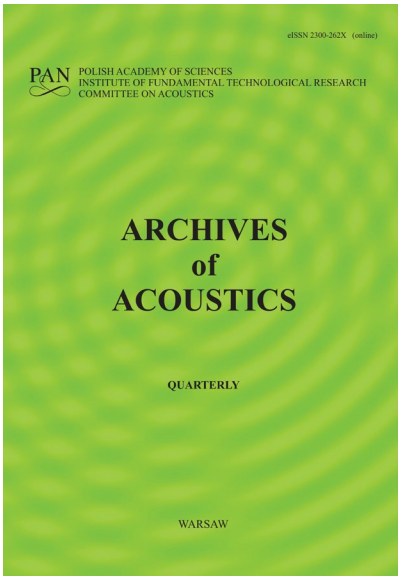
JOURNAL PRE-PROOF

This is an early version of the article, published prior to copyediting, typesetting, and editorial correction. The manuscript has been accepted for publication and is now available online to ensure early dissemination, author visibility, and citation tracking prior to the formal issue publication.

It has not undergone final language verification, formatting, or technical editing by the journal's editorial team. Content is subject to change in the final Version of Record.

To differentiate this version, it is marked as "PRE-PROOF PUBLICATION" and should be cited with the provided DOI. A visible watermark on each page indicates its preliminary status.

The final version will appear in a regular issue of *Archives of Acoustics*, with final metadata, layout, and pagination.



Title: TimeGAN and Coordinated Attention Prototype Network Based Prediction Model for Infrasound Signal

Author(s): Quanbo Lu, Xiaojuan Huang, Rao Li, Mei Li, Dong Zhu

DOI: https://doi.org/10.24423/archacoust.2026.4229

Journal: *Archives of Acoustics*

ISSN: 0137-5075, e-ISSN: 2300-262X

Publication status: In press

Received: 2025-04-08

Revised: 2025-11-02

Accepted: 2025-11-13

Published pre-proof: 2025-01-15

Please cite this article as:

Lu Q., Huang X., Li R., Li M., Zhu D. (2026), TimeGAN and Coordinated Attention Prototype Network Based Prediction Model for Infrasound Signal, *Archives of Acoustics*, https://doi.org/10.24423/archacoust.2026.4229

TimeGAN and Coordinated Attention Prototype Network Based Prediction Model for Infrasound Signal

Quanbo LU[1]*, Xiaojuan HUANG[2], Rao LI[1], Mei LI[3], Dong ZHU[4]

[1] College of Communication Engineering, Chongqing Polytechnic University of Electronic Technology, Chongqing, China

[2] School of Mechanical Engineering, Chongqing Three Gorges University, Chongqing, China

[3] School of Information Engineering, China University of Geosciences, Beijing, China

[4] Sevnce Robotics Co, Ltd, Chongqing, China

*Corresponding Author: luquanbo@cqcet.edu.cn

Due to the complexity of the infrasound environment and the high costs associated with data collection, frequent acquisition of infrasound data is often impractical, resulting in a limited amount of labeled data. To address the challenge of low classification prediction accuracy caused by data scarcity, this paper proposes an infrasound prediction model based on a Time-series Generative Adversarial Network (TimeGAN) and Coordinated Attention Prototype Network (CAPN) (TimeGAN-CAPN). The model begins by introducing TimeGAN, where the generative network is trained using a combination of unsupervised and supervised learning. This approach enables the network to operate within the latent space of temporal features and generate time-series data that closely aligns with the distribution of the original data. These generated samples are then combined with the original data to form an augmented dataset. Subsequently, the augmented data is input into the CAPN, which enhances the sample size per class, allowing for more precise class prototypes and improving the prediction accuracy of the model. Furthermore, the quality and diversity of the data generated by TimeGAN are quantitatively and qualitatively assessed using Maximum Mean Discrepancy (MMD) and t-Distributed Stochastic Neighbor Embedding (t-SNE), facilitating a comparison and verification of the generated data's performance. Experimental results show that TimeGAN-CAPN significantly outperforms the CAPN model in classification tasks with limited infrasound data, achieving a 7.15% increase in accuracy. This demonstrates that the proposed method is highly effective for predicting infrasound-related disasters, particularly in scenarios with small sample sizes.

Keywords: infrasound signal; time-series generative adversarial network; coordinated attention prototype network; maximum mean discrepancy.

1. Introduction

Infrasound (⩽20Hz) refers to sound waves with frequencies below the human hearing range, and is characterized by long propagation distances and strong penetration ability (Sovilla *et al.*, 2025; Lu *et al.*, 2023). Many extreme events, such as earthquakes, tsunamis, and explosions, generate infrasound waves. Globally, infrasound monitoring has been widely applied in the prediction and prevention of natural disasters. Infrasound event detection is the foundation of infrasound monitoring, with its main goal being to extract infrasound events from a large amount of background noise and determine the event's scope (Dong *et al.*, 2024). Event detection is significant for subsequent research such as event classification and localization. Therefore, improving the effectiveness of event detection has become a key issue in the field of infrasound research.

The importance of infrasound event detection algorithms in infrasound monitoring has led to rapid advancements in their technological research. Many scholars have conducted studies on infrasound event detection methods, and new methods continue to be introduced. Baeza *et al*. (2022) explored the potential health impacts of infrasound and advocates for improvements in housing conditions to mitigate these effects. Watson *et al*. (2022) reviewed the advancements in volcano infrasound research and outlines future directions for further investigation and application in volcanic monitoring. Friedrich *et al*. (2023) examined how infrasound affects the perception of low-frequency sounds and its potential influence on human perception and response. Hupe *et al*. (2022) discussed the use of infrasound data products from the International Monitoring System for atmospheric studies and various civilian applications. Macpherson *et al*. (2023) explored the use of local infrasound to estimate seismic velocity and earthquake magnitudes, offering a new approach for seismic monitoring. Listowski *et al*. (2022) investigated the use of infrasound for remotely monitoring Mediterranean hurricanes, highlighting its potential for early detection and tracking. Zajamsek *et al*. (2023) explored how infrasound influences the detectability of amplitude-modulated tonal noise, focusing on its impact on human perception. Wilson *et al*. (2023) presented findings from a long-term microphone array deployment in Oklahoma, analyzing infrasound and low-audible acoustic detections for various environmental and geophysical applications. Yang *et al*. (2025) examined

the correlation between gas desorption processes and infrasound signals, investigating the underlying mechanisms that link the two phenomena. However, the above methods do not consider the prediction of infrasound signals in small sample scenarios.

To address the challenge of low classification prediction accuracy caused by the scarcity of labeled infrasound signal samples, this paper proposes an infrasound prediction model based on TimeGAN-CAPN. The model first expands the temporal infrasound data using TimeGAN, then combines the generated data with the original dataset to train the prediction model, thereby enhancing its performance. Subsequently, drawing on the principles of metric learning, a coordinated attention mechanism is integrated into the traditional prototype network to extract more discriminative feature information, facilitating the accurate construction of metric prototypes for various types of infrasound. Inspired by the biological binocular system, a deep mutual learning framework is introduced to integrate convolutional neural networks with CAPN, further improving the model's prediction accuracy. Experimental results demonstrate that the proposed method outperforms other approaches in classification performance, significantly enhancing disaster early warning rates and advancing the practical application of infrasound detection algorithms.

The structure of this paper is as follows: Sec. 2 provides a brief overview of the basic theories behind TimeGAN, CAPN, and TimeGAN-CAPN, which are used in this study; Sec. 3 presents a performance comparison of different methods through experiments; finally, conclusions are drawn in Sec. 4.

2. Methods

2.1. TimeGAN

Yoon et al. proposed the TimeGAN by combining the flexibility of unsupervised learning with the strong control of the training process in supervised learning (Yoon *et al.*, 2019). Its training process is essentially a process of solving the min-max problem of a binary function. The model consists of two networks: the reconstruction network and the embedding network; and two generative models: the discriminator and the generator. It uses three different loss functions: the generation loss function, the supervised loss function, and the unsupervised loss function to train the network.

The Time-GAN model uses gradient descent for parameter optimization, with the

generator typically taking random noise and vectors as input. The loss function is expressed as follows (Sharma *et al.*, 2024):

$$L_G(Z) = E_{Z\sim P_Z(Z)}\left[\log\left(1 - D\left(G(Z)\right)\right)\right], \tag{1}$$

where $L_G(\cdot)$ is the generator's loss function, $E(\cdot)$ is the embedding network's expected loss, $G(\cdot)$ is the generator function, $D(\cdot)$ is the discriminator function, $P_Z(\cdot)$ is the noise data distribution, Z is the random variable for noise input.

The input variables for the discriminator are synthetic data and real data to be distinguished, and the loss function is expressed as (Vuletic *et al.*, 2024) :

$$L_D(x) = E_{X\sim P_i(x)}[\log D\,(x)] + E_{X\sim P_G(x)}\left[\log\left(1 - D(x)\right)\right] \tag{2}$$

where $L_D(\cdot)$ is the real data variable, i is the fake data variable, $P_i(\cdot)$ is the real data distribution, x is the input random variable.

2.2. CAPN

The CAPN model is shown in Fig. 1. It consists of two parallel views: in the global view, a convolutional neural network (CNN) is used to capture inter-class relationships, while in the local view, a prototype network with a coordinated attention mechanism focuses more on matching details (Jiang *et al.*, 2025). The two views are then aggregated through a deep mutual learning framework, implicitly exploring useful knowledge from each other. The training process aims to find the best hyperparameter settings and leverage prior knowledge to better train specific test tasks. Finally, during the testing process, the collaborative features from both views are combined to perform classification tasks, thereby improving the accuracy of few-shot classification prediction. The model is mainly divided into three parts: the global view, the local view, and cross-view mutual learning.

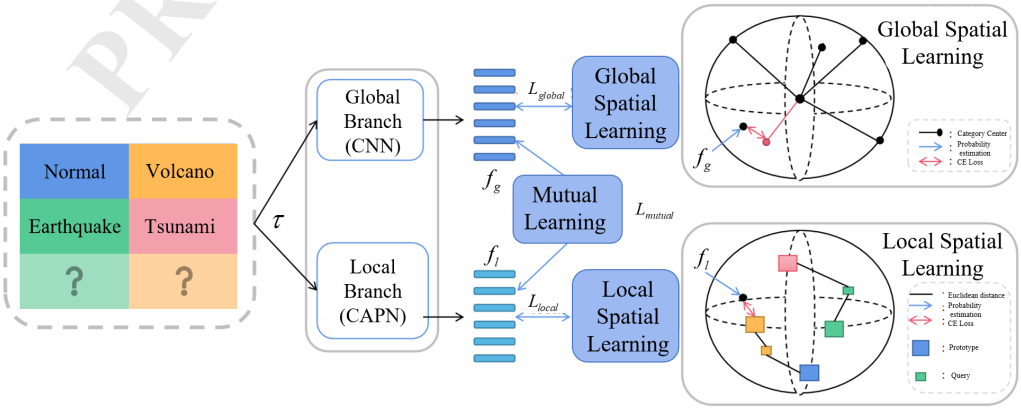


Fig. 1. Structure of CAPN.

2.2.1. The global view

In the global view, a one-dimensional convolutional network is used for training. Specifically, for a given task Γ, a global learner $A_{\emptyset}^{G}$ is trained to map each data sample x_i in the Γ set to a high-dimensional space. The probability distribution of x_i is expressed as follows (Zhang *et al.*, 2024):

$$P(y_i = y|x_i) = \sigma\left(A_{\emptyset}^{G}(x_i)\right), \tag{3}$$

where σ is the softmax activation function. It serves to combine the extracted features for nonlinear activation, outputting the probability distribution of each class, which is then used for classification.

The loss function is calculated using cross-entropy, i.e., the negative logarithm of the probability $P(y_i = y|x_i)$. Therefore, the loss function for the global view is as follows (Tang *et al.*, 2023):

$$L_{\text{global}} = E_{(x_i,y_i)\in T} \cdot \sum_{i=1}^{N} y_i \log P\,(y_i = y|x_i). \tag{4}$$

2.2.2. The local view

In the local view, a prototype network is used to match each query sample with the class prototype from the support set in the embedding space. Therefore, in the local view, a prototype network with a coordinated attention mechanism is applied.

This model consists of three parts: feature embedding, prototype generation, and feature distance-based classification. The structure is shown in Fig. 2. The first step is to use a feature embedding module with an attention mechanism for feature embedding. Support set and query set samples are passed through the convolutional layers. By adding the coordinated attention mechanism, both spatial and channel information are extracted, and by embedding position information in channel attention, accurate position details and long-range dependencies are captured. This allows the feature embedding to focus more on useful local feature information, enhancing the feature representation ability of the feature embedding network. The second step is to compute the class prototype features by averaging the feature maps of samples from the same class. The mean feature serves as the class prototype feature. The third step is to measure the distance between the category prototype features learned by the feature embedding network

and the query sample features using a selected distance metric, such as Euclidean distance. According to the principle that similar samples are close and dissimilar samples are far apart, the closest prototype to the query set output is selected as the predicted result, and the network is trained until it meets the required model and label prototypes. Classification prediction is then performed using the saved optimal coordinated attention prototype network.

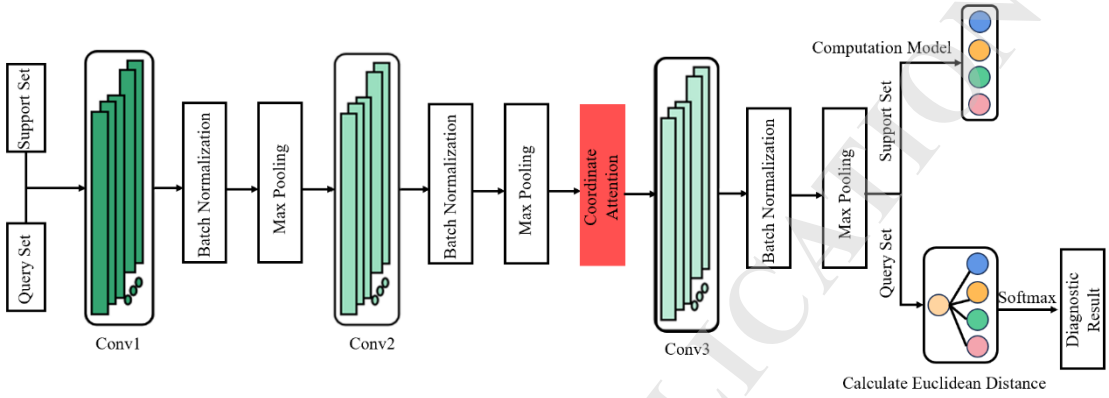


Fig. 2. Structure of CAPN for the classification prediction.

2.2.3. The cross-view mutual learning

In addition to learning within each individual view, the global and local views also mutually promote each other through cross-view interaction in the deep mutual learning network. Specifically, for each view, in addition to completing its own training task, the view also minimizes the imitation loss from the other view. This imitation loss uses the Kullback-Leibler divergence to quantify the match between the prediction probabilities of the two networks, which aids in implicit knowledge transfer. The mutual loss is shown in Eq. (5), which includes two sub-items (Ji *et al.*, 2020):

$$L_{\text{mutual}} = D_{KL}(F_1 \| F_g) + D_{KL}((F_G \| F_1)) \tag{5}$$

$$D_{KL}(F_1 \| F_g) = F_1 \, log \frac{F_1}{F_g} \tag{6}$$

$$D_{KL}(F_g \| F_1) = F_g \, log \frac{F_g}{F_1} \tag{7}$$

where $F()$ represents the feature distribution computed by $\sigma(A_\emptyset(x))$, and the interaction problem is considered from the perspective of feature distribution consistency. The learning in Euclidean space focuses on relative relationships rather than hard constraints like mean square error. This is because overly strong supervision signals are not conducive to preserving the specificity of both views.

Thus, the final loss function of the model is given by (Ruddick *et al.*, 2024):

$$L_{\text{total}} = \alpha L_{\text{global}} + \beta \hat{L}_{11} + \gamma L_{\text{mutual}}, \tag{8}$$

where α, β, and γ are the weighting factors. The optimal loss weights α, β, and γ are determined through a systematic hyperparameter tuning process using grid search combined with cross-validation. This paper explores various combinations of α, β, and γ within a pre-defined range, informed by prior work on similar models and the characteristics of infrasound data. The model's performance is evaluated based on classification accuracy and loss using a validation set, with the aim of balancing the three loss terms – reconstruction loss, supervised loss, and unsupervised loss – while avoiding overfitting. After multiple iterations, the values that resulted in the highest overall performance are selected, ensuring the model effectively captured both temporal and discriminative features of the infrasound signals.

2.3. The proposed approach

The TimeGAN-CAPN infrasound prediction model proposed in this paper is illustrated in Fig. 3. It consists of three main components: data preprocessing, data generation, and infrasound prediction. In the data preprocessing phase, missing values in the sensor-collected data are imputed using the nearest neighbor interpolation method. The data is then normalized using min-max normalization, ensuring consistent dimensions and complete features, which enhances its usability. In the data generation phase, a TimeGAN model is constructed, and the collected data samples are fed into the generative model. Through an adversarial process between the generator and discriminator in the latent space, the loss function is computed to update the model parameters, ultimately generating high-quality infrasound data samples. These generated samples are then combined with the original data to form an augmented dataset. In the infrasound prediction phase, the synthesized dataset is split into a training set and a test set. The training set is used to train the CAPN-based infrasound prediction model. The final model is then applied to infrasound prediction tasks for early disaster detection.

2.3.1. Data preprocessing

The collected infrasound dataset contains valuable infrasound characteristics but is presented in various forms, lacking uniformity, which makes it unsuitable for direct use in machine learning models. Consequently, data preprocessing is essential to extract useful

parameters and convert the infrasound data into a standardized format that can be effectively recognized by learning algorithms. Initially, missing values are imputed to ensure the completeness of the dataset. Following this, the input vectors are normalized to standardize the units, thereby preventing issues such as disproportionately large feature weights that could lead to increased model training time or gradient explosion problems.

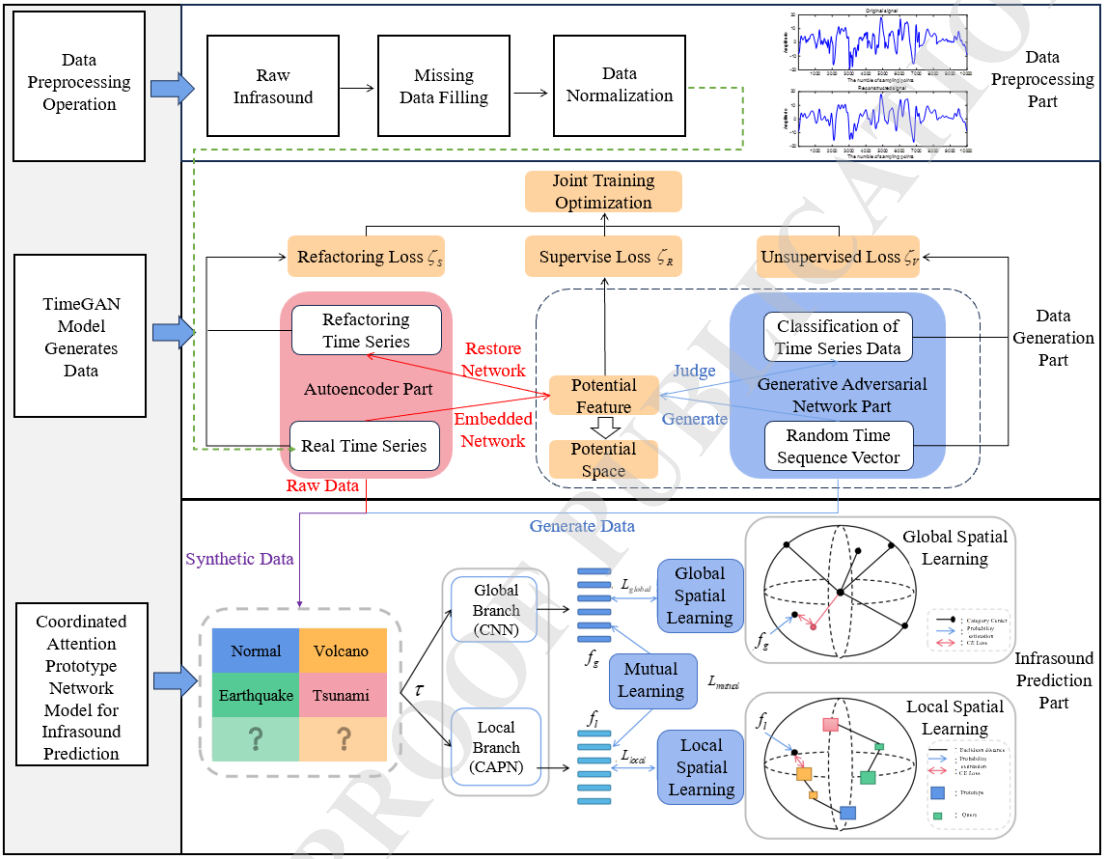


Fig. 3. Framework of the proposed approach.

In the experiments, the nearest neighbor interpolation method is used to fill missing data. The nearest neighbor interpolation method uses the previous and next values of the missing data. Let the value at time t_1 be x_1, at time t_2 be x_2, and at time t_3 be x_3. The missing value x_2 can be expressed as follows (Sehar *et al.*, 2025):

$$\frac{x_2-x_1}{x_3-x_1}=\frac{y_2-y_1}{y_3-y_1} \tag{9}$$

For normalization, the min-max normalization method is used to map the results to the interval $[0, 1]$, as shown in Eq. (10). The original data x is normalized to x^*, where $x_{\min}$ and $x_{\max}$ are the minimum and maximum values in the original data, respectively (Mitropoulos *et al.*, 2022):

$$x^* = \frac{x - x_{\min}}{x\min_{\max}}. \tag{10}$$

2.3.2. Data generation

In the data generation phase, the TimeGAN model is composed of four primary components: the embedding network, the recovery network, the sequence generator, and the sequence discriminator. The embedding and recovery networks fall under the autoencoder category, while the sequence generator and discriminator are part of the generative adversarial network framework. As a result, TimeGAN involves joint training of both the autoencoding and adversarial components. In the autoencoding section, the embedding network maps high-dimensional data into a lower-dimensional vector, or latent space, to capture essential feature information. The recovery network then reconstructs the data from this latent space back to its original dimensionality, minimizing the reconstruction loss L_R to optimize the representation of the latent space. Following the principle that the dynamics of complex systems are often driven by a smaller set of lower-dimensional factors, the adversarial component trains the sequence generator and discriminator within the latent space produced by the embedding network. This approach alleviates the challenges associated with high-dimensional data during the adversarial training process.

The embedding and recovery functions achieve the mapping from feature space to latent space, enabling the adversarial network to learn the potential time characteristics of the data through low-dimensional representations. Let H_S represent the latent vector space containing time-related feature S, and similarly, let H_X represent the latent vector space for static feature X. The role of the embedding function e is to encode real-time sequences into the latent space, defined as $S \times \prod_t X \to H_S \times \prod_t H_X$. This function uses a recurrent neural network (RNN) to perform the mapping, encoding both static and temporal features into low-dimensional latent vectors $h_S, h_{1:T} = e(S, X_{1:T})$ that are easier for the network to learn. The embedding function is expressed as (Ruddick *et al.*, 2024):

$$\begin{cases} h_S = e_S(S) \\ h_t = e_X(h_S, h_{t-1}, X_t)' \end{cases} \tag{11}$$

where $e_S: S \to H_S$ is the embedding function for static features, aimed at converting static features S into low-dimensional static features h_S through mapping, and $e_X: H_S \times H_X \times X \to H_X$ is the RNN-based embedding function for temporal features, aiming to map temporal features X_t into low-dimensional static features h_t. It follows causal ordering, meaning each

step's output depends only on the preceding information.

The recovery function γ performs decoding, defined as $H_S \times \prod_t H_X \to S \times \prod_t X$. It uses a feedforward neural network (FNN) to restore the low-dimensional latent code back into high-dimensional static and temporal features $\tilde{S}, \tilde{X}_{1:T} = \gamma(h_S, h_{1:T})$ (Ruddick *et al.*, 2024):

$$\begin{cases} \tilde{S} = \gamma_S(h_S) \\ \tilde{X}_t = \gamma_X(h_t) \end{cases}, \tag{12}$$

where $\gamma_S: H_S \to S$is the recovery function for static features, which is the inverse mapping of h_S, and similarly, $\gamma_X: H_X \to X$represents the recovery network for temporal feature embeddings, which is the inverse mapping of h_t.

In the autoencoding part, the embedding function maps high-dimensional static and temporal features into a low-dimensional latent space, and the recovery function maps them back to high-dimensional features. Therefore, the embedding function and recovery function are reversible mappings existing between feature space and latent space. They can accurately represent the high-dimensional reconstructed data $\tilde{S}, \tilde{X}_{1:T}$ using high-dimensional original data $S, X_{1:T}$ and low-dimensional latent vectors $h_S, h_{1:T}$. The reconstruction loss L_R of the autoencoder part is shown in Eq. (13), which represents the autoencoder's understanding of the intrinsic patterns in the input data (JI *et al.*, 2020). By optimizing the reconstruction, the autoencoder can generate higher-quality low-dimensional latent representations:

$$L_R = E_{S,X_{1:T\sim P}}\left[\left\|S - \tilde{S}\right\|_2 + \sum_t \left\|X_t - \tilde{X}_t\right\|_2\right]. \tag{13}$$

During TimeGAN's training, two types of data are input into the sequence generator. In the open-loop mode, the low-dimensional data $\tilde{h}_S, \tilde{h}_{1:T}$ generated by the generator is input into the sequence generator to obtain the next generated vector $\hat{h}_t$. Then, by optimizing the unsupervised loss L_U, the probability of correctly classifying the real data $h_S, h_{1:T}$ and generated data $\tilde{h}_S, \tilde{h}_{1:T}$ is increased (Ji *et al.*, 2020):

$$L_U = E_{S,X_{1:T\sim P}}[\log y_S + \sum_t \log y_t] + E_{S,X_{1:T\sim P}}\left[\log(1 - \tilde{y}_S) + \sum_t \widetilde{\left(1 - \log y_t\right)}\right]. \tag{14}$$

Due to insufficient adversarial feedback from the sequence discriminator, the sequence generator does not fully capture the conditional distribution of the time steps in the real data. Therefore, TimeGAN introduces supervised loss to further constrain the model and alternates training in the closed-loop mode. The low-dimensional temporal latent sequence $h_{1:t-1}$ encoded by the embedding network is input into the sequence generator to obtain the latent vector for the next time step. Then, the supervised loss is optimized using the maximum likelihood method, which reflects the similarity between the data generated by the sequence generator and the data encoded by the autoencoder. This loss measures the difference between

distributions $p(H_t|H_S, H_{1,t-1})$ and $\hat{p}(H_t|H_S, H_{1,t-1})$. The supervised loss L_S obtained using the maximum likelihood method (Tang *et al.*, 2023):

$$L_S = E_{S,X_{1:T\sim P}}[\sum_t \|h_t - g_X(h_S, h_{t-1}, z_t)\|_2]. \quad (15)$$

At each training step, the difference between the next latent vector from the embedding function and the next latent vector synthesized by the sequence generator needs to be evaluated. Although the unsupervised loss L_U can guide the sequence generator to create real sequences, the supervised loss L_S ensures that it generates smooth transitions.

2.3.3. Infrasound prediction

The specific process of the coordinated attention mechanism is shown in Fig. 4, which includes two steps: information embedding and attention generation.

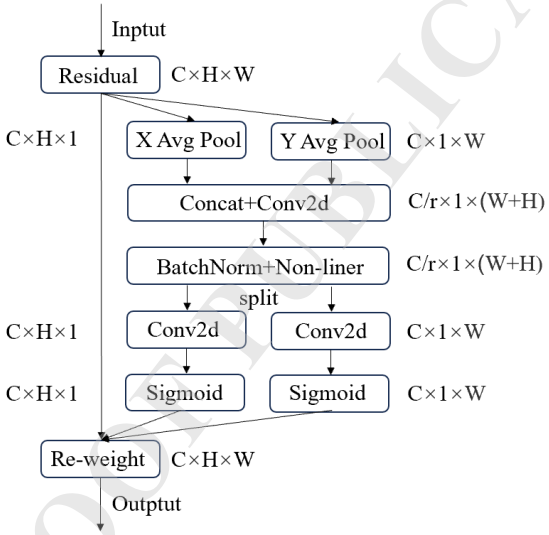


Fig. 4. Schematic of the coordinated attention generation process

The information embedding component plays a crucial role in enhancing the attention module's ability to capture a higher-quality global receptive field while preserving the accuracy of positional encoding. In traditional channel attention, global pooling is commonly used to encode spatial information. However, this approach often compresses global spatial data into channels, making it challenging to retain precise positional information. To address this issue, the information embedding operation decomposes the global average pooling step by pooling separately along both the horizontal and vertical axes of the input features. This technique aggregates features from both spatial directions, resulting in two feature maps that retain directional information. As shown in Fig. 5, by performing transformations along both directions, long-range dependencies along one spatial axis and positional information along the other are captured by the attention module, enabling the network to more effectively localize

key targets.

Fig. 5. Schematic of the coordinated attention information embedding operation.

Specifically, in the horizontal direction, the global average pooling operation uses a pooling kernel of size $H \times 1$ to compress the input feature X dimensions from $H \times W \times C$ to $H \times 1 \times C$ (Tang *et al.*, 2023):

$$Z_c^h(h) = \frac{1}{W}\sum_{0 \le a \le W} x_C(h, a), z_C^h \in R^{C*H*1}. \tag{16}$$

In the vertical direction, the global average pooling operation uses a pooling kernel of $1 \times W$ size to compress the input feature dimensions from $H \times W \times C$ to $H \times 1 \times C$ (Zhang *et al.*, 2024):

$$Z_c^w(w) = \frac{1}{H}\sum_{0 \le b \le H} x_C(b, w), z_C^w \in R^{C*1*W}. \tag{17}$$

The attention generation operation aims to fully utilize the positional information encoded in the embedding operation and capture the regions of interest and relationships between channels. Specifically, the feature maps from the two directions, Z_C^h and Z_C^w, are concatenated along the channel dimension, and then convolution operations are applied using a shared convolutional transformation function F_1, obtaining intermediate feature maps f, $f \in R^{\frac{C}{r}\times(H+W)}$, which encode both horizontal and vertical directions (Zhang *et al.*, 2024):

$$f = \delta\left(F_1\left(\left[z^h, z^w\right]\right)\right), f \in R^{\frac{C}{r}*1*(H+W)}, \tag{18}$$

where δ is the non-linear activation function Relu，$[\cdot\ ,\ \cdot]$ represents the concatenation along the spatial dimension.

Then, the intermediate feature map f is split along the spatial dimensions into two feature maps f^h and f^w, $f^h \in R^{\frac{c}{r}\times H}, f^w \in R^{\frac{c}{r}\times W}$. Each feature map is upsampled using convolution operations F_h and F_w, obtaining two directional attention weights g^h and g^w, $g^h \in R^{C\times H\times 1}, g^w \in R^{C\times 1\times W}$, as follows (Jiang *et al.*, 2025):

$$g^h = \sigma\left(F_h\left(f^h\right)\right), \tag{19}$$

$$g^w = \sigma\left(F_w(f^w)\right). \tag{20}$$

Finally, the attention weights g^h and g^w are multiplied with the original features x_C to obtain the scaled features y_C (Jiang *et al.*, 2025) :

$$y_C(a,b) = x_C(a,b) \times g_C^h(a) \times g_C^w(b). \tag{21}$$

2.4. Data det

This study utilizes infrasound data provided by the international monitoring system (IMS) with support from the Comprehensive Nuclear-Test-Ban Treaty Beijing National Data Center. A total of 611 infrasound data sets are collected from six distinct infrasound sensor arrays located globally. These data sets are categorized into three types of infrasound events: Earthquake, Tsunami, and Volcano. All infrasound recordings have a sampling frequency of 20 Hz. Table 1 presents the details of the infrasound data collected from various regions, while Fig. 6 illustrates the geographical distribution of the infrasound stations.

Table 1. Information of infrasound data.

Event type	Data source (IMS Station Code)	Geographic coordinate	Number of signals	Total	Sampling frequency [Hz]
Earthquake	I14CL	(–33.65, –78.79)	74	203	20
	I30JP	(35.31,140.31)	124		20
	I59US	(19.59, –155.89)	6		20
Tsunami	I10CA	(50.20, –96.03)	4	218	20
	I22FR	(–22.18,166.85)	53		20
	I30JP	(35.31,140.31)	113		20
	I52GB	(–7.38,72.48)	66		20
Volcano	I30JP	(35.31,140.31)	189	189	20

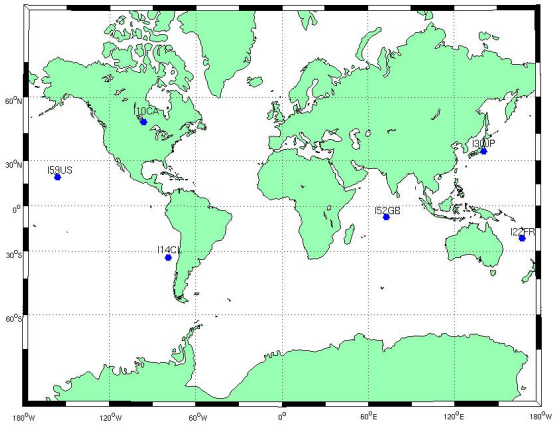


Fig. 6. Map of the infrasound station.

3. Experiments

3.1 Experiments setup

The operating system used in this study is Windows 11, with CUDA 10.0 and cuDNN 7.4 for accelerated training. The hardware used includes an NVIDIA Quadro P4000 (8 GB memory). The network development framework is TensorFlow 1.14, and programming is done in Python. The CPU used is an Intel(R) Core(TM) i5-11320H CPU @ 3.20 GHz, 2.5 GHz. As described in Subsec. 2.2, the key parameters of the CAPN are summarized in Table 2. The simulation validation focuses on applying the infrasound signal data to assess the feature learning performance of the proposed CAPN model. Each infrasound signal consists of 10 400 data points. The dataset is divided into training and testing samples. The input map size for the CAPN model is 128 × 128 × 1. The number of iterations is set to 60.

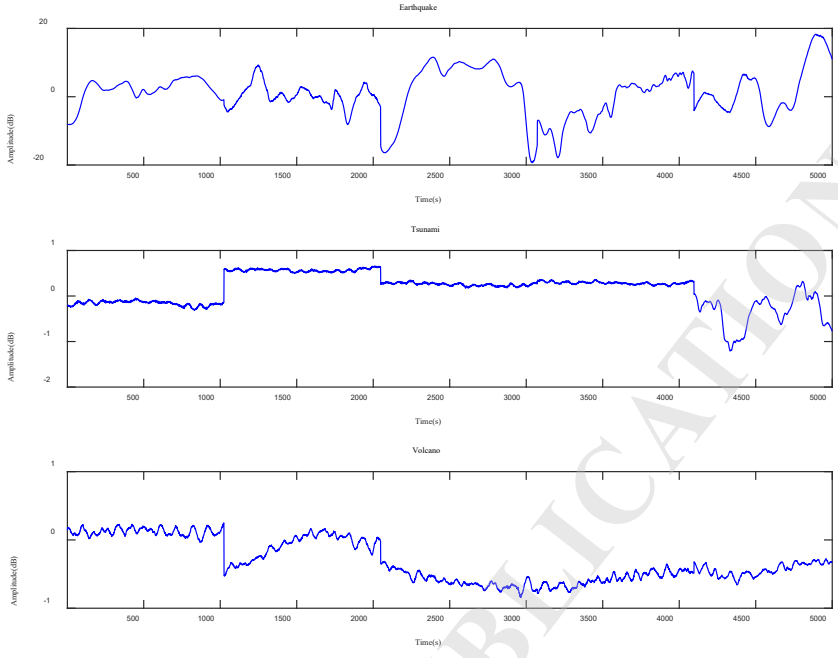
Table 2. Parameter of CAPN.

Number of layer	Layer type	Kernel size	Filters
1	Convolution 1	12 × 12	4
2	Maxpooling 1	5 × 5	–
3	Convolution 2	7 × 7	4
4	Maxpooling 2	5 × 5	–
5	Convolution 3	5 × 5	8
6	Maxpooling 3	5 × 5	–
7	Flatten	–	–
8	Fully-connected	–	–
9	Softmax	–	–

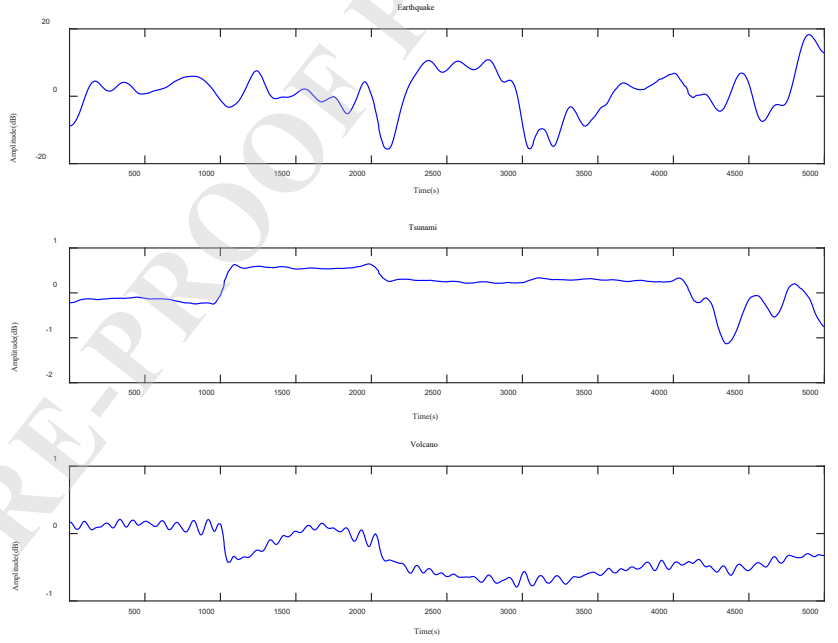
3.2 Data preprocessing

The infrasound data collected in this study are smoothed to effectively eliminate noise. Fig. 7a displays the original infrasound signal, which contains substantial noise. To reduce computational complexity, a moving average filtering method is applied for smoothing, with the resulting signal shown in Fig. 7b. Details of the moving average filtering method can be found in (Mitropoulos *et al*., 2022). A total of 70 % of the smoothed data are used as the training

set, while the remaining 30 % are allocated as the testing set. Finally, data standardization and normalization are performed using Eqs. (9) and (10).



a)



b)

Fig. 7. Original (a) and preprocessing (b) signal.

3.3 Data generation

To evaluate the quality of the data generated by the model, both fidelity and diversity are

taken into account. Fidelity refers to the degree to which the generated samples resemble the real data, while diversity ensures that the generated samples do not exhibit excessive similarity to each other. Thus, the performance of the generated model is assessed from both qualitative and quantitative perspectives.

3.3.1. Discriminator score

The performance of the generative model is quantitatively assessed from the quality and diversity of the generated samples. In this study, the maximum mean discrepancy (MMD) metric is used to evaluate the generative model based on the difference in sample distributions. MMD is used to measure the distance between two distributions in Hilbert space. Thus, for the generative model, this metric can measure the distance between the original data distribution P_o and the generated data distribution P_g. The smaller the MMD distance, the more similar the distributions of the original and generated data are, indicating higher quality of the generated samples and better model performance.

When calculating the MMD distance, the Gaussian kernel function $K(x, y)$ is used to map the two samples into a real number (Wang *et al.*, 2021):

$$K(x, y) = \exp(-\|x - y\|^2). \tag{22}$$

The MMD distance $D_{\mathrm{MMD}}(P_o, P_g)$ is expressed as Eq. (23) (Wang *et al.*, 2021) :

$$D_{\mathrm{MMD}}(P_o, P_g) = E_{x,x'\sim P_o}[K(x, x')] - 2E_{x\sim P_o, y\sim P_g}[K(x, y)] + E_{y,y'\sim P_g}[K(y, y')]. \tag{23}$$

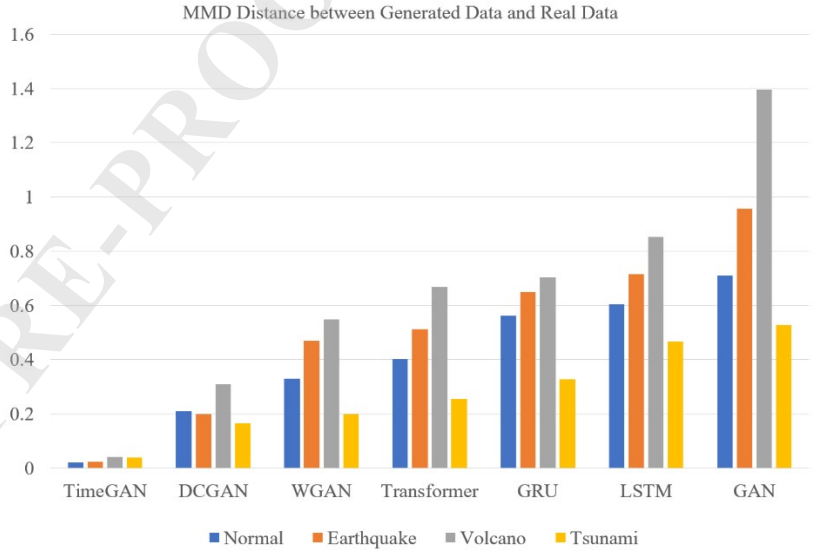


Fig. 8. MMD distances between generated data from different models and real data.

In this experiment, both the original data and generated data have four state types: normal, earthquake, volcano, and tsunami. Therefore, the distribution of the original data is denoted as

$P_{oi,}i = (1,2,3,4)$, and the distribution of the generated data is denoted as $P_{gi,}i = (1,2,3,4)$. As shown in Fig. 8, the MMD distance between the original and generated data is calculated for infrasound data after applying several models, including GAN, LSTM, GRU, Transformer, WGAN, DCGAN, and TimeGAN. Compared to the other generative adversarial network models, TimeGAN exhibits the smallest MMD distance between the original and generated data. Notably, for the volcano data, the MMD distance is 1.364 times smaller for TimeGAN than for GAN, indicating that the distribution of TimeGAN-generated data closely matches the original data distribution, resulting in superior model performance. In contrast, the GAN model shows the largest discrepancy between the generated and original data, making it the least effective model. For the tsunami data, all four generative models show relatively high performance due to the distinct infrasound characteristics. However, for volcano data, where infrasound features are less pronounced, all models exhibit the largest MMD distance, suggesting a greater challenge in accurately modeling such data. Therefore, it can be concluded that TimeGAN generates relatively high-quality samples, outperforming the other models in terms of data fidelity.

In addition, the MMD metric is also used to evaluate the diversity of the generated samples, albeit with a slightly different focus. Here, the goal is to measure the variability between the sample distributions within the generated data. Specifically, the MMD distance between the distributions of individual samples is calculated, and the mean of these distances is taken as the internal MMD distance of the generated data. A higher value of this distance indicates greater variability between the samples, reflecting higher diversity in the generated data and superior performance of the GAN.

Let the number of distribution samples be 1, and let P_i and P_j represent the source distributions of two different samples. The MMD distance is given by (Wang *et al.*, 2021):

$$D_{\mathrm{MMD}}(P_i, P_j) = 2 - 2E_{x_i \sim P_i, x_j \sim P_j}\left[K(x_i, x_j)\right], i \neq j. \tag{24}$$

Then, the internal MMD distance of the generated data, which measures the diversity within the samples, is given by (WANG *et al.*, 2021):

$$D_r = \frac{1}{1+2+\cdots+N-1}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} D_{MMD}\left(P_i, P_j\right). \tag{25}$$

In the experiment, the internal MMD distances of the generated data from three classical generative adversarial networks and the proposed TimeGAN model are calculated, with the results presented in Fig. 9. The analysis reveals that, among the seven generative models, TimeGAN produces data with the largest internal MMD distance, indicating that it generates

data with higher diversity. This is particularly evident in the tsunami data. In contrast, GAN and WGAN perform the worst in generating diverse samples. For example, in the tsunami case, the internal MMD distance of the data generated by GAN is 0.0024 smaller than that of TimeGAN, WGAN is 0.002 smaller, and LSTM is 0.0018 smaller than TimeGAN. These results demonstrate that TimeGAN outperforms other models in terms of generating diverse and varied infrasound samples.

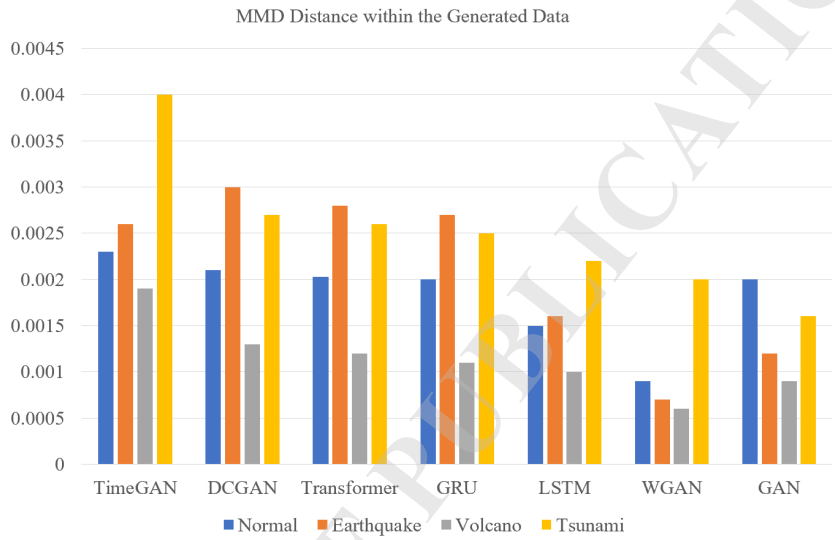


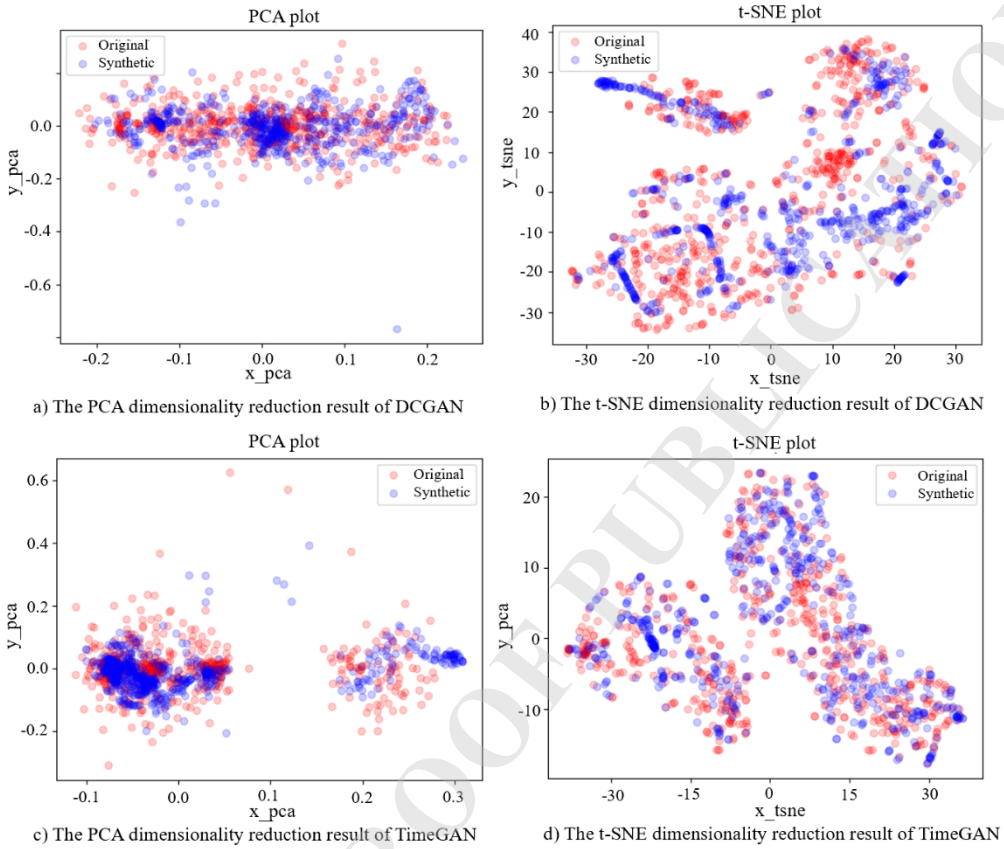
Fig. 9. Internal MMD distances of generated data from different models.

A comprehensive analysis of the MMD metric reveals that, in most cases, GAN, WGAN, and LSTM models only offer rough approximations of the original data, with limited quality and diversity in the generated samples. In contrast, TimeGAN, DCGAN, Transformer, and GRU models generate data with higher quality and greater diversity. To further validate the effectiveness of TimeGAN in infrasound prediction, a detailed performance comparison is conducted between these models.

3.3.2. Visualization

In the previous MMD analysis, TimeGAN and DCGAN demonstrated superior performance, and thus, these two models are the focus of further analysis. To qualitatively evaluate the effectiveness of the proposed method, the t-SNE and PCA techniques are applied to visualize the distribution of the generated and original samples in a two-dimensional space. Figure 10 presents the results of the PCA and t-SNE visualizations, where red points represent

the feature distribution of the real infrasound data, and blue points represent the feature distribution of the generated infrasound data. The closer the two sets of points are to each other, the better the model's performance, indicating that the distribution of the generated samples closely matches that of the real data.



a) The PCA dimensionality reduction result of DCGAN

b) The t-SNE dimensionality reduction result of DCGAN

c) The PCA dimensionality reduction result of TimeGAN

d) The t-SNE dimensionality reduction result of TimeGAN

Fig. 10. WGAN and TimeGAN data visualization.

From the analysis, the feature distribution of the data generated by TimeGAN closely aligns with the feature profile of the original data, demonstrating a high degree of similarity. In contrast, DCGAN fails to generate certain features that are present in the original data, resulting in a mismatch between the feature distribution of the generated and original data. When augmenting time-dependent data, TimeGAN significantly outperforms DCGAN, showcasing its superior performance in capturing temporal dynamics.

3.4 Infrasound prediction

Considering that the model is designed for infrasound prediction tasks with small sample sizes, the generated samples must be as effective as the real data samples. These samples should

serve to augment the dataset, ensuring that each sample contains sufficient information to improve the performance of the infrasound prediction model. The generated data is combined with the original data, with the amount of synthetic data being twice the size of the original dataset. Additionally, three consecutive data points are grouped together to form a single sample. The synthesized dataset is then used to train the diagnostic model, which is subsequently tested on a separate test set. The infrasound prediction accuracy after training is used as an evaluation metric to assess whether the inclusion of the generated data enhances the model's predictive capability, particularly in small sample scenarios.

The results shown in Fig. 11 represent the accuracy after 60 iterations of the model. From the figure, it can be observed that the data generated by the classical GAN-CAPN and LSTM-CAPN model reduces the accuracy by 5.76 % and 3.64 % compared to CAPN, which decreases the performance of the infrasound prediction model. Apart from the GAN-CAPN and LSTM-CAPN model, datasets that were not augmented with a generative model perform poorly in infrasound prediction after training. Compared to the DCGAN-CAPN, WGAN-CAPN, Transformer-CAPN, and GRU-CAPN models, the TimeGAN-CAPN model generates higher-quality data by considering the internal temporal correlations in the data, effectively addressing the issue of insufficient information in the samples. The prediction accuracy improved by 5.62 % compared to when no augmentation was performed. Therefore, using the TimeGAN model to augment the infrasound data and inputting the augmented data into the CAPN infrasound prediction model can significantly improve the infrasound prediction accuracy.

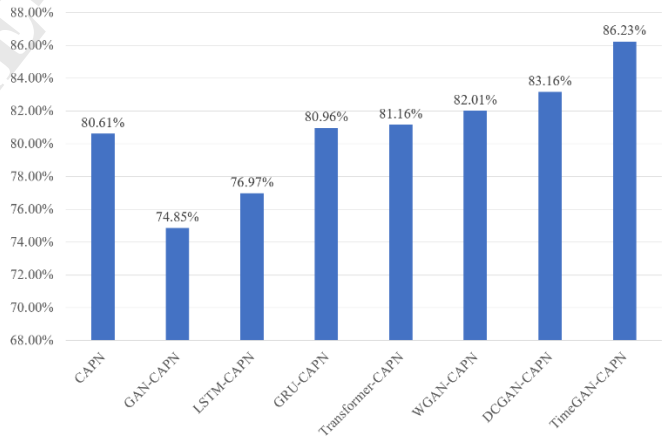


Fig. 11. Comparison of diagnostic accuracy after sample augmentation.

Table 3 shows a comparison of the classification performance of eight methods for infrasound signals. The experimental results indicate that TimeGAN-CAPN achieves the best overall classification Precision, reaching 84.62 %. In addition, TimeGAN-CAPN also significantly outperforms the other seven classification methods in terms of F1-score and Recall, with values of 86.02 % and 87.26 %, respectively.

Table 3. Comparison of classification results for four types of infrasound events by different classification networks [%].

Method	*F*1-score	Recall	Precision
CAPN	80.07	79.62	79.16
GAN-CAPN	74.39	73.68	73.07
LSTM-CAPN	76.27	75.89	75.25
GRU-CAPN	80.29	79.68	78.97
Transformer-CAPN	80.76	80.92	79.13
WGAN-CAPN	81.96	82.17	80.86
DCGAN-CAPN	83.11	84.06	81.71
TimeGAN-CAPN	86.02	87.26	84.62

Further analysis shows that, compared to other classification networks, TimeGAN-CAPN exhibits higher classification accuracy and more stable classification performance in infrasound classification. To provide a comprehensive evaluation of its performance, Fig. 12 presents the ROC curve for different methods. From the figure, it is evident that the AUC value of TimeGAN-CAPN reaches 0.8451, significantly higher than the other seven networks, further validating its superior performance in the infrasound signal classification task.

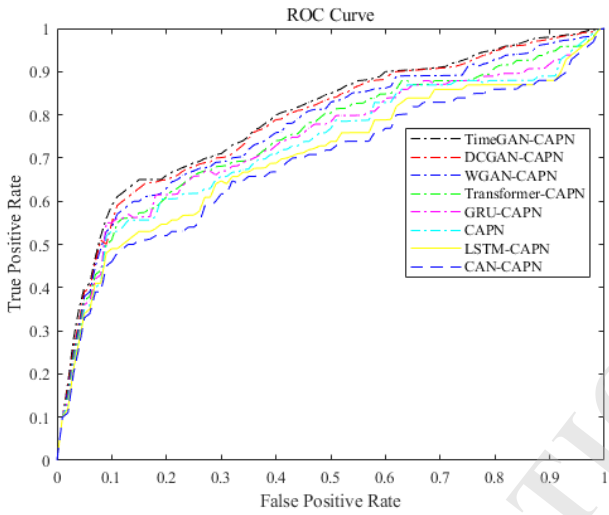
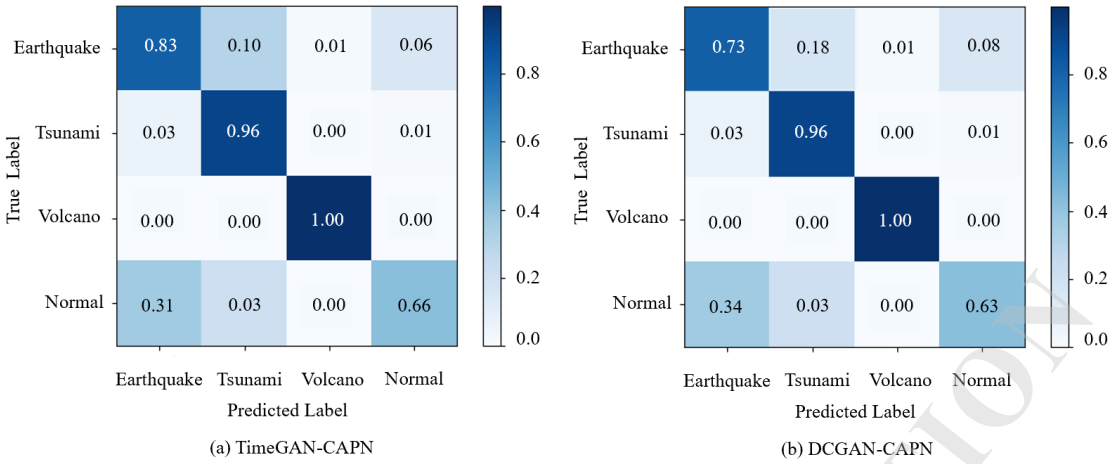


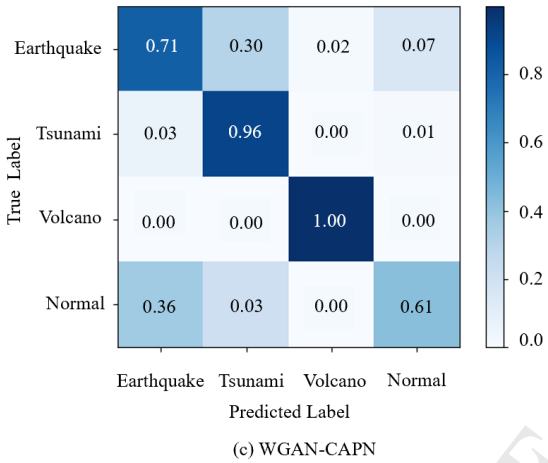
Fig. 12. ROC curve of different classification methods.

To analyze the infrasound recognition performance, eight classification models were evaluated using 6-fold cross-validation to obtain the accuracy of real labels and predicted labels from six validation runs. The confusion matrix for infrasound classification is shown in Fig. 13. From the perspective of single-class classification performance, TimeGAN-CAPN demonstrates significant advantages in classifying earthquake, tsunami, and volcanic infrasound signals. This result thoroughly confirms the robustness and generalization ability of the proposed method in handling different types of infrasound signals.

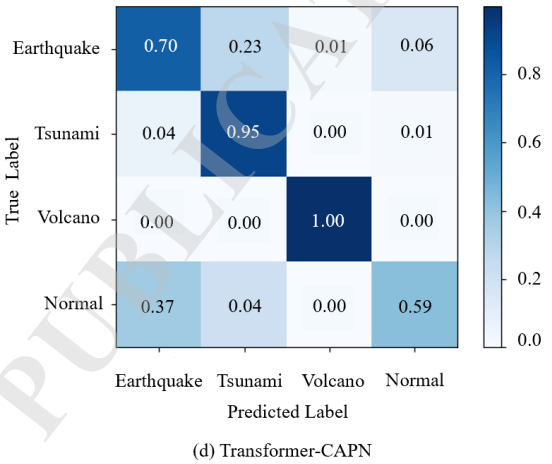


(a) TimeGAN-CAPN

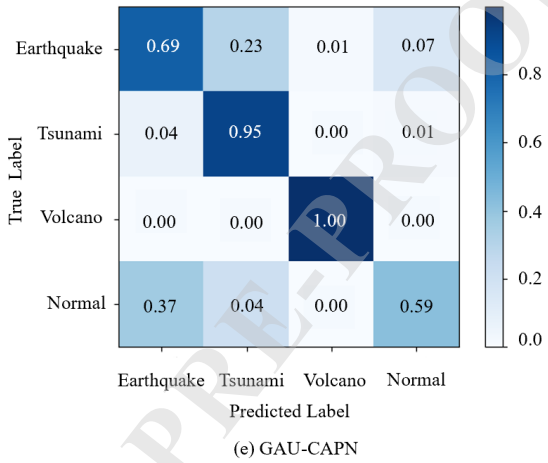
(b) DCGAN-CAPN



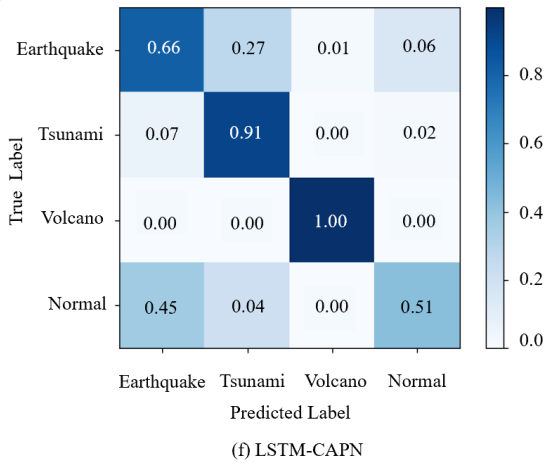
(c) WGAN-CAPN



(d) Transformer-CAPN



(e) GAU-CAPN



(f) LSTM-CAPN

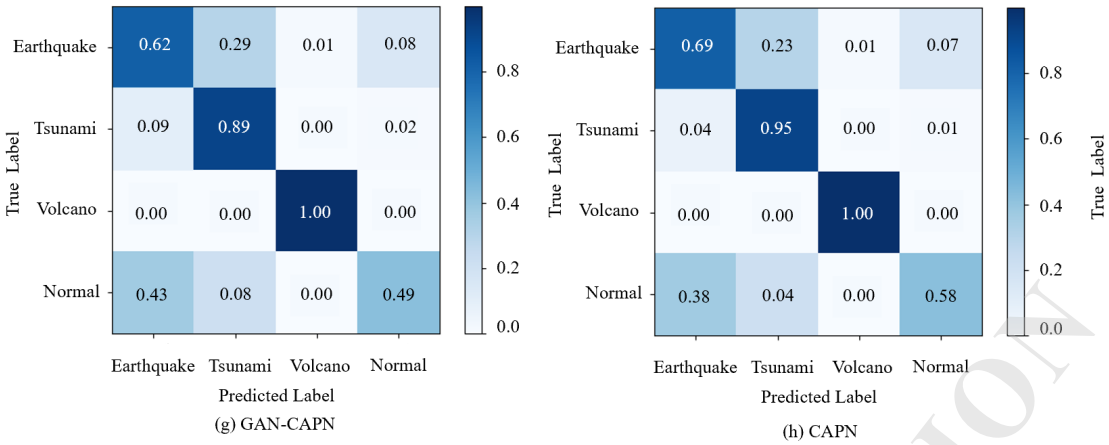


Fig. 13. Confusion matrix for infrasound classification.

4. Conclusion and future work

To further enhance the accuracy of infrasound disaster prediction, this paper proposes the TimeGAN-CAPN prediction model. The TimeGAN-CAPN model combines unsupervised and supervised learning, where the autoencoder component provides an embedding space for temporal features. The generative component operates within this embedding space to produce high-quality sequential data. By augmenting the sample data, the model increases the information content, and these new samples are then input into the CAPN to more effectively capture class prototypes, further improving prediction performance. The quality and diversity of the generated data are quantitatively and qualitatively assessed using the MMD metric and visualization methods, demonstrating that TimeGAN-CAPN generates data that closely approximates the original distribution. Comparative experiments highlight the superior predictive performance of TimeGAN-CAPN.

Although the TimeGAN-CAPN model improves infrasound prediction accuracy, it does not transform data into two-dimensional images as seen in traditional fault diagnosis models due to the limited quantity and lack of periodicity in the data. As a result, the TimeGAN-CAPN model is specifically tailored for infrasound data. Future research could explore the use of transfer learning techniques to apply the trained model to different disaster datasets, thereby improving the model's generalizability.

Fundings

This work was supported by the Project Supported by the Chongqing Natural Science Foundation General Project (Grant No. CSTB2025NSCQ-GPX0115), the Scientific and Technological Research Program of Chongqing Municipal Education Commission (Grant No. KJQN202403106), and the 'Spark' Program Project for Teachers' Independent Innovation at Chongqing Polytechnic University of Electronic Technology (Grant No. 25XJJSCX11).

Acknowledgments

Many thanks for the Comprehensive Nuclear-Test-Ban Treaty Beijing National Data Center providing the data.

References

1. BAEZA MOYANO D., GONZALEZ LEZCANO R. A. (2022), Effects of infrasound on health: Looking for improvements in housing conditions, *International Journal of Occupational Safety and Ergonomics*, **28**(2): 809-823, doi: 10.1080/10803548.2020.1831787.
2. DONG H., LIU S., LIU D., TAO Z., FANG L., PANG L., ZHANG Z. (2024), Enhanced infrasound denoising for debris flow analysis: Integrating empirical mode decomposition with an improved wavelet threshold algorithm, *Measurement*, **235**: 114961, doi: 10.1016/j.measurement.2024.114961.
3. FRIEDRICH B., JOOST H., FEDTKE T., VERHEY J. L. (2023), Effects of infrasound on the perception of a low-frequency sound, *Acta Acustica*, **7**: 60, doi: 10.1051/aacus/2023061.
4. HUPE P., CERANNA L., LE PICHON A., MATOZA R. S., MIALLE P. (2022), International monitoring system infrasound data products for atmospheric studies and civilian applications, *Earth System Science Data Discussions*, **2022**: 1-40, doi: 10.5194/essd-14-4201-2022.
5. JIANG Y. H., QIU Z. J., ZHENG L. J., DONG Z. L., JIAO W. D., TANG C., SUN J. F., XUAN Z. Y., (2025), Recursive prototypical network with coordinate attention: A model for few-shot cross-condition bearing fault diagnosis, *Applied Acoustics*, **231**: 110442, doi: 10.1016/j.apacoust.2024.110442.
6. JI Z., LIU X., PANG Y., OUYANG W., LI X. (2020), Few-shot human-object interaction

recognition with semantic-guided attentive prototypes network, *IEEE Transactions on Image Processing*, **30**: 1648-1661, doi: 10.1109/TIP.2020.3046861.

7. LU Q. B., LI M. (2023), VMD and CNN-based classification model for infrasound signal. *Archives of Acoustics*, **48**(3): 403-412, doi: 10.24425/aoa.2023.145247.
8. LISTOWSKI C., FORESTIER E., DAFIS S., FARGES T., DE CARLO M., GRIMALDI F., PICHON A. L., VERGOZ J., HEINRICH P., CLAUD C. (2022), Remote monitoring of mediterranean hurricanes using infrasound, *Remote Sensing*, **14**(23): 6162, doi: 10.3390/rs14236162.
9. MACPHERSON K. A., FEE D., COLWELL J. R., WITSIL A. J. (2023), Using local infrasound to estimate seismic velocity and earthquake magnitudes, *Bulletin of the Seismological Society of America*, **113**(4): 1434-1456, doi: 10.1785/0120220237.
10. MITROPOULOS S., TOULAS V., DOULIGERIS C. (2022), A prototype network monitoring information system: modelling, design, implementation and evaluation, *International Journal of Information and Communication Technology*, **21**(2): 111-136, doi: 10.1504/IJICT.2022.124807.
11. RUDDICK K. G., BRANDO V. E., CORIZZI A., DOGLIOTTI A. I., DOXARAN D., GOYENS C., KUUSK J., VANHELLENONT Q., VANSTEENWEGEN D., BIALEK A., DE VIS P., LAVIGNE H., BECK M., FLIGHT K., GAMMARU A., GONZALEZ VILAS L., LAIZANS K., ORTENZIO F., PERNA P., PIEGARI E., RUBINSTEIN L., SINCLAIR M., VAN DER ZANDE D. (2024), WATERHYPERNET: a prototype network of automated in situ measurements of hyperspectral water reflectance for satellite validation and water quality monitoring, *Frontiers in Remote Sensing*, **5**: 1347520, doi: 10.3389/frsen.2024.1347520.
12. SOVILLA B., MARCHETTI E., KYBURZ M. L., KOHLER A., HUGUENIN P., CALIC I., KOHLER M. J., SURINACH E., PEREZ-GUILLEN C. (2025), The dominant source mechanism of infrasound generation in powder snow avalanches, *Geophysical Research Letters*, **52**(2): e2024GL112886, doi: 10.1029/2024GL112886.
13. SHARMA S. K., ALENIZI A., KUMAR M., ALFARRAJ O., ALOWAIDI M. (2024), Detection of real-time deep fakes and face forgery in video conferencing employing generative adversarial networks, *Heliyon*, **10**(17): e37163, doi: 10.1016/j.heliyon.2024.e37163.
14. SEHAR U., XIONG J., XIA Z. (2025), Automatic tooth labeling after segmentation using prototype-based meta-learning, *Machine Intelligence Research*, **51**: 1-14, doi:

10.1007/s11633-024-1520-6.

15. TANG T., WANG J., YANG T., QIU C., ZHAO J., CHEN M., WANG L. (2023), An improved prototypical network with L2 prototype correction for few-shot cross-domain fault diagnosis, *Measurement*, **217**: 113065, doi: 10.1016/j.measurement.2023.113065.
16. VULETIC M., PRENZEL F., CUCURINGU M. (2024), Fin-gan: Forecasting and classifying financial time series via generative adversarial networks, *Quantitative Finance*, **24**(2): 175-199, doi: 10.1080/14697688.2023.2299466.
17. WATSON L. M., IEZZI A. M., TONEY L., MAHER S. P., FEE D., MCKEE K., ORTIZ H. D., MATOZA R. S., GESTRICH J. E., BISHOP J. W., WITSIL A. J. C., ANDERSON J. F., JOHNSON J. B. (2022), Volcano infrasound: Progress and future directions, *Bulletin of Volcanology*, **84**(5): 44, doi: 10.1007/s00445-022-01544-w.
18. WILSON T. C., PETRIN C. E., ELBING B. R. (2023), Infrasound and low-audible acoustic detections from a long-term microphone array deployment in Oklahoma, *Remote Sensing*, **15**(5): 1455, doi: 10.3390/rs15051455.
19. WANG W., LI H., DING Z., NIE F., CHEN J., DONG X., WANG Z. (2021), Rethinking maximum mean discrepancy for visual domain adaptation, *IEEE Transactions on Neural Networks and Learning Systems*, **34**(1): 264-277, doi: 10.1109/TNNLS.2021.3093468.
20. YANG S., CHENG Y., LEI Y., LU Z., CHENG X., WANG H., ZHU K. (2025), Correlation between and mechanisms of gas desorption and infrasound signals, *Natural Resources Research*, **34**(1): 515-537, doi: 10.1007/s11053-024-10417-2.
21. YOON J., JARRETT D., VAN DER SCHAAR M. (2019), Time-series generative adversarial networks, *Advances in neural information processing systems*, **32**:1-11.
22. ZAJAMSEK B., HANSEN K. L., NGUYEN P. D., LECHAT B., MICIC G., CATCHESIDE P. (2023), Effect of infrasound on the detectability of amplitude-modulated tonal noise, *Applied Acoustics*, **207**: 109361, doi: 10.1016/j.apacoust.2023.109361.
23. ZHANG B., XU M., ZHANG Y., YE S., CHEN Y. (2024), Attention-ProNet: A prototype network with hybrid attention mechanisms applied to zero calibration in rapid serial visual presentation-based brain–computer interface, *Bioengineering*, **11**(4): 347, doi: 10.3390/bioengineering11040347.