Data augmentation techniques applied to the classification of multi-station seismic-volcanic signals

Franz Yupanqui Machaca[a,b,c], Pablo Salazar Reinoso[b,d], and Claudio Meneses Villegas[e]

[a]PhD in Sustainable Engineering, Universidad Católica del Norte, Antofagasta, Chile
[b]Millennium Institute on Volcanic Risk Research – Ckelar Volcanoes, Antofagasta, Chile
[c]4D Perceptio, La Paz, Bolivia
[d]Department of Geological Sciences, Universidad Católica del Norte, Antofagasta, Chile
[e]Department of Computing and Systems Engineering, Universidad Católica del Norte, Antofagasta, Chile

***Abstract*—This paper presents and implements different data augmentation techniques applied to volcanic seismic signals. The datasets generated considered two types of data: real and augmented data that balance the original dataset, and datasets with fully augmented data. Using the datasets generated by these data augmentation techniques, volcanic seismic event classification models were trained in order to identify the data augmentation technique that generates the best performing models. In addition, by applying similarity and diversity metrics to the data generated by the data augmentation techniques, these characteristics were analysed in the different datasets, which does not seem to have too much impact on model performance.**

***Index Terms*—Deep Learning, Transfer Learning, Data Augmentation, Machine learning classification, Seismic volcanic signals, FID, KID, MS-SSIM.**

I. Introduction

A. The problem of recognition

The signatures are not dangerous; however, volcanic and seismic events may be dangerous events that occur unexpectedly, although they present some signals before they occur. Globally, these events are continuously registered using seismometers. Afterward, expert geophysicists classify these data.

Seismic-volcanic activity has been studied for many years. Currently, this type of activity is registered by sensors that record signals in three dimensions. The seismic-volcanic pattern recognition process is typically done manually, although there are state-of-the-art studies that make the process in a semi-automatic o automatic form [1], [2], [3], [4], [5], [6]. These studies regularly consider one of the three available channels, one of several stations, and one volcano. In this context, classical machine learning (ML) and deep learning (DL) algorithms applied to the automatic classification of seismic-volcanic signatures are becoming an emerging research field to overcome the limitations of manual classification.

This study proposes and applies a series of data augmentation (DA) techniques to volcanic seismic events and analyzes their impact on the performance of event classification models. A database of events collected from the Láscar volcano over an 8-month period and labelled by CKELAR-VOLCANES experts was considered. Five classes (see Table I) were included according to the classification given by CKELAR-VOLCANES; see also [1]. Given the nature of the seismic phenomena that occur in and around the volcano, the database is unbalanced, requiring a data balancing technique; therefore, DA techniques were considered. DA techniques were applied to generate two types of datasets: real and augmented data that balance the dataset and datasets with fully augmented data. Similarity and diversity metrics were applied to the data generated by the DA techniques to analyze these characteristics. Classification models were trained using DL and Transfer Learning (TL) techniques. The models were evaluated using precision, accuracy, recall, F1 score, and confusion matrix metrics.

Each seismic event is represented by its respective spectrogram, so that these images can be processed. Using the Transfer Learning strategy, event classification models were trained, considering the pre-trained AlexNet deep artificial neural network. Event classification models are trained on different datasets generated by different Data Augmentation techniques, and their performance is reviewed.

The results of the experiments are presented in terms of performance metrics, comparing the impact of each data augmentation method on the classification of seismic-volcanic events. The evaluation included models trained on real and augmented data, as well as models trained on entirely augmented datasets. The influence of similarity and diversity metrics on model generalization was also analyzed.

The remainder of this paper is organized as follows. The most recent bibliography on this subject is presented in Section II. Section III presents the methodology. The methods used are described in Section IV. The main results are presented in Section V. Section VI discusses the results of this study, and conclusions are presented in Section VII.

F. Yupanqui Machaca, frayup@gmail.com
C. Meneses Villegas, cmeneses@ucn.cl
P. Salazar Reinoso, pasalaz@ucn.cl
Manuscript received April 19, 2005; revised September 17, 2025.

II. Literature review

A. A summary of methods used in classificaction of volcanic seismic events

The seismic-volcanic signals are time series that are not stationary and contain information about the physical processes happening in the volcanic system. Magma movement, gas emissions, and rock fracturing are just a few of the sources that generate these signals. Understanding volcanic activity and assessing eruption risks requires the analysis of these signals. In the analysis and treatment of seismic-volcanic time signals, transformations are usually applied to represent the time domain observations in terms of the frequency and cepstral domains, and using statistical and information theory metrics, among others [3], [6], which consider statistical measures and information theory, among others. Among the methods used for the analysis of this type of signal, the more recurrent are the Hidden Markov model (HMM)[4], [7], [8] with its improvements, such as the Hidden Semi-Markov Models (HSMM), the Continuous Wavelet Transform (CWT)[7] in the detection of singularities, Gaussian Mixture Model (GMM)[9], and others, such as the HMM-based generative embedding [4].

Currently, there are several works that propose the automatic classification of seismic-volcanic signals that consider supervised Machine Learning (ML) methods such as Support Vector Machines (SVM)[3], [5], [6], [9], [10], [11], [12]; Multilayer Perceptron(MLP)[6], [9], [10]; Random Forest[3], [6], [13]; Linear Discriminant Analysis[6]; Convolutional Neural Networks (CNN)[2], [13], [14]; Temporal Convolutional Neural Network(TCN)[13], [15]; Bayesian Neural Networks[16]; Tensor Learning Framework with Multilinear Principal Component Analysis(MPCA) and Support Tensor Machine(STM)[17] techniques on multichannel and multistation data; Scalable Variational Inference for Gaussian Process (SVGP) and Deep Gaussian Processes (DGP)[18]; Bag of Words in seismic document classification as a set of seismic words[19].

With unsupervised methods we have Kohonen Self-Organizing Map (SOM)[9], [20], Cluster Analysis(CA)[9], Balanced iterative reducing and clustering using hierarchies(BIRCH)[21].

Transfer Learning (TL) was applied in the classification of volcanic seismic events using CNN[13], [22].

Deep Learning (DL) has been applied to seismic-volcanic signal classification, experimenting with different CNN[22] and Recurrent Neural Networks(RNN) architectures[2], [23], [24], and through Deep Gaussian Processes (DGP)[18].

RNNs provide real-time tracking capabilities for volcanic activity, even if the seismic sources change over time. It has been applied to seismic-volcanic signal classification and tested several architectures[2]. In detection and classification, the following RNN architectures have been applied: vanilla-RNN, Long Short-Term Memory (LSTM), and Gated-Recurrent Unit (GRU)[25]; in [15] vanilla-RNN and LSTM, in addition to identifying the degree of specialization of the neurons.

An Autoencoder (AE) is an excellent unsupervised learning algorithm. The AE is considered a nonlinear compression structure that reduces the signal size and dimensionality reduction towards another latent dimension. In [22] various AE architectures generated reconstructed and encoded seismic events, which were classified using different DL and TL model architectures. In [15] AEs were applied to obtain the segmentation mask of a seismic event.

DA involves generating new valid data from existing data following certain rules and restrictions. Amplitude deformation, frequency deformation, noise addition, and horizontal shifts were applied to the seismic signals [14]. In [1] the importance of DA in avoiding biased models in the classification of seismic-volcanic events was shown.

In reference to volcanic seismic data, most studies consider a single station channel for the classification of volcanic seismic events, but other studies consider several channels and stations [5], [6], [23], [26]. However, [13] and [23] showed that using a multi-station scheme is more precise than a single-station approach.

III. Methodology

Currently, Machine Learning is used for the recognition of seismic-volcanic signal patterns. In fact, our previous research developed[1], [22] models that implemented a new methodology for classifying these types of seismic signals using Deep Learning, Transfer Learning, and spectrograms. The rotation technique was used as a data augmentation technique to avoid imbalanced data. Current research on recognizing seismic-volcanic signal patterns considers data from a single Chilean volcano. The Millennium Institute on Volcanic Risk Research – Ckelar Volcanoes[1] monitors many of these volcanos by seismic stations provided with triaxial sensors.

A. Láscar volcano

The Láscar volcano is located in northern Chile (23°22' S, 67 °44' W; 5592 m a.s.l.), 270 km NE from Antofagasta and 70 km SE from San Pedro de Atacama. CKELAR-VOLCANES has been monitoring the area using a network of 11 three-component seismic stations. Short-period 2 Hz seismometers were continuously monitored at 200 Hz. A signal database was built using only the Z channel. The stations are distributed around the volcano, as shown in Figure 1.

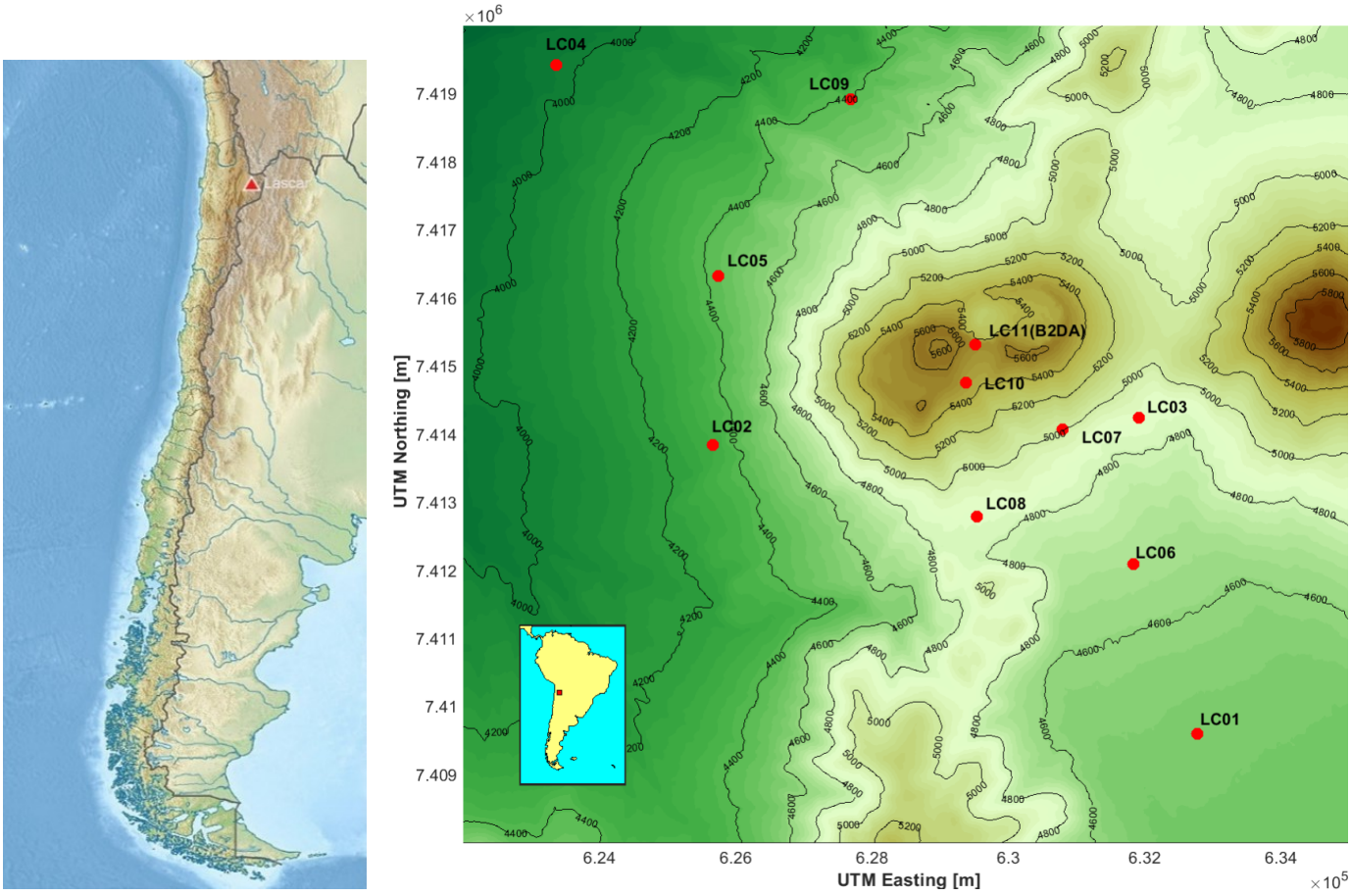
B. Lascar's database

The Láscar database contains 6,145 seismic events classified by geophysicist experts belonging to CKELAR-VOLCANES. The data correspond to the period between March and October 2018. The next table shows the data distribution.

IV. Methods

We proposed a solution based on transfer learning because this helps to speed up training and improve the performance of deep learning models [13], [16], in addition to having few examples to generate the model again. For TL, the pre-trained AlexNet[27] model was considered. The input data were spectrograms of the data labeled (waveforms) for each seismic event class. The processing sequence consists of five steps.

[1] https://ckelar.org



(a) (b)

Fig. 1: a) Location map of Láscar volcano. b) Red dots indicate the positions of the seismic stations.

TABLE I: Contribution of events from each station to the Lascar database.

Event\Station	LC01	LC02	LC03	LC04	LC05	LC06	LC07	LC08	LC09	LC10	B2DA	Total
HY	0	0	0	7	45	4	0	3	12	136	6	213
LP	11	0	4	2	142	28	5	60	23	195	107	577
TC	9	0	11	97	284	36	157	127	144	1,171	162	2,198
TR	1	0	5	29	21	4	61	13	35	292	10	471
VT	2	0	18	52	84	6	206	14	41	2,249	14	2,686
TOTAL	23	0	38	187	576	78	429	217	255	4,043	299	6,145

TABLE II: Statistics of the events belonging to the Láscar volcano.

Events	Quantity (%)	Quantity	Maximum (s)	Minimum (s)	Mean (s)	Median (s)	Mode (s)	Standard Deviation (s)
HY	3%	213	265	10	57.2	59	40	35.3
LP	9%	577	600	9	82.3	67	25	63.5
TC	36%	2,198	670	9	78.5	65	20	56.2
TR	8%	471	216	9	46.2	31	20	33.8
VT	44%	2,686	189	5	19.6	17	10	12.3
TOTAL	100%	6,145	670	5	49.9	28.0	20	50.0

1) describe the pre-process that applies to the time series (earthquake);
2) detail the generation of spectrograms from the earthquakes labeled;
3) use the data augmentation technique to increase the amount of data in training and test data sets;
4) the construction of the prediction model; and, finally,
5) estimate the performance of the model.

In the following paragraphs, each step is explained.

1) Pre-processing: The seismometers register data at 200Hz, 500Hz or 100Hz. We resampled at 100 Hz each event. The data were centered at the mean. A Butterworth Highpass filter was applied at 1 Hz. Then, we applied a Butterworth Bandpass filter at frequencies between 1 Hz and 10 Hz. The next step was to normalize the data. This pre-processing task was performed using the ObsPy Python toolbox [28].
2) The spectrograms were calculated applying a short-time Fourier transform with formula

$$S[x(t)](n,k) = \left| \sum_{m=0}^{N-1} x(m) \cdot w(m-n) \cdot e^{i2\pi mk} \right|^2$$

where x(t) and y(t) represent the seismic signal and short-time Fourier transform sliding window, respectively. The sliding window size was set to 1 with a 95% overlap. The generated spectrograms are RGB images of 224×224 pixels.
3) Data augmentation techniques are considered to avoid unbalanced dataset, and for that reason to avoid biased models. Many data augmentation techniques were implemented, which are presented in the next section.
4) The pre-trained model AlexNet[27] was considered for the being retrained over spectrograms, using the scheme of transfer learning.
5) The model performance is evaluated for metrics described in the next sections. The metrics is implemented on the TorchMetrics python toolbox[2].

Codes for reproducing our method are available at.[3]

A. Data Similarity

Kullback–Leibler divergence (KL divergence) has been used to measure similarities between synthetic and real datasets, and is defined as

$$D_{KL}(P||Q) = \sum_i P(x_i) \cdot log\left(\frac{P(x_i)}{Q(x_i)}\right)$$

where P and Q are the probability distributions whose distance is calculated over the samples x_i of the distribution[29]. Kullback–Leibler divergence (KL divergence) is not a symmetric distance, so it can be symmetrized to give rise to the so-called Jensen–Shannon divergence (JS divergense), defined as

$$JS(P||Q) = D_{KL}(P||(P+Q)/2) + D_{KL}(Q||(P+Q)/2)$$

A measurement to quantify the distance between time series distributions. It is based on the calculation of the Wasserstein distance between time-series data. This metric is defined by measuring the Wasserstein distance of the energy between frequencies. The probability distributions have a Wasserstein–Fourier distance, which is calculated as follows[29]:

$$WF([x],[y]) = W_2(s_x, s_y)$$

where s_x and s_y are the normalized power spectral densities of the distributions.

B. Data Augmentation

1) Rotation: This transformation considers the rotation of the spectrogram around the time axis within a given percentage range of the signal length and at a given axis position. The axis position can be at the beginning, middle, or end of a spectrogram. When applying this technique, the combination of the percentage rotation value, location of the rotation axis, and spectrogram of the event to which it will be applied is obtained. When generating new instances, care was taken to ensure that these three values were not repeated, thus ensuring unique instances in the dataset.

2) Jittering: Jittering is a data augmentation technique commonly employed in time series analysis[29], [30]. It is considered one of the simplest yet most effective transformation-based methods, as it involves adding noise to a time series. Formally, an original time series $x = x_1, ..., x_t, ..., x_T$ is transformed into $x' = x_1 + \epsilon_1, ..., x_t + \epsilon_t, ..., x_T + \epsilon_T$ by introducing noise at each time step t, where ϵ typically represents Gaussian noise distributed as $N(\theta, \sigma^2)$. The standard deviation σ of the added noise is a hyperparameter that must be defined in advance[30]. This approach assumes that the time series patterns of a dataset are inherently noisy. Such cases may include data derived from sensors, audio signals, or electroencephalograms (EEG). For instance, jittering has been applied to wearable sensor data for Parkinson's disease monitoring.

3) Drifting: One challenge in both online and offline settings is data drift, defined as a change in the data distribution over time. A random walk is often used to simulate such drift. In mathematics, a random walk is classified as a stochastic process that represents a path composed of successive random steps in a mathematical space, such as the integers or the reals. A simple example is the integer number line Z, where the walk begins at θ and moves either $+1$ or -1 with equal probability at each step. More generally, a random move of $+a$ or $-a$ with equal probability can occur in any vector space, such as the real space, where $a \in \mathrm{R}$ denotes the step magnitude[31]. Let d define a path starting at position $d_1 = \phi(\tau)$. A random walk can then be modeled as:

$$d_t = d_{t-1} + \phi(\tau)$$

where ϕ is the random variable describing the probability distribution for the next step, and τ is the time interval between consecutive steps. Consequently, an original time series $x = x_1, ..., x_t, ..., x_T$ can be transformed into $x' = x_1 + d_1, ..., x_t + d_t, ..., x_T + d_T$. Data drift simulation can

[2] https://lightning.ai/docs/torchmetrics/stable
[3] https://github.com/franznet/daseismicsignals

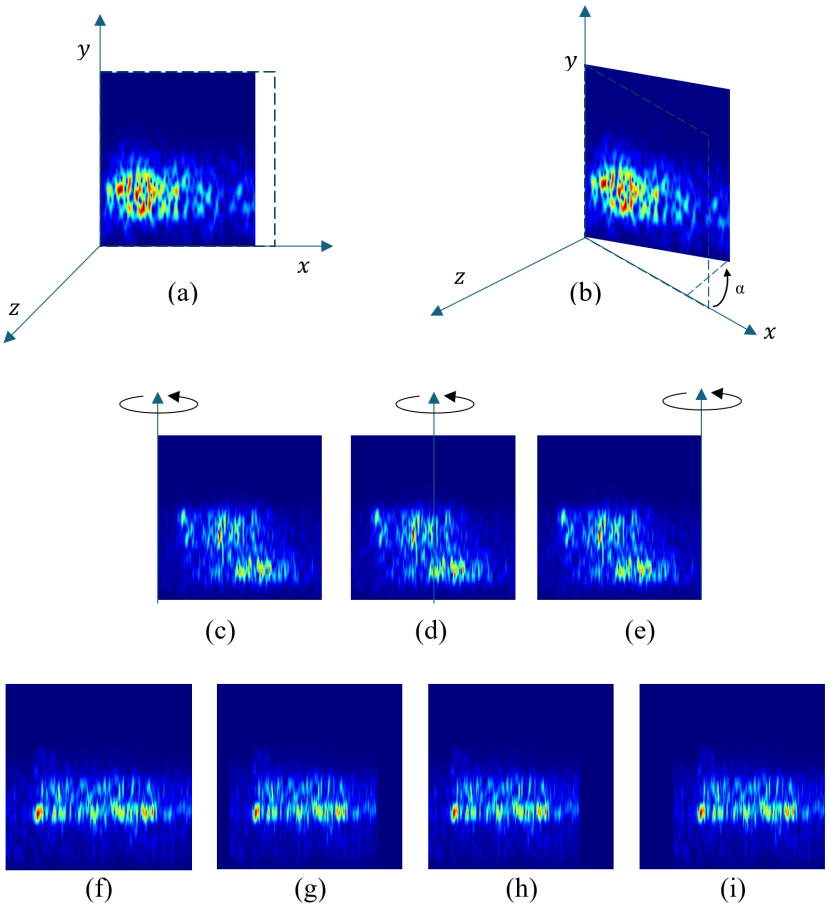


Fig. 2: Transformation by rotations, the time stretching (a) is produced when we apply the angle of rotation (b); The possible rotation axes are located at the beginning (c), center (d), and end (e) of the spectrogram, respectively. Application examples include: (f) original spectrogram, (g) 19% rotation around the central axis, (h) 23% rotation around the initial axis, and (i) 24% rotation around the final axis.

be used to generate more robust training datasets, thereby enhancing model resilience and reducing sensitivity to drift when it arises in real-world data.

4) Genetic Algorithms: Genetic Algorithms (GA) belong to the broader class of Evolutionary Algorithms (EA) and are inspired by the principle of natural selection. Their primary objective is to generate high-quality solutions to complex problems using bio-inspired operators such as selection, crossover, and mutation[32]. In [33], GA have been applied to data augmentation of time series derived from centers of pressure for the detection of neuropathies. Building on GA operators, a seismogram data augmentation technique was developed with the following procedure:

- *Selection:* Real seismograms of the same class are selected, without repetition, to serve as parents.
- *Crossover:* Genetic information from two or more parent seismograms is combined to generate a new instance. This is achieved by exchanging time series segments between the parents' chromosomes.
- *Mutation:* Small modifications are introduced in a limited portion of the population. In this case, a smoothing operation is applied to the scales of the exchanged segments.
- *Validation:* KL and JS divergences are used to measure similarity between datasets. Values closer to zero for both KL and JS divergences indicate greater similarity in the frequency-domain power distribution between the original and augmented seismic signals.

Two strategies were considered for the crossover operation:

- *Without time adjustment:* The segments exchanged between parents may differ in length, resulting in a child seismogram with a duration that can vary significantly from that of the original seismograms.
- *With time adjustment:* The exchanged segments are rescaled to equal length by applying a time-scaling transformation, ensuring that the generated seismogram maintains a duration comparable to the original signals.

5) SpecAugment: SpecAugment is a data augmentation technique designed for Automatic Speech Recognition (ASR). Its primary purpose is to help the network learn useful features that are robust to time warping, to the partial loss of frequency information, and to the partial loss of small signal segments. It is a simple and computationally inexpensive method, as it acts directly on the log-mel spectrogram as if it were an image, allowing for online application during training[34]. SpecAugment policies consist of three main components: frequency masking, time masking, and time warping[35], [36].

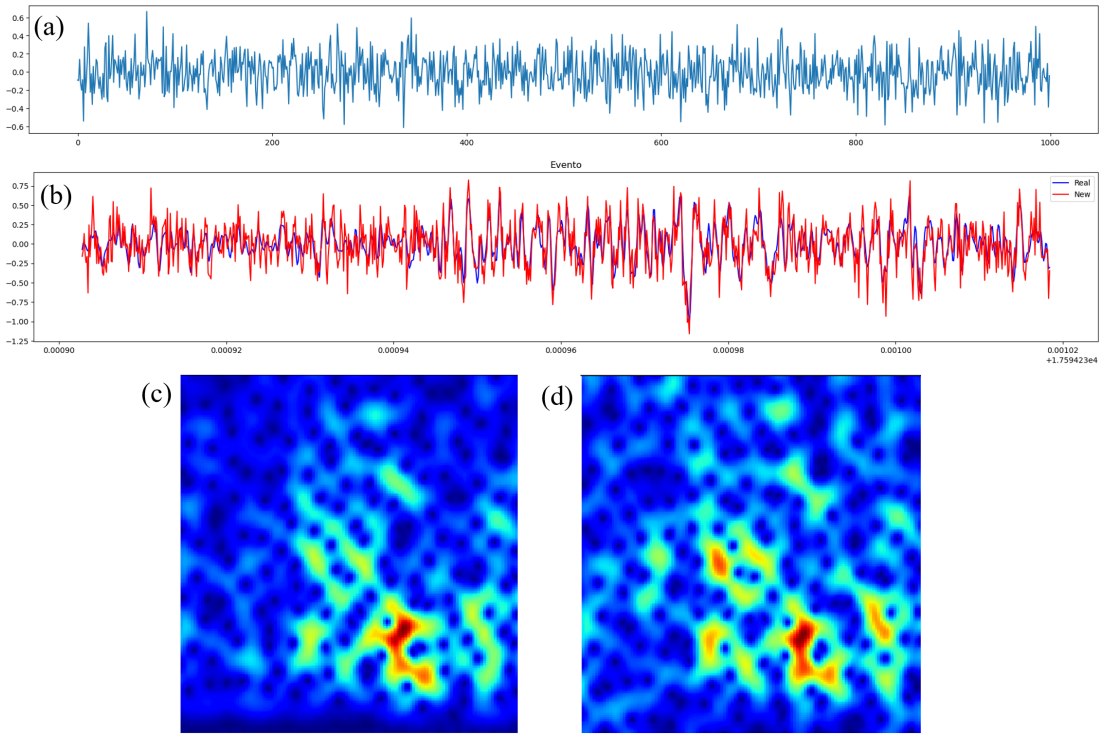


Fig. 3: Generation of a new signal using jittering. (a) Gaussian noise with distribution $N(\mu = 0, \sigma = 0.2)$. (b) Original (blue) and generated (red) signals. (c) Spectrogram of the original signal, and (d) spectrogram of the jittered signal.

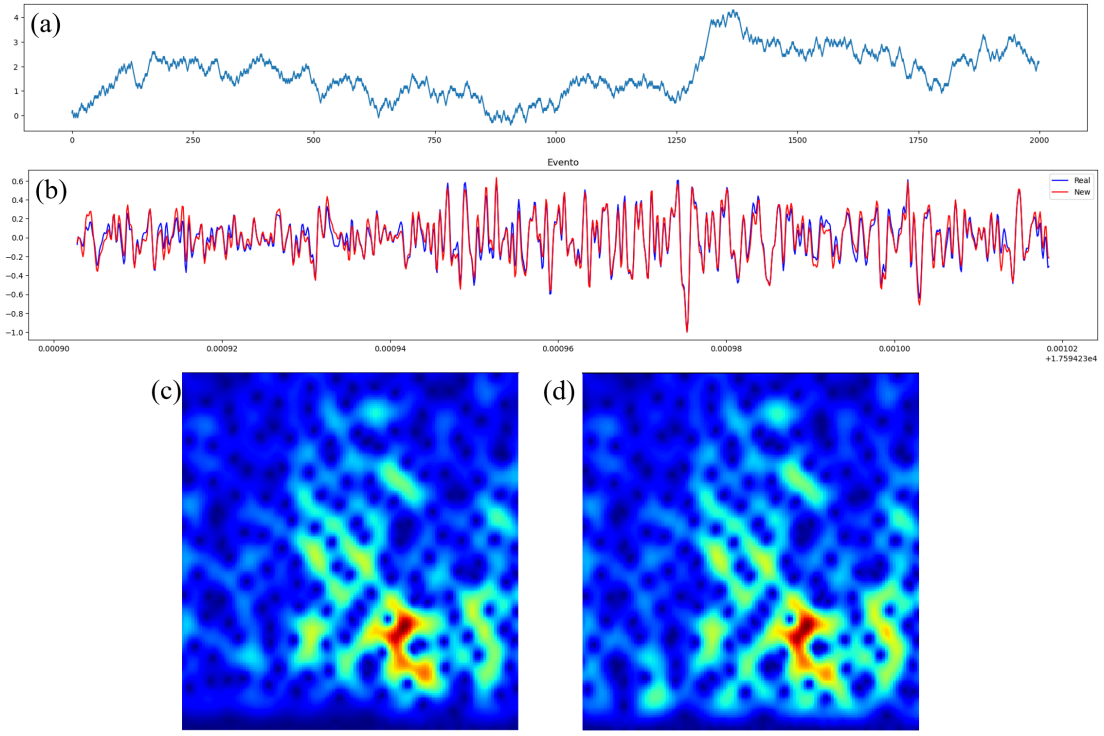


Fig. 4: Generation of a new signal using drifting. (a) Drifting signal with $\tau = 0.2$. (b) Original signal (blue) and generated signal (red). (c) Spectrogram of the original signal, and (d) spectrogram of the drifted signal.

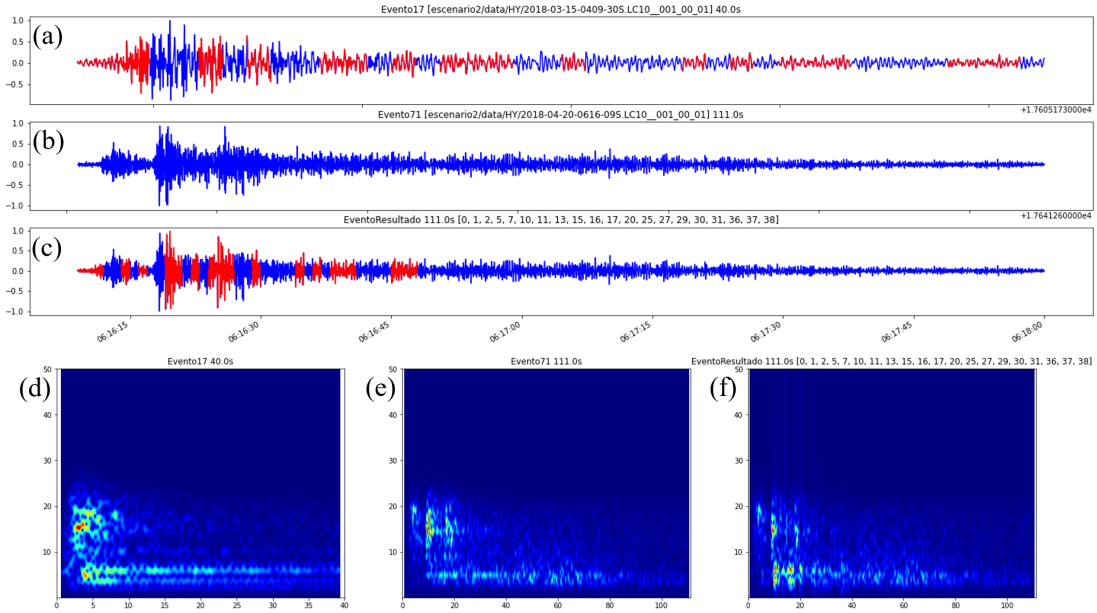


Fig. 5: Generation of a new signal using a Genetic Algorithm without time adjustment. (a) and (b) show the parent signals; the red color highlights the exchanged segments. (c) The resulting signal generated by the Genetic Algorithm. (d) and (e) show the spectrograms of the parent signals. (f) Spectrogram of the generated signal.

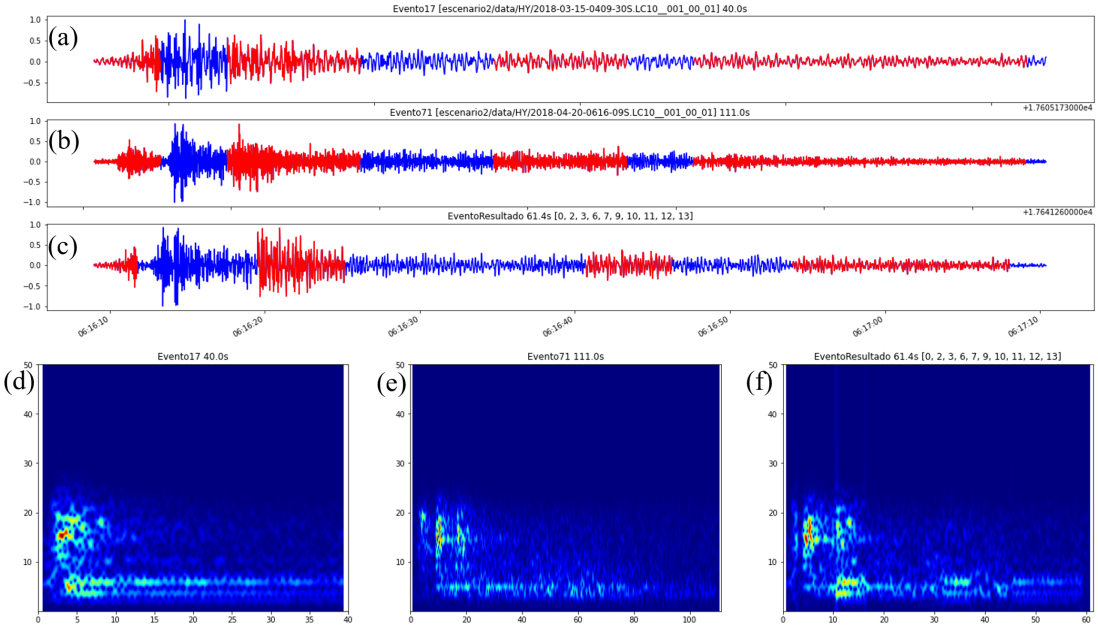


Fig. 6: Generation of a new signal using a Genetic Algorithm with time adjustment. (a) and (b) show the parent signals; the red color highlights the exchanged segments. (c) The resulting signal generated by the Genetic Algorithm. (d) and (e) display the spectrograms of the parent signals. (f) Spectrogram of the generated signal.

TABLE III: Confusion matrix.

		Predicted	
		Positive (P)	Negative (N)
Actual	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)

In this work, we apply SpecAugment on the spectrograms using frequency masking, time masking, and time warping policies.

6) Interpolation AEs: Autoencoders can interpolate in certain situations by decoding the convex combination of the latent codes of two data points, thereby producing an output that semantically blends the features of the data points[37]. First, an input $x \in \mathrm{R}^{d_x}$ is passed through an encoder $z = f_\theta(x)$, parameterized by θ, to obtain a latent code $z \in \mathrm{R}^{d_z}$. The latent code is then passed through a decoder $\hat{x} = g_\phi(z)$, parameterized by ϕ, to produce an approximate reconstruction $\hat{x} \in \mathrm{R}^{d_x}$ of the input x. We consider the case in which both f_θ and g_ϕ are implemented as multilayer neural networks. The encoder and decoder are trained jointly with respect to θ and ϕ to minimize a distance measure between the input x and the output $\hat{x}$, such as the squared L_2 distance $|x - \hat{x}|^2$. Interpolation with an autoencoder refers to the process of using the decoder g_ϕ to reconstruct a mixture of two latent codes. Typically, the latent codes are combined via a convex combination, so that interpolation amounts to computing $x_\alpha = g_\phi(\alpha z_1 + (1-\alpha)z_2)$ for some $\alpha \in [0,1]$, where $z_1 = f_\theta(x_1)$ and $z_2 = f_\theta(x_2)$ are the latent codes corresponding to the data points x_1 and x_2.

C. Model performance - Metrics

Machine learning metrics make it possible to quantify the performance of a trained model and to compare it with other generated models or with the current baseline. Evaluating the learning algorithm is a crucial part of any project. While classification accuracy is most commonly used to assess model performance, it is insufficient on its own to provide a complete evaluation. Therefore, various complementary evaluation metrics are considered. Among the different classification metrics, we can mention:

- *Confusion Matrix:* A tabular representation of true labels versus model predictions. Although not a performance metric in itself, it serves as the foundation for deriving many other evaluation measures.
  Where TP denotes true positives, TN true negatives, FP false positives, and FN false negatives.
- *Recall (Sensitivity):* Measures the proportion of actual positive cases that the model correctly identifies.
$$Recall = \frac{TP}{TP + FN}$$
- *Precision:* Measures the proportion of positive predictions that are correct, indicating the model's reliability in classification tasks.
$$Precision = \frac{TP}{TP + FP}$$
- F_β: Combines precision and recall into a single value, facilitating comparison across models. The parameter $\beta > 0$ adjusts the relative importance of recall versus precision: values $\beta > 1$ favor recall, while $\beta < 1$ favor precision. This metric is particularly useful for imbalanced datasets.
$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{\beta^2 \cdot precision + recall}$$
- *Acurracy:* Measures the overall proportion of correctly classified cases. However, it can be misleading when dealing with imbalanced datasets.
$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$
- *Specificity:* Measures the proportion of actual negative cases that the model correctly identifies.
$$Specificity = \frac{TN}{TN + FP}$$

D. Evaluation metrics for data augmentation

Data augmentation techniques have become a widely adopted strategy for improving the generalization capacity of deep neural networks. An important consideration in this context is the balance between similarity and diversity in the data [38]. Similarity reflects the degree of resemblance between the original and augmented data, whereas diversity captures the variation in complexity introduced through augmentation.

1) Fréchet Inception Distance (FID): The Fréchet Inception Distance (FID) is a widely used metric for evaluating the quality of images generated by Generative Adversarial Networks (GANs)[39]. Its primary objective is to quantify the similarity between generated and real images[40]. Compared to earlier metrics such as the Inception Score (IS), the FID offers an improvement because it leverages statistics from real-world samples for comparison, whereas the IS does not[41]. The core idea of the FID is to embed both real and generated images into a vision-relevant feature space and then compute the distance between the distributions of these embeddings. In practice, this feature space is typically the penultimate layer (pool3, with 2048 features) of an Inception-V3 classifier pre-trained on ImageNet[41].

$$FID(\mu_r, \Sigma_r, \mu_g, \Sigma_g) = \|\mu_r - \mu_g\|_2^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{\frac{1}{2}})$$

where (μ_r, Σ_r) and (μ_g, Σ_g) denote the sample main and covariance of the embeddings of the real and generated data, respectively, and $Tr(\cdot)$ indicates the matrix trace[41]. The theoretical range of the FID is $[0, \infty)$, as it is a distance measure. Lower values correspond to higher-quality images[42]. In theory, an FID of 0 indicates that the feature distributions of the generated and real images are identical in terms of mean and covariance under the Gaussian approximation[40]. Empirical studies have shown that FID correlates well with human judgments of image fidelity[40].

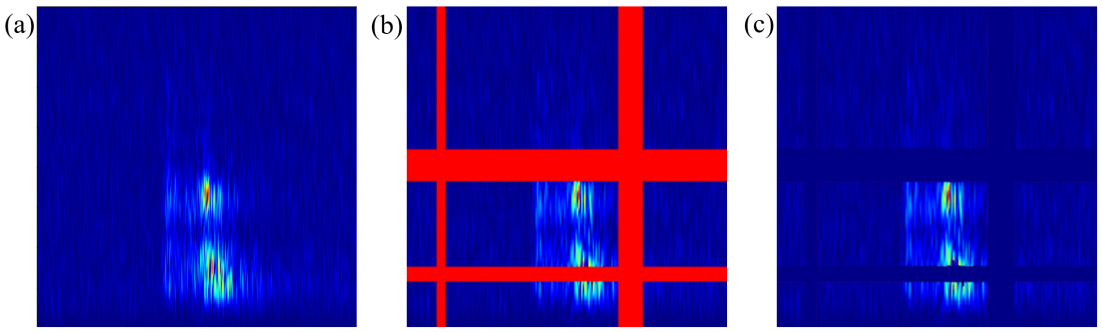


Fig. 7: Generating new signal with SpecAugment. (a) Original spectrogram. Spectrogram with two time-frequency distortions. The distortions are shown in red (b), and with white noise in (c). Generation of a new signal using SpecAugment. (a) The original spectrogram. (b) Spectrogram displaying two time-frequency distortions (highlighted in red). (c) Spectrogram displaying the same distortions with white noise.

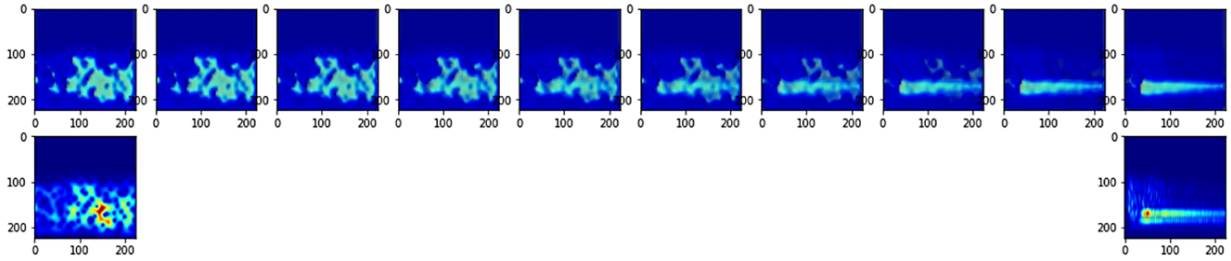


Fig. 8: Interpolating events (HY) at different intensity levels yields greater diversity.

2) Kernel Inception Distance (KID): The squared Maximum Mean Discrepancy (MMD) is used by the Kernel Inception Distance (KID) to measure the difference between the feature representations of real and generated images. These features are extracted from the Inception network. Unlike the Fréchet Inception Distance (FID)[40], KID employs an unbiased estimator, which makes it more reliable, particularly when the number of test images is smaller than the dimensionality of the Inception features. A lower KID score indicates greater visual similarity between real and generated images[43]. KID use a polynomial kernel $k(x, y) = (\frac{1}{d}x^T \cdot y+1)^3$ where d is the dimension of the Inception representation vector, and x and y are two feature vectors. The kernel $K(x, y) = k(\phi(x), \phi(y))$ can then be applied as an MMD for input images, where images are mapped into Inception representations through ϕ[44]. Since KID is a distance metric, a value of 0 indicates a perfect match between the distributions of real and generated features. Theoretically, if the two distributions are identical, the distance equals zero. A key advantage of KID lies in its simple and unbiased estimator, which distinguishes it from FID[44].

3) Multi-Scale Structural Similarity (MS-SSIM): MS-SSIM is a multi-scale improvement of SSIM that aims to eliminate aspects of an image that are not crucial for human perception [45], [46]. The calculation of SSIM, which serves as the basis for MS-SSIM, compares three principal components: luminance (intensity), contrast, and structure. It employs basic statistical moments, such as the mean and standard deviation, within local windows to derive a similarity score.

$$MS{-}SSIM(x, y) = [l_M(x, y)]^{\alpha_M} \cdot \prod_{j=1}^{M} [c_j(x, y)]^{\beta_j} \cdot [s_j(x, y)]^{\gamma_j}$$

where $l_M(x, y)$ represents the luminance component between images x and y, $c_j(x, y)$ denotes the contrast component at scale j, and $s_j(x, y)$ reflects the structural similarity in scale j. The parameters α_M, β_j, and γ_j define the relative importance of the luminance, contrast, and structure components, respectively. Lastly, M signifies the number of scales employed in the MS-SSIM calculation[47]. The range of MS-SSIM values is between 0 and 1, and images with higher MS-SSIM values are perceived as more similar[48], with 1 indicating perfect similarity. Lower MS-SSIM scores indicate higher diversity[42], [49]. Unlike metrics such as FID (Fréchet Inception Distance) or KID (Kernel Inception Distance), MS-SSIM does not require a pre-trained Inception network[50]. Therefore, the use of IS, KID, or FID is not recommended, as these metrics inherently utilize weights from the Inception network, which are validated for images similar to those in the ImageNet dataset. Consequently, these metrics cannot be applied to datasets outside ImageNet. MS-SSIM was applied to a fingerprint dataset[50]. Lower FID values indicate superior image quality, and lower MS-SSIM scores reflect increased diversity [49]. Smaller FID and KID values correlate with more realistic images in image translation [51]. Considering that our spectrogram dataset is not part of the ImageNet dataset, we utilized the MS-SSIM score.

TABLE IV: Statistics related to the transfer learning over AlexNet pre-trained model.

Dataset	Data type	Data Augmentation	Balanced dataset	Epochs	Total data	Train data	Test data	Training time (min.)
(i)	Real	No	NO	10	6,145	4,916	1,229	4.2
(ii)	Real + Augmented	Rotation(5%-25%)	YES	20	13,430	10,744	2,686	18.7
(iii)	Augmented	Rotation(5%-25%)	YES	20	13,430	10,744	2,686	20.9
(iv)	Real + Augmented	Jittering(0.2)	YES	20	13,430	10,744	2,686	16.8
(v)	Augmented	Jittering(0.2)	YES	20	13,430	10,744	2,686	17.2
(vi)	Real + Augmented	Drifting Drift(0.01-0.1)	YES	20	13,430	10,744	2,686	17.9
(vii)	Augmented	Drifting Drift(0.01-0.1)	YES	20	13,430	10,744	2,686	20.3
(viii)	Real + Augmented	Drifting Drift(0.001-0.01)	YES	20	13,430	10,744	2,686	21.7
(ix)	Augmented	Drifting Drift(0.001-0.01)	YES	20	13,430	10,744	2,686	21.0
(x)	Real + Augmented	Drifting Drift(0.0001-0.001)	YES	20	13,430	10,744	2,686	18.1
(xi)	Augmented	Drifting Drift(0.0001-0.001)	YES	20	13,430	10,744	2,686	20.9
(xii)	Real + Augmented	Genetic Algorithm Segment Size(5s) Crossover Segments(3)	YES	20	13,430	10,744	2,686	16.7
(xiii)	Augmented	Genetic Algorithm Segment Size(5s) Crossover Segments(3)	YES	20	13,430	10,744	2,686	17.0
(xiv)	Real + Augmented	Genetic Algorithm Segment Size(5s) Crossover Segments(5)	YES	20	13,430	10,744	2,686	16.6
(xv)	Augmented	Genetic Algorithm Segment Size(5s) Crossover Segments(5)	YES	20	13,430	10,744	2,686	16.9
(xvi)	Real + Augmented	Genetic Algorithm Signal Percent(40%) Crossover Segments(10)	YES	20	13,430	10,744	2,686	16.6
(xvii)	Augmented	Genetic Algorithm Signal Percent(40%) Crossover Segments(10)	YES	20	13,430	10,744	2,686	17.3
(xviii)	Real + Augmented	Genetic Algorithm Signal Percent(45%) Crossover Segments(10)	YES	20	13,430	10,744	2,686	16.5
(xix)	Augmented	Genetic Algorithm Signal Percent(45%) Crossover Segments(10)	YES	20	13,430	10,744	2,686	17.1
(xx)	Real + Augmented	SpecAugment Frequency Percent(0%-10%) Frequency Masks(2) Time Percent(0%-10%) Time Marks(2)	YES	20	13,430	10,744	2,686	16.5
(xxi)	Augmented	SpecAugment Frequency Percent(0%-10%) Frequency Masks(2) Time Percent(0%-10%) Time Marks(2)	YES	20	13,430	10,744	2,686	17.0
(xxii)	Real + Augmented	SpecAugment Frequency Percent(0%-15%) Frequency Masks(2) Time Percent(0%-15%) Time Marks(2)	YES	20	13,430	10,744	2,686	16.6
(xxiii)	Augmented	SpecAugment Frequency Percent(0%-15%) Frequency Masks(2) Time Percent(0%-15%) Time Marks(2)	YES	20	13,430	10,744	2,686	16.9
(xxiv)	Real + Augmented	Interpolation AE Signal percent(40%-60%)	YES	20	13,430	10,744	2,686	16.6
(xxv)	Augmented	Interpolation AE Signal percent(40%-60%)	YES	20	13,430	10,744	2,686	17.9

TABLE V: Performance of transfer learning models trained to 1-20 [Hz].

Dataset	Data augmentation	Accuracy	F1	Recall	Precision	Cohen Kappa	Class Accuracy				
							HY	LP	TC	TR	VT
(i)	No	62.8	63.9	62.8	66.7	67.6	38.9	74.8	73.5	33.3	93.4
(ii)	Rotation(5%-25%)	91.5	91.4	91.5	91.4	89.2	98.4	92.4	83.5	93.3	89.6
(iii)	Rotation(5%-25%)	94.2	94.0	94.2	94.0	92.6	99.8	96.3	83.1	96.1	95.8
(iv)	Jittering(0.2)	90.2	90.0	90.2	90.2	87.5	94.0	93.6	77.9	95.9	89.4
(v)	Jittering(0.2)	91.1	91.1	91.1	91.2	88.8	97.1	95.3	85.2	93.3	84.3
(vi)	Drifting Drift(0.01-0.1)	84.1	83.5	84.1	83.7	79.6	94.4	90.7	62.1	82.4	90.8
(vii)	Drifting Drift(0.01-0.1)	87.7	87.6	87.7	87.9	84.6	98.7	88.5	78.1	91.3	81.7
(viii)	Drifting Drift(0.001-0.01)	90.8	90.6	90.8	90.6	88.3	96.9	95.7	77.7	90.2	93.6
(ix)	Drifting Drift(0.001-0.01)	93.5	93.4	93.5	93.5	91.8	98.9	95.9	83.0	97.0	92.8
(x)	Drifting Drift(0.0001-0.001)	93.8	93.5	93.8	93.7	92.0	99.8	100.0	81.0	98.0	90.0
(xi)	Drifting Drift(0.0001-0.001)	97.6	97.6	97.6	97.6	97.0	100.0	99.0	93.4	100.0	95.6
(xii)	Genetic Algorithm Segment Size(5s) Crossover Segments(3)	87.8	87.6	87.8	87.7	84.6	95.8	91.5	73.6	91.7	86.7
(xiii)	Genetic Algorithm Segment Size(5s) Crossover Segments(3)	86.4	86.2	86.4	86.3	82.8	92.6	90.7	74.6	85.6	88.6
(xiv)	Genetic Algorithm Segment Size(5s) Crossover Segments(5)	86.4	86.1	86.4	86.2	82.7	94.6	88.5	72.0	86.0	91.0
(xv)	Genetic Algorithm Segment Size(5s) Crossover Segments(5)	85.8	85.5	85.8	85.9	81.9	90.4	94.4	75.1	77.3	91.8
(xvi)	Genetic Algorithm Signal Percent(40%) Crossover Segments(10)	84.9	84.7	84.9	84.7	80.8	88.1	91.5	74.1	80.0	90.8
(xvii)	Genetic Algorithm Signal Percent(40%) Crossover Segments(10)	83.0	82.9	83.0	83.0	78.6	89.2	83.1	77.2	73.9	91.8
(xviii)	Genetic Algorithm Signal Percent(45%) Crossover Segments(10)	83.7	83.4	83.7	83.8	79.3	94.0	84.3	75.0	73.2	91.8
(xix)	Genetic Algorithm Signal Percent(45%) Crossover Segments(10)	83.2	83.2	83.2	83.3	78.8	89.5	86.0	75.3	76.9	88.2
(xx)	SpecAugment Frequency Percent(0%-10%) Frequency Masks(2) Time Percent(0%-10%) Time Marks(2)	90.5	90.2	90.5	90.2	87.9	98.2	95.5	77.9	90.2	90.4
(xxi)	SpecAugment Frequency Percent(0%-10%) Frequency Masks(2) Time Percent(0%-10%) Time Marks(2)	91.7	91.5	91.7	91.5	89.4	97.5	94.8	83.5	89.5	93.2
(xxii)	SpecAugment Frequency Percent(0%-15%) Frequency Masks(2) Time Percent(0%-15%) Time Marks(2)	88.6	88.6	88.6	88.5	85.7	94.9	86.8	80.5	89.5	91.4
(xxiii)	SpecAugment Frequency Percent(0%-15%) Frequency Masks(2) Time Percent(0%-15%) Time Marks(2)	86.6	86.4	86.6	86.4	83.1	96.0	85.6	75.8	87.2	88.2
(xxiv)	Interpolation AE Signal percent(40%-60%)	87.5	87.4	87.5	87.6	84.1	89.3	90.9	76.9	89.3	91.0
(xxv)	Interpolation AE Signal percent(40%-60%)	93.1	93.1	93.1	93.3	91.3	97.6	96.5	83.7	96.5	91.4

V. RESULTS

We identified an unbalanced data distribution, comprising 3% (HY), 9% (LP), 36% (TC), 8% (TR), and 44% (VT), as shown in Table II. To address this imbalance, we generated new data through various data augmentation techniques, including rotation, jittering, drifting, scaling, flipping, genetic algorithms, SpecAugment, and interpolation. Each data augmentation (DA) technique was applied to balance the existing dataset, resulting in datasets containing 13,430 events (2,686 examples per class). Furthermore, we created balanced datasets consisting solely of augmented data, ensuring an equal number of events, as detailed in Table IV. Multiple experiments were designed based on each dataset generated by the respective DA methods (refer to Table V). For each newly balanced dataset, a deep learning model was trained utilizing a transfer learning strategy with the pre-trained AlexNet model. For each balanced dataset generated using the aforementioned data augmentation techniques, a deep learning model was trained using a transfer learning strategy with the pre-trained AlexNet model. The training process involved 20 epochs, a mini-batch size of 8, and an initial learning rate of 0.00005 with optimization algorithm Adam. The training and validation performance scores for selected experiments are illustrated in Figure 10. The trained models were subsequently evaluated using the original unbalanced dataset of 6,145 events, with the results presented in Table VI. Additionally, confusion matrices for selected experiments are depicted in Figure 11.

A. Real

The original unbalanced dataset, comprising 6,145 events, was utilized, divided into proportions of 80% for training and 20% for testing the model. This experiment aims to highlight the influence of an unbalanced dataset on the accuracy of the probabilistic model.

TABLE VI: Test of trained models over the real dataset of 6,145 events.

Dataset	Data augmentation	Accuracy	F1	Recall	Precision	Cohen Kappa	Class Accuracy				
							HY	LP	TC	TR	VT
(i)	No	-	-	-	-	-	-	-	-	-	-
(ii)	Rotation(5%-25%)	96.6	96.2	96.6	95.9	95.7	97.7	95.3	97.0	94.9	98.1
(iii)	Rotation(5%-25%)	93.7	88.9	93.7	85.5	86.9	100.0	96.0	82.5	93.8	96.0
(iv)	Jittering(0.2)	96.2	96.0	96.2	95.8	95.0	94.8	96.7	95.8	96.2	97.5
(v)	Jittering(0.2)	93.9	87.3	93.9	82.9	86.2	99.5	97.1	86.0	95.8	91.3
(vi)	Drifting Drift(0.01-0.1)	91.5	89.4	91.5	88.1	90.0	93.0	95.5	90.0	81.5	97.6
(vii)	Drifting Drift(0.01-0.1)	66.6	53.5	66.6	55.2	48.3	69.5	96.5	42.0	56.5	68.4
(viii)	Drifting Drift(0.001-0.01)	95.0	95.4	95.0	95.8	94.7	93.0	97.1	94.9	91.3	98.8
(ix)	Drifting Drift(0.001-0.01)	93.7	86.7	93.7	82.4	85.5	98.1	97.4	83.4	97.7	91.8
(x)	Drifting Drift(0.0001-0.001)	98.3	96.7	98.3	95.3	96.0	99.5	100.0	95.5	98.5	97.8
(xi)	Drifting Drift(0.0001-0.001)	97.2	93.8	97.2	91.0	92.9	100.0	98.3	93.3	99.6	94.9
(xii)	Genetic Algorithm Segment Size(5s) Crossover Segments(3)	95.9	93.7	95.9	91.8	93.8	96.7	97.6	94.1	94.3	97.0
(xiii)	Genetic Algorithm Segment Size(5s) Crossover Segments(3)	82.7	73.6	82.7	70.0	71.8	86.9	93.4	65.2	78.3	89.8
(xiv)	Genetic Algorithm Segment Size(5s) Crossover Segments(5)	95.2	93.6	95.2	92.2	93.6	95.3	96.7	93.2	92.4	98.3
(xv)	Genetic Algorithm Segment Size(5s) Crossover Segments(5)	75.2	67.3	75.2	64.9	64.3	77.5	91.2	57.3	61.4	88.8
(xvi)	Genetic Algorithm Signal Percent(40%) Crossover Segments(10)	93.3	94.0	93.3	94.9	93.8	90.6	98.1	95.5	84.1	98.3
(xvii)	Genetic Algorithm Signal Percent(40%) Crossover Segments(10)	81.4	74.2	81.4	70.7	72.4	85.4	83.9	67.9	78.6	91.1
(xviii)	Genetic Algorithm Signal Percent(45%) Crossover Segments(10)	94.0	94.0	94.0	94.2	93.8	95.3	94.8	95.1	86.4	98.5
(xix)	Genetic Algorithm Signal Percent(45%) Crossover Segments(10)	80.1	68.1	80.1	64.9	65.8	88.7	92.5	56.9	76.6	85.9
(xx)	SpecAugment Frequency Percent(0%-10%) Frequency Masks(2) Time Percent(0%-10%) Time Marks(2)	96.7	96.4	96.7	96.1	95.4	98.6	98.6	95.8	92.4	98.2
(xxi)	SpecAugment Frequency Percent(0%-10%) Frequency Masks(2) Time Percent(0%-10%) Time Marks(2)	95.4	91.1	95.4	87.8	89.8	100.0	97.2	88.1	96.6	95.1
(xxii)	SpecAugment Frequency Percent(0%-15%) Frequency Masks(2) Time Percent(0%-15%) Time Marks(2)	95.5	95.7	95.5	95.9	94.9	97.7	90.5	96.3	94.9	98.4
(xxiii)	SpecAugment Frequency Percent(0%-15%) Frequency Masks(2) Time Percent(0%-15%) Time Marks(2)	93.6	87.5	93.6	83.7	85.2	100.0	95.1	82.8	97.5	92.4
(xxiv)	Interpolation AE Signal percent(40%-60%)	90.4	91.8	90.4	93.4	92.0	82.2	90.5	94.7	86.4	98.1
(xxv)	Interpolation AE Signal percent(40%-60%)	80.7	71.0	80.7	67.5	67.8	88.7	85.8	60.4	80.7	88.1

B. Rotation

The original unbalanced dataset of 6,145 events has been considered and subsequently balanced using rotation as a data augmentation technique. This approach increases the number of examples from underrepresented classes, resulting in a balanced dataset of 13,430 events. This dataset will demonstrate the effectiveness of this data augmentation technique in constructing a classification model. Additionally, a balanced dataset consisting solely of 13,430 augmented events was generated, with 2,686 examples for each class, for comparative purposes. The rotation parameters employed ranged from 5% to 25%, utilizing random rotation axes (start, center, and end) without repetition.

C. Jittering

The original unbalanced dataset, comprising 6,145 events, was utilized, and data augmentation techniques were applied to achieve balance by increasing the number of examples from underrepresented classes. This process resulted in a balanced dataset of 13,430 events. The parameters employed include Gaussian noise, following a distribution of $N(\mu = 0, \sigma = 0.2)$. Additionally, a balanced dataset of 13,430 events was generated entirely from augmented data.

D. Drifting

The original unbalanced dataset of 6,145 events was considered, and the data augmentation technique was applied in a manner that balanced the dataset by increasing examples from underrepresented classes, yielding a total of 13,430 events. The parameters utilized include random values sampled from defined real intervals $[a, b]$. Similarly, a balanced dataset consisting exclusively of 13,430 augmented events was generated.

E. Genetic Algorithm

Utilizing the data augmentation technique of Genetic Algorithms, a balanced dataset of 13,430 augmented events was generated, comprising 2,686 examples for each class. Various values for the signal proportion parameters of the parent events, as well as the number of crossover segments, were considered for generating the new dataset. This dataset aims to demonstrate the effectiveness of this data augmentation technique in developing the model. This data augmentation technique includes two variants:

- *Without Time Adjustment:* In this variant, the parents are segmented into the specified number of crossover segments, with crossover occurring at defined times. The parameters used include three crossover segments, each lasting 5 seconds.
- *With Time Adjustment:* In this variant, both the number of crossover segments and the signal proportion for the crossover are defined. The segment times are calculated proportionally for both parents.

F. SpecAugment

Using the SpecAugment data augmentation technique, a balanced dataset of 13,430 augmented events was generated, comprising 2,686 examples for each class. This dataset aims to demonstrate the effectiveness of this data augmentation technique in developing the model. To generate the new dataset, different values for the maximum percentage parameters of each time and frequency mask were considered, along with the number of masks applied. The parameters employed include two time and frequency masks, each covering 10% of the signal.

G. Interpolation AEs

This technique was utilized to generate a balanced dataset of 13,430 purely augmented events. For this technique, a randomly selected parameter $\alpha \in [0.4, 0.6]$ was used in conjunction with a specifically designed convolutional autoencoder to reconstruct the interpolated signal, as illustrated in Figure 9.

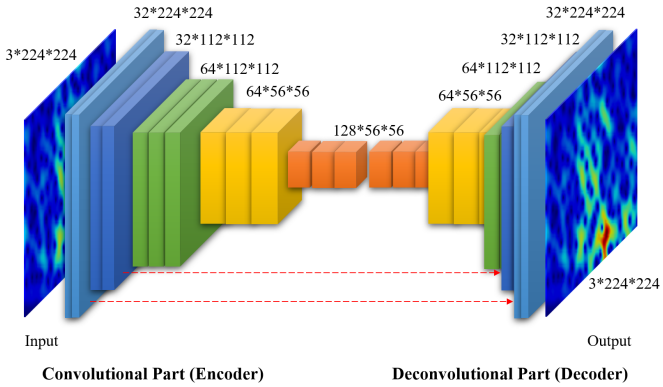


Fig. 9: Convolutional AutoEncoder of spectrograms.

The experimental evidence underscores the importance of data augmentation techniques in enhancing the precision of deep learning models. Models trained without these data augmentation techniques exhibited low precision, as illustrated in Table IV. To compare our models with the state-of-the-art, we considered performance metrics, including accuracy, precision, recall, specificity, and F1 score.

VI. Discussion

The various experiments conducted in this research highlight the performance of the probabilistic model on unbalanced datasets. Specifically, dataset (i) produced a model biased toward the LP, TC, and VT events, as shown in Table V. The results indicate that balanced datasets yield higher-performing models, as they expand the examples within the event feature space. The models were subsequently tested on real data, demonstrating competitive classification levels in most cases. Transfer learning facilitated the development of deep learning models in a reduced timeframe, as evidenced in Table V. In Tables V and VI, it is evident that the drifting data augmentation technique achieved the best results across nearly all pre-trained models. This success may be attributed to the drifting technique's ability to effectively cover the data space. The most notable result is from drifting (xi), which, with a drift range of 0.0001-0.001, achieved an accuracy of 97.6% over an entirely augmented dataset. An additional noteworthy result is the rotation (iii) technique, which, with a range of 5%-25%, achieved an accuracy of 94.2In both cases, the F1 score metrics closely correspond to their respective accuracies. Although both models were derived from entirely augmented data, they performed well in real data tests, achieving accuracies of 97.2% for (xi) and 93.7% for (iii) (refer to Table VI). However, the highest accuracy of 98.3% was recorded for model (x), which utilized both real and augmented data. It can be observed that the dataset composed of real and augmented data (x) generated by drifting in the range of 0.0001-0.001 reaches an accuracy of 93.8%. Then the dataset of augmented data (ix) generated by drifting in range 0.001-0.01, reaches an accuracy of 93.5%. Then we have the augmented data dataset (xxv) generated by Autoencoder Interpolation in the range 40%-60%, it obtains an accuracy of 93.1% (Table V). These models also show good performance in tests with real data, with accuracies of 98.3% for (x), 93.7% for (ix) and 80.7% for (xxv) (Table VI). Only the last model (xxv) has a large distance between training and test accuracy on real data. To obtain a generalized dataset, data from several stations were considered to generate a multi-station generalized probabilistic model. Furthermore, it was observed that models trained with completely augmented data achieved high recognition accuracy with real data in most cases (Table VI). Figure 12 shows that drifting, rotation, interpolation, specaugment and jittering have slightly better accuracy values than the other techniques. In general, the trained models perform better when tested on real data than during training.

Regarding the similarity and diversity metrics, it can be seen that the MS-SSIM values for the rotation and drifting techniques are similar, although the former shows more diversity, depending on the generation parameters (Table VII). It can also be observed that Jittering exhibits higher diversity in datasets (v) and (iv), with MS-SSIM values of 0.26 and 0.29, respectively. The dataset with the highest similarity and lowest diversity is Drifting (0.01-0.1) (vii), which has an MS-SSIM index of 0.78. This is followed by Interpolation (xxv) and Drifting (vi), with MS-SSIM values of 0.42 and 0.41, respectively. Figure 13 illustrates the concentration of points representing the MS-SSIM similarity metric against the accuracy of model training. Notably, two distant points are evident: the dataset comprising only real data (i), which has low accuracy due to being unbalanced, and the drifting dataset (vii) with a range of 0.01-0.1, which shows higher similarity. Upon zooming into Figure 14, it is clear that almost all datasets demonstrate similar levels of diversity, including the real dataset; however, the augmented Jittering dataset (v) shows greater diversity. When comparing augmented datasets with real and augmented datasets within the same data augmentation (DA) method, it can be noted that augmented datasets exhibit more diversity and superior training performance, especially with Jittering, SpecAugment (xx, xxii), and Rotation. In Figure 15, certain augmented and real datasets demonstrate better diversity and testing accuracy values compared to their counterparts within the same DA method. Notable examples include Drifting (x, xi), Rotation (ii, iii), and SpecAugment (xxii, xxiii). By generating a simple index that accounts for the model training accuracy (which is better when higher) and the MS-SSIM metric value (which is more favorable when lower, indicating greater diversity), we can observe (see Figure 16) that models trained on the Drifting (x, xi), Jittering (iv, v), SpecAugment (xx, xxii), and Rotation (ii, iii) datasets are more relevant in terms of model accuracy and dataset diversity. Conversely, models with low indices, such as the Drifting dataset (vii) and the real event dataset (i), are disadvantaged by their lower diversity and unbalanced nature, respectively.

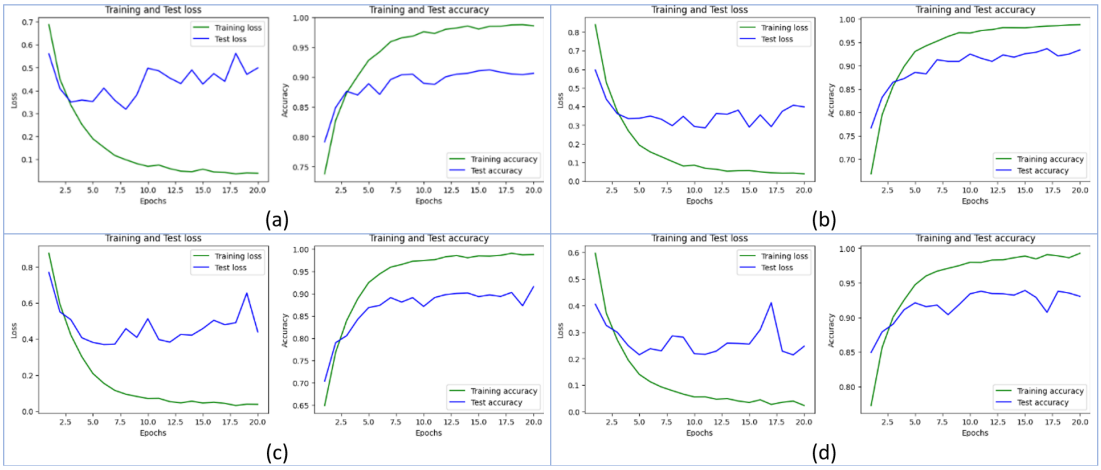


Fig. 10: The training, and validation performance scores of the transfer learning using the AlexNet for the experiment: a) training and test loss for the experiment (viii), b) (ix), c) (xxi), and d) (xxv).

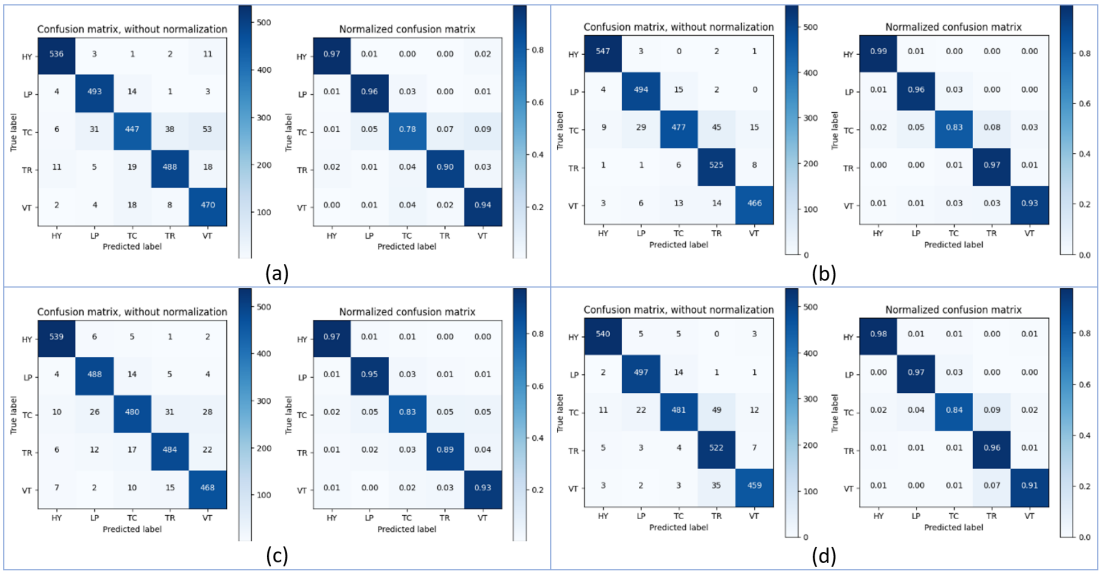


Fig. 11: Confusion matrix without normalization and normalized for experiments a) (viii), b) (ix), c) (xxi) and d) (xxv).

TABLE VII: Similarity and Diversity metrics for experimental datasets.

Dataset	Data augmentation	Fréchet Inception Distance (FID)						Kernel Inception Distance (KID)						Multiscale Structural Similarity Index (MS-SSIM)					
		HY	LP	TC	TR	VT	FID	HY	LP	TC	TR	VT	KID	HY	LP	TC	TR	VT	MS
(i)	No	0.16	0.10	0.02	0.05	0.02	0.03	0.14	-0.75	3.78	0.31	-0.23	1.21	0.34	0.41	0.33	0.29	0.34	0.34
(ii)	Rotation(5%-25%)	0.70	0.61	0.50	0.45		0.56	4.81	-1.40	4.41	2.56		2.59	0.38	0.42	0.36	0.29		0.36
(iii)	Rotation(5%-25%)	0.03	0.01	0.04	0.02	0.03	0.02	1.62	0.15	1.16	-1.03	2.34	0.85	0.36	0.42	0.36	0.33	0.35	0.37
(iv)	Jittering(0.2)	10.09	5.46	7.65	8.88		8.02	56.99	18.80	40.98	48.72		41.37	0.28	0.36	0.27	0.26		0.29
(v)	Jittering(0.2)	0.01	0.01	0.02	0.02	0.03	0.02	-2.50	-2.35	3.38	0.24	-2.77	-0.80	0.26	0.33	0.26	0.21	0.23	0.26
(vi)	Drifting Drift(0.01-0.1)	34.80	30.48	35.93	30.01		32.81	102.53	71.83	89.02	83.60		86.75	0.40	0.46	0.42	0.35		0.41
(vii)	Drifting Drift(0.01-0.1)	0.01	0.02	0.01	0.01	0.01	0.01	-0.06	-0.10	0.29	0.51	-0.11	0.11	0.80	0.81	0.81	0.76	0.71	0.78
(viii)	Drifting Drift(0.001-0.01)	1.50	1.23	2.69	0.84		1.56	3.25	3.61	3.84	1.01		2.93	0.37	0.42	0.38	0.31		0.37
(ix)	Drifting Drift(0.001-0.01)	0.03	0.01	0.02	0.03	0.01	0.02	1.53	-1.02	0.30	1.14	-0.80	0.23	0.42	0.48	0.43	0.37	0.36	0.41
(x)	Drifting Drift(0.0001-0.001)	0.02	0.02	0.03	0.02		0.02	-4.42	4.68	-3.96	0.79		-0.73	0.35	0.40	0.33	0.29		0.34
(xi)	Drifting Drift(0.0001-0.001)	0.01	0.01	0.02	0.01	0.01	0.01	-5.60	-1.65	2.42	3.00	1.74	-0.02	0.35	0.40	0.32	0.31	0.34	0.34
(xii)	Genetic Algorithm Segment Size(5s) Crossover Segments(3)	0.89	0.41	0.39	0.28		0.49	2.36	2.98	-1.94	2.55		1.49	0.38	0.40	0.35	0.30		0.36
(xiii)	Genetic Algorithm Segment Size(5s) Crossover Segments(3)	0.02	0.01	0.00	0.01	0.01	0.01	0.45	-2.49	-0.40	-2.06	2.54	-0.39	0.35	0.40	0.37	0.30	0.36	0.36
(xiv)	Genetic Algorithm Segment Size(5s) Crossover Segments(5)	1.65	0.83	0.76	1.03		1.07	1.51	4.59	2.90	1.29		2.57	0.35	0.41	0.35	0.30		0.35
(xv)	Genetic Algorithm Segment Size(5s) Crossover Segments(5)	0.02	0.01	0.01	0.02	0.01	0.01	1.06	-0.48	-0.32	-2.37	-1.77	-0.77	0.35	0.40	0.37	0.33	0.40	0.37
(xvi)	Genetic Algorithm Signal Percent(40%) Crossover Segments(10)	0.90	0.63	0.55	0.41		0.63	3.99	1.91	0.08	4.25		2.56	0.36	0.41	0.35	0.30		0.35
(xvii)	Genetic Algorithm Signal Percent(40%) Crossover Segments(10)	0.02	0.02	0.04	0.01	0.01	0.02	-0.28	-3.37	-0.14	1.87	-0.70	-0.52	0.34	0.41	0.32	0.29	0.34	0.34
(xviii)	Genetic Algorithm Signal Percent(45%) Crossover Segments(10)	0.97	0.67	0.73	0.36		0.68	4.43	-0.28	3.54	0.97		2.16	0.35	0.41	0.33	0.30		0.35
(xix)	Genetic Algorithm Signal Percent(45%) Crossover Segments(10)	0.01	0.00	0.00	0.01	0.01	0.01	0.91	0.75	-0.73	2.75	0.86	0.91	0.31	0.41	0.35	0.30	0.34	0.34
(xx)	SpecAugment Frequency Percent(0%-10%) Frequency Masks(2) Time Percent(0%-10%) Time Marks(2)	1.53	1.48	1.85	1.21		1.52	6.23	4.41	6.33	2.06		4.76	0.37	0.41	0.33	0.30		0.35
(xxi)	SpecAugment Frequency Percent(0%-10%) Frequency Masks(2) Time Percent(0%-10%) Time Marks(2)	0.00	0.01	0.02	0.01	0.01	0.01	-1.21	2.46	-0.33	-1.04	1.03	0.18	0.34	0.40	0.35	0.31	0.34	0.35
(xxii)	SpecAugment Frequency Percent(0%-15%) Frequency Masks(2) Time Percent(0%-15%) Time Marks(2)	3.84	3.45	3.08	3.22		3.40	16.73	11.34	7.82	6.15		10.51	0.35	0.41	0.33	0.31		0.35
(xxiii)	SpecAugment Frequency Percent(0%-15%) Frequency Masks(2) Time Percent(0%-15%) Time Marks(2)	0.01	0.01	0.01	0.00	0.01	0.01	0.31	1.29	0.47	0.72	-0.61	0.44	0.35	0.40	0.36	0.32	0.39	0.37
(xxiv)	Interpolation AE Signal percent(40%-60%)	2.31	2.46	2.18	1.88		2.21	9.48	3.04	4.03	7.12		5.92	0.38	0.44	0.36	0.33		0.38
(xxv)	Interpolation AE Signal percent(40%-60%)	0.01	0.01	0.01	0.01	0.01	0.01	-0.22	0.62	-0.89	-0.44	0.03	-0.18	0.43	0.48	0.41	0.37	0.41	0.42

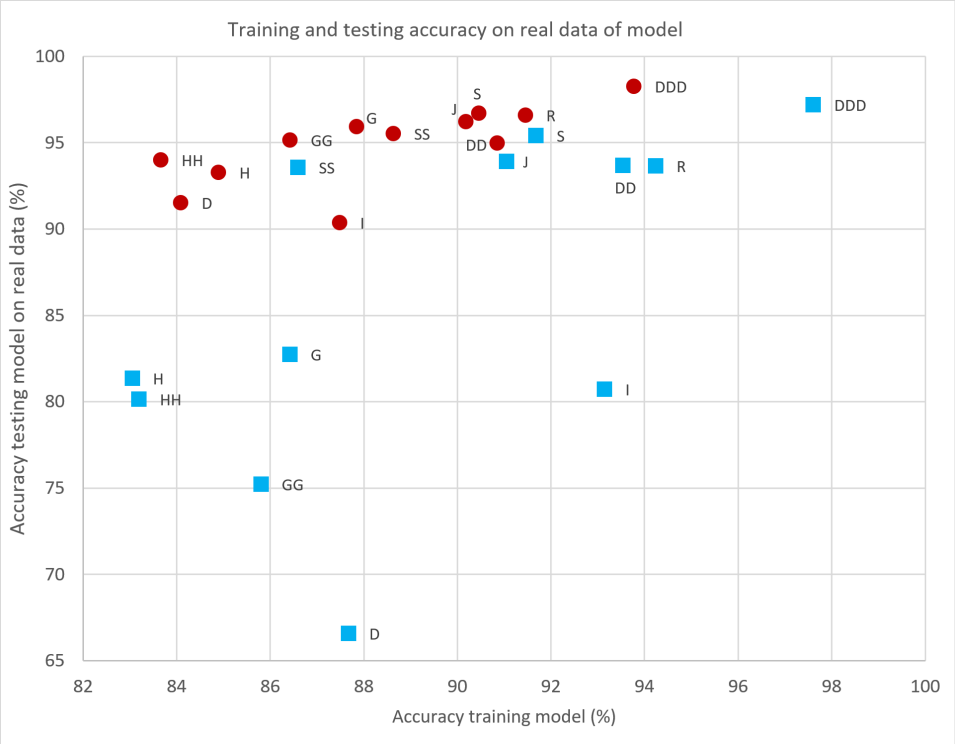


Fig. 12: Accuracy of model training and testing on real data. **Green** shape indicates real dataset with 62.8% of accuracy, **red** circles indicate datasets with real and augmented data, and **turquoise** squares are entirely augmented datasets. The codes are described in Table VIII.

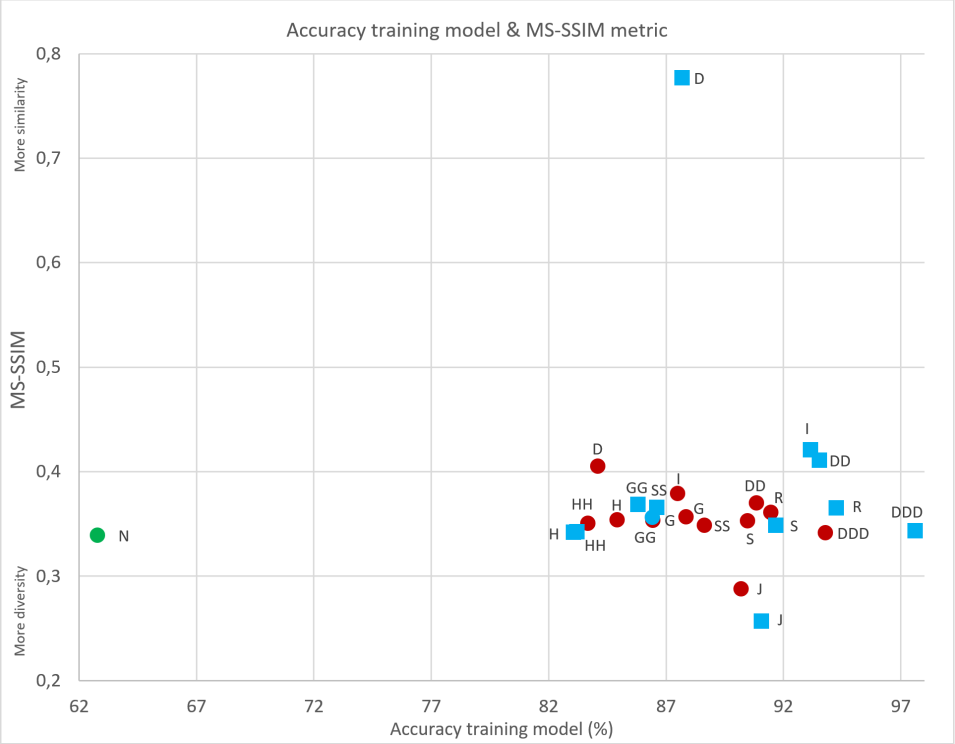


Fig. 13: Accuracy of model training and MS-SSIM metric. **Green** shape indicates real dataset, **red** circles indicate datasets with real and augmented data, and **turquoise** squares are entirely augmented datasets. The codes are described in Table VIII.

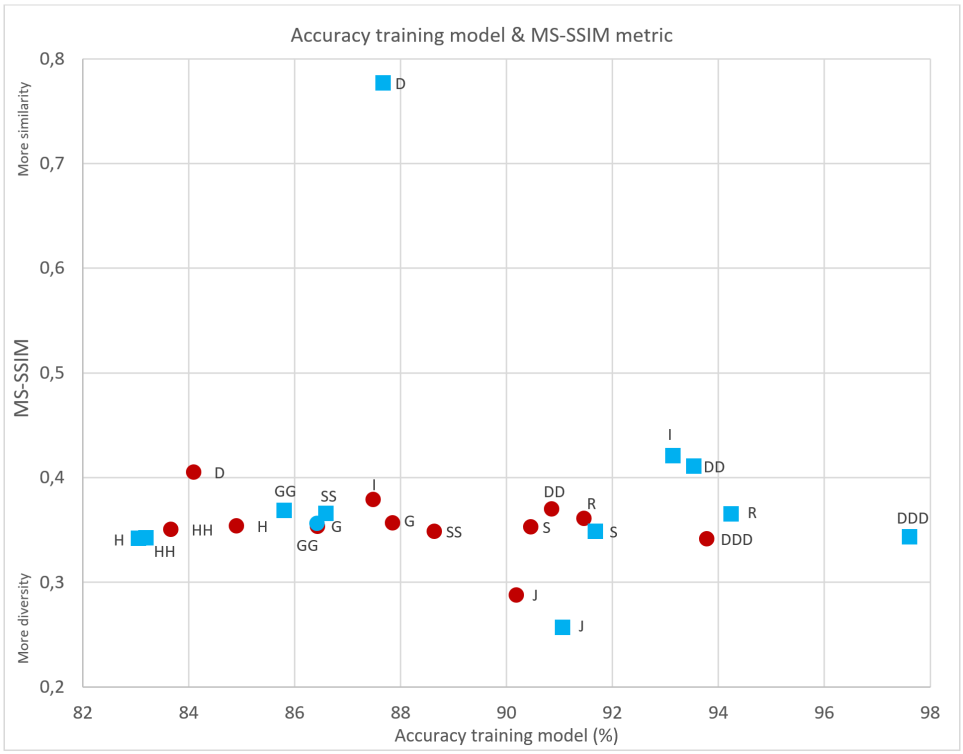


Fig. 14: Zooming at figure of model training accuracy and MS-SSIM metric. **Green** shape indicates real dataset, **red** circles indicate datasets with real and augmented data, and **turquoise** squares are entirely augmented datasets. The codes are described in Table VIII.

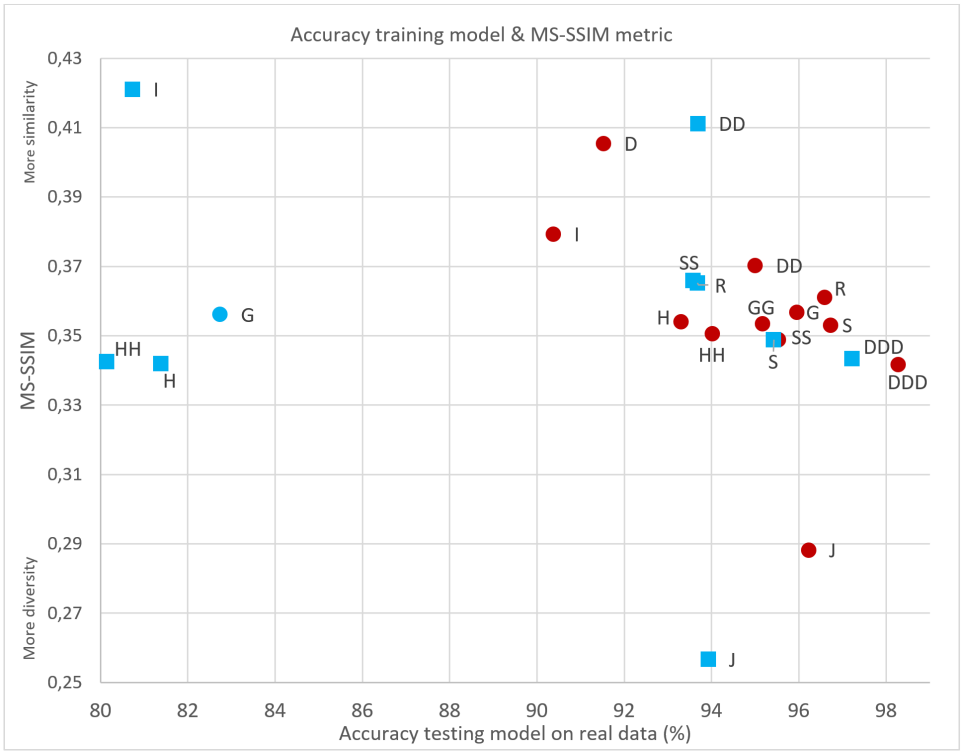


Fig. 15: Accuracy of model testing and MS-SSIM metric. **Green** shape indicates real dataset, **red** circles indicate datasets with real and augmented data, and **turquoise** squares are entirely augmented datasets. The codes are described in Table VIII.

TABLE VIII: Relationship between Data Augmentation and Dataset.

Code	Data augmentation	Datasets	
		Real + Augmented	Augmented
N	Real dataset	i	
R	Rotation (5%-25%)	ii	iii
J	Jittering (0.2)	iv	v
D	Drifting (0.01-0.1)	vi	vii
DD	Drifting (0.001-0.01)	viii	ix
DDD	Drifting (0.0001-0.001)	x	xi
G	Genetic Algorithm Segment Size(5s) Crossover Segments(3)	xii	xiii
GG	Genetic Algorithm Segment Size(5s) Crossover Segments(5)	xiv	xv
H	Genetic Algorithm Signal Percent(40%) Crossover Segments(10)	xvi	xvii
HH	Genetic Algorithm Signal Percent(45%) Crossover Segments(10)	xviii	xix
S	SpecAugment Frequency Percent(0%-10%) Frequency Masks(2) Time Percent(0%-10%) Time Marks(2)	xx	xxi
SS	SpecAugment Frequency Percent(0%-15%) Frequency Masks(2) Time Percent(0%-15%) Time Marks(2)	xxii	xxiii
I	Interpolation AE Signal percent(40%-60%)	xxiv	xv

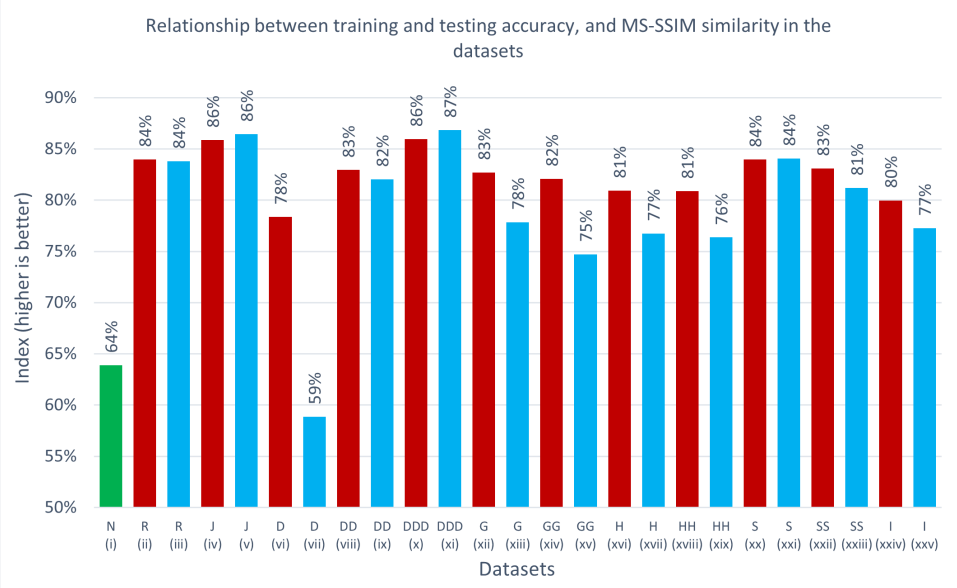


Fig. 16: Index that relates model training accuracy (higher is better), model testing accuracy on real data (higher is better), and the value of the MS-SSIM metric (lower is better). **Green** shape indicates real dataset, **red** indicate datasets with real and augmented data, and **turquoise** are entirely augmented datasets. The codes are described in Table VIII.

VII. Conclusion

In this work, several data augmentation techniques have been described, developed, and applied to seismic-volcanic signals. Initially, these techniques were employed to balance the datasets, and subsequently to generate entirely augmented datasets. Deep learning classification models were then trained on these datasets, utilizing transfer learning with the pre-trained AlexNet model. The performance of the trained classification models was evaluated, revealing that data augmentation techniques enhance model performance. It was observed that the trained classification models, alongside the considered data augmentation methods, are competitive with the baseline presented in previous research. Additionally, the following results were achieved:

- The integration of data from multiple stations on the Lascar volcano facilitated the construction of more generalized probabilistic models, thereby mitigating biases associated with individual stations through various data augmentation techniques.
- Data augmentation techniques significantly improve the performance metrics of deep learning models, as they enable the generation of additional examples within the feature space of volcanic seismic events.
- Similarity and diversity metrics were utilized to analyze the characteristics of the various data augmentation techniques presented.
- Transfer learning significantly expedited the generation of deep learning models compared to training from scratch.

The findings of this work can be used to enhance the performance of deep learning models for recognizing seismic-volcanic signals, taking into account the presented data augmentation techniques. Moreover, the results indicate that models trained exclusively on augmented data demonstrate strong performance when tested on real data, marking a substantial contribution to the field of seismic-volcanic signal recognition.

A repository is available on GitHub for data verification at: https://github.com/franznet/daseismicsignals.

VIII. Future Work

Future research could also focus on optimising the parameters of existing data augmentation methods to improve model performance. Furthermore, evaluating the effectiveness of these techniques on different types of seismic events and with varying levels of noise would provide deeper insight into their applicability. Finally, integrating these data augmentation strategies with other machine learning approaches, such as ensemble learning or reinforcement learning, could lead to the development of more robust and accurate models for seismic and volcanic signal recognition. A large dataset could be generated using various data augmentation techniques from different channels, stations, and volcanoes.

References

[1] P. Salazar, F. Yupanqui, C. Meneses, S. Layana, and G. Yáñez, "Multi-station automatic classification of seismic signatures from the lascar volcano database," *Natural Hazards and Earth System Sciences Discussions*, vol. 2022, pp. 1–27, 2022.

[2] A. Salazar, R. Arroyo, N. Pérez, and D. Benítez, "Deep-learning for volcanic seismic events classification," in *2020 IEEE Colombian Conference on Applications of Computational Intelligence (IEEE ColCACI 2020)*, IEEE, 2020, pp. 1–6.

[3] M. Malfante, M. Dalla Mura, J. I. Mars, J.-P. Métaxian, O. Macedo, and A. Inza, "Automatic classification of volcano seismic signatures," *Journal of Geophysical Research: Solid Earth*, vol. 123, no. 12, pp. 10–645, 2018.

[4] M. Bicego, C. Acosta-Munoz, and M. Orozco-Alzate, "Classification of seismic volcanic signals using hidden-markov-model-based generative embeddings," *IEEE transactions on geoscience and remote sensing*, vol. 51, no. 6, pp. 3400–3409, 2012.

[5] M. Curilem et al., "Pattern recognition applied to seismic signals of llaima volcano (chile): An evaluation of station-dependent classifiers," *Journal of Volcanology and Geothermal Research*, vol. 315, pp. 15–27, 2016.

[6] P. E. E. Lara et al., "Automatic multichannel volcano-seismic classification using machine learning and emd," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 1322–1331, 2020.

[7] M. Beyreuther, C. Hammer, J. Wassermann, M. Ohrnberger, and T. Megies, "Constructing a hidden markov model based earthquake detector: Application to induced seismicity," *Geophysical Journal International*, vol. 189, no. 1, pp. 602–610, 2012.

[8] R. Carniel, "Characterization of volcanic regimes and identification of significant transitions using geophysical data: A review," *Bulletin of Volcanology*, vol. 76, pp. 1–22, 2014.

[9] H. Langer, S. Falsaperla, M. Masotti, R. Campanini, S. Spampinato, and A. Messina, "Synopsis of supervised and unsupervised pattern classification techniques applied to volcanic tremor data at mt etna, italy," *Geophysical Journal International*, vol. 178, no. 2, pp. 1132–1144, 2009.

[10] F. Giacco, A. M. Esposito, S. Scarpetta, F. Giudicepietro, and M. Marinaro, "Support vector machines and mlp for automatic classification of seismic signals at stromboli volcano," in *Neural Nets WIRN09*, IOS Press, 2009, pp. 116–123.

[11] M. Curilem et al., "Improving the classification of volcanic seismic events extracting new seismic and speech features," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 22nd Iberoamerican Congress, CIARP 2017, Valparaíso, Chile, November 7–10, 2017, Proceedings 22*, Springer, 2018, pp. 177–185.

[12] M. Malfante, M. Dalla Mura, J.-P. Métaxian, J. I. Mars, O. Macedo, and A. Inza, "Machine learning for volcano-seismic signals: Challenges and perspectives," *IEEE Signal Processing Magazine*, vol. 35, no. 2, pp. 20–30, 2018.

[13] M. Titos, A. Bueno, L. García, C. Benítez, and J. C. Segura, "Classification of isolated volcano-seismic events based on inductive transfer learning," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 5, pp. 869–873, 2019.

[14] M. Curilem, J. P. Canário, L. Franco, and R. A. Rios, "Using cnn to classify spectrograms of seismic events from llaima volcano (chile)," in *2018 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2018, pp. 1–8.

[15] A. B. Rodriguez, C. Benitez, L. Zuccarello, S. De Angelis, and J. M. Ibáñez, "Bayesian monitoring of seismo-volcanic dynamics," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2021.

[16] A. Bueno, C. Benitez, S. De Angelis, A. D. Moreno, and J. M. Ibáñez, "Volcano-seismic transfer learning and uncertainty quantification with bayesian neural networks," *IEEE transactions on geoscience and remote sensing*, vol. 58, no. 2, pp. 892–902, 2019.

[17] A. A. T. Peixoto et al., "Tensor-based learning framework for automatic multichannel volcano-seismic classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 4517–4529, 2021.

[18] M. López-Pérez, L. García, C. Benítez, and R. Molina, "A contribution to deep learning approaches for automatic classification of volcano-seismic events: Deep gaussian processes," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 5, pp. 3875–3890, 2020.

[19] M. Bicego, J. M. Londoño-Bonilla, and M. Orozco-Alzate, "Volcano-seismic events classification using document classification strategies," in *Image Analysis and Processing—ICIAP 2015: 18th International Con-*

ference, Genoa, Italy, September 7-11, 2015, Proceedings, Part I 18, Springer, 2015, pp. 119–129.

[20] A. M. Esposito et al., "Unsupervised neural analysis of very-long-period events at stromboli volcano using the self-organizing maps," *Bulletin of the Seismological Society of America*, vol. 98, no. 5, pp. 2449–2459, 2008.

[21] A. Duque et al., "Exploring the unsupervised classification of seismic events of cotopaxi volcano," *Journal of Volcanology and Geothermal Research*, vol. 403, p. 107 009, 2020.

[22] F. Yupanqui-Machaca, "Clasificación de señales sísmico volcánicas mediante técnicas de transferencia de aprendizaje y aprendizaje profundo," Available at: https://ucn.primo.exlibrisgroup.com/discovery/fulldisplay?docid=alma990005547010703&context=L&vid=56UCN_NUI:UCN_lang=es, M.S. thesis, Universidad Católica del Norte, Avenida Angamos 0610, Antofagasta, Chile, 2021.

[23] A. Ferreira, M. Curilem, W. Gomez, and R. Rios, "Deep learning and multi-station classification of volcano-seismic events of the nevados del chillán volcanic complex (chile)," *Neural Computing and Applications*, vol. 35, no. 35, pp. 24 859–24 876, 2023.

[24] J. P. Canario, R. Mello, M. Curilem, F. Huenupan, and R. Rios, "In-depth comparison of deep artificial neural network architectures on seismic events classification," *Journal of Volcanology and Geothermal Research*, vol. 401, p. 106 881, 2020.

[25] M. Titos, A. Bueno, L. García, M. C. Benítez, and J. Ibañez, "Detection and classification of continuous volcano-seismic signals with recurrent neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 4, pp. 1936–1948, 2018.

[26] A. Maggi, V. Ferrazzini, C. Hibert, F. Beauducel, P. Boissier, and A. Amemoutou, "Implementation of a multistation approach for automated event classification at piton de la fournaise volcano," *Seismological Research Letters*, vol. 88, no. 3, pp. 878–891, 2017.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[28] M. Beyreuther, R. Barsch, L. Krischer, T. Megies, Y. Behr, and J. Wassermann, "Obspy: A python toolbox for seismology," *Seismological Research Letters*, vol. 81, no. 3, pp. 530–533, 2010.

[29] G. Iglesias, E. Talavera, Á. González-Prieto, A. Mozo, and S. Gómez-Canaval, "Data augmentation techniques in time series domain: A survey and taxonomy," *Neural Computing and Applications*, vol. 35, no. 14, pp. 10 123–10 145, 2023.

[30] B. K. Iwana and S. Uchida, "An empirical survey of data augmentation for time series classification with neural networks," *Plos one*, vol. 16, no. 7, e0254841, 2021.

[31] T. Fields, G. Hsieh, and J. Chenou, "Mitigating drift in time series data with noise augmentation," in *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, IEEE, 2019, pp. 227–230.

[32] P. Nethravathi and K. Karibasappa, "Augmentation of the customer's profile dataset using genetic algorithm," *International Journal of Research and Scientific Innovation (IJRSI), IV*, pp. 33–39, 2017.

[33] C. M. Villegas, J. L. Curinao, D. C. Aqueveque, J. Guerrero-Henríquez, and M. V. Matamala, "Data augmentation and hierarchical classification to support the diagnosis of neuropathies based on time series analysis," *Biomedical Signal Processing and Control*, vol. 95, p. 106 302, 2024.

[34] D. S. Park et al., "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[35] D. S. Park et al., "Specaugment on large scale datasets," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 6879–6883.

[36] M. Soni, A. Panda, and S. K. Kopparapu, "Generalized specaugment: Robust online augmentation technique for end-to-end automatic speech recognition," in *2024 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, IEEE, 2024, pp. 1–5.

[37] D. Berthelot, C. Raffel, A. Roy, and I. Goodfellow, "Understanding and improving interpolation in autoencoders via an adversarial regularizer," *arXiv preprint arXiv:1807.07543*, 2018.

[38] S. Yang, S. Guo, J. Zhao, and F. Shen, "Investigating the effectiveness of data augmentation from similarity and diversity: An empirical study," *Pattern Recognition*, vol. 148, p. 110 204, 2024.

[39] S. A. Güven, E. Şahin, and M. F. Talu, "Image-to-image translation with cnn based perceptual similarity metrics," *Computer Science*, vol. 9, no. 1, pp. 84–98, 2024.

[40] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[41] T. Kynkäänniemi, T. Karras, M. Aittala, T. Aila, and J. Lehtinen, *The role of imagenet classes in fréchet inception distance*, 2023. arXiv: 2203.06026 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2203.06026

[42] W. Wang, Y. Sun, and S. Halgamuge, *Improving mmd-gan training with repulsive loss function*, 2019. arXiv: 1812.09916 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1812.09916

[43] J. Kim, M. Kim, H. Kang, and K. Lee, "U-gat-it: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation," *arXiv preprint arXiv:1907.10830*, 2019.

[44] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying mmd gans," *arXiv preprint arXiv:1801.01401*, 2018.

[45] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assess-

ment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, Ieee, vol. 2, 2003, pp. 1398–1402.

[46] K. Ma et al., "Group mad competition-a new methodology to compare objective image quality models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1664–1673.

[47] M. Arsénio, R. Vigário, and A. M. Mota, "Recovering image quality in low-dose pediatric renal scintigraphy using deep learning," *Journal of Imaging*, vol. 11, no. 3, p. 88, 2025.

[48] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *International conference on machine learning*, PMLR, 2017, pp. 2642–2651.

[49] Y. Guo, Q. Chen, J. Chen, Q. Wu, Q. Shi, and M. Tan, "Auto-embedding generative adversarial networks for high resolution image synthesis," *IEEE Transactions on Multimedia*, vol. 21, no. 11, pp. 2726–2737, 2019. DOI: 10.1109/TMM.2019.2908352

[50] M. A.-N. I. Fahim and H. Y. Jung, "A lightweight gan network for large scale fingerprint generation," *IEEE Access*, vol. 8, pp. 92 918–92 928, 2020.

[51] D. Torbunov et al., "Rethinking cyclegan: Improving quality of gans for unpaired image-to-image translation," Mar. 2023. DOI: 10.48550/arXiv.2303.16280

Franz Yupanqui Machaca earned his Master's degree in Computer Engineering from the Universidad Católica del Norte in Chile in 2023. He is currently pursuing his Ph.D. in Sustainable Engineering at the Universidad Católica del Norte, focusing on the development of advanced algorithms for seismic-volcanic signal recognition. His research interests include deep learning techniques, transfer learning, data augmentation, computational intelligence in signals, data analytics, and more.

Pablo Salazar Reinoso Biography text here.

Claudio Meneses Villegas Biography text here.