

Article

Identification of Greek Orthodox Church Chants Using Fuzzy Entropy

Lazaros Moysis ^{1,*}, Konstantinos Karasavvidis ², Dimitris Kampelopoulos ², Achilles D. Boursianis ², Sotirios Sotiroudis ², Spiridon Nikolaidis ², Christos Volos ¹, Panagiotis Sarigiannidis ³, Mohammad Abdul Matin ⁴ and Sotirios K. Goudos ^{2,*}

- ¹ Laboratory of Nonlinear Systems-Circuits & Complexity, Physics Department, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece; volos@physics.auth.gr
- ² ELEDIA@AUTH, School of Physics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece; kokarasa@physics.auth.gr (K.K.); dkampelo@physics.auth.gr (D.K.); bachi@physics.auth.gr (A.D.B.); ssoti@physics.auth.gr (S.S.); snikolaid@physics.auth.gr (S.N.)
- ³ Department of Electrical and Computer Engineering, University of Western Macedonia, 50100 Kozani, Greece; psarigiannidis@uowm.gr
- ⁴ Department of Electrical and Computer Engineering, North South University, Dhaka 1229, Bangladesh; mohammad.matin@northsouth.edu
- * Correspondence: lmousis@physics.auth.gr (L.M.); sgoudo@physics.auth.gr (S.K.G.)

Abstract: In this work, a comparison of Greek Orthodox religious chants is performed using fuzzy entropy. Using a dataset of chant performances, each recitation is segmented into overlapping time windows, and the fuzzy entropy of each window in the frequency domain is computed. We introduce a novel audio fingerprinting framework by comparing the variations in the resulting fuzzy entropy vector for the dataset. For this purpose, we use the correlation coefficient as a measure and dynamic time warping. Thus, it is possible to match the performances of the same chant with high probability. The proposed methodology provides a foundation for building an audio fingerprinting method based on fuzzy entropy.

Keywords: audio fingerprinting; fuzzy entropy; Greek Orthodox Church chants; music identification; music information retrieval



Academic Editor: Leandros Maglaras

Received: 18 December 2024

Revised: 16 January 2025

Accepted: 23 January 2025

Published: 27 January 2025

Citation: Moysis, L.; Karasavvidis, K.; Kampelopoulos, D.; Boursianis, A.D.; Sotiroudis, S.; Nikolaidis, S.; Volos, C.; Sarigiannidis, P.; Matin, M.A.; Goudos, S.K. Identification of Greek Orthodox Church Chants Using Fuzzy Entropy. *Computers* **2025**, *14*, 39. <https://doi.org/10.3390/computers14020039>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Music Identification

In the field of Music Information Retrieval (MIR), song identification is one of the most prominent applications [1,2]. It refers to the task of identifying the origin track of a short audio recording, usually a few seconds long. Another problem closely related to song identification is that of cover song identification, which refers to matching performances of the same song by different artists, which can often differ significantly in musical style. Especially in genres like traditional and religious music, there may be numerous different renditions of a song or chant, performed by many artists and across many decades, so song matching can be challenging. These tasks have numerous commercial applications, such as song identification for music listeners, broadcast monitoring for advertising purposes, and even support of legal violation claims such as copyright infringement and plagiarism detection [3].

1.2. Related Works

The task of audio identification relies on the extraction of an audio's fingerprint. This process of fingerprinting refers to the generation of a signature from the signal. This

signature is small in size, and can be used to identify the signal. Different signals will have distinct signatures, but the signatures of similar signals will also have similarities. Thus, instead of comparing audio signals directly, their corresponding signatures can be compared, in order to find similarities faster.

The fingerprint is usually computed in the frequency domain by considering the Fast Fourier Transform (FFT) of the signal. For example, in [4], the FFT was applied and a fingerprint was computed from the spectrogram using four frequencies. The effect of noise was also studied. In [5], the use of the two-dimensional Fourier transform was applied to the binarized spectrogram of a signal to derive a fingerprint. The effect of noise was also considered. Both [4,5] consider recognition of Orthodox hymn recitations. In [6], fingerprints are computed by peaks in the spectrogram and are considered for music recommendation, taking artist and genre into account. In [7], fingerprints are extracted by finding peaks in the Mel spectrogram and used for matching song recordings. In [8], fingerprints are extracted from the Mel-Frequency Cepstral Coefficients and considered for the problem of music identification in broadcast monitoring to address copyright issues. Other approaches consider machine learning or generative AI for this and similar tasks [9–11].

Among the various approaches developed in the last 20 years for identifying songs, the use of entropy seems highly efficient. Entropy is a measure used to characterize the complexity and unpredictability of a system [12]. Over the years, entropy has been used to characterize several types of physical system, as well as signals of any type, such as text, audio, and video.

For audio processing, entropy-based fingerprinting has previously been considered for song identification in [13–18] and broadcast monitoring in [19]. The process involves segmenting an audio signal, transforming it into the frequency domain, and measuring the entropy along different frequency bins. The derivative across time segments is then used to derive a binary fingerprint, which can be used for identification. The present work will consider a similar approach but using fuzzy entropy.

Fuzzy entropy [20] is a variation of entropy, where the principle of fuzzy distance is applied to derive a soft, non-binary (equal/not equal) comparison between time series. The theory of fuzzy sets was introduced by Lotfi Zadeh in the 1960s [21], and has long been established as a highly applicable theory to all aspects of engineering science. Examples include control theory, decision-making, optimization, signal processing, and more [22].

Fuzzy entropy has gained more attention in the last decade or so and has been used in a plethora of applications. Examples include the characterization of biomedical signals, such as electroencephalogram (EEG) [23–25], electrocardiogram (ECG) [23], and electromyography (EMG) signals [20,26,27]. It can also be used effectively to compare chaotic time series [26–28]. Moreover, it has been considered a feature for voice activity detection [29] and microphone identification [30].

1.3. Motivation

In this work, motivated by the use of entropy in audio identification and the increasing applicability of fuzzy entropy in signal characterization, the problem of identification of different performances using fuzzy entropy is studied for Greek Orthodox Church chants. The most important research challenge in this musical field stems from the fact that in contrast to other genres of music, there is not a single version of a given chant. For example, in genres like pop, classic, rock, etc, there is a single official version of a song, usually published in an album, and performed by a single artist. So, the challenge of identification is to match a short recording to its source track, which is an identical song. This is different in religious chants, as there is not a single official recording of a song, but rather multiple

renditions performed by different chanters. Thus, the chant identification task here is not to match a short recording to its identical track, but rather to identify similarities between numerous different renditions of the same chant.

We consider a newly created dataset of 50 different popular religious chants, each chant having several different performances. To find the fuzzy entropy vector, each track is divided into windows that overlap. The Fast Fourier Transform (FFT) is then used to change the time domain to the frequency domain, and the fuzzy entropy of the size of each window is found. The similarity between the fuzzy entropy vectors for all tracks is then computed using the correlation coefficient. To improve performance, an additional Dynamic Time Warping (DTW) process [31] is performed before computing the similarity measures. With this addition, the accuracy can exceed 90%.

This work is based on previous work on entropy-based identification [13–18]. The main contributions of this work are the following:

- The measure of entropy has been swapped with that of fuzzy entropy, which has been reported in the literature to be more robust in comparing signals.
- Following [17,32], a pre-emphasis filter is applied in the preprocessing stage, which has not been considered in all earlier works.
- The frequency domain signal is not segmented in frequency bins, but rather its entropy as a whole is computed. This results in a less detailed but much shorter vector entropy measure, as opposed to a matrix entropy feature (entropygram).
- The comparison is performed on the entropy vector rather than on a binary fingerprint.
- Following [31], the correlation coefficient is calculated after applying DTW to the entropy vectors, which significantly improves performance.

To the best of our knowledge, this is the first time a fuzzy entropy method has been applied to a chant recognition problem. We must note that the same methodology can easily be extended to other similar recognition problems, e.g., to other vocal music problems. The remainder of the work is structured as follows: Section 2 presents the method used to calculate the fuzzy entropy measure and the method employed to compare the entropy vectors. Section 3 presents the comparison results in the dataset considered. Finally, Section 4 concludes the work with a discussion of future research goals.

2. Characterization of Byzantine Chants Using Fuzzy Entropy

2.1. The Dataset

The dataset consists of fifty different most popular hymns, and for each one, there is a varying number of performances available from different chanters. Therefore, there are more than 5000 different tracks available. Example track titles are 'Ταις πρεσβείαις της Θεοτόκου', 'Σώσον Ημάς', 'Άγιος ο Θεός', 'Εις άγιος εις κύριος', 'Είδομεν το φώς', 'Χριστός ανέστη', 'Χριστός γεννάται', and 'Δόξα σοι το δείξαντι το φως'.

The sampling frequency of all tracks was 44100 Hz. Initial preprocessing included cropping the tracks to remove the silent parts at the beginning and end. The tracks range from a couple of seconds to 2 min. The duration of the complete dataset is around 3 h.

2.2. Signal Preprocessing

For all the signals in the dataset, preprocessing was performed. The procedure is described in Figure 1 and consists of the following steps:

1. For dual-channel signals, an averaged single-channel audio signal is computed.

- Following [17,32], a pre-emphasis filter is applied to the signal, which can emphasize human voice patterns. For a signal $s(k)$, where k denotes its samples, the filter's function is given by

$$y(k) = s(k) - a \cdot s(k - 1) \quad (1)$$

where $y(k)$ is the filtered signal and $a \in [0, 1]$ is an equalization parameter. Here, we choose $a = 0.95$. Figure 2 shows the difference between the original signal and the filtered signal.

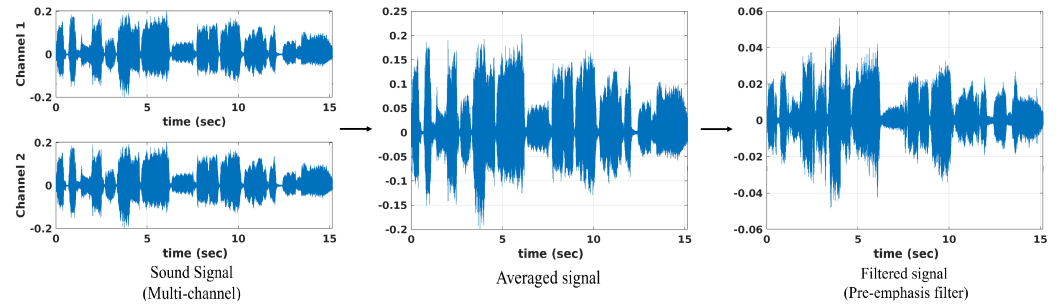


Figure 1. Outline of the signal preprocessing procedure.

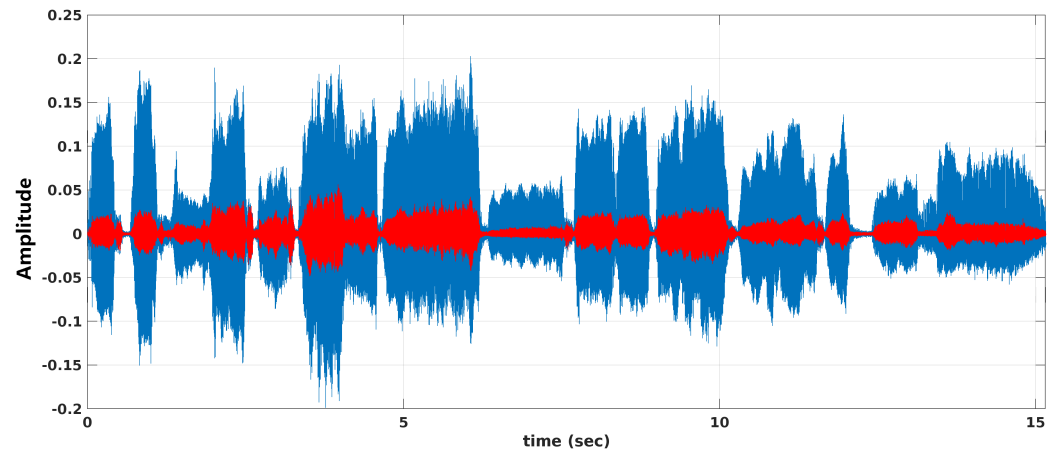


Figure 2. The averaged signal (blue) and the filtered one (red) using a pre-emphasis filter.

2.3. Computation of Fuzzy Entropy

Consider the following time series $u(i)$, $i = 1, \dots, N$ of length N . The fuzzy entropy of the time series is computed as follows [20,24,26,29]:

- Construct the vectors

$$X_i^m = [u(i), u(i+1), \dots, u(i+m-1)] - u_0(i), \quad i = 1, 2, \dots, N - m + 1 \quad (2)$$

where $m \leq N - 2$, and $u_0(i)$ is the average of the vector

$$u_0(i) = \frac{1}{m} \sum_{j=0}^{m-1} u(i+j), \quad (3)$$

and the subtraction is element-wise.

- The distance d_{ij}^m between two vectors X_i^m and X_j^m is computed, which is the maximum absolute difference of the following two vectors:

$$d_{ij}^m = d(X_i^m, X_j^m) = \max_{k \in (0, m-1)} [|(u(i+k) - u_0(i)) - (u(j+k) - u_0(j))|],$$

$$i, j = 1, \dots, N - m + 1, \quad i \neq j \quad (4)$$

- A fuzzy membership function is applied to measure the similarity degree between two vectors X_i^m and X_j^m . Here, the exponential function is considered, given by

$$D_{ij}^m = \mu(d_{ij}^m, n, r) = e^{\left(\frac{-(d_{ij}^m)^n}{r}\right)} \quad (5)$$

where n and r are the gradient and width of the function.

- The following function is computed.

$$\phi^m(n, r) = \frac{1}{N - m} \sum_{i=1}^{N-m} \left(\frac{1}{N - m - 1} \sum_{j=1, j \neq i}^{N-m} D_{ij}^m \right) \quad (6)$$

- The previous steps are repeated for vectors X_i^{m+1} , in order to compute

$$\phi^{m+1}(n, r) = \frac{1}{N - m} \sum_{i=1}^{N-m} \left(\frac{1}{N - m - 1} \sum_{j=1, j \neq i}^{N-m} D_{ij}^{m+1} \right) \quad (7)$$

- The fuzzy entropy is finally defined as

$$\text{FuzzEn}(m, n, r, N) = \lim_{N \rightarrow \infty} (\ln \phi^m(n, r) - \ln \phi^{m+1}(n, r)) \quad (8)$$

and since N is finite for a given time series, the entropy is simply expressed as

$$\text{FuzzEn}(m, n, r) = \ln \phi^m(n, r) - \ln \phi^{m+1}(n, r) \quad (9)$$

2.4. Extraction of Fuzzy Entropy Measure

Consider a given recording as a time series $y(k)$, sampled at a frequency freq_s , after its preprocessing described above is performed. Computing its fuzzy entropy (FuzzEn) would provide only a single measurement, which would not be enough to characterize the recording. Thus, the following procedure is followed, which is also described in Figure 3.

- The time series $y(k)$ is broken down into overlapping segments w_i of duration $d = 0.1$ s. The overlap between consecutive segments is taken as 50%.
- For each segmented part w_i , a Hanning window is applied, and the signal is then transformed into the frequency domain, using an FFT transform. The resulting signal is denoted as f_{w_i} , and it is a complex vector.
- The fuzzy entropy $\text{FuzzEn}(m, n, r)$ of magnitude $|f_{w_i}|$ is calculated for each segment. Since $|f_{w_i}|$ is symmetric, only its first half is considered for this. The resulting entropy values are added to a vector. This results in the fuzzy entropy vector of the recording, which represents the changes between the FuzzEn values across all segments of the track.
- Finally, the entropy vector is normalized so that its values lie in the interval $[0, 1]$. Normalization of a vector x is performed as

$$x_i^{\text{norm}} = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (10)$$

where x_i denotes the vector elements.

Fuzzy entropy is computed using the code developed in [23]. The fuzzy function considered is exponential, with parameters $r = 0.15 \cdot \text{std}(|f_{w_i}|)$, $m = 2$, $n = 2$, $loc = 1$, and $\tau = 1$, where $\text{std}(|f_{w_i}|)$ denotes the standard deviation of the window under consideration. Figure 4 shows two fuzzy entropy vectors for two performances of the first chant in the dataset. Although the vectors have unequal lengths, similarities between them can be observed, for example, peaks, falls, and plateaus. In the next section, the comparison process is described.

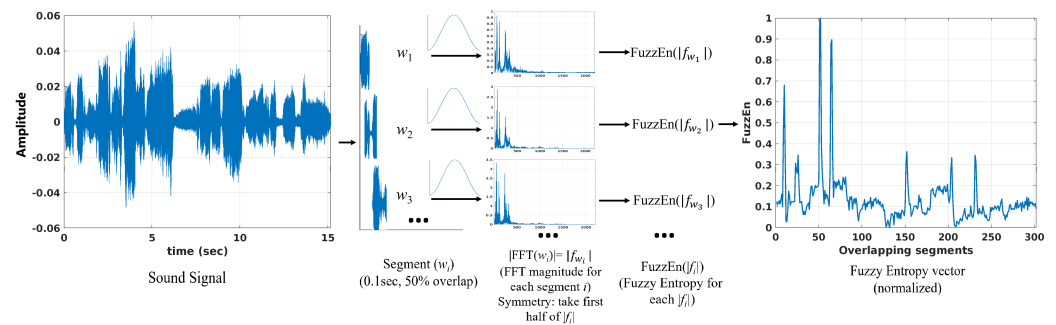


Figure 3. Outline of the procedure for extracting the FuzzEn information vector for a track.

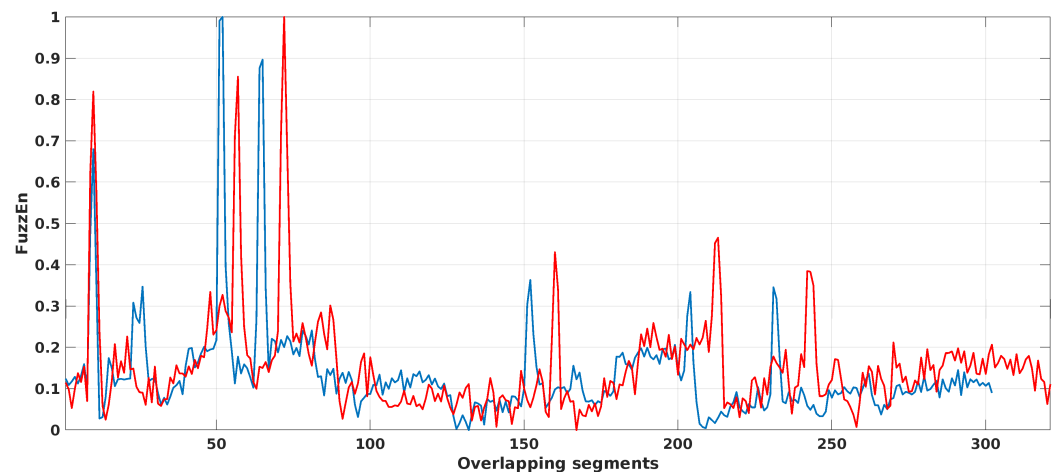


Figure 4. Fuzzy entropy vectors for two performances of the first chant in the dataset. The red line shows the entropy vector of the first performance, the blue line shows the entropy vector of the second performance.

2.5. Track Comparison

To compare the fuzzy entropy vectors of all tracks in the dataset, the statistical measure of the correlation coefficient is used. The correlation coefficient $\rho(X, Y)$ between two time series X and Y of length N is given by

$$\rho(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} \quad (11)$$

where $\text{cov}(X, Y)$ denotes the covariance between the signals, and σ_X and σ_Y their standard deviations. This must be applied to vectors of equal length. Thus, two different approaches are considered for comparison.

2.5.1. Comparison of Whole Tracks

In the first approach, the complete fuzzy entropy vectors of all tracks are compared to each other. The reasoning behind this comparison is to see if a complete recording is similar as a whole to a part of another track. Since vectors have unequal lengths, a sliding window is considered to compute the statistical measure. The complete process is described below and is also visually depicted in Figure 5.

1. Let y_1 and y_2 be two different recordings in the dataset, and F_1 and F_2 their corresponding fuzzy entropy vectors. Let $l_1 = \text{length}(F_1)$, $l_2 = \text{length}(F_2)$, and assume, without loss of generality, that $l_1 < l_2$, so the second recording is longer.
2. For the longer vector F_2 , segments of length l_1 , that is, $F_2^i = F_2(i : i + l_1 - 1)$, $i = 1, z + 1, 2z + 1, \dots$ are considered. Here, i denotes the iteration step and z denotes the sliding window jump. Its default value is $z = 1$, but higher values can be considered to improve the execution speed. Here, we choose $z = 20$. This corresponds to a 1.05 s jump. The notation $i : i + l_1 - 1$ denotes the elements from the position i to the position $i + l_1 - 1$.
3. A DTW is applied on all pairs of F_1 and F_2^i to stretch the two vectors into the new vectors F_1^{dtw} and $F_2^{i,dtw}$, whose Euclidean distance is the smallest. Note that, in general, the stretched signals have a length longer than l_1 , so some entries may be repeated.
4. The vector F_1^{dtw} is compared with each segment $F_2^{i,dtw}$. For this, the correlation coefficients $\rho(F_1^{dtw}, F_2^{i,dtw})$ are computed. The maximum value among these is taken as the highest similarity index between the two tracks y_1 and y_2 .

Note that another approach for comparison would be to apply zero padding on all the fuzzy entropy vectors to make them of equal length. However, because of the high variation in the lengths of the recordings, this would make the comparison unreliable and inaccurate, so it is avoided.

Figures 6 and 7 show examples of comparisons between F_1 and F_2^i or $F_2^{i,dtw}$. Note that when DTW is applied, the subvector $F_2^{i,dtw}$ that will show the highest correlation with F_1 may be different from the subvector found without the use of DTW.

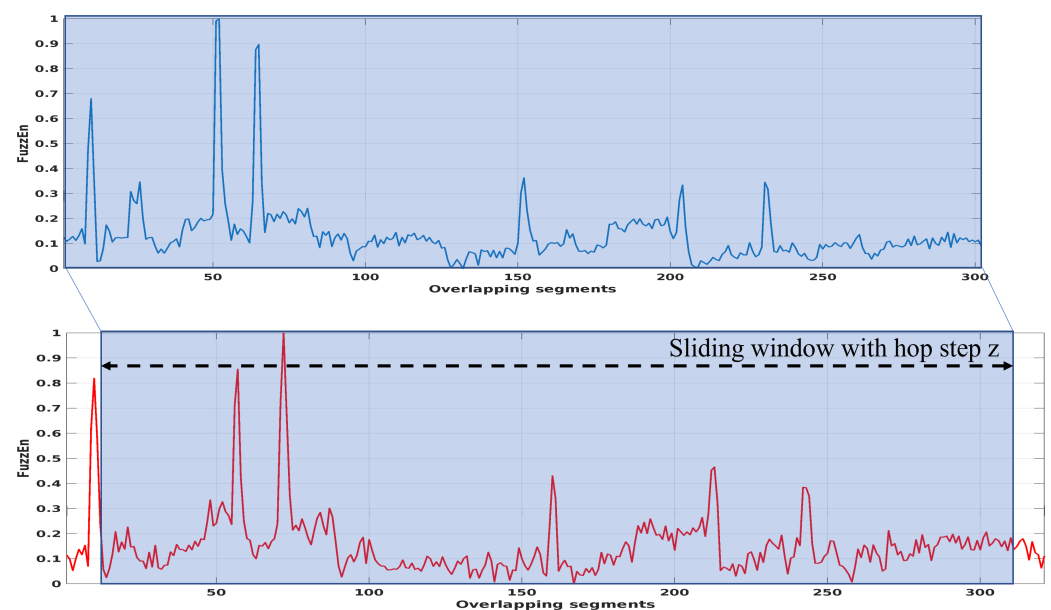


Figure 5. Visual outline of the comparison approach for whole recordings.

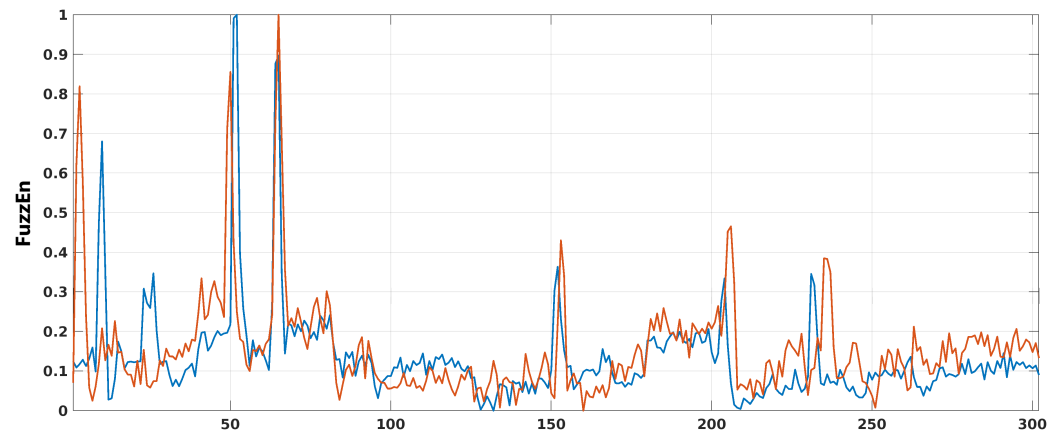


Figure 6. Example of entropy vectors F_1 and F_2^i that showcase the highest correlation coefficient (0.47) among all the segments of F_2 . The red line shows the entropy vector of the first performance, the blue line shows the entropy vector of the second performance.

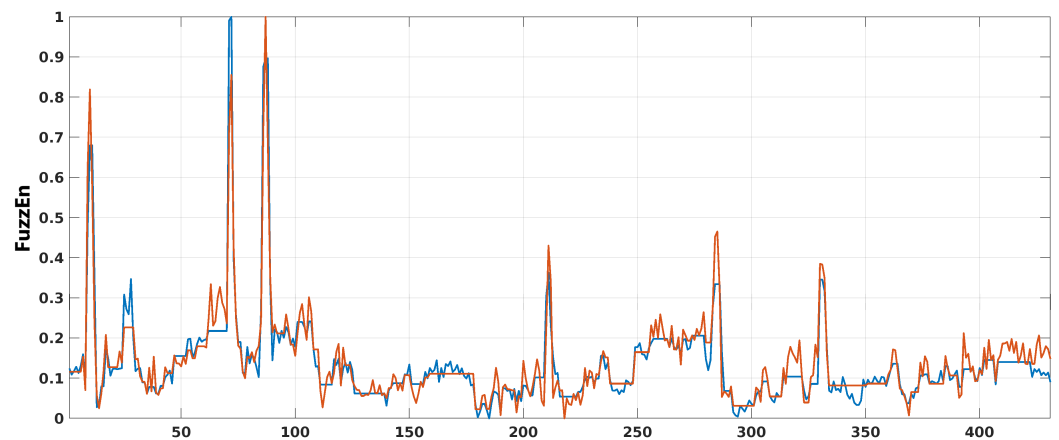


Figure 7. Example of entropy vectors F_1^{dtw} and $F_2^{i,dtw}$ that showcase the highest correlation coefficient (0.94) among all the segments of F_2 . The red line shows the entropy vector of the first performance, the blue line shows the entropy vector of the second performance.

2.5.2. Comparison of Segments

As a second approach, segments from both entropy vectors of two tracks are compared with each other. The reason behind this comparison is to see if a part of one track is similar to a part of another track. Here, we consider the comparison of 5 s segments between tracks. For the given window size and overlap percentage, this corresponds to the entropy sub-vectors of 99 elements, which is rounded to 100 elements for simplicity. For this comparison, a sliding window is applied to both entropy vectors, and all segments with a length of 100 elements are compared. The process is described in the following and is also visually depicted in Figure 8.

1. Let y_1 and y_2 be two different recordings in the dataset, and F_1 and F_2 their corresponding fuzzy entropy vectors. Let $l_1 = \text{length}(F_1)$ and $l_2 = \text{length}(F_2)$.
2. For both vectors F_1 and F_2 , segments of 100 values are considered, that is, $F_1^i = F_1(i : i + 100 - 1)$, $F_2^j = F_2(j : j + 100 - 1)$, $i, j = 1, z + 1, 2z + 1, \dots$. Here, i and j denote the iteration step and z denotes the jump of the sliding window. Here, we choose $z = 40$.
3. A DTW is applied to all pairs of F_1^i and F_2^j to stretch the two vectors into the new vectors $F_1^{i,dtw}$, and $F_2^{j,dtw}$, whose Euclidean distance is the smallest. Note that, in general, the stretched signals have lengths longer than 100, so some entries may be repeated.

- Each entropy vector $F_1^{i,dtw}$ is compared to each $F_2^{j,dtw}$. For this, the correlation coefficients $\rho(F_1^{i,dtw}, F_2^{j,dtw})$ are calculated. The maximum value among these is taken as the highest similarity index between the two tracks y_1 and y_2 .

Note that since this comparison takes longer, a sliding window jump of $z = 40$ values is taken, which corresponds to a jump of 2.05 s.

Figures 9 and 10 show examples of comparisons between F_1^i and F_2^j and $F_1^{i,dtw}$ and $F_2^{j,dtw}$. As in the previous case, the positions i and j with the highest similarity may differ between the use of DTW and those without it.

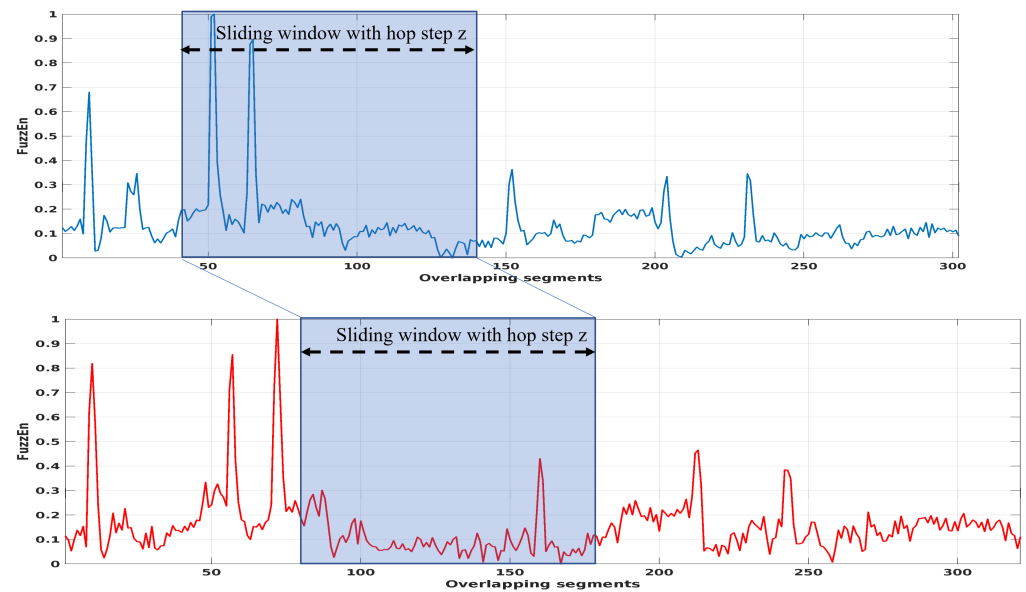


Figure 8. Visual outline of the comparison approach for 5 s segments.

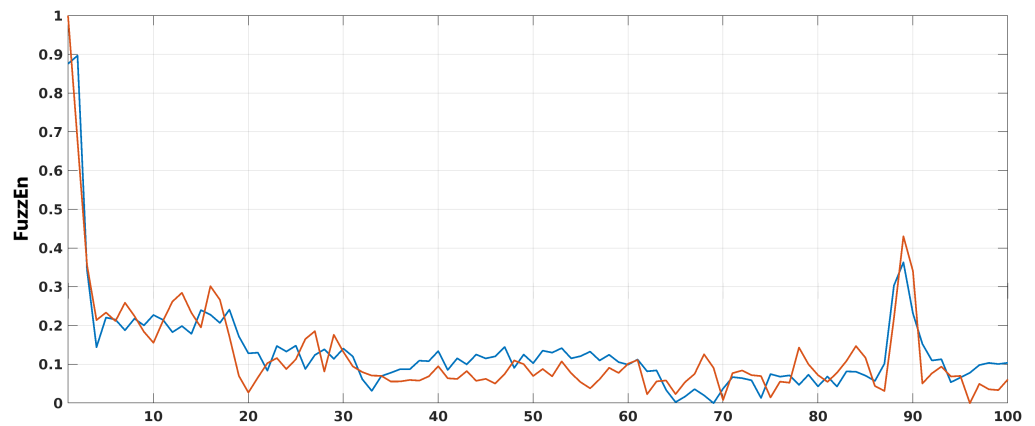


Figure 9. Example of entropy vectors F_1^i and F_2^j that showcase the highest correlation coefficient (0.90) among all i and j segments. The red line shows the entropy vector of the first performance, the blue line shows the entropy vector of the second performance.

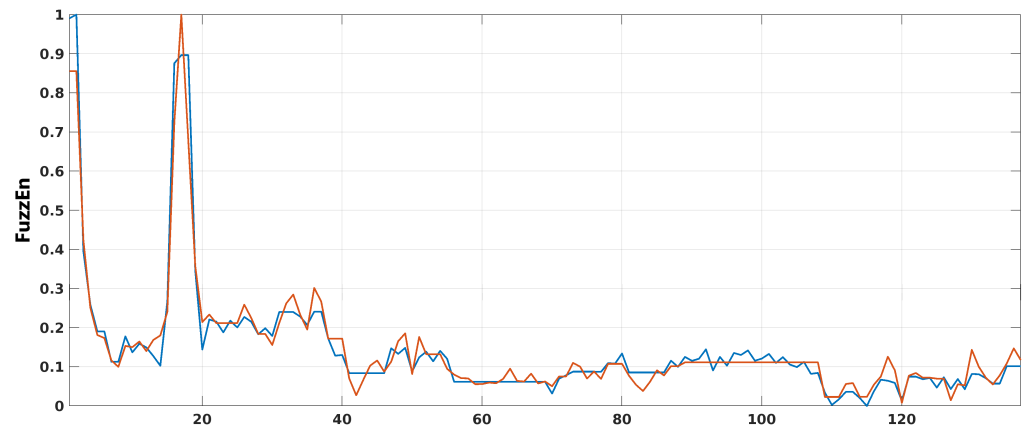


Figure 10. Example of entropy vectors $F_1^{i,dtw}$ and $F_2^{j,dtw}$ that showcase the highest correlation coefficient (0.98) among all i and j segments. The red line shows the entropy vector of the first performance, the blue line shows the entropy vector of the second performance.

3. Comparison Results

When comparing the similarity between tracks in the dataset, the objective is to see if performances of the same chant showcase the highest similarity between all tracks in the dataset. For this, once the similarity measures between all tracks in the database are computed, the following simple procedure is followed to identify the performances of the same chant.

1. For a track y_i , the similarity measures (maximum correlation coefficient) that were computed between all songs are sorted in descending order.
2. If the top three most similar entries include a performance of the same chant, the identification is considered successful. Self-matches are, of course, excluded.

Table 1 lists the results of the comparison of the whole tracks with each other. In addition to the use of fuzzy entropy described above, the performance of the algorithms was also evaluated under the use of entropy, which is computed as in [13,17,19]:

$$H = \ln(2\pi e) + \frac{1}{2} \ln(\sigma_{xx}\sigma_{yy} - \sigma_{xy}^2) \quad (12)$$

where σ_{xx} and σ_{yy} are the variances of the real and imaginary part of the spectrum, and σ_{xy} their covariance. Results without the use of the DTW step are also shown. When taking the DTW to compare pairs of entropy vectors, the default metric is the Euclidean distance. Yet, there are other alternative metrics that can be used, like the sum of absolute differences (absolute), the square of the Euclidean metric (squared), and the symmetric Kullback–Leibler metric (symmkl). These metrics are available in MATLAB’s `dtw` built-in function, so the performance under these alternatives was also considered. Moreover, when computing the DTW measure, an optional restriction has been considered, where the warping path is limited to within Z samples of a straight line fit between the two vectors. These cases are denoted as DTW-lim Z in the table. The accuracy is also displayed for the case where the song that is the most similar to the top is associated with the same chant.

Without the use of the DTW step, the accuracy with fuzzy entropy can reach 56% for the top three results, and 51% for the top one result. This means that for more than half of the recordings in the dataset, performances of the same chant indeed yield the highest similarity among all the recordings. When considering entropy, this percentage is slightly higher: 63% for the top three results, and 54% for the top one result. However, with the application of the DTW and the default Euclidean distance as a metric, the accuracy is significantly increased—up to 95% for the top three, and 90% for the top one. When

considering the absolute distance, the results are the same as the Euclidean case for all the tests. When considering the squared metric, the results are slightly improved to 96% for the top three and 93% for the top one. The symmkl metric also yields good results, but as discussed later, it is significantly slower than the other metrics. When entropy is used, the results are around 90–92% for the top three, and around 86–87% for the top one, so the performance is below that of fuzzy entropy. The use of DTW gives a very promising result, although it comes at a trade-off with execution speed, as will be seen in the next section. By limiting the warping path to $Z = 100, 80, 60$ values, there is an insignificant change in performance. For $Z = 40, 20$, the drop becomes larger. Although the use of this constraint can damage performance, it can help improve execution speed, as will be discussed next.

Table 1. Identification accuracy. Comparing whole tracks.

Method	Metric	FuzzEn		En	
		Top 3	Top 1	Top 3	Top 1
no-DTW	-	56.82%	51.25%	63.23%	54.03%
DTW	Euclidean	95.54%	90.52%	91.92%	87.18%
DTW	Absolute	95.54%	90.52%	91.92%	87.18%
DTW	Squared	96.10%	93.03%	92.20%	86.90%
DTW	Symmkl	93.59%	88.30%	90.52%	86.07%
DTW-lim100	Euclidean	95.82%	90.25%	91.64%	87.18%
DTW-lim100	Absolute	95.82%	90.25%	91.64%	87.18%
DTW-lim100	Squared	96.37%	92.75%	91.92%	87.18%
DTW-lim100	Symmkl	94.98%	89.97%	91.64%	87.46%
DTW-lim80	Euclidean	95.82%	90.25%	91.36%	86.62%
DTW-lim80	Absolute	95.82%	90.25%	91.36%	86.62%
DTW-lim80	Squared	96.37%	92.47%	91.36%	86.90%
DTW-lim80	Symmkl	94.98%	89.69%	91.36%	87.18%
DTW-lim60	Euclidean	94.98%	89.13%	91.36%	84.95%
DTW-lim60	Absolute	94.98%	89.13%	91.36%	84.95%
DTW-lim60	Squared	95.54%	91.08%	90.52%	84.95%
DTW-lim60	Symmkl	95.26%	89.97%	92.20%	85.23%
DTW-lim40	Euclidean	93.03%	86.62%	91.08%	84.67%
DTW-lim40	Absolute	93.03%	86.62%	91.08%	84.67%
DTW-lim40	Squared	93.87%	88.02%	90.25%	83.00%
DTW-lim40	Symmkl	94.70%	87.46%	89.97%	82.17%
DTW-lim20	Euclidean	86.62%	80.77%	85.79%	75.48%
DTW-lim20	Absolute	86.62%	80.77%	85.79%	75.48%
DTW-lim20	Squared	86.90%	81.05%	84.12%	73.81%
DTW-lim20	Symmkl	88.85%	81.89%	84.67%	74.09%

Table 2 lists the result of the comparison of 5 s segments between all tracks. Here, the results for fuzzy entropy without the use of DTW are significantly improved compared to the entire track comparison, with an accuracy of 85% for the top three and 73% for the top one. When using entropy, the result is lower: 80% for the top three and 69% for the top one. The use of DTW again increased the accuracy to 92% for the top three, and 85% for the top one result in fuzzy entropy under the Euclidean distance. For entropy, the accuracy is also increased, but it is lower than fuzzy entropy: 88% and 77%. The use of absolute distance gives the same results as the Euclidean case. The squared and symmkl metrics give very small variations in the performance. Here, the use of constraints on the warping path is seen to be able to produce a slight performance improvement. One explanation behind

this can be that this constraint limits the high repetition of some entries in the warping path, which could falsely result in a high correlation coefficient value.

Overall, with the exception of comparing whole tracks without the use of DTW, the use of fuzzy entropy yields better results compared to entropy. The use of different metrics in the DTW gives small variations in the results. For simplicity, though, the default Euclidean distance can be used.

Table 2. Identification accuracy. Comparison of 5 s intervals.

Method	Metric	FuzzEn		En	
		Top 3	Top 1	Top 3	Top 1
no-DTW	-	85.51%	73.53%	80.50%	69.08 %
DTW	Euclidean	92.47%	85.23%	88.57%	77.99%
DTW	Absolute	92.47%	85.23%	88.57%	77.99%
DTW	Squared	93.59%	86.07%	87.46%	79.10%
DTW	Symmkl	93.59%	86.35%	87.74%	79.66%
DTW-lim40	Euclidean	92.47%	85.23%	88.57%	78.83%
DTW-lim40	Absolute	92.47%	85.23%	88.57%	78.83%
DTW-lim40	Squared	93.59%	86.35%	87.74%	79.94%
DTW-lim40	Symmkl	93.59%	86.07 %	88.30%	81.05%
DTW-lim20	Euclidean	93.31%	86.07%	90.25%	80.77%
DTW-lim20	Absolute	93.31%	86.07%	90.25%	80.77%
DTW-lim20	Squared	93.59%	86.35%	89.41%	81.33%
DTW-lim20	Symmkl	94.42%	86.07%	88.57%	83.00%
DTW-lim10	Euclidean	96.37%	89.97%	92.75%	85.79%
DTW-lim10	Absolute	96.37%	89.97%	92.75%	85.79%
DTW-lim10	Squared	96.10%	89.69 %	93.31%	86.35%
DTW-lim10	Symmkl	95.26%	89.69 %	93.03%	87.46%
DTW-lim05	Euclidean	94.70%	90.25%	92.20%	86.62%
DTW-lim05	Absolute	94.70%	90.25 %	92.20%	86.62%
DTW-lim05	Squared	94.98%	89.69%	91.92%	87.74%
DTW-lim05	Symmkl	94.70%	88.57 %	93.03%	87.46%
DTW-lim02	Euclidean	92.75%	86.35%	92.20%	81.33%
DTW-lim02	Absolute	92.75%	86.35%	92.20%	81.33%
DTW-lim02	Squared	91.92%	86.35%	90.80%	81.05%
DTW-lim02	Symmkl	91.92%	86.35%	90.80%	81.89%

Execution Time

The execution time to compare all tracks in the dataset and derive identification results for all different techniques when using fuzzy entropy is listed in Table 3. The tests were carried out using MATLAB Online, using an ASUS laptop computer with the Windows 11 operating system, a 13th Gen Intel Core i5-1335U 1.30 GHz processor, and 16GB of RAM. However, it should be noted that when running MATLAB Online, the simulations are not performed on the computer, but on MATLAB's cloud servers.

When comparing whole tracks, the correlation coefficient only requires 15 s for the dataset, and around 11 min when the DTW is applied. The Euclidean, absolute, and squared metrics in the DTW have relatively close execution times, with some seconds in difference. The symmkl metric, on the other hand, has a significantly higher execution time, and should thus be avoided. The execution time can be reduced when the warping path is restricted, for $Z = 100$ to around 6.5 min, and even less for stricter limits, at the cost of performance.

When comparing 5 s segments, the time is increased to around 2 min without the DTW step, and around 11 min with it. Different levels of path constraints from $Z = 40, 20, 10, 5, 2$ can further limit the execution time. The symmkl metric is again the slowest, and should be avoided.

Of course, these execution times can be improved in the future by considering several modifications and improvements. For example, the window sliding jump can be increased, which improves execution time at the cost of accuracy. Another modification would be to consider comparing only the middle part of the track for recordings that are longer than 30 s. This, though, may come at a cost of accuracy.

Table 3. Execution time (seconds).

Method	Metric	Whole Track	Method	Metric	5 s Interval
no-DTW	-	15	no-DTW	-	120
DTW	Euclidean	642	DTW	Euclidean	651
DTW	Absolute	669	DTW	Absolute	650
DTW	Squared	666	DTW	Squared	647
DTW	Symmkl	2095	DTW	Symmkl	1372
DTW-lim100	Euclidean	380	DTW-lim40	Euclidean	590
DTW-lim100	Absolute	393	DTW-lim40	Absolute	603
DTW-lim100	Squared	395	DTW-lim40	Squared	608
DTW-lim100	Symmkl	1003	DTW-lim40	Symmkl	1150
DTW-lim80	Euclidean	354	DTW-lim20	Euclidean	552
DTW-lim80	Absolute	364	DTW-lim20	Absolute	557
DTW-lim80	Squared	380	DTW-lim20	Squared	557
DTW-lim80	Symmkl	917	DTW-lim20	Symmkl	883
DTW-lim60	Euclidean	326	DTW-lim10	Euclidean	546
DTW-lim60	Absolute	335	DTW-lim10	Absolute	513
DTW-lim60	Squared	351	DTW-lim10	Squared	522
DTW-lim60	Symmkl	780	DTW-lim10	Symmkl	706
DTW-lim40	Euclidean	299	DTW-lim05	Euclidean	507
DTW-lim40	Absolute	329	DTW-lim05	Absolute	497
DTW-lim40	Squared	318	DTW-lim05	Squared	511
DTW-lim40	Symmkl	585	DTW-lim05	Symmkl	611
DTW-lim20	Euclidean	262	DTW-lim02	Euclidean	496
DTW-lim20	Absolute	262	DTW-lim02	Absolute	478
DTW-lim20	Squared	279	DTW-lim02	Squared	472
DTW-lim20	Symmkl	430	DTW-lim02	Symmkl	542

4. Conclusions

In this work, a method was developed to match the performances of Greek Orthodox chants using the fuzzy entropy of recordings in the frequency domain. With the use of statistical measures and DTW stretching, the identification accuracy may exceed 90%.

The methodology provided in this work can be further built upon and modified to improve its performance. Therefore, several goals are set for future studies, many of which are currently being developed. One goal is to make a thorough comparison of the different types of entropy that can be considered to see which is more suitable in characterizing the hymns. Examples include cross entropy [33], phase entropy [34], and fuzzy distribution entropy [35]. Moreover, in addition to using the correlation coefficient to evaluate similarity, other measures could be studied, like cosine similarity. Also, the derivation of a binary version of the entropy vector can be considered, using different binarization techniques.

The use of a binary vector can help reduce the comparison speed, but due to carrying limited information, it may negatively affect the accuracy. The combination of Mel frequency cepstral coefficients and entropy is also interesting. So, in general, there are further studies that can be performed to find a balance between the execution time for deriving the entropy vectors, their size, the comparison method, and the accuracy. It is also of interest to test the technique in other datasets of music genres where singing is the focus, like a capella, or other forms of religious chanting.

Another modification that can be considered is to develop a more detailed entropy measure. In the approach considered here, the entropy for the FFT magnitude over a whole time window is taken, leading to an entropy feature vector that is short in size but may be too crude to characterize more complex signals, like musical pieces. So, the limitation here is that the vector carries limited information, especially for more complex tracks that feature many instruments. To obtain more information, it would be better to divide the magnitude of the FFT signal into bins for each segmented interval, like in [13,16,32]. This will result in a matrix of entropy values, each corresponding to a time window and a frequency bin, called the entropygram. This entropygram can then be used to design a binary fingerprint for each track, which can be used for faster matching. Building on this, it may be possible to consider the 2D FFT transform in the fingerprint [36] as an advanced approach to applying image processing techniques to find similarities between performances.

So, in general, there are several improvements to be made in the future, which will greatly increase the potential implementability of this technique to commercial applications relevant to music identification, music teaching, and cultural tourism, especially when considering the dataset under study. Finally, although machine learning methods were not considered here, it would also be interesting to explore if the developed techniques could be integrated in other AI- or machine learning-based architectures for audio characterization.

In the future, it is within the authors' scope to implement the algorithm as a standalone GUI, for ease of use by any interested party. There is also a long-term goal of developing a mobile application for the algorithm, dedicated to the genre of orthodox chants, with accompanying historical and cultural information about each song, as well as information about famous orthodox churches and monasteries in Greece. Seeing the rise of cultural tourism in recent years, and the digitization of museums, historic locations, and cultural knowledge in general, there is fruitful ground for developing applications dedicated to the promotion of culture-related content. This will benefit a plethora of different groups, including music listeners, musicians, sound engineers, educators, historians, and other groups adjacent to the tourism industry. Our future studies aim at contributing further towards that direction.

Author Contributions: Conceptualization, L.M. and S.K.G.; methodology, L.M.; software, L.M.; validation, K.K., D.K., and S.K.G.; formal analysis, L.M., K.K., D.K. and S.K.G.; data curation, K.K.; writing—original draft preparation, L.M., K.K. and D.K.; writing—review and editing, L.M., K.K., D.K. and S.K.G.; visualization, L.M.; supervision, A.D.B., S.S., S.N., C.V., P.S., M.A.M. and S.K.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was carried out as part of the project 'Recognition and direct characterization of cultural items for the education and promotion of Byzantine Music using artificial intelligence' (Project code: KMP6-0078938) under the framework of the Action 'Investment Plans of Innovation' of the Operational Program 'Central Macedonia 2014 2020', which is co-funded by the European Regional Development Fund and Greece.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: The authors are thankful to the anonymous reviewers for their constructive feedback.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wang, A. An industrial strength audio search algorithm. In Proceedings of the Ismir, Washington, DC, USA, 26–30 October 2003; Volume 2003; pp. 7–13.
2. Son, W.; Cho, H.T.; Yoon, K.; Lee, S.P. Sub-fingerprint masking for a robust audio fingerprinting system in a real-noise environment for portable consumer devices. *IEEE Trans. Consum. Electron.* **2010**, *56*, 156–160.
3. Borkar, N.; Patre, S.; Khalsa, R.S.; Kawale, R.; Chakurkar, P. Music plagiarism detection using audio fingerprinting and segment matching. In Proceedings of the IEEE 2021 Smart Technologies, Communication and Robotics (STCR), Sathyamangalam, India, 9–10 October 2021; pp. 1–4.
4. Karasavvidis, K.; Kappelopoulos, D.; Moysis, L.; Boursianis, A.D.; Nikolaidis, S.; Sarigiannidis, P.; Goudos, S.K. Recognition of Greek Orthodox Hymns Using Audio Fingerprint Techniques. In Proceedings of the IEEE 2023 8th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM), Piraeus, Greece, 10–12 November 2023; pp. 1–6.
5. Kappelopoulos, D.; Moysis, L.; Karasavvidis, K.; Boursianis, A.D.; Goudos, S.K.; Nikolaidis, S. Byzantine Hymn Recognition with Audio Fingerprints Resistant to Noise, Tempo and Scale Changes. In Proceedings of the IEEE 2024 13th International Conference on Modern Circuits and Systems Technologies (MOCASST), Sofia, Bulgaria, 26–28 June 2024; pp. 1–4.
6. Su, X.; Nongpong, K. Audio Fingerprinting Based Music Recommendation Algorithm. In Proceedings of the 2023 6th International Conference on Algorithms, Computing and Artificial Intelligence, Sanya, China, 22–24 December 2023; pp. 52–56.
7. Kishor, K.; Venkatesh, S.; Koolagudi, S.G. Audio fingerprinting system to detect and match audio recordings. In Proceedings of the International Conference on Pattern Recognition and Machine Intelligence, Kolkata, India, 12–15 December 2023; Springer: Berlin/Heidelberg, Germany, 2023; pp. 683–690.
8. Htun, M.T.; Oo, T.T. Broadcast Monitoring System using MFCC-based Audio Fingerprinting. In Proceedings of the 2023 IEEE Conference on Computer Applications (ICCA), Cairo, Egypt, 28–30 November 2023; pp. 243–247.
9. Kritopoulou, P.; Stergiaki, A.; Kokkinidis, K. Optimizing human computer interaction for byzantine music learning: Comparing HMMs with RDFs. In Proceedings of the IEEE 2020 9th International Conference on Modern Circuits and Systems Technologies (MOCASST), Bremen, Germany, 7–9 September 2020; pp. 1–4.
10. Fang, J.T.; Day, C.T.; Chang, P.C. Deep feature learning for cover song identification. *Multimed. Tools Appl.* **2017**, *76*, 23225–23238.
11. Jin, Y.; Cai, W.; Chen, L.; Zhang, Y.; Doherty, G.; Jiang, T. Exploring the Design of Generative AI in Supporting Music-based Reminiscence for Older Adults. In Proceedings of the CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, 11–16 May 2024; pp. 1–17.
12. Tsallis, C. Entropy. *Encyclopedia* **2022**, *2*, 264–300.
13. Camarena-Ibarrola, A.; Chávez, E. Identifying music by performances using an entropy based audio-fingerprint. In Proceedings of the Mexican International Conference on Artificial Intelligence (MICAI), Apizaco, Mexico, 13–17 November 2006.
14. Yin, C.; Li, W.; Luo, Y.; Tseng, L.C. Robust online music identification using spectral entropy in the compressed domain. In Proceedings of the 2014 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Istanbul, Turkey, 6–9 April 2014; pp. 128–133.
15. Li, W.; Liu, Y.; Xue, X. Robust audio identification for MP3 popular music. In Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Geneva, Switzerland, 18–23 July 2010; pp. 627–634.
16. Camarena-Ibarrola, A.; Chávez, E. *Robust Audio-Fingerprinting With Spectral Entropy Signatures*; Universidad Michoacana de San Nicolás de Hidalgo: Morelia, Mexico 2007.
17. Camarena-Ibarrola, A.; Figueroa, K.; Tejeda-Villela, H. Entropy per chroma for Cover song identification. In Proceedings of the 2016 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC), Ixtapa, Mexico, 9–11 November 2016; pp. 1–6.
18. Ibarrola, A.C.; Chávez, E. A robust entropy-based audio-fingerprint. In Proceedings of the 2006 IEEE International Conference on Multimedia and Expo, Toronto, ON, Canada, 9–12 July 2006; pp. 1729–1732.
19. Camarena-Ibarrola, A.; Chávez, E.; Tellez, E.S. Robust radio broadcast monitoring using a multi-band spectral entropy signature. In Proceedings of the Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 14th Iberoamerican Conference on Pattern Recognition, CIARP 2009, Guadalajara, Jalisco, Mexico, 15–18 November 2009; Proceedings 14; Springer: Berlin/Heidelberg, Germany, 2009; pp. 587–594.
20. Chen, W.; Wang, Z.; Xie, H.; Yu, W. Characterization of surface EMG signal based on fuzzy entropy. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2007**, *15*, 266–272.

21. Zadeh, L. Fuzzy sets. *Inf. Control* **1965**, *8*, 338–353.
22. Zimmermann, H.J. *Fuzzy Set Theory—And Its Applications*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2011.
23. Azami, H.; Li, P.; Arnold, S.E.; Escudero, J.; Humeau-Heurtier, A. Fuzzy entropy metrics for the analysis of biomedical signals: Assessment and comparison. *IEEE Access* **2019**, *7*, 104833–104847.
24. Xiang, J.; Li, C.; Li, H.; Cao, R.; Wang, B.; Han, X.; Chen, J. The detection of epileptic seizure signals based on fuzzy entropy. *J. Neurosci. Methods* **2015**, *243*, 18–25.
25. Li, P.; Karmakar, C.; Yearwood, J.; Venkatesh, S.; Palaniswami, M.; Liu, C. Detection of epileptic seizure based on entropy analysis of short-term EEG. *PLoS ONE* **2018**, *13*, e0193691.
26. Chen, W.; Zhuang, J.; Yu, W.; Wang, Z. Measuring complexity using FuzzyEn, ApEn, and SampEn. *Med Eng. Phys.* **2009**, *31*, 61–68.
27. Xie, H.B.; Chen, W.T.; He, W.X.; Liu, H. Complexity analysis of the biomedical signal using fuzzy entropy measurement. *Appl. Soft Comput.* **2011**, *11*, 2871–2879.
28. Dong, C.; Rajagopal, K.; He, S.; Jafari, S.; Sun, K. Chaotification of Sine-series maps based on the internal perturbation model. *Results Phys.* **2021**, *31*, 105010.
29. Johny Elton, R.; Vasuki, P.; Mohanalin, J. Voice activity detection using fuzzy entropy and support vector machine. *Entropy* **2016**, *18*, 298.
30. Baldini, G.; Amerini, I. An evaluation of entropy measures for microphone identification. *Entropy* **2020**, *22*, 1235.
31. Pandria, N.; Kugiumtzis, D. Testing the correlation of time series using dynamic time warping. In Proceedings of the 27th Panhellenic Conference of Statistics, Thessaloniki, Greece, 13–15 June 2014.
32. Ramírez-Hernández, J.I.; Manzo-Martínez, A.; Gaxiola, F.; González-Gurrola, L.C.; Álvarez-Oliva, V.C.; López-Santillán, R. A Comparison Between MFCC and MSE Features for Text-Independent Speaker Recognition Using Machine Learning Algorithms. In *Fuzzy Logic and Neural Networks for Hybrid Intelligent System Design*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 123–140.
33. Laney, R.; Samuels, R.; Capulet, E. Cross entropy as a measure of musical contrast. In Proceedings of the Mathematics and Computation in Music: 5th International Conference, MCM 2015, London, UK, 22–25 June 2015; Proceedings 5; Springer: Berlin/Heidelberg, Germany, 2015; pp. 193–198.
34. Rohila, A.; Sharma, A. Phase entropy: A new complexity measure for heart rate variability. *Physiol. Meas.* **2019**, *40*, 105006.
35. Zhang, T.; Chen, W.; Li, M. Fuzzy distribution entropy and its application in automated seizure detection technique. *Biomed. Signal Process. Control* **2018**, *39*, 360–377.
36. Seetharaman, P.; Rafii, Z. Cover song identification with 2D Fourier transform sequences. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 616–620.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.