Generative AI for Short Sound Message Transmission in the Internet of Things

Manuele Favero*, Alessandro Buratto*, Leonardo Badia*, Sergio Canazza*, and Luciano Murrone†
*Dept. of Information Engineering, University of Padova, via Gradenigo 6/b, 35131 Padua, Italy
† Ogenus Srl, via Uruguay 20, 35127 Padua, Italy
Email: {faveromanu,burattoale,badia,canazza}@dei.unipd.it, luciano.murrone@ogenus.it

***Abstract*—We leverage the latest advancements in generative AI for music creation to develop an automated system producing short sound messages. These sound-based messages, referred to as Transmit In Sound code (TIScode), are brief audio sequences lasting 5 seconds that carrying digital information. They can be recognized by a specific smartphone application in an Internet of Audio Things (IoAuT) scenario. We describe the methodologies of the TIScode pipeline, which includes generation, transmission, and ultimately, reception and decoding. For the generation phase, we use MusicGen, a state-of-the-art autoregressive transformer model, and we introduce a channel coding system based on the quantization of sound features and high-level features extracted through convolutional neural networks (CNNs). The extracted features are mapped to create a unique bitmap for each TIScode, simplifying the decoding process. We present an algorithm for the recognition phase, combining sound feature analysis with frequency-based peak analysis to enhance detection accuracy. Experimental results, obtained through simulation and field tests, demonstrate the effectiveness of the system in retrieving the digital information encoded within sound messages.**

***Index Terms*—Generative AI, Internet of Audio Things, Audio Classification, Channel Coding, Digital Signal Processing**

I. Introduction

The convergence of IoT and audio technologies has led to the emergence of innovative concepts such as the Internet of Audio Things (IoAuT) [1]. These paradigms integrate devices capable of producing, analyzing, and transmitting audio data in real time, enabling a wide range of applications.

The concept of TIScode (Transmit In Sound Code) emerges from the fusion of these technologies [2]. TIScode is a distinctive jingle that blends audio elements from various genres and styles, creating a diverse and dynamic sound. Each TIScode last 5 seconds and it is capable of carrying information. At the time of its creation, a digital piece of information is assigned to it, which can later be retrieved once the TIScode is decoded by a smartphone application. This process occurs seamlessly, without requiring the user to unlock the device, turn on the recorder, or activate the camera. The app detects an opening marker, a brief sound preceding the actual audio containing the information, which triggers autonomous recording. Upon unlocking the device, the user gains access to the digital content embedded within the TIScode.

This technology has numerous applications, ranging from transmitting information through radio broadcasts in cars, to receiving ambient information in a context-rich environment [2]. In advertisement, TIScode integrates with radio/TV broadcast as well as movies or concerts to deliver promotional content directly to personal mobile devices [3]. In localized marketing, it facilitates targeted promotions in shopping centers and malls. Events and exhibitions benefit from digital catalogs and real-time updates via ambient acoustic signals [4]. In hospitality and tourism, the system provides automatic updates on restaurants and accommodations [5]. Museums and cultural institutions can use it for interactive tours and exhibits. In transportation hubs, it can deliver real-time travel updates [6].

TIScode also supports healthcare, providing regulatory information and patient instructions, and education, enabling seamless distribution of institutional announcements. It enhances accessibility for visually impaired users and those with disabilities, ensuring inclusive communication [7]. TIScode enables automated, passive interaction, making it a powerful tool for contextual and location-based information dissemination [8]. It can also be useful for new types of multiple-factor authentication that differ from usual visual codes [9], without aiming the camera at a screen and with automatic handling and directional tracking, ensuring a new layer of security [10].

The IoAuT is an emerging research field where AI-powered sound recognition technologies are advancing rapidly. These developments enable applications such as urban noise monitoring, environmental surveillance [11], anomaly detection [12], and data sonification [13], which involves converting information into sound. Semantic audio technologies further enhance interoperability by extracting structured data from audio signals, while web-based tools facilitate real-time audio processing for interactive and distributed applications [14].

Many of these applications rely on field-programmable gate arrays (FPGAs), which are power-efficient and highly suitable for computationally intensive tasks like convolutional neural networks (CNNs) [15]. Some works also consider the use of transformer-based architectures for sound recognition and classification tasks, as done in [16] with the introduction of a multi-resolution attention mechanism for audio samples. Other works employ transformers to improve the classification in the presence of considerable ambient noise [17]. In this paper, we outline the process of generating TIScodes using existing Generative AI algorithms. We also introduce a feature selection and coding algorithm designed to maximize the

This work has been supported by the Italian PRIN project 2022PNRR "DIGIT4CIRCLE," project code P2022788KK.

accuracy of TIScode decoding. To validate our approach, we present performance evaluations based on both simulations and real-world experiments. The practical implementation of IoAuT-related technologies remains relatively unexplored in the existing literature, highlighting the novelty and significance of our contribution.

The reminder of this paper is organized as follows: in Section II, we introduce the role of Generative AI in the creation of TIScodes. Section III covers the methodologies necessary for TIScode channel coding. In Section IV, we present and analyze the results of these techniques, demonstrating the performance of our system's channel coding. Additionally, we highlight the effectiveness of various AI techniques in both generating and classifying short sound messages. Finally, we conclude in Section V.

II. Sound Message Transmission

The use of sound messages for communication is common in various contexts, such as in wireless acoustic sensor networks, where low-power nodes with microphones enable event detection through sound classification and localization [3]. Similarly, sonification employs microphones in production machines to detect anomalies by analyzing acoustic signals, with classification performed via cloud-based machine learning [13]. Here, we refer to short sound messages as TIScodes, a patented technology by Ogenus S.R.L. These are brief 5-second jingles received by IoAuT devices, such as smartphones and classified through a cloud-based ML algorithm. Once classified, each jingle is mapped to a pointer in a database, allowing retrieval of the associated digital information [2].

Fig. 1 shows the TIScode pipeline, comprising:

Generation: the system generates a prompt and uses it as input for MusicGen [18], which creates a short musical piece paired with the digital information to transmit.

Channel Encoding: the system creates a bitmap that uniquely identifies the TIScode using various sound features. It saves the bitmaps in a database along with an analysis of the frequency peaks.

Transmission: users can now access the TIScode and decide when and where to transmit it.

Channel Decoding: a device receives the TIScode and decodes it using a minimum Hamming distance encoding. It enhances recognition by verifying the frequency peaks.

Information Retrieval: the system matches the TIScode to create a pointer to the database, allowing it to retrieve the originally paired digital information.

Managing a large number of short sound messages is essential for a TIScode platform. Manually generating each sound is impractical and may not provide sufficient variability, especially considering that each TIScode is only 5 seconds long. Sound generation must also ensure maximum diversity among TIScodes. To address these challenges, we leverage MusicGen by Meta, an autoregressive transformer-based model capable of generating high-quality audio from text descriptions and melodic cues [18]. MusicGen transformer process sequences of tokens $(x_1, ..., x_T)$ of arbitrary length T. Transformer first embeds the tokens to obtain a sequence $(e_1, ..., e_T)$. It is a succession of blocks with residual connections. Each block is made of the composition of a multi-head self-attention module and a multi-layer perceptron. Importantly, the latter acts on each token separately, whereas multi-head self-attention mixes tokens, and corresponds to applying vanilla self-attention in parallel [19]. More precisely, each multi-head self attention is parametrized by a collection of weight matrices representing query, key, value and output $(Q^h, K^h, V^h, O^h)_{1 \leq h \leq H}$ and returns:

$$\left(\sum_{h=1}^{H} O^h \sum_{t'=1}^{t} A^h_{t,t'} V^h e_{t'} \right)_{t \in \{1,\ldots,T\}} \tag{1}$$

where $A^h_{t,t'}$ is the attention matrix defined as [20]:

$$A^h_{t,t'} = \text{softmax}\left(Q^h e_t, K^h e_{t'}\right) \ . \tag{2}$$

The token system of MusicGen represents discrete musical units allowing to model complex musical structures and generate results that align with user requests. In this generation process we can manage degrees of freedom related to: 16 different genres (pop, rock, blues...), 8 different production types (Lo-Fi, 8-bit...), 16 different musical instruments (acoustic guitar, electric guitar, flute...), 4 tempo settings (slow, medium, moderate, fast), the key of the track and 8 different moods (energetic, relaxing, intense...). This gives approximately 20 bits for generating audio tracks within this dataset, enabling the creation of $2^{20} \simeq 10^6$ different TIScodes. The dataset generated with MusicGen that will be used for all tests conducted in this work consists of 1000 publicly available audio tracks.[1] These tracks were generated with a maximum token limit of 512 per track, which translates to a duration of about 5 seconds. One hundred of these tracks were generated using the *unconditional generation* function, which involves MusicGen generating random pieces without any input. These were created both to test this type of generation and to introduce more variability into the dataset to stress the tests described later. The remaining 900 tracks were generated using 900 text AI-generated prompts. The prompts are structured following this template:

Genre + *Production Type* + *One or more instruments* +
Mood + *Tempo setting* + *Key* .

We summarize in Tab. I all the information on the dataset.

III. Channel Coding in TIScode

Differently from typical redundancy coming from CRC or FEC codes, we leverage the correlation and inter-dependencies of multiple audio features. The idea is to analyze features and characteristics of our TIScode, then quantize them to create a bitmap for efficient channel encoding and decoding. We extract the features using Librosa [21] and MIRtoolbox [22], which are platforms dedicated to the extraction from audio files of

[1] https://www.kaggle.com/datasets/manuelefavero/tiscode-dataset

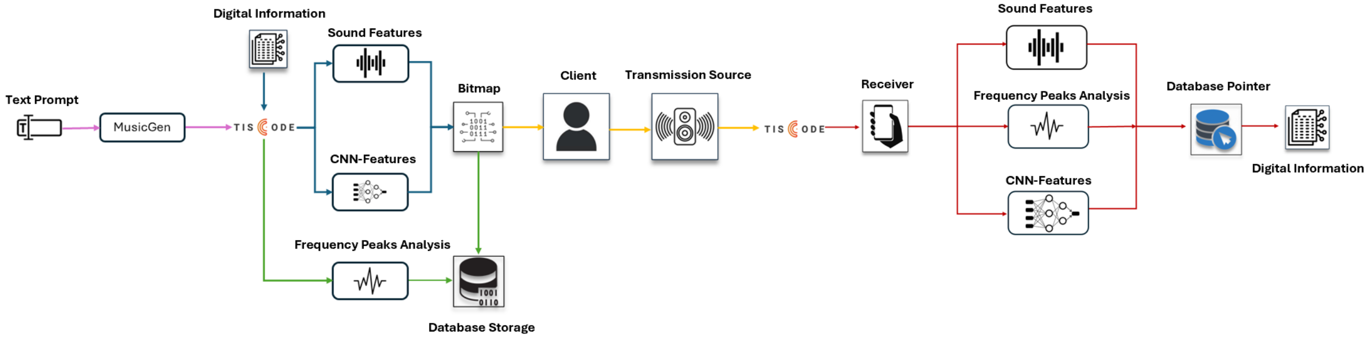

Fig. 1. TIScode Pipeline

TABLE I
TISCODE DATASET SUMMARY

Data	Description
Conditioned Generated Tracks	900
Unconditioned Generated Tracks	100
Audio Format	Mono
Extension Format	.WAV
Token Length	512
Time Length	5 seconds
Min. Degree of Freedom	20 bits
Available combinations	1 048 576

musical features written, respectively, in Python and Matlab. We report the analyzed features in Tab. II.

We selected features that are robust to noise while also able to characterize the TIScode from a perceptual sound perspective. This ensures that differences in the generated bitmaps correspond to perceptual differences in sound. Audio features play a crucial role in characterizing sound by capturing its tonal, rhythmic, and textural properties. For instance, chroma features identify dominant pitch classes, helping to determine the key and harmonic structure, while the tonal centroid represents tonal relationships in a multidimensional space. Spectral contrast distinguishes between harmonic and percussive elements, shaping timbre perception. ZCR and roughness provide insights into signal texture, differentiating smooth from percussive or dissonant sounds. Together, these features enable comprehending musical and audio content.

Among all these features, three are derived using a CNN: Key, Genre, and Top Instrument. Indeed, correctly recognizing the key in the presence of high noise is challenging, thus we used a CNN that is more resilient to noise than traditional methods. Specifically, we applied Convolutional Representation for Pitch Estimation (CREPE), a state-of-the-art tool for pitch detection [15], [23]. CREPE consists of a deep convolutional neural network which operates directly on the time-domain audio signal to produce a pitch estimate. The architecture consist of six convolutional layers that result in a 2048-dimensional latent representation, which is then connected densely to the output layer with sigmoid activations corresponding to a 360-dimensional output vector $\hat{y}$. Each dimension in the output vector represents a frequency bin covering 20 cents (a unit representing musical intervals relative to a reference pitch) [23].

To soften the penalty for near-correct predictions, the target is Gaussian-blurred in frequency, such that the energy surrounding a ground truth frequency decays with a standard deviation of 25 cents:

$$y_i = \exp\left(-\frac{(\check{c}_i - \check{c}_{\text{true}})^2}{2 \cdot 25^2}\right), \tag{3}$$

This way, high activations in the last layer indicate that the input signal is likely to have a pitch that is close to the target pitches of the nodes with high activations [15].

We classify Genre and Top-Instrument using wav2vec [20], a convolutional feature encoder trained on different datasets [24], [25]. The encoder comprises blocks with temporal convolutions, layer normalization, and GELU activation. The output is fed into a transformer-based context network. In self-supervised training, the encoder output z is discretized using product quantization, selecting representations from G codebooks with V entries, concatenating them, and applying a linear transformation [20]. The Gumbel softmax enables choosing discrete codebook entries in a fully differentiable way. We used the straight-through estimator with G hard Gumbel softmax operations [26]. The feature encoder output z is mapped to $l \in \mathbb{R}^{G \times V}$ logits and the probabilities for choosing the v-th codebook entry for group g are

$$p_{g,v} = \frac{\exp(l_{g,v} + n_v)/\tau}{\sum_{k=1}^{V} \exp(l_{g,k} + n_k)/\tau}, \tag{4}$$

where τ is a non-negative temperature, $n = -\log(-\log(u))$ and u are uniform samples from $U(0,1)$. During the forward pass, codeword i is chosen by $i = \arg\max_j p_{g,j}$ and in the backward pass, the Gumbel gradient of the outputs is used.

We analyzed and tested the percentage variation of each feature against every other TIScode in the dataset. Based on this analysis, we selected the 28 features reported in Tab. II (including the mode, which indicates whether the key is in the major or minor scale) as the most suitable for constructing bitmaps that identify TIScodes maximizing their pair-wise distances. In other words, we maximize the minimum Hamming distance between the bitmaps, represented

TABLE II
FEATURES COMPARED FOR SOUND ANALYSIS WITH THE SELECTED FEATURES HIGHLIGHTED

Attack Time	Attack Slope	Average Zero-Crossing Rate	Brightness	**Centroid**
Contrast	Decay Slope	Decay Time	Event Density	**Flatness**
Genre	Harmonic Ratio	Inharmonicity	Irregularity	**Key**
Kurtosis of Audio Signal	**Low Energy**	Low Energy Ratio	Maximum Bandwidth	Maximum Flatness
Maximum Peak Prominence	Maximum Roll-Off	Maximum Strength	**Mean of Audio Signal**	**Mean Pulse Curve**
Mean Roughness	Mean Strength	Mean Tempogram	Minimum Bandwidth	Minimum Centroid
Minimum Contrast	Minimum Flatness	Minimum Roll-Off	Minimum Strength	Minimum Tempogram
Modulation Index	**Musical Mode**	Pulse Clarity	**Root Mean Square (RMS)**	**Skewness of Audio Signal**
Spectral Centroid	**Spectral Entropy**	**Spectral Roll-Off**	**Spectral Spread**	**Max Pulse Curve Peak**
Tempo (BPM)	Temporal Centroid	**Tonal Centroid**	**Top Chroma Note 1**	**Top Chroma Note 2**
Top Instrument	**Top Tonal Network Class**	**Variance of Audio Signal**	**Width of Top Correlation Peak**	**Zero Crossing Rate**

as binary strings. We mapped all numerical features with 8 bit to obtain a final 193 bit bitmap that uniquely identifies every TIScode in our dataset.

The obtained Hamming distance can be used for error control (detection or correction) [27]. In this work, we exploit the redundancy due to the large number of features used to recover the TIScode through minimum Hamming distance decoding, so as to fix errors caused by ambient noise or erroneous matches of some features [28]. We also save spectral information of our short sounds separately from the features to improve the recognition precision. To analyze the spectrograms in the frequency domain, we divide them into multiple windows using the short-time Fourier transform (STFT):

$$\text{STFT}\{x(t)\}(m,\omega) = \sum_{n=-\infty}^{\infty} x[n] \cdot w[n-m] \cdot e^{-j\omega n} . \quad (5)$$

We set a 0.5 seconds window $w[n]$ and, for each window, we select the 10 most significant frequency peaks, saving their frequencies and the time intervals between them [29]. The map created from the peaks and their temporal positions will further contribute during the recognition phase by introducing redundancy, enabling more accurate identification of best match candidates. To test the recognition capabilities of our system, we design Algorithm 1. For the experiments, we select two noisy datasets: the "Ambient Dataset"[2] which is less noisy, featuring sounds like rain, thunder, and wind, and the "Hospital Dataset" [30], which is noisier, including sounds like screams, small and very loud rooms, etc.

In the audio recognition literature, the frequency responses of device microphones are often simplified as low-pass filters or not considered at all [14], [31]. To introduce a representation closer to real-world conditions we have conducted experiments in an anechoic chamber, recording a sweep from 20 Hz to 20 kHz in the Mel scale using various smartphones placed one meter from the sound source.[3] With the sweep recorded by the phone $y(t)$ and its original version $x(t)$, we first convert the two signals in the frequency domain as $X(f)$ and $Y(f)$, respectively. Then, the frequency response $H(f)$ is

$$H(f) = \frac{Y(f)}{X(f)} . \quad (6)$$

Algorithm 1 Simulations Recognition Algorithm

```
Select a number $N$ of simulations
for each simulation do
    Choose a random TIScode from the dataset
    Choose a random noise sample from the noise dataset
    Set an attenuation level in dB for the TIScode
    Set an attenuation level in dB for the noise sample
    Overlay the TIScode and noise
    if a phone's frequency response is selected then
        Apply the frequency response of the phone's microphone to the overlaid sound
    end if
    Extract features and perform frequency peaks analysis
    Match the audio with the highest-scoring TIScode in the database
end for
Compute the total number of matches
```

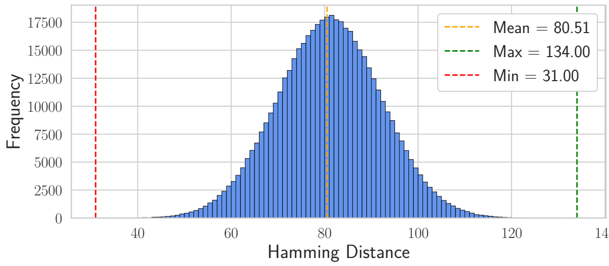

Fig. 2. Hamming distances distribution within the TIScode dataset.

Finally, we derive a general representation of the microphones impulse response $h(t)$ using the inverse discrete Fourier transform (IFFT) [32].

IV. RESULTS

In this section, we present the results of the tests conducted for the various methodologies listed in the previous sections, highlighting how the proposed techniques are suitable for an IoAuT system that leverages communication through short sound messages. First, we create a bitmap for each TIScode in our dataset, as described earlier and we compute and analyze how the Hamming distances are distributed within our dataset. The Hamming distances computed between all bitmap pairs and the frequencies are reported in Fig. 2. With

[2] https://www.kaggle.com/datasets/solorzano/ambient-noise/data
[3] https://github.com/manuelefavero/DSP

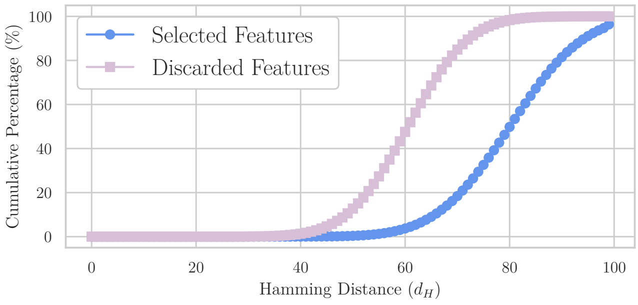

Fig. 3. CDF of Hamming Distances

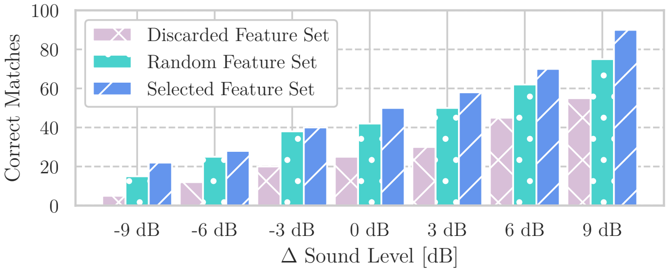

Fig. 4. Recognition benchmark of the three feature sets under Ambient Noise.

the generated mapping, we achieve an average distance of 80 bits, a maximum distance of 134, and a minimum distance of $d_{\min} = 31$. We thus can correct up to $\left\lfloor \frac{d_{\min}}{2} \right\rfloor = 15$ errors, which correspond to the bits of almost two entire features being detected incorrectly due to noise or distortion. By correcting these errors, we obtain the TIScode identifier as close as possible to the original codeword, thus pointing to the digital information database with which that sound was originally generated.

We observe that not all minimum distances are clustered near the Hamming distance of 31. In fact, this is a minimum value that occurs only in a few cases. As shown in Fig. 3, the probability of generating a TIScode with a Hamming distance below 50 is under 2%, and the probability of obtaining one with a distance below 60 bits is less than 10%. Given these results, to improve the minimum distance we can choose to intervene at the source, during the post-generation phase. We adopt a brute-force approach where, once a TIScode is generated, its Hamming distance is tested. If the new minimum Hamming distance does not exceed a previously set threshold, it is discarded and regenerated iteratively. To strengthen the selection of our features, we verify their impact during the recognition phase. In Fig. 4, we present a recognition test performed on three different feature pools: one using our selected features, a second using discarded features, and a third one using a random mix of 14 features from the selected set and 14 features from the discarded set.

The x-axis represents the difference in sound levels between the two audio signals. Positive values indicate a higher intensity level for the TIScode, while negative values signify that the noise has a greater intensity. Our selected feature set outperforms the other two, thus confirming the validity of our feature selection. Once our features have been validated, we proceed with recognition simulations, where the final result is obtained by combining the previously described decoding rule and the saved frequency peaks map. In Tabs. III and IV we report the simulations carried out using the frequency response of three selected phones for this recognition test: iPhone 13, OPPO A54, and Honor 9X. Honor 9X and OPPO A54 are new smartphones used only for these tests, while iPhone13 is a 3-years-old phone used by people on a day-to-day basis. The results come from five sets of 100 simulations each and should be considered with a 2% confidence interval. As shown in the case of the Ambient dataset, the success rate never drops below 50%, and without sound attenuation, it remains above 75% for correct matching. The noisier Hospital dataset, on the other hand, yields worse results, although it never falls below 50% at 0 dB. These results also highlight a dependence on the quality and condition of the microphone capsule. The Honor 9X proves to be the best in terms of correct match percentage, while the OPPO A54, a low-end smartphone, performs the worst. iPhone 13, although not producing excellent results, probably due to its microphone's capsule degradation, outperforms the OPPO. Overall, the system delivers solid results, especially considering the absence of any noise attenuation system in these experiments.

Finally, we present field tests conducted in three different scenarios. In all three, the smartphone, in this case a OnePlus 8T, is placed 40 cm away from a DELL Inspiron 16Plus 7620 laptop equipped with a NVIDIA GeForce RTX 3050. TIScodes were emitted by the laptop speaker, recorded with the smartphone, and subsequently decoded on the laptop. The recognition of a single TIScode, leveraging CUDA 12.6 for CNN-based feature extraction, takes an average of 7–8 seconds. The first scenario is a indoor room, with no significant background noise. The second scenario is inside the same room, but this time the TIScode is played with music in the

TABLE III
CORRECT MATCHES PERFORMANCES IN AMBIENT NOISE DATASETS

	Ambient Dataset						
	-9 dB	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB
Without $H(f)$	79%	85%	88%	95%	96%	97%	98%
OPPO A54	51%	66%	68%	75%	77%	82%	88%
iPhone 13	50%	52%	63%	76%	85%	92%	95%
Honor 9X	68%	85%	87%	88%	91%	94%	95%

TABLE IV
CORRECT MATCHES PERFORMANCES IN HOSPITAL NOISE DATASETS

	Hospital Dataset						
	-9 dB	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB
Without $H(f)$	72%	79%	81%	92%	95%	97%	98%
OPPO A54	32%	37%	49%	55%	62%	76%	82%
iPhone 13	42%	49%	52%	56%	69%	81%	83%
Honor 9X	62%	75%	85%	87%	89%	91%	92%

TABLE V
SUMMARY OF SCENARIOS WITH CORRECT MATCHES PERCENTAGES

Scenario	Location	PC Volume	Correct Matches
Scenario 1	Room	30%	96%
Scenario 2	Room w/ background music	30%	92%
Scenario 3	Outdoor area w/ people speaking	50%	91%

background from the same PC at an intensity level half that of the TIScode. The third scenario is an outdoors area near the university with tables around and people talking. In this case, the percentages of correct matches reported in Tab. V were produced after 100 recognition tests.

V. Concluding Remarks and Future Works

In this work, we introduced TIScode, a novel technology that uses audio for information transmission. We demonstrated the integration of various artificial intelligence technologies to automate audio generation and recognition processes. Specifically, we employed MusicGen [18], an autoregressive transformer-based model, for sound generation; CREPE [23], a deep convolutional neural network, for robust key recognition; and wav2vec [20], a multi-layer convolutional feature encoder, for genre and instrument classification.

Our proposed channel coding leverages sound features to create a bitmap, uniquely identifying TIScodes. Initial results are promising, though the absence of a noise cancellation or attenuation mechanism remains a limitation that must be addressed to enhance system performance [33]. Further real-world testing is required to validate the virtual simulation results. Additionally, incorporating methods to introduce redundancy and improve error correction, such as Reed-Solomon codes and audio watermarking algorithms leveraging Discrete Cosine Transform coefficients, will be valuable in optimizing the system for practical applications. [27], [34].

References

[1] L. Turchet, G. Fazekas, M. Lagrange, H. S. Ghadikolaei, and C. Fischione, "The Internet of audio things: State of the art, vision, and challenges," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10 233–10 249, 2020.

[2] L. Murrone. TIScode : Notes on application. Ogenus S.R.L. [Online]. Available: https://www.tiscode.eu/features

[3] A. Bertrand, "Applications and trends in wireless acoustic sensor networks: A signal processing perspective," in *Proc. IEEE SCVT*, 2011.

[4] P. C. Verhoef, A. T. Stephen, P. Kannan, X. Luo, V. Abhishek *et al.*, "Consumer connectivity in a complex, technology-enabled, and mobile-oriented world with smart products," *J. Interact. Marketing*, vol. 40, no. 1, pp. 1–8, 2017.

[5] F. Badia, G. Galeone, and M. Shini, "Sustainable strategies of industrial tourism in the agri-food business: an exploratory approach," *British Food J.*, vol. 126, no. 1, pp. 327–346, 2024.

[6] L. Badia, N. Bui, M. Miozzo, M. Rossi, and M. Zorzi, "Improved resource management through user aggregation in heterogeneous multiple access wireless networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 9, pp. 3329–3334, 2008.

[7] N. Caporusso, L. Mkrtchyan, and L. Badia, "A multimodal interface device for online board games designed for sight-impaired people," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 2, pp. 248–254, 2009.

[8] C. Nalmpantis, L. Vrysis, D. Vlachava, L. Papageorgiou, and D. Vrakas, "Noise invariant feature pooling for the Internet of audio things," *Multim. Tools Appl.*, vol. 81, no. 22, pp. 32 057–32 072, 2022.

[9] J. Rouillard, "Contextual QR codes," in *Proc. IEEE ICCGI*, 2008, pp. 50–55.

[10] D. Salvati, S. Canazza, A. Rodà *et al.*, "A sound localization based interface for real-time control of audio processing," in *Proc. Int. Conf. Digit. Audio Effects*, 2011, pp. 177–184.

[11] J. Vandendriessche, N. Wouters, B. da Silva, M. Lamrini, M. Y. Chkouri, and A. Touhafi, "Environmental sound recognition on embedded systems: From FPGAs to TPUs," *MDPI Electronics*, vol. 10, no. 21, p. 2622, 2021.

[12] E. C. Nunes, "Anomalous sound detection with machine learning: A systematic review," *arXiv preprint arXiv:2102.07820*, 2021.

[13] C. Nadri, S. Ko, C. Diggs, M. Winters, S. Vattakkandy, and M. Jeon, "Sonification use cases in highly automated vehicles: designing and evaluating use cases in level 4 automation," *Int. J. Hum.-Comput. Interact.*, vol. 40, no. 12, pp. 3122–3132, 2024.

[14] K. Altwlkany, S. Delalić, A. Alihodžić, E. Selmanović, and D. Hasić, "Application of audio fingerprinting techniques for real-time scalable speech retrieval and speech clusterization," in *Proc. IEEE MIPRO*, 2024, pp. 91–96.

[15] X. Riley and S. Dixon, "CREPE notes: A new method for segmenting pitch contours into discrete notes," *arXiv preprint arXiv:2311.08884*, 2023.

[16] X. Liu, H. Lu, J. Yuan, and X. Li, "CAT: Causal audio transformer for audio classification," in *Proc. IEEE ICASSP*, 2023, pp. 1–5.

[17] S. Wyatt, D. Elliott, A. Aravamudan, C. E. Otero, L. D. Otero, G. C. Anagnostopoulos, A. O. Smith, A. M. Peter, W. Jones, S. Leung *et al.*, "Environmental sound classification with tiny transformers in noisy edge environments," in *Proc. IEEE WF-IoT*, 2021, pp. 309–314.

[18] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," *Adv. Neural Inf. Process. Syst.*, vol. 36, pp. 47 704–47 720, 2024.

[19] M. E. Sander, R. Giryes, T. Suzuki, M. Blondel, and G. Peyré, "How do transformers perform in-context autoregressive learning?" *Proc. ICML*, 2024.

[20] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *Proc. Interspeech*, 2019.

[21] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "LIBROSA: Audio and music signal analysis in Python," in *Proc. Python Sci. Conf.*, 2015, pp. 18–24.

[22] O. Lartillot. Mirtoolbox. [Online]. Available: https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe-/materials/mirtoolbox

[23] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "CREPE: A convolutional representation for pitch estimation," in *Proc. IEEE ICASSP*, 2018, pp. 161–165.

[24] B. L. Sturm, "An analysis of the GTZAN music genre dataset," in *Proc. MIRUM*, 2012, pp. 7–12.

[25] E. Humphrey, S. Durand, and B. McFee, "OpenMIC-2018: An open data-set for multiple instrument recognition." in *Proc. ISMIR*, 2018, pp. 438–444.

[26] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[27] M. Rossi, L. Badia, and M. Zorzi, "On the delay statistics of SR ARQ over Markov channels with finite round-trip delay," *IEEE Trans. Wireless Commun.*, vol. 4, no. 4, pp. 1858–1868, 2005.

[28] B. Tomasi, P. Casari, L. Badia, and M. Zorzi, "A study of incremental redundancy hybrid ARQ over markov channel models derived from experimental data," in *Proc. ACM Int. Wkshp Underwater Netw. (WuWNet)*, 2010.

[29] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system." in *Proc. ISMIR*, vol. 2002, 2002, pp. 107–115.

[30] S. Cammarasana, P. Nicolardi, and G. Patanè, "Real-time denoising of ultrasound images based on deep learning," *Med. Biol. Eng. Comput.*, vol. 60, no. 8, pp. 2229–2244, 2022.

[31] S. Chang, D. Lee, J. Park, H. Lim, K. Lee, K. Ko, and Y. Han, "Neural audio fingerprint for high-specific audio retrieval based on contrastive learning," in *IEEE ICASSP*, 2021, pp. 3025–3029.

[32] R. Wang, Z. Chen, and F. Yin, "Adaptive frequency response calibration method for microphone arrays," *IEEE Sensors J.*, vol. 20, no. 13, pp. 7118–7128, 2020.

[33] Z. Mushtaq, S.-F. Su, and Q.-V. Tran, "Spectral images based environmental sound classification using cnn with meaningful data augmentation," *Appl. Acoust.*, vol. 172, p. 107581, 2021.

[34] Y.-Y. Tai and M. Mansour, "Audio watermarking over the air with modulated self-correlation," in *Proc. IEEE ICASSP*, 2019.