.

Pereira, D. R., Piteri, M. A., Souza, A. N., Papa, J. P., & Adeli, H. (2020). FEMa: A finite element machine for fast learning. *Neural Computing and Applications*, *32*(10), 6393–6404.

Rafiei, M. H., & Adeli, H. (2017). A new neural dynamic classification algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(12), 3074–3083.

Rafiei, M. H., & Adeli, H. (2018). A novel unsupervised deep learning model for global and local health condition assessment of structures. *Engineering Structures*, *156*, 598–607.

Rafiei, M. H., Gauthier, L. V., Adeli, H., & Takabi, D. (2024). Self-supervised learning for electroencephalography. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(2), 1457–1471.

Ramteke, S. M., Chelladurai, H., & Amarnath, M. (2022). Diagnosis and classification of diesel engine components faults using time–frequency and machine learning approach. *Journal of Vibration Engineering & Technologies*, *10*(1), 175–192.

Rao, K. S., & Manjunath, K. E. (2017). *Speech recognition using articulatory and excitation source features*. Springer.

Saleem, M. R., Park, J.-W., Lee, J.-H., Jung, H.-J., & Sarwar, M. Z. (2021). Instant bridge visual inspection using an unmanned aerial vehicle by image capturing and geo-tagging system and deep convolutional neural network. *Structural Health Monitoring*, *20*(4), 1760–1777.

Takahashi, N., Gygli, M., Pfister, B., & Van Gool, L. (2016). Deep convolutional neural networks and data augmentation for acoustic event detection. *arXive preprint arXiv:1604.07160*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *arXive preprint arXiv:1706.03762*.

Virtanen, T., Plumbley, M. D., & Ellis, D. (Eds.) (2018). *Computational analysis of sound scenes and events*. Springer International Publishing.

Wang, Z., Qian, K., Liu, H., Hu, B., Schuller, B. W., & Yamamoto, Y. (2023). Exploring interpretable representations for heart sound abnormality detection. *Biomedical Signal Processing and Control*, *82*, 104569.

Yin, Y., Yu, Q., Hu, B., Zhang, Y., Chen, W., Liu, X., & Ding, X. (2023). A vision monitoring system for multipoint deflection of large–span bridge based on camera networking. *Computer-Aided Civil and Infrastructure Engineering*, *38*(13), 1879–1891.

Yu, F., & Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXive:1511.07122*.

Yuyama, S., Yokoyama, K., Niitani, K., Ohtsu, M., & Uomoto, T. (2007). Detection and evaluation of failures in high-strength tendon of prestressed concrete bridges by acoustic emission. *Construction and Building Materials*, *21*(3), 491–500.

Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhang, L., Han, J., & Shi, Z. (2020). Learning temporal relations from semantic neighbors for acoustic scene classification. *IEEE Signal Processing Letters*, *27*, 950–954.

Zhu, Y.-F., Ren, W.-X., & Wang, Y.-F. (2022). Structural health monitoring on Yangluo Yangtze River Bridge: Implementation and demonstration. *Advances in Structural Engineering*, *25*(7), 1431–1448.

Zhutovsky, S., & Douglas Hooton, R. (2017). Experimental study on physical sulfate salt attack. *Materials and Structures*, *50*(1), 54.

Zoph, B., & Le, Q. V. (2016). Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*.

How to cite this article: Farhadi, S., Corrado, M., & Ventura, G. (2024). Automated acoustic event-based monitoring of prestressing tendons breakage in concrete bridges. *Computer-Aided Civil and Infrastructure Engineering*, 1–21. https://doi.org/10.1111/mice.13321