Interspeech 2024
1-5 September 2024, Kos, Greece

Enhancing Speech and Music Discrimination Through the Integration of Static and Dynamic Features

Liangwei Chen[1], *Xiren Zhou*[1], *Qiang Tu*[2], *Huanhuan Chen*[1]

[1]University of Science and Technology of China, Hefei, China
[2]Anhui Provincial Hospital, Hefei, China

clw2018@mail.ustc.edu.cn, zhou0612@ustc.edu.cn, 78198019@qq.com, hchen@ustc.edu.cn

Abstract

Audio is inherently temporal data, where features extracted from each segment evolve over time, yielding dynamic traits. These dynamics, relative to the acoustic characteristics inherent in raw audio features, primarily serve as complementary aids for audio classification. This paper employs the reservoir computing model to fit the audio feature sequences efficiently, capturing feature-sequence dynamics into the readout models, and without the need for offline iterative training. Additionally, stacked autoencoders further integrate the extracted static features (i.e., raw audio features) with the captured dynamics, resulting in more stable and effective classification performance. The entire framework is called Static-Dynamic Integration Network (SDIN). The conducted experiments demonstrate the effectiveness of SDIN in speech-music classification tasks.

Index Terms: reservoir computing model, stacked autoencoder, speech-music classification, audio processing

1. Introduction

Classifying audio into specific categories, notably speech and music, serves as a preliminary step in audio indexing and retrieval. Manually annotating these two types of audio requires significant human effort and time, which has spurred research into automatic Speech-Music Classification (SMC) systems. The SMC task [1, 2] involves analyzing audio content to determine its belonging to either of the two fundamental categories. This paper focuses on employing the simple and efficient Static-Dynamic Integration Network (SDIN) to capture dynamic changes inherent in audio features, thereby enhancing the distinction between speech and music.

Numerous audio features have been utilized in SMC tasks, often encapsulating certain acoustic characteristics, especially the loudness and timbre. These two characteristics exhibit differences between speech and music, thus serving as discriminative cues. The time-domain feature, Root-Mean-Square Energy (RMS) [3], provides insight into the loudness by characterizing the envelope of the waveform. Besides, Zero-Crossing Rate (ZCR) [4] gauges the rate of change in the audio waveform and is often paired with other features for a comprehensive audio description. In the realm of frequency domain features, the Mel-Frequency Cepstral Coefficients (MFCCs) [5] emerge as a prevalent choice. While the lower-dimensional MFCCs represent the spectral envelope, their higher-dimensional counterparts delve into intricate spectral details, enriching the portrayal of the audio's timbre.

Audio exhibits variations within each segment over time, leading to dynamic information in the corresponding features. These dynamics reflect alterations in acoustic characteristics and could potentially offer valuable cues for distinguishing between speech and music. Existing studies have transformed audio into image forms, such as spectrograms, with the x-axis denoting time and the y-axis indicating frequency, encapsulating dynamic spectral information. Vision-based models, including Convolutional Neural Networks (CNNs) [6] and Vision Transformers (ViTs) [7], have been adapted for audio classification in light of this. Nevertheless, challenges persist: 1) Transposing audio into spectrograms might inadvertently omit some temporal variations present in the original audio format; 2)Additionally, the absence of connections between audio features and their inherent dynamics could hinder the full exploitation of their distinctive capabilities for SMC.

In this study, statistical metrics (i.e., mean, variance, etc.) of audio features such as MFCCs, RMS, and ZCR are referred to as audio's "*static features*". On the other hand, the contextual information arising from the temporal variations of these features is designated as the audio's "*dynamic features*". As depicted in Figure 1, the proposed Static-Dynamic Integration Network (SDIN), leverages reservoir computing [8] for the efficient capture of these dynamic features. To elaborate, SDIN scans the audio with a fixed-size window, extracting audio features within each window. These features are arranged in sequential, hereinafter referred to as feature sequences. Subsequently, each feature sequence is fitted by reservoir computing to capture its inherent dynamic features, and represented using the fitted readout model [9], all without the need for offline iterative training. Previous studies have demonstrated that temporal signal or data could be more effectively discerned in the "dynamic feature space" [10, 11, 12], suggesting that the dynamic variation of sequences could be efficiently captured and well represented by the fitted model.

One distinctiveness of audio lies in the contribution of its static features (that portray acoustic characteristics) to audio discrimination [13], surpassing the reliance solely on dynamically evolving information. Thus, SDIN emphasizes effectively integrating both the audio's static and dynamic features to achieve superior classification. SDIN utilizes two Stacked Autoencoders (SAEs) [14] to perform dimensionality reduction on static and dynamic features separately. Subsequently, it integrates the information from both aspects to derive the final classification result. This approach ensures the retention of both the acoustic characteristics of the audio and its variation information. Experimental findings indicate that SDIN is comparable to state-of-the-art methods in SMC tasks.

The main contributions are summarized as follows:

- Efficiently capturing the dynamic feature of audio without the need for iterative training, mitigating dependency on extensive computational resources and data.
- Integrating audio static features with their dynamic features, fully leveraging the discriminative information they provide

10.21437/Interspeech.2024-1596

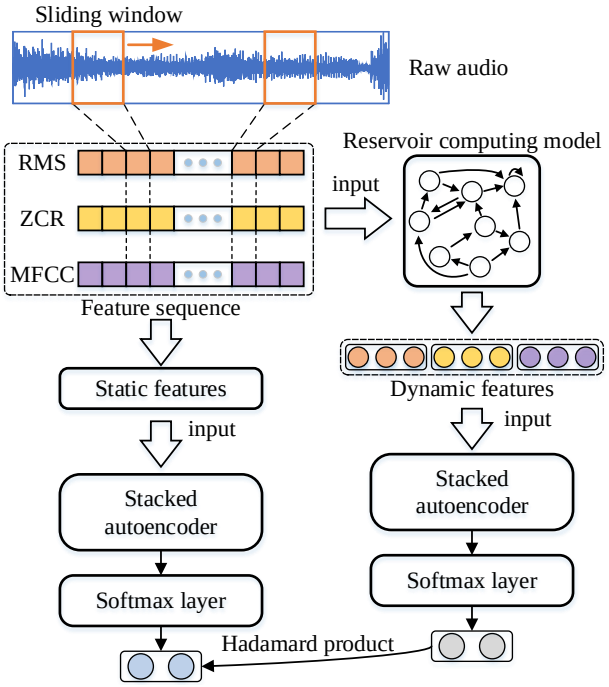

Figure 1: *The structure of SDIN. We scan the raw audio using a sliding window, where audio features within each window are extracted, and sequentially organized.*

for classification tasks.

- The introduced SDIN boasts flexibility, adeptly processing audio of diverse lengths by swiping each audio with a fixed-size window.

2. Methodology

This section presents SDIN in three parts. The first part introduces the selected static audio features, followed by an explanation in the second part of how the reservoir computing model is used to capture dynamic features. The final part elaborates on the process of integrating static and dynamic features using SAEs.

2.1. Static Features and Feature Sequences

Speech and music exhibit differences in acoustic characteristics including loudness and timbre. This paper focuses on audio features that well represent these characteristics, including Root-Mean-Square Energy (RMS), Zero-Crossing Rate (ZCR), and Mel-Frequency Cepstral Coefficients (MFCCs). 1) RMS characterizes the envelope of the audio signal in the time domain and exhibits greater resistance to outliers compared to the amplitude envelope. It reflects the loudness of the audio. 2) ZCR is used to measure the rate of change and noise level in the audio signal. It is often combined with other features to provide a comprehensive description of audio characteristics, such as when combined with MFCCs to represent audio timbre. 3) MFCCs describe the distribution of energy in the sound signal across different frequency ranges. Lower-dimensional MFCCs correspond to the envelope information of the spectrum, while higher-dimensional MFCCs capture spectral details.

Referencing Figure 1, within each sliding window, the extraction of the trio of features mentioned above takes place.

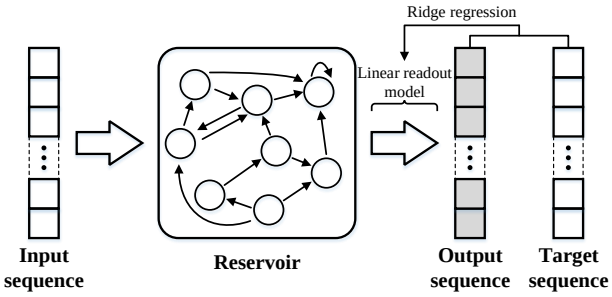

Figure 2: *The structure of ESN. The ESN includes an input layer, a reservoir, and an output layer. The input layer in this paper is the feature sequence of the audio. This work employs the next-item prediction task to train the ESN.*

Subsequently, the means of these features are computed, bestowing us with the audio's static features. Also, organizing these features sequentially produces what we term as "feature sequences", further fitted to capture their dynamic features.

2.2. Capturing Dynamic Features within Each Feature Sequence

The Echo State Network (ESN) [15, 16] is used to fit each feature sequence, for two main reasons: 1) Its efficiency, eliminating the need for iterative training like gradient descent, and relying merely on ridge regression [17] for fitting; 2) As a reservoir computing model [18], ESN's proven capability in fitting diverse nonlinear sequential data and capturing their inherent dynamic features, making it apt for time-series classification and representations in areas like healthcare and industry [19, 20].

As shown in Figure 2, ESN consists of an input layer, a reservoir, and an output layer. The reservoir is a high-dimensional and sparse recurrent structure, with its architecture and the weights determined randomly based on Echo State Properties [8]. For a sequence a, assuming the input at time/item t is denoted as $a(t) \in \mathbb{R}^n$, the equations for its processing through the reservoir and the output layer can be expressed as follows:

$$\begin{cases} r(t) = f(W_{res}r(t-1) + W_{in}a(t)), \\ g(t) = W_{out}r(t) + d = F(r(t)), \end{cases} \tag{1}$$

where $r(t) \in \mathbb{R}^m$ is the reservoir state at time t, $g(t) \in \mathbb{R}^l$ is the output. $W_{in} \in \mathbb{R}^{m\times n}$ represents the input weight matrix, $W_{res} \in \mathbb{R}^{m\times m}$ is the reservoir weight matrix. $W_{out} \in \mathbb{R}^{l\times m}$ is the output weight matrix (also the readout model to be computed), and $d \in \mathbb{R}^l$ is a bias term. f is a non-linear activation function. In ESN, W_{in} and W_{res} are randomly initialized, and the only weight matrix that needs to be computed is W_{out}.

To capture the dynamic features within the sequence, the "next-item prediction" task [9] is employed to solve the readout model W_{out}. Specifically, since the feature sequence is constructed using a sliding window, the input vector at t, denoted as $a(t)$, corresponds to the features of the t-th window. Assuming the input feature sequence is represented as:

$$a_{L_0\sim L-1} = \{a(L_0), a(L_0+1), ..., a(L-1)\}, \tag{2}$$

each $a(t)$ is fed into the reservoir through Equation (1), obtaining $r(t)$. Based on the relationship between $r(t)$ and $g(t)$ as

described in Equation (1), the output sequence g can be represented[1] as follows:

$$g_{L_0 \sim L-1} = \{W_{out} r(L_0), W_{out} r(L_0+1), ..., W_{out} r(L-1)\}. \tag{3}$$

For the next-item prediction task, ridge regression is eventually performed using the output sequence g and the input sequence $a_{L_0+1 \sim L}$:

$$W_{out} = \arg\min_{W_{out}} \sum_{t=L_0}^{L-1} \|W_{out} r(t) - a(t+1)\|^2 + \lambda \|W_{out}\|^2, \tag{4}$$

where λ is a non-negative regularization parameter. W_{out} is used to characterize the dynamic information of the feature sequence. Using ridge regression, W_{out} can be calculated:

$$W_{out} = (R^T R + \lambda I)^{-1} R^T G, \tag{5}$$

where R is a matrix of the reservoir states, I is the identity matrix. G is a matrix of the target value, composed of all the values from $a_{L_0+1 \sim L}$.

It is worth noting that directly calculating the distance between the parameters of the readout model depends on the specific model parameters and merely reflects the distance between the model parameters. In this paper, we denote the readout model corresponding to the sequence a_i as $F_i(r)$. For two distinct sequences a_1 and a_2 processed through the same reservoir, the resulting readouts are denoted as $F_1(r)$ and $F_2(r)$, respectively. As shown in the following equation:

$$\begin{aligned} F_1(r) &= W_{out_1} r + d_1, \\ F_2(r) &= W_{out_2} r + d_2, \end{aligned} \tag{6}$$

where $r \in \mathbb{R}^m$ represents the state vector, and $d \in \mathbb{R}^l$ denotes the bias vector. W_{out_1} and W_{out_2} are the respective output weight matrices within the ESN framework. The L_2 distance between them [9] in the dynamic feature space can be calculated as

$$L_2(F_1, F_2) = \left(\int_{\mathbb{D}} \|F_1(r) - F_2(r)\|^2 d\mu(r) \right)^{\frac{1}{2}}, \tag{7}$$

where $\mu(r)$ denotes the probability density function for r, with $\mathbb{D}$ representing the domain of integration. Given that f in Equation (1) is the tanh function, it follows that $\mathbb{D} = [-1, +1]^m$. Under the assumption of a uniform distribution for r, Chen *et al.* [9] elaborate on a scaling method to ascertain the squared distance between F_1 and F_2 as follows:

$$\frac{1}{3} \sum_{j=1}^{m} \sum_{i=1}^{l} w_{i,j}^2 + \|d\|^2,$$

where $w_{i,j}$ refers to the element at position (i, j) in $W = W_{out_1} - W_{out_2}$, and d is calculated as $d_1 - d_2$. This equation allows the calculation of the model distance. With this foundation, it is possible to apply distance-based classification techniques within the dynamic feature space.

Through the above process, the dynamic features within the feature sequences are captured into the readout models, denoted as D-RMS, D-ZCR, and D-MFCCs, respectively.

[1]The bias term d can be incorporated into r, so it is not explicitly written out separately.

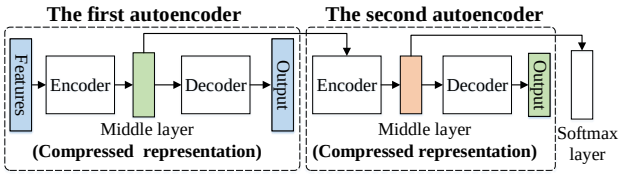

Figure 3: *This figure illustrates the structure of the employed SAEs, consisting of two autoencoders and a softmax layer. These three components are trained separately.*

2.3. Integrating Static and Dynamic Features

SDIN uses Stacked Autoencoders (SAEs) [14] to process both static and dynamic features, enhancing audio classification. For instance, when audio samples exhibit low volume or blurring, relying solely on static features like loudness and timbre might fall short. Dynamic features provide additional context, aiding in distinguishing between speech and music. Hence, integrating both static and dynamic features could be essential for precise audio classification.

SAEs offer the capability to encode complex multi-dimensional features into compressed representations [21]. We leverage this property to process static features like RMS, ZCR, and MFCCs, as well as the corresponding dynamic features captured. Figure 3 provides a schematic of our two-layered autoencoder structure. The input of the first autoencoder is either the static or dynamic features:

$$\text{input} = \begin{cases} [\text{RMS}, \text{ZCR}, \text{MFCCs}], \\ [\text{D-RMS}, \text{D-ZCR}, \text{D-MFCCs}]. \end{cases} \tag{8}$$

The first autoencoder compresses the input features, and its middle layer output is used as an input to the second autoencoder. The final compressed representation from the second autoencoder is directed into a softmax layer, producing probability distributions for classification[2].

As illustrated in Figure 3, both static and dynamic features are individually processed through the SAEs. From this processing, we derive two probability distributions: P_{static} for static features and P_{dynamic} for dynamic features. To amalgamate the insights from both these features, we compute the Hadamard product [22] of P_{dynamic} and P_{static}:

$$P = P_{\text{static}} \circ P_{\text{dynamic}}, \tag{9}$$

where P is subsequently normalized to serve as the conclusive probability distribution used for classification tasks. Experimental results corroborate that the proposed SDIN proficiently amalgamates the static and dynamic features in SMC tasks, leading to an enhanced classification performance.

3. Experiments

This section tests SDIN on SMC tasks using two public datasets. The experiments focus on: 1) Showcasing SDIN's proficiency in amalgamating audio static features with their dynamics for enhanced classification. 2) Verifying the Reservoir Computing Model's (i.e., the ESN's) ability to capture and represent the dynamic features.

[2]Each autoencoder layer is trained individually in an unsupervised manner using the cross-entropy loss function.

Table 1: *Comparison of SDIN with state-of-the-art methods in terms of accuracy(%).*

Method	GTZAN	MUSAN
chromagram visual features [24]	96.88	98.48
chromagram spectral features [24]	87.50	95.44
Visual features+SAEs [25]	94.73	98.07
SDIN	**99.17**	**98.89**

Table 2: *Comparison of SDIN with state-of-the-art methods in terms of F1-score(%).*

Method	GTZAN	MUSAN
Papakostas-CNN [26]	89.76±3.16	**99.36±0.76**
MSD-ASPT-LSPT [27]	94.17±2.19	98.10±0.05
CBoW-ASPT-LSPT [27]	95.25±2.24	98.99±0.04
SDIN	**99.18±3.41**	99.03±0.77

3.1. The Utilized Datasets

3.1.1. The GTZAN Dataset

The GTZAN dataset [23] comprises 120 songs and is designed for speech/music discrimination. Each category consists of 64 tracks (audios), and each track has a fixed length of 30 seconds.

3.1.2. The MUSAN Dataset

The MUSAN dataset [5] is more extensive, consisting of over 100 hours of audio data, distributed across 426 speech tracks and 660 music tracks, all of varying lengths.

3.2. Specific Parameters and Environment

The MATLAB environment utilized the voicebox[3] and MIRtoolbox[4] toolkits. The reservoir size and spectral radius of the ESN are set to 50 and 0.8, respectively. The number of trees in the random forest classifier is chosen as 100. The value range of the ridge regression parameter λ is $\{10^{-5}, 10^{-4}, \ldots, 10^{1}\}$.

3.3. Baseline Methods

Both accuracy and F1-score metrics are utilized to evaluate the model's classification performance. This paper selected several methods for comparison. 1) In the accuracy comparison section: Music specific chromagram representation was employed for SMC in [24], and Kumar et al. [25] introduced the application of stacked auto-encoders for speech/music discrimination. 2) In the F1-score comparison section: The key contributions revolved around CNNs and transfer learning for training audio classifiers [26], and two time-frequency features were proposed in [27] for SMC. The experiment employs 10-fold cross-validation to obtain the results.

3.4. Experimental Results and Specific Analysis

Tables 1 and 2 showcase SDIN's classification performance in the SMC task against several state-of-the-art methods. SDIN demonstrates prominent accuracy and F1-score values, emphasizing its capability to seamlessly integrate static and dynamic audio features for classification.

[3]http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html

[4]https://www.jyu.fi/hytk/fi/laitokset/mutku/en/research/materials/mirtoolbox

Table 3: *The classification results obtained by combining static and dynamic features separately with SAEs on GTZAN.*

Method	Accuracy	F1-score
Static features+SAEs	97.63±3.82	97.46±4.11
Dynamic features+SAEs	88.34±8.38	87.74±9.01

Table 4: *The classification results obtained by combining static and dynamic features separately with SAEs on MUSAN.*

Method	Accuracy	F1-score
Static features+SAEs	96.22±1.97	96.05±2.03
Dynamic features+SAEs	95.95±1.38	95.71±1.60

In Table 1, the method cited as [25] also employs SAEs for the SMC task, taking audio's time-frequency representations (like spectrograms and chromagrams) as inputs. However, SDIN surpasses its accuracy on the GTZAN dataset by several percentage points. One reason might be the potential loss of temporal details when audio is converted into spectrograms or chromagrams. SDIN, on the other hand, effectively captures dynamic features by considering both time-domain (D-RMS and D-ZCR) and frequency-domain (D-MFCCs) variations. Such comprehensive features prove particularly potent for discerning speech from music, especially in audios with reduced volume or muddled properties.

While examining Table 2, we notice CNNs outperforming on the MUSAN dataset compared to their results on GTZAN. This can be explained by the sheer volume of samples in MUSAN versus GTZAN. Deep learning models typically thrive with more extensive data for training. However, SDIN, utilizing ESN and the SAEs, benefits from the inherent advantages of these models. Specifically, ESN sidesteps the need for offline iterative training, while the SAEs are kept lean with just two layers. SDIN, therefore, concentrates on inherent audio characteristics and their inside variations, without demanding hefty computational power or vast datasets. Consequently, its performance remains fairly consistent regardless of the dataset's size.

To evaluate the enhancement in SMC performance by integrating dynamic and static features, experiments were conducted using each feature set individually in conjunction with SAEs. Tables 3 and 4 display results from both datasets. When paired with SAEs, either dynamic or static features demonstrate competence in discerning between speech and music. Static features showcased superior classification performance, while dynamic features fell slightly short. This reiterates that inherent acoustic characteristics are more pivotal in differentiating between speech and music than their temporal variations. Yet, when both static and dynamic features are amalgamated, they outperform the results achieved using them individually.

4. Conclusion

The Static-Dynamic Integration Network (SDIN), integrates audio's static and dynamic features effectively, enhancing their discriminative power for classification. Experimental results demonstrated that SDIN performs comparably to state-of-the-art methods in SMC tasks. Overall, SDIN provides a robust approach to SMC, with potential applications in multimedia content analysis and retrieval, contributing to the advancement of audio classification technology.

5. Acknowledgments

This research is supported by National Key R&D Program of China (No. 2021ZD0111700), National Nature Science Foundation of China (No. 62137002, 62176245).

6. References

[1] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," in *1997 IEEE international conference on acoustics, speech, and signal processing*, vol. 2. IEEE, 1997, pp. 1331–1334.

[2] Y. Lavner and D. Ruinskiy, "A decision-tree-based algorithm for speech/music classification and segmentation," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2009, pp. 1–14, 2009.

[3] G. Sell and P. Clark, "Music tonality features for speech/music discrimination," in *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2014, pp. 2489–2493.

[4] M. Bhattacharjee, S. M. Prasanna, and P. Guha, "Time-frequency audio features for speech-music classification," *arXiv preprint arXiv:1811.01222*, 2018.

[5] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[6] P. Lopez-Meyer, J. A. del Hoyo Ontiveros, H. Lu, and G. Stemmer, "Efficient end-to-end audio embeddings generation for audio classification on target applications," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 601–605.

[7] Y. Gong, C.-I. Lai, Y.-A. Chung, and J. Glass, "Ssast: Self-supervised audio spectrogram transformer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 10, 2022, pp. 10 699–10 709.

[8] H. Jaeger, "Adaptive nonlinear system identification with echo state networks," *Advances in neural information processing systems*, vol. 15, 2002.

[9] H. Chen, P. Tiňo, A. Rodan, and X. Yao, "Learning in the model space for cognitive fault diagnosis," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 1, pp. 124–136, 2013.

[10] H. Chen, F. Tang, P. Tino, and X. Yao, "Model-based kernel for efficient time series analysis," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 392–400.

[11] H. Chen, F. Tang, P. Tino, A. G. Cohn, and X. Yao, "Model metric co-learning for time series classification." in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*. AAAI Press, 2015, pp. 3387–3394.

[12] L. Chen, X. Zhou, and H. Chen, "Audio scanning network: Bridging time and frequency domains for audio classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 10, 2024, pp. 11 355–11 363.

[13] L. Nanni, Y. M. Costa, D. R. Lucio, C. N. Silla Jr, and S. Brahnam, "Combining visual and acoustic features for audio classification tasks," *Pattern Recognition Letters*, vol. 88, pp. 49–56, 2017.

[14] J. Zabalza, J. Ren, J. Zheng, H. Zhao, C. Qing, Z. Yang, P. Du, and S. Marshall, "Novel segmented stacked autoencoder for effective dimensionality reduction and feature extraction in hyperspectral imaging," *Neurocomputing*, vol. 185, pp. 1–10, 2016.

[15] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note," *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, vol. 148, no. 34, p. 13, 2001.

[16] A. Rodan and P. Tino, "Minimum complexity echo state network," *IEEE transactions on neural networks*, vol. 22, no. 1, pp. 131–144, 2010.

[17] A. E. Hoerl and R. W. Kennard, "Ridge regression: applications to nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 69–82, 1970.

[18] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer science review*, vol. 3, no. 3, pp. 127–149, 2009.

[19] Z. Gong, H. Chen, B. Yuan, and X. Yao, "Multiobjective learning in the model space for time series classification," *IEEE transactions on cybernetics*, vol. 49, no. 3, pp. 918–932, 2018.

[20] X. Zhou, S. Liu, A. Chen, Q. Chen, F. Xiong, Y. Wang, and H. Chen, "Underground anomaly detection in gpr data by learning in the c3 model space," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–11, 2023.

[21] G. Liu, H. Bao, and B. Han, "A stacked autoencoder-based deep neural network for achieving gearbox fault diagnosis," *Mathematical Problems in Engineering*, vol. 2018, pp. 1–10, 2018.

[22] G. P. Styan, "Hadamard products and multivariate statistical analysis," *Linear algebra and its applications*, vol. 6, pp. 217–240, 1973.

[23] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on speech and audio processing*, vol. 10, no. 5, pp. 293–302, 2002.

[24] G. K. Birajdar and M. D. Patil, "Speech/music classification using visual and spectral chromagram features," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 329–347, 2020.

[25] A. Kumar, S. S. Solanki, and M. Chandra, "Stacked auto-encoders based visual features for speech/music classification," *Expert Systems with Applications*, vol. 208, p. 118041, 2022.

[26] M. Papakostas and T. Giannakopoulos, "Speech-music discrimination using deep visual feature extractors," *Expert Systems with Applications*, vol. 114, pp. 334–344, 2018.

[27] M. Bhattacharjee, S. M. Prasanna, and P. Guha, "Speech/music classification using features from spectral peaks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1549–1559, 2020.