Data Mining and Knowledge Discovery
https://doi.org/10.1007/s10618-022-00835-2

An adaptive meta-heuristic for music plagiarism detection based on text similarity and clustering

Delfina Malandrino[1] · Roberto De Prisco[1] · Mario Ianulardo[1] · Rocco Zaccagnino[1]

Received: 14 August 2021 / Accepted: 6 April 2022
© The Author(s) 2022, corrected publication 2022

Abstract

Plagiarism is a controversial and debated topic in different fields, especially in the *Music* one, where the commercial market generates a huge amount of money. The lack of objective metrics to decide whether a song is a plagiarism, makes music plagiarism detection a very complex task: often decisions have to be based on subjective argumentations. Automated music analysis methods that identify music similarities can be of help. In this work, we first propose two novel such methods: a *text similarity-based* method and a *clustering-based* method. Then, we show how to combine them to get an improved (hybrid) method. The result is a novel *adaptive meta-heuristic* for music plagiarism detection. To assess the effectiveness of the proposed methods, considered both singularly and in the combined meta-heuristic, we performed tests on a large dataset of ascertained plagiarism and non-plagiarism cases. Results show that the meta-heuristic outperforms existing methods. Finally, we deployed the meta-heuristic into a *tool*, accessible as a Web application, and assessed the effectiveness, usefulness, and overall user acceptance of the tool by means of a study involving 20 people, divided into two groups, one of which with access to the tool. The study consisted in having people decide which pair of songs, in a predefined set of pairs, should be considered plagiarisms and which not. The study shows that the group supported by our tool successfully identified all plagiarism cases, performing all tasks with no errors.

Responsible editor: Johannes Fürnkranz.

✉ Delfina Malandrino
dmalandrino@unisa.it

Roberto De Prisco
robdep@unisa.it

Mario Ianulardo
mianulardo@unisa.it

Rocco Zaccagnino
rzaccagnino@unisa.it

[1] University of Salerno, Via Giovanni Paolo II, 132, 84084 Fisciano, SA, Italy

Published online: 12 May 2022

The whole sample agreed about the usefulness of an automatic tool that provides a measure of similarity between two songs.

Keywords Music plagiarism detection · Multi-objective optimization · Text similarity · Clustering · Evaluation study

1 Introduction

Music plagiarism, that is the act of copying, without permission, existing music covered by a copyright, is a controversial and debated topic. In some cases, the high amount of money at stake makes the issue quite important. Recently, famous singers were sued for alleged plagiarism (e.g. Ed Sheeran,[1] Justin Bieber[2]). Easy-to-use and effective automatic mechanisms for plagiarism detection could be helpful.

Legislations are unclear regarding the rules, or general criteria, that can be applied in order to detect a plagiarism. We focus on the similarity of the melodies. Even though laws do not state that the *melody* is the main deciding element, it is certainly one of the most crucial aspects. Moreover, we focus our attention on *pop music*. Thus we consider the problem of identifying similarities between melodies of pop music. In a pop song the melody is the crucial element.

Our contributions Our goal is that of building an automated system able to take as input two melodies, as MusicXML files, and provide an indication of their similarity (a percentage). The tool can be of help in the detection of plagiarisms cases. For this reason it can be seen as a plagiarism detection system. We propose a meta-heuristic which is the combination of two novel plagiarism detection methods. The first of such two methods is a *text similarity-based* method; the second one is a *clustering-based* method. These methods can be used singularly for detecting a music plagiarism. However, their combination into a meta-heuristic outperforms each one.

A distinctive characteristic that makes the proposed meta-heuristic different from other methods proposed in literature is its *adaptive* nature, i.e., it changes its behavior based on the *knowledge-base* on which it was built. Indeed, the detection mechanism is not based on pre-defined rules built by observing specific plagiarism examples, but rather on rules extracted directly from the *knowledge-base*. The knowledge-base, therefore, can be viewed as the set of *examples from which the system learns* how to discover music plagiarisms. This allows to overcome the limitations of existing methods, which have a poor generalization capability. To reach this result, both the text similarity-based and the clustering-based methods are built according to a reward mechanism that depends on the knowledge-base.

Other contributions are the *textual representation* for melodies used by the text similarity-based method and the *vectorial representation* used by the cluster similarity-based method, for representing melodies. Both the representations are novel.

[1] http://www.rollingstone.com/music/news/ed-sheeran-sued-for-more-than-20-million-for-alleged-plagiarism-20160608.

[2] http://www.billboard.com/articles/news/6605154/justin-bieber-usher-song-plagiarism-lawsuit-revived.

To assess the effectiveness of the proposed methods, we have carried out experiments on a testing set containing famous plagiarism cases. Results show that the adaptive meta-heuristic outperforms existing methods.

Finally, we deployed the meta-heuristic into a *tool*, made available as a Web application accessible as an online service. The tool allows to compare two songs using the proposed meta-heuristic and provides visual feedback to the user. Intended users of the tool are judges of court trials involving music plagiarism, music producers, music composers, or anyone interested in comparing the similarities of two songs. We have also conducted a preliminary study to evaluate its effectiveness, usefulness, and overall acceptance. We have involved 20 people, partitioned into two equally-sized groups. We asked both groups to listen to 6 pairs of songs and provide feedback about the existence of potential plagiarisms. We gave access to the online service to only one of the groups, and thus only people in this group have used the tool to get support for the judgment.

Limitations As already said, our approach takes into consideration only the melody. Considering only the melody is certainly a limitation. There are other criteria that can be considered. For example, one can carry out an accurate analysis of scores and recordings, that, beside the melody, can involve harmonic and rhythmic considerations, comparative listening, and other aspects, such as a similar (or same) title, similar lyrics, the likeliness of the defendant's knowledge about the plaintiff's song, and also factors depending on the psychological perception of melodic similarity.[3] Also the *genre* of the music can play a role in identifying the important aspects that need to be considered.

Some studies (Yuan et al. 2020; Flexer and Lallai 2019; Flexer and Grill 2016; Flexer 2014) show that aspects different from the melody can play a crucial role in assessing the case, although they are difficult to be described with quantitative metrics of similarity. These studies provide a deep analysis of the intrinsic difficulties that arise in the determination of the similarity between a pair of songs. These difficulties are tied to the subjective aspects of the problem. In particular, it is demonstrated that judgments of different listeners often disagree and even the same person can change his own judgments when repeating trials involving the same examples. The tool that we propose clearly is not able to overcome these intrinsic difficulties. Nevertheless, it can be of help in making the decision.

As discussed in Wolf and Müllensiefen (2011), music copyright laws vary, to some extent, from country to country. In the USA, the Copyright Law for Musical Works is found in Title 17 of the United States Code and is administered by the US Copyright Office. In the UK the relevant law is the Copyright, Designs and Patents Act 1988. In Italy, copyright is largely regulated by the Law of 22 April 1941, n. 633. There have been also attempts to harmonize copyright laws: the directive 2001/29/EC on the harmonization of certain aspects of copyright and related rights in the information society which was implemented by Legislative Decree of April 9, 2003, n. 68. To the best of our knowledge, none of the law jurisdictions provide explicit "rules" to recognize music plagiarism cases. Actually, legislations have only general laws for copyright infringement, that are applied to the specific case being judged.

[3] An interesting study about the psychological aspect of similarity, which addresses the perception of similarity and different psychological methodologies to measure it, can be found in Wolf and Müllensiefen (2011).

In several court cases the similarity of the melody is the fundamental aspect. As a concrete example, we bring the controversy that involved Michael Jackson and the Italian singer Albano Carrisi. In 1991, Jackson released the song *Will You Be There*, which has a melody almost equal to that of the 1987 song *I Cigni di Balaka* by Albano. In a first verdict, Jackson was sentenced to pay 4 million dollars. But at the end of a long and fluctuating trial held in Italy, in which Michael Jackson himself participated, it was established that both songs resumed a third song, namely *Bless you*, recorded by the Ink Spots in 1939, which in turn was based on a very old and traditional melody. In this case the melody was crucial, as stated in the verdict ("Corte d'Appello", Milan, Nov 24, 1999[4]).

In other law jurisdictions, there are numerous well-known cases of plagiarism in which the most relevant deciding element is the melody. We cite the case of Chuck Berry who sued the Beach Boys for the song *Surfin* whose melody is exactly the same as the one of *Sweet Little Sixteen*. As a result, the credits for writing the song were changed from Brian Wilson (Beach Boys) to Chuck Berry.

The rest of the paper is organized as follows: in Sect. 2 we briefly cite existing literature, and we highlight the key differences with our work. Then we present our music plagiarism detection method: in Sect. 3 we provide details about the approach based on text-similarity techniques, while in Sect. 4 we provide details about the approach based on clustering. Section 5 shows how the text similarity-based and the clustering-based plagiarism detection methods have been combined to obtain the proposed adaptive meta-heuristic. Section 6 describes a web application offering the proposed system as on online service. The section describes also a study that evaluates the effectiveness, usefulness, and the overall user acceptance of the tool. Finally, in Sect. 7 we conclude with some final remarks and future directions.

2 Related work

Several interesting attempts to classify existing algorithms that measure melodic similarity have been proposed (Müllensiefen et al. 2004; Müllensiefen and Pendzich 2009; Wolf and Müllensiefen 2011). Based on these studies, in this section, first, we identify some crucial classification criteria useful for the definition of a comprehensive classification scheme, and then, we discuss algorithms proposed in literature, by highlighting for each of them, the position within the defined scheme, and the main differences with the approach described in this work.

Classification scheme. With the aim of describing the main features of the proposed approach, we have identified the following three *classification criteria*:

1. *"how melody is represented?"*
2. *"how melodic similarity is perceived?"*
3. *"is the similarity dependent on the context in which it is measured?"*

The first criterion (*"how melody is represented?"*) focuses on the melody representation. As discussed in Müllensiefen et al. (2004), an algorithmic based similarity

[4] https://www.robertocaso.it/wp-content/uploads/2020/10/appello-milano-24-novembre-1999-carrisi-c.-jackson-sentenza-integrale.pdf.

measure uses an *abstract* representation of a melody sounding in time and pitch space. According to this definition, a melody can be handled as a time series (t_n, p_n), where t_n and p_n are the *rhythm* and *pitch*, respectively, at time n. Starting from this abstract representation, an algorithm for measuring melodic similarity can "view" a melody as a: (i) string, (ii) numeric feature-based entity (usually representable in space), and (iii) musical feature-based entity. In the first case (melody as string), we say that the algorithm uses a `symbolic` representation; this class coincides with the symbolic representations proposed in Müllensiefen et al. (2004). In the second case (melody as numeric entity), we say that the algorithm uses a `numeric` representation; this class includes the *vector measures* proposed in Müllensiefen et al. (2004), but also *geometric* measures (Ó'Maidin 1998), and applications of the Earth Mover's Distance (Typke et al. 2007). In the third case (melody as musical entity), we say that the algorithm uses a `musical` representation; this class includes the musical measure proposed in Müllensiefen et al. (2004).

The second criterion (*"how melodic similarity is perceived?"*) focuses on the perception of the melodic similarity. As discussed in Wolf and Müllensiefen (2011), algorithms can be classified in two general classes: *feature-based* (Tversky 1977), i.e., based on the perception of features that two objects share and the salience of these features, or *spatial* (Shepard 1957), i.e., for which the comparison of two objects is based on points in space and on distance metrics. Accordingly, we classify a measure as `feature-based`, in the first case, and as `spatial`, in the second one.

The third criterion (*"is the similarity dependent on the context in which it is measured?"*) focuses on the *nature* (adaptive or static) of the melodic similarity. If the perceived similarity changes based on the specific context in which the melodies are used (e.g., learned examples), then the measure has an *adaptive nature*, otherwise it has a *static nature*. Accordingly, an algorithm can be classified as either `adaptive` or `static`.

Using the above described criteria, in the following, we provide classification details of known algorithms. As we will see, an algorithm can also belong to more than one class.

Algorithms for measuring melodic similarity. Several algorithms for measuring the similarity between *monophonic* melodies, have been proposed in literature. As discussed in Müllensiefen and Pendzich (2009), they range from relatively simple and widely-used string matching-based techniques, to algorithms which make use of statistical information about the prevalence of musical structures in large music collections, up to geometric measures, and so on.

The most basic example of string matching-based technique is the well-known *edit distance*, defined over two strings of symbols from the same alphabet as the minimum number of operations, i.e., insertions, deletions, and substitutions, needed to transform one of the strings into the other one. Several attempts of applying it and its "weighted" version (in which the weights are the duration of the notes) to melodies represented as strings, can be found in literature (e.g., Mongeaul and Sankoff 1990; Crawford et al. 1998).

Another family of algorithms widely used in modern text retrieval is that based on the notion of *n-gram*.[5] Several applications of such algorithms to music have been proposed (e.g., Downie 1999; Uitdenbogerd 2002). The performance of some n-gram algorithms, such as *Ukkonen* and *Sum Common*, have been also compared with other similarities measures in Müllensiefen and Pendzich (2009); Ukkonen reflects the notion of "difference" between two strings by summing the differences of the frequencies of all the n-grams not occurring in both strings, while Sum Common reflects the notion of similarity, by summing frequencies of n-grams occurring in both strings. Algorithms based on the edit distance and on n-grams, can be classified as `symbolic`, `spatial`, and `static`.

Based on the idea of similarity as a function of the complexity required to transform the representation of one into the representation of the other (Hahn et al. 2003), a music application of the *Earth Mover's Distance* has been proposed in Typke et al. (2007). Using a mapping of (usually) onset time and pitch onto a subset of a two-dimensional space, where the note's duration is represented by the weight of each point, the similarity of two melodies is measured as the work required to convert one melody into the other one, given by the effort needed to change the location and weight of the point set representing the first melody. This algorithm can be classified as `numeric`, `spatial`, and `static`.

Tversky (1977) proposed a different concept of similarity, based on a theory that takes into account the perception of features that two objects share and the salience of these features. According to this theory, the human perception is mainly influenced by well-defined features and also the salience of these features, where the salience is also the result of habituation, knowledge, and interest in a specific domain. Based on this idea of similarity, an interesting weighting scheme developed from a large corpus of pop melodies has been proposed in Müllensiefen and Pendzich (2009). The assumption is that two melodies sharing more frequent features are less similar than two melodies sharing rather infrequent features. Several variations of the Tversky algorithm have been proposed in Müllensiefen and Pendzich (2009). An interesting application of this approach can be found in Cason and Müllensiefen (2012), in which the authors observed that involving a corpus of music in the similarity comparison makes the similarity value depend not only on the two melodies being compared, but also on the corpus from which the perceptual weights are derived; and this represents a desirable feature for a similarity comparison approach, since it models an average and idealized listeners' stylistic knowledge and expectation. The basic Tversky algorithm can be classified as `symbolic`, `feature-based`, and `static`, while its weighting variants are `symbolic`, `feature-based`, and `adaptive`.

Several works in literature have exploited the algorithms described so far, to propose more elaborate techniques for measuring melodic similarity.

As an example, in Robine et al. (2007) the authors observed that most of the existing systems for the comparison of symbolic music are based on string matching algorithms and represent music as sequences of notes. However, the direct adaptation to the musical context often does not lead to an accurate detection algorithm, since very different sequences can represent very similar musical pieces. To face this problem, they pro-

[5] An n-gram is any substring of n consecutive symbols of a string.

posed a system which mainly considers melody but takes also into account music theory elements. Such a system can be classified as `spatial`, and `static`, according to the second and third criteria described above, respectively. However, concerning the first criterion, the integration of musical elements in the string representation can be considered both `symbolic` and `musical`.

A different idea (than those described up to now) of melodic similarity, can be found in the family of measures named *geometric* in Müllensiefen and Pendzich (2009), that can be classified as `numeric`. As an example, in Ó'Maidin (1998), the authors proposed a geometric measure of the difference between two melodies. The melodies are modelled as monotonic pitch-duration rectilinear functions of time, and the difference between the two melodies is given by the minimum area between the two polygonal chains, allowing vertical translations. This is a typical example of `numeric` algorithm. Furthermore, it is also `spatial` and `static`.

Other approaches exploit visualization techniques (e.g., De Prisco et al. 2016, 2017a, c; Park et al. 2019). Several representations of the melodic relations among songs have been used and studies investigating the intuitiveness and accuracy of the proposed representations, highlighted important aspects: (i) the performance is closely related to the ability of a melodic representation to grasp the most useful aspects for discovering melodic relations, (ii) the combination of different approaches or representations often provides unexpected and interesting results [as observed also in De Prisco et al. (2017c) and in Robine et al. (2007)], and (iii) as observed in Cason and Müllensiefen (2012), the definition of an *adaptive* approach, in which the similarity depends not only on the melodies being compared but also on information about the corpus in which the melodies are defined, could be decisive for a similarity comparison approach.

The proposed approach The main characteristic of the proposed approach, is that it tries to merge the advantages offered by several of the classes of algorithms described above. Specifically, the idea is to use a very simple representation, "close" to the original musical one but at the same time we want to exploit in a more efficient way the existing clustering algorithms by using a "vectorial" transformation of the proposed textual representation, in order to obtain a greater generalization capability. To the best of our knowledge, our meta-heuristic is one of the first attempts at defining an adaptive method based on different types of representations. Such a meta-heuristic is `spatial` and `adaptive` according to the second and third criteria, respectively, and both `symbolic` and `numeric` according to the first criterion. We remark that, the specific textual representation, and other technical details that the reader can find in the following sections, are also novel.

We recall, as discussed in Sect. 1, that we focus only on the melody, without considering other musical aspects such as harmony, sound properties of specific performances (timbre, time, dynamics), and auditory perception of similarity.

Table 1 summarizes the comparison between our work and the previous ones. Specifically, we report details about the classification criteria described above, and named in the table as: (i) **representation**—*"how melody is represented?"* (`symbolic`, `numeric`, `musical`), (i) **perception**—*"how melodic similarity is perceived?"* (`spatial`, `feature-based`), (iii) **nature**—*"is the similarity dependent on the context in which it is measured?"* (`adaptive`, `static`). When necessary,

Table 1 Algorithms classification

References	Representation	Perception	Nature
Mongeaul et al. (1990)◇	symbolic	spatial	static
Crawford et al. (1998)◇	symbolic	spatial	static
Ó'Maidin (1998)•	numeric	spatial	static
Downie (1999)⋇	symbolic	spatial	static
Uitdenbogerd (2002)⋇	symbolic	spatial	static
Typke et al. (2007)⊕	numeric	spatial	static
Robine et al. (2007)	symbolic/musical	spatial	static
Müllensiefen et al. (2009)⋇⋇	symbolic	spatial	adaptive
Müllensiefen et al. (2012)+	symbolic	feature-based	adaptive
De Prisco et al. (2016)	symbolic	spatial	static
De Prisco et al. (2017)	numeric/musical	feature-based	static
Our work	symbolic/numeric	spatial	adaptive

◇ (edit distances), ⋇ (*n*-gram), ⋇⋇ (ukkonen, sum common),
• (geometric), + (weighted Tversky), ⊕ (earth mover's distance)

for each work we also highlighted the specific basic approach to which we are interested. For example, in Müllensiefen and Pendzich (2009), several algorithms have been compared. However in this specific case, we are interested in the Ukkonen and Sum Common algorithms (highlighted with ⋇⋇) used in the experiments.

3 The text similarity-based method

In this section, we present our first music plagiarism detection method, the one based on text-similarity techniques. Figure 1 shows an overall view.

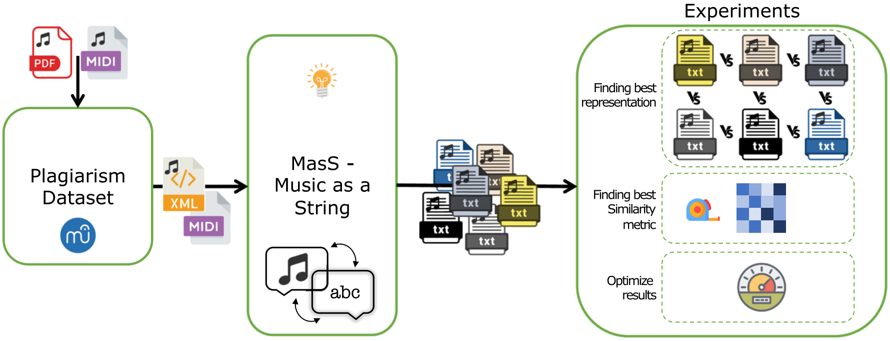


Fig. 1 The methodology followed to build the text similarity-based method for music plagiarism detection: (i) *Knowledge-base construction*, (ii) *MasS—Music as String*, (iii) *Experiments*

3.1 Knowledge-base construction

The knowledge-base can be thought of as the set of examples from which to learn the detection rules for discovering music plagiarisms. The examples can be *positive*, i.e., plagiarism cases, and *negative*, i.e., non-plagiarism cases. The choice of this set is crucial.

Unfortunately, there is no publicly available digitized dataset of plagiarism cases, in which the music sheets have a uniform format that is easy to use. For our experiments, we have used, as a starting point, a database made publicly available by the *George Washington University Law School & Columbia Law School*[6]; the database contains 296 plagiarism disputes brought to the United States federal court since 1840s.

Since among these 296 cases, there are very few cases whose verdict is based "exclusively" on the melody (recall that, as explained in the Introduction, we consider only the melody), we have selected only the cases whose verdict considered the melody as one of the decisive elements, completely excluding cases where only other aspects like lyrics, sound, and accompaniment, were decisive. The filtered set so obtained consisted of 164 cases.

We have split such 164 cases in *positive* and *negative* examples. Specifically, in this subset of positive examples we have included: (i) *effective* plagiarism cases, i.e., cases concluded with a plagiarism sentence, and (ii) *alleged* plagiarism cases, i.e., cases concluded with an agreement between the parties. Conversely, in the set of negative examples, we have included all other cases.

Since the starting database did not contain, for every song, all the needed information, we have carried out exhaustive searches on the web to recover music sheets and, when possible, documents related to the judicial sentences. In some cases we have transcribed the melody. In total, we have collected music sheets of songs involved in 52 positive examples and in 112 negative examples. We have digitalized the music representation, using the `musicxml`[7] format. As we will see in Sect. 3.3, several experiments have been performed to study how the selection of positive and negative examples (and their size) to be included in the knowledge-base, can affect the tuning of the proposed method.

3.2 MasS: Music as String

The music sheet of each melody contained in the knowledge-base has been converted into a text string through a *textual representation* technique. We have experimented with several representations. We started with simple ones that considered only the notes (e.g. *p-C5-B4-D5-C5-B4-Bb4-G#4-A4-A4-A4...*) or only the intervals between notes (e.g. *-1+3-2-1-1-2+1+0+0-1+2-1-1-1...*). Simple representations, however, suffer of several drawbacks. For example in the notes-only representation a melody transposed into a different key will be considered completely different from the original one; the intervals representation ignores completely the note lengths and notes lengths are an important part of the melody. Thus we tried several other representations, some of

[6] https://blogs.law.gwu.edu/mcir/.

[7] https://www.musicxml.com/.

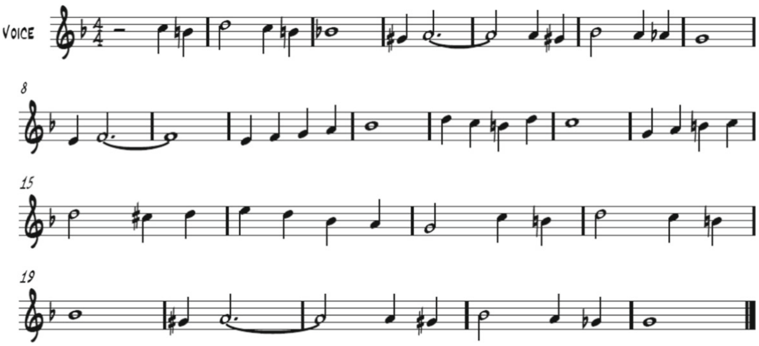


Fig. 2 A fragment of a melody's music sheet in the knowledge-base: *"Near you"*, by Francis Craig

them that used symbols equivalent to the ones used in the actual music score (e.g., a *note-length* alphabet $D = \{\ w,\ h,\ q,\ e,\ s\}$ containing a symbol for each possible note length: $w =$ *whole* (4/4), $h =$ *half*, 2/4, $q =$ *quarter*, 1/4, $e =$ *eight*, 1/8, and $s =$ *sixteenth*, 1/16), and finally used the following PINL representation.

PINL(P*itch*-I*nterval with* N*ormalized* L*ength*). The melody is represented as a text string that uses the following symbols. A "*p*" (pause) for the rests, an integer that represents the interval, measured in semitones, with respect to the previous note (the first note is ignored). The integer is negative for jumps downwards and positive for jumps upwards; there is no need for a sign for an interval of 0 (repeated notes). After the integer, there is a string of one or several "*b*". The "*b*" encodes a *base* note length and is repeated as many times as needed. For example, if the symbol "*b*" represents a quaver note length, the length of a dotted quarter note is represented as "*bbb*". The base note length value represented by the "*b*" is computed in a dynamic manner. Given a melody $m = \{n_1, \ldots, n_k\}$, we first compute the minimum note length d_{min} in m. Then, we compute the frequency f_{min} of the notes with length d_{min} in m. If $f_{min} < \delta$, where δ is a *fixed minimum frequency*, then the notes with length d_{min} are simply ignored, and we repeat the same argument on the remaining notes. The parameter δ can be fine-tuned, testing the results with experiments; we have used $f_{min} = 0.03$.

As an example, consider the melody in Fig. 2. For this melody the base note length, represented by b, is the quarter note and the string representation of the melody is:

pbb-1b+3bb-2b-1b-1bbbb-2b-1bbbbb0b-1b+2bb-1b-1b-1bbbb-3b-2bbbbbbb-1b
+1b+2b+2b +1bbbb+4b-2b-1b+3b-2bbbb-5b+2b+2b+1b+2bb-1b+1b+2b-2b-
4b-1b-2bb+5b-1b+3bb-2b-1b-1bbbb-2b-1bbbbb0b-1b+2bb-1b-3b+1bbbb

We remark that the PINL representation is the result of a study that involved several other representations and the specific choices made for this representation are attempts to have a better behavior with respect to specific aspects. For example the dynamic normalized note length representation helps in identifying similarity of scores written with different time signatures.

Table 2 PINL evaluation

	Thr	tp	fp	fn	tn	acc	rec	pre
CS	0.70	35	2	17	50	0.83	0.67	1.00
JRS	0.60	50	4	2	48	0.90	0.96	0.92
DC	0.70	34	2	18	50	0.82	0.65	1.00
OC	0.90	44	11	8	41	0.92	0.84	0.89
JS	0.40	44	3	8	49	0.89	0.84	0.91

3.3 Experiments

We have considered the following well-known text-similarity metrics (see Gomaa et al. (2013) for a survey on text-similarities metrics): *Cosine similarity* (CS), *Jaro similarity* (JRS), *Dice's coefficient* (DC), *Overlap coefficient* (OC), and the *Jaccard similarity* (JS). We recall that for each of the considered text-similarity metrics we can obtain a value in the range [0, 1], where 0 means "different melodies" while 1 means "same melody". Since the metric will be used to provide a binary answer, we need to decide a threshold t, that will determine the answer: if $Metric(m_1, m_2) > t$ then melodies m_1 and m_2 are similar, while if $Metric(m_1, m_2) \leq t$, they are classified as different. Our goal is to select the best choice of the pair ($Metric$,t).

To this end, for each $Metric \in \{$CS,JRS,DC,OC,JS$\}$ we proceeded as follows:

1. Compute the similarity value for each pair of melodies in the knowledge base, using *Metric*. Let $t_1, t_2, t_3, \ldots$ be all such values.
2. Consider each t_i as a possible threshold for the plagiarism detection with *Metric*.
3. Evaluate the "behavior" of the pair ($Metric$,t_i) with respect to the knowledge base: That is, take all the plagiarism cases (m_1, m_2) and check whether $Metric(m_1, m_2) > t_i$; these are the true positives. Take all non plagiarisms cases (m_1, m_2) and check whether $Metric(m_1, m_2) > t_i$; these are the false positives.
4. Consider the best threshold Thr, that is the value t_i that maximizes the true positives and minimize the false positives. Since this is a multi-objective optimization problem, consider the *Pareto front*[8];
5. Among all the solutions in the Pareto front, select the one that minimizes the number of false positives; if such a solution is not unique, select one that reaches the highest number of true positives.

Table 2 shows the results obtained. Specifically, the first part of the table reports the similarity threshold (Thr), and the results obtained in terms of *true positives* (*tp*), *false positives* (*fp*), *true negatives* (*tn*), and *false negatives* (*fn*), while the second part reports the *accuracy* (*acc*), *recall* (*rec*), and *precision* (*pre*), for each text-similarity metric considered.

Using these values, we can decide the best metric, as the one that maximizes the true positives and minimizes the false positive. As can be seen from the table, the Pareto front contains CS and JRS. To choose a single metric, we look for the one with the best false positive rate, which is CS.

[8] The Pareto front is a standard technique used to compare elements using more than one optimization criterion. All "non-dominated" elements are kept as best solutions.

Improvement We considered the following question: instead of defining a method based on the application of a single text-similarity metric, is it possible to automatically create an ensemble of metrics that allows to optimize performance, based on the examples contained in the knowledge-base? To tackle such a problem, we first define the notion of *best ensemble* of a knowledge-base:

1. Let $\mathcal{K} = \mathcal{K}_p \cup \mathcal{K}_n$ be the knowledge-base, where $\mathcal{K}_p$ is the set of *plagiarism* cases, and $\mathcal{K}_n$ is the set of *non-plagiarism* cases. Let $\mathcal{M}$ be a set of text-similarity metrics.
2. Let

$$\mathcal{E} = \bigcup_{\substack{k=1,\ldots,|\mathcal{M}| \\ s_j \in \mathcal{M} \\ \forall s_i, s_j, s_i \neq s_j}} (s_1, \ldots, s_k)$$

   be the set of possible *ensembles* which can be constructed on $\mathcal{M}$, i.e., the set of possible k-tuple of distinct metrics in $\mathcal{M}$, with $k = 1, \ldots, |\mathcal{M}|$.
3. For each ensemble $e = (s_1, \ldots, s_k)$ in $\mathcal{E}$, apply each text-similarity metric $s_i \in e$ on $\mathcal{K}$, for $i = 1, \ldots, k$. Let th_i be the threshold for each s_i; we define the *threshold for the ensemble e* as the vector $t_e = [th_1, \ldots, th_k]$.
4. Select the best ensemble, i.e., the ensemble that maximizes the number of true positives and minimizes the number of false negative. In case of more ensembles with the same best performance, choose the one with the smallest k. To this end, compute the Pareto front on the results obtained by all the ensembles in $\mathcal{E}$. If more ensembles occur in such a set, select the one with the lowest number of false positives.

The overall method Given the best ensemble, we establish whether a pair of melodies (m_1, m_2) are a plagiarism case, as follows: let $e = (s_1, \ldots, s_k)$ be the best metric ensemble, let t_e be the threshold for e. Compute the value for each metric $s_i \in e$ on (m_1, m_2). Let t_i be the threshold of s_i on (m_1, m_2), if $t_i \geq t_e[i]$ for every $i = 1, \ldots, k$, then m_1 and m_2 are a case of plagiarism.

The choice of $\mathcal{K}_p$ and $\mathcal{K}_n$ Observe that the performance of the proposed approach depends on the tuning of the parameters and the thresholds, which, in turn, depends on $\mathcal{K}$. In order to understand how these tuned values depend on $\mathcal{K}$, starting from the knowledge-base described in Sect. 3.1, consisting of 52 plagiarism cases and 112 non-plagiarism cases, we have built several $\mathcal{K}_p$ (as subsets of the 52 plagiarism cases) and $\mathcal{K}_n$ (as subsets of the 112 non-plagiarism cases) of different size. Specifically, we have considered $|\mathcal{K}_p| = 10, 20, 30, 40, 52$, and $|\mathcal{K}_n| = 10, 20, 30, 40, 52, 60, 70, 80, 90, 100, 112$. Notice that we considered the case $|\mathcal{K}_n| = 52$ instead of the more natural choice of 50, because this is the largest size for $\mathcal{K}_p$. Moreover we verified that the case with cardinality $|\mathcal{K}_p| = 50$ returned substantially the same results as the case with cardinality $|\mathcal{K}_p| = 52$. The same choice has been made for the largest size of $|\mathcal{K}_n| = 112$. For each combination of $\mathcal{K}_p$ and $\mathcal{K}_n$, with each set consisting of randomly chosen elements, we have a *configuration* that we indicate with $\langle |\mathcal{K}_p|, |\mathcal{K}_n| \rangle$. For each configuration, we have applied 10 times the overall method described above, using $\mathcal{M} = \{$cs, jrs, dc, oc, js$\}$.

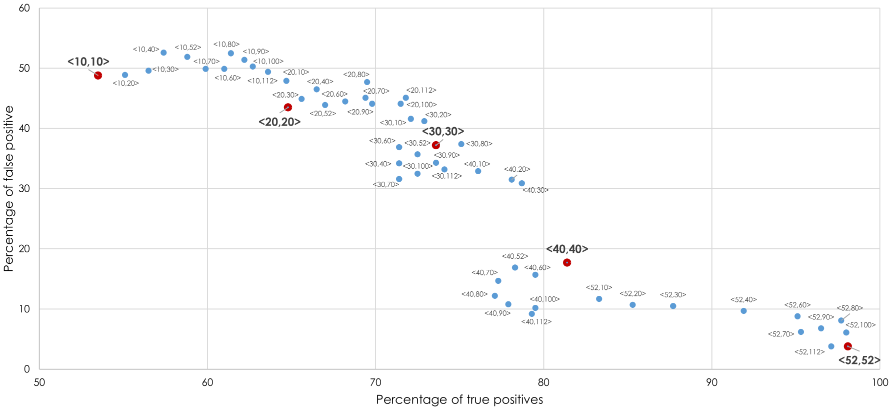


Fig. 3 Results of experiments performed to study how the performance of the proposed approach and the tuning of parameters and thresholds are affected from $\mathcal{K}$. Each point corresponds to (and is labeled with) a "$\langle\mathcal{K}_p$,$\mathcal{K}_n\rangle$" configuration, and the x-coordinate (resp. y-coordinate) is the percentage of true positives (resp. false positive) detected by the overall method. Balanced configuration ($|\mathcal{K}_p| = |\mathcal{K}_n|$) are highlighted with red points (Color figure online)

Figure 3 reports the average results obtained. Each point represents one specific configuration $\langle|\mathcal{K}_p|, |\mathcal{K}_n|\rangle$ and is labeled with the text "$\langle|\mathcal{K}_p|, |\mathcal{K}_n\rangle|$". For example, the point labeled with $\langle 10, 40\rangle$ corresponds to the configuration using $|\mathcal{K}_p| = 10$ and $|\mathcal{K}_n| = 40$. For each point The x-coordinate (resp. y-coordinate) is equal to the percentage of true positives (resp. false positive) detected by the overall method, when the $\langle|\mathcal{K}_p|, |\mathcal{K}_n|\rangle$ configuration corresponding to such a point is used. The percentage of true positives detected is computed with respect to $|\mathcal{K}_p|$, as $\frac{tp}{|\mathcal{K}_p|} \times 100$, where tp is the total number of true positives detected. The percentage of true negatives is computed similarly, but considering $|\mathcal{K}_n|$. For example, for the point corresponding to (and labeled with) $\langle 10, 40\rangle$, the x-coordinate is 57.4 and the y-coordinate is 52.6. This means that, by using the $\langle 10, 40\rangle$ configuration, the overall method is able to detect 57.4% of true positives, obtaining a 52.6% of false positives.

As described above, the goal of the overall method is to maximize the number of true positives and, at the same time, minimize the number of false positives. From a graphical point of view, in Fig. 3, the quality of a point increases as its position drifts down and to the right. For example, the point labeled with $\langle 30, 70\rangle$ corresponds to a better configuration than the point labeled with $\langle 10, 40\rangle$. Furthermore, balanced configurations, that is configurations with sets of the same size, are indicated with larger red points and boldface labels. As we can see, in general, as the value of $|\mathcal{K}_p|$ increases, the ability to detect more true positives tends to increase. Indeed, it is possible to see how the x-coordinates of the points tend to go to the right as we go from $|\mathcal{K}_p| = 10$ to $|\mathcal{K}_p| = 20$, and so on up to $|\mathcal{K}_p| = 52$. This behavior means that an increase in examples of plagiarism cases in the knowledge-base actually corresponds to an increase in the ability to detect cases of plagiarism. However, it is not possible to always observe the same behavior for $\mathcal{K}_n$. As we can see, for the configurations $\langle|\mathcal{K}_p|, |\mathcal{K}_n|\rangle$ with $|\mathcal{K}_p| = 10, 20$, the results appear unstable, i.e., increasing the $|\mathcal{K}_n|$ value does not always correspond to a better ability to avoid false positives (i.e. decrease the percentage of false positives). This instability tends to lessen for $|\mathcal{K}_p| > 30$. In particular, for $|\mathcal{K}_p| > 30$, we can see how balanced configurations $\langle|\mathcal{K}_p|, |\mathcal{K}_n|\rangle$ are always better than all configurations $\langle|\mathcal{K}_p|, z\rangle$, for each $z \neq |\mathcal{K}_p|$. For example, the configuration $\langle 52, 52\rangle$ is always better than $\langle 52, z\rangle$, with $z = 10, 20, 30, 40, 52, 60, 70, 80, 90, 100, 112$. For such a configuration, we always obtained $e = (\textsc{cs}, \textsc{dc}, \textsc{oc})$ as the best performing ensemble, with $tp = 51$ and $fp = 2$.

In conclusion, from this experimental data, we conclude that it is always preferable to work with balanced configurations. Since the largest balanced configuration we can build is $\langle 52, 52\rangle$, in the following experiments we have used the knowledge-base described in Sect. 3.1, $\mathcal{K} = \mathcal{K}_p \cup \mathcal{K}_n$, with $|\mathcal{K}_p| = |\mathcal{K}_n| = 52$ (with the elements of $\mathcal{K}_n$ randomly chosen among the 112 available).

Finally, we remark that the method described is general, and thus it is possible to consider other sets for $\mathcal{K}$ and $\mathcal{M}$.

4 The clustering-based method

In this section, we present our second music plagiarism detection method, the one based on the clustering technique. It consists of the following three phases, illustrated in Fig. 4:

- *Knowledge-base construction* identification of a reliable source of music plagiarism and non-plagiarism cases and digitization of the songs involved (Sect. 4.1);
- *MasV - Music as Vector* definition of an *embedding* technique that converts the text-based representation of one melody into a vector of real numbers (Sect. 4.2);
- *Experiments* use of well-known clustering models to evaluate which is the most effective clustering model for discovering music plagiarisms (Sect. 4.3).

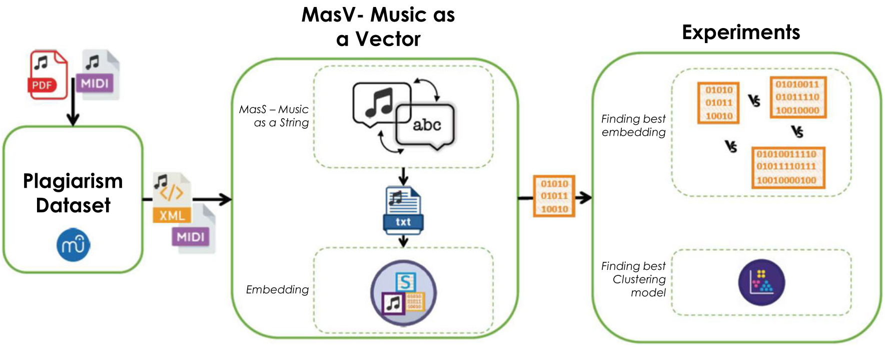


Fig. 4 The methodology followed to build the clustering-based method for music plagiarism detection: (i) *Knowledge-base construction*, (ii) *MasV—Music as Vector*, (iii) *Experiments*

4.1 Knowledge-base construction

The knowledge-base used for the construction of the clustering-based method is the same as the one proposed in Sect. 3.1, and it has the same function: allow the system to learn from the examples.

4.2 MasV: Music as Vector

In this phase, each melody in the knowledge-base described in Sect. 3.1 has been first converted into text by using the PINL representation described in Sect. 3.2. Then, the textual representation of each melody has been converted into a vector of real numbers.

Several works have been proposed in literature to transform a string into a vector. Some simple models represent words with random vectors or with integer numbers. The drawback of such models is that they do not represent any semantic connections between words. Other techniques, such as `word2vec`[9] or `Glove`,[10] called *semantic models*, are trained on a large corpus of existing texts, and as a result, the words in similar contexts are represented by proximal vectors.

However, semantic models contain some information about the meaning of words extracted while training on large texts corpora, but they operate with a fixed vocabulary. This is a significant drawback since, if a text contains a significant number of words that do not belong to the vocabulary of a semantic language model, then the model is non-efficient. To solve the problem, it would be appropriate to create embeddings based only on words spelling and relate similar vectors to similarly spelled words. One of the most interesting techniques is `char2vec`, a recent *word embedding technique*, particularly suitable to learn words not belonging to any dictionary. For example, `char2vec` proved to be efficient in learning some patterns hidden within HTTP packets (Wang et al. 2018), or to understand the use of words, abbreviations, codes not present in the dictionary and strings used in biology (Hussain et al. 2018). By empirical observations during our experiments, we have noticed that such a technique is particularly suitable for embedding music representations such as those described in Sect. 3.2. Thus, we have decided to use such a technique for embedding the strings obtained using the PINL representation.

Specifically, given a string, `char2vec` produces a vector of a fixed *embedding* size. The `char2vec` library allows to use custom sizes. For our experiments we used the following ones: 50, 100, 150, 200, and 300. Recall that we use $\textsc{pinl}(m)$ to represent the melody m as a string; we denote $\textsc{cv}_l(\textsc{pinl}(m))$, or just $\textsc{cv}_l$ for brevity, the vector of size l produced by applying `char2vec` to the string $\textsc{pinl}(m)$.

4.3 Experiments

In this section, we describe the experiments carried out define the details of the proposed method. As for the text similarity-based method described in Sect. 3, also in this

[9] https://www.tensorflow.org/tutorials/text/word2vec.

[10] https://nlp.stanford.edu/projects/glove/.

Table 3 Results obtained with the CV_{50} representation

Clustering model	sc	tp	#cluster	Setting
`birch`	0.53	28	8	threshold = *0.5*, branching_factor = *50*
`gaussianmixturemeasure`	0.55	34	10	covariance_type = *"full"*
`minibatch(100)`	0.58	26	9	batch_size = *100*
`minibatch(250)`	0.60	28	9	batch_size = *250*
`miniBatch(500)`	0.67	35	10	batch_size = *500*
`minibatch(1000)`	0.66	31	11	batch_size = *1000*
`agglomerative`	0.62	31	11	affinity = *"euclidean"*, linkage = *"ward"*
`k-means`	0.73	31	9	algorithm = *"elkan"*
`spectral`	0.80	50	9	eigen_solver = *"arpack"*, affinity = *"rbf"*

sc silhouette coefficient, *tp* average number of true positives, *#cluster* number of clusters used, *setting* variant of the model (in terms of set parameters), which showed the best silhouette coefficient

case, the *adaptive aspect* consists in the fact that the proposed method will be defined on a knowledge-base.

Preliminary analysis We have used the following well-known clustering models: `gaussianmixturemodel`, `k-means`, `birch` (Zhang et al. 1996), `agglomerative`, `minibatch` (Sculley 2010), and `spectral`. $\mathcal{K} = \mathcal{K}_p \cup \mathcal{K}_n$ is the same as before. The interested reader should refer to Xu and Wunsch (2005) for more information about clustering algorithms.

1. For each combination *"MasV representation/clustering model"*, and for a number of clusters in $1, \ldots, |\mathcal{K}_p| = 52$, we have trained 20 times the clustering model on $\mathcal{K}_p$.
2. We selected the best combination *"MasV representation/clustering model"*, i.e, the one which best learned the plagiarism examples in $\mathcal{K}_p$; specifically, we built the *pareto front* by using two optimization criteria: *maximize* the average number of *true positives* and *maximize* the average *silhouette coefficient*. In this case, a true positive is a case in $\mathcal{K}_p$, whose pair of melodies are correctly inserted in the same cluster.

All the clustering models have been implemented using Scikit-learn,[11] the Python library used for machine learning-based applications. Regardless of the clustering model used, the best results have always been obtained by using the representation CV_{50}. In Table 3 we provide the results. For each model, we report the *setting*, i.e., is the variant of the model in terms of set parameters which showed the best silhouette coefficient (see Python Scikit-learn library documentation for further details). As we can see, the best result for each clustering model has been obtained by setting the value for the *number of cluster* parameter in the range [8, 11]. In particular, the best clustering model turned out to be `spectral` with *#cluster* = 9. Such a model, indeed, is able to successfully detect 97% of the plagiarism cases in $\mathcal{K}_p$ ($tp = 50$).

In Fig. 5, we also show an overview of the application of the clustering models trained using the vectors obtained by using the CV_{50} representation. Each bar graph is

[11] https://scikit-learn.org/stable/.

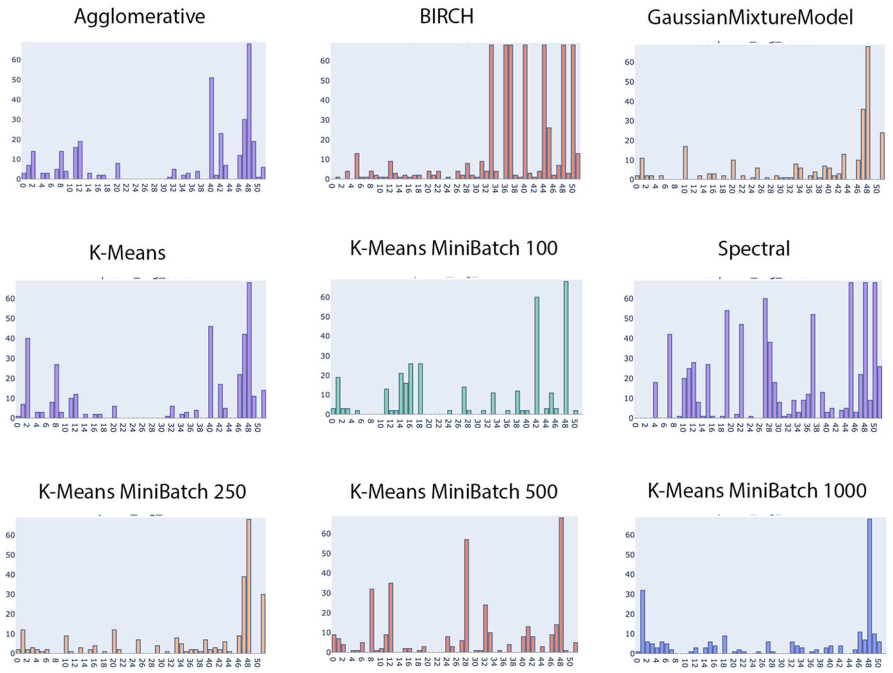


Fig. 5 Overview of the application of the clustering models trained using the vectors obtained by using the CV_{50} representation. Each bar graph is structured as follows: on the X axis there are all the plagiarism cases in $\mathcal{K}_p$, while on the Y axis the number of times the model was able to insert a plagiarism case in $\mathcal{K}_p$ in the same cluster

structured as follows: on the X axis there are all the plagiarism cases in $\mathcal{K}_p$, while on the Y axis the number of times the model was able to insert a plagiarism case in $\mathcal{K}_p$ in the same cluster. To get a "visual intuition" of the performance for each model, just look at the frequency of high values (long bars) along the entire X axis. As we can see, the `spectral` model has the higher frequency of high values along the entire X axis.

In conclusion, the clustering-based method for music plagiarism detection that we use is the one obtained by training a `spectral` model on the CV_{50} representations of the melodies involved in the plagiarism cases contained in $\mathcal{K}$. Finally, to test whether a pair of melodies (m_1, m_2) is a plagiarism case, we just need to check whether m_1 and m_2 belong to the same cluster. We remark that this does not mean that clustering is reduced to a similarity function between two melodies, but, instead that we use clustering to check if the two melodies are in the same cluster (where it is possible also to find other melodies).

5 The adaptive meta-heuristic

In this section, we present the combined meta-heuristic for music plagiarism detection. The meta-heuristic is a combination of the text similarity-based method described in

Sect. 3, and of the clustering-based method described in Sect. 4. By combining the two methods we are able to improve the performance in discovering music plagiarisms.

There are several choices to be made to construct such a combination. First of all the order in which the two methods are applied; then, for the text similarity, we have to choose all the thresholds for each of the similarity techniques used. We have conducted experiments, making various choices. Then, by empirical observation, judging each combination on its behavior on a set of known cases, we ended up using the following criterion: given a pair of melodies, (i) we first use the text similarity-based method to have an "high level" plagiarism/non-plagiarism classification, and then, if necessary, (ii) we use the clustering-based method to refine the previous classification.

Formally, the adaptive meta-heuristic works as follows:

1. Let $\mathcal{K} = \mathcal{K}_p \cup \mathcal{K}_n$ be the knowledge-base, where $\mathcal{K}_p$ is the set of *plagiarism* cases, and $\mathcal{K}_n$ is the set of *non-plagiarism* cases.
2. Let $e = (s_1, \ldots, s_k)$, be the best ensemble on $\mathcal{K}$ and representation PINL, and let t_e be the *threshold* for e, as defined in Sect. 3.
3. Let $\mathcal{C}$ be the set of clustering models, and let $c \in \mathcal{C}$ be the best clustering model trained on $\mathcal{K}_p$ using CV$_{50}$ on the PINL representations of the melodies.
4. Given a pair of melodies (m_1, m_2) we compute the value for each metric $s_i \in e$ on (m_1, m_2). Then we compute t_i as the threshold of s_i on (m_1, m_2). If $t_i \geq t_e[i]$ for each $i = 1, \ldots, k$, then m_1 and m_2 are a plagiarism case; otherwise we run c on (m_1, m_2), and if m_1 and m_2 belong to the same cluster then m_1 and m_2 are a plagiarism case, otherwise m_1 and m_2 are not a plagiarism case.

To summarize, the proposed meta-heuristic is built upon the knowledge-base $\mathcal{K} = \mathcal{K}_p \cup \mathcal{K}_n$ defined in Sect. 3.1 ($\mathcal{K}_p$ consists of 52 plagiarism cases, while $\mathcal{K}_n$ consists of 52 non-plagiarism cases). Uses the text-based representation PINL, optimized over the set of metrics $\mathcal{M} = \{$CS, JRS, DC, OC, JS$\}$, with $e =$(CS, DC, OC) being the best ensemble, and $t_e = [0.70, .0.70, 0.90]$. Uses the vector representation CV$_{50}$, with the `spectral` clustering model, which gave the best results for 9 clusters. With such a setting, we obtained full accuracy on the test set, that is $tp = 52$ (100%) and $tn = 52$ (100%), which is an improvement with respect to the result obtained by applying only the text similarity-based technique ($tp = 51$ and $tn = 50$). This also proves the positive impact of the clustering-based method to refine the previous classification.

5.1 Experiments with famous cases

We tested the proposed meta-heuristic on 25 plagiarism cases and 25 non-plagiarism cases and compared the results with those of existing methods.

We have considered the set of 105 cases in the list of plagiarism disputes provided by Wikipedia,[12] and by carrying out, on each of these cases, a thorough search of the verdicts. In order to assess the generalization capability of the proposed approach. Among such 105 cases, we have selected 25 *positive* cases (plagiarism cases) and 25 *negative* cases (not-plagiarism cases), satisfying the following two conditions: (i) they were not included in the knowledge-base $\mathcal{K}$ used for training the meta-heuristic (see

[12] https://en.wikipedia.org/wiki/List_of_songs_subject_to_plagiarism_disputes.

Sect. 3.1) and (ii) the melody was one of the elements considered for the decisions. The choice about the number (25) was driven by the availability of the material (e.g., music sheets, audio recordings).

Comparison We selected the following methods: the adaptive meta-heuristic (AM) proposed in Sect. 5, the text similarity-based method (TM) proposed in Sect. 3, the clustering-based method (CM) proposed in Sect. 4, the *Fuzzy vectorial* (fuzzy) proposed in De Prisco et al. (2017c), and the *Tversky's feature-based* (tversky), *Ukkonen* (ukk), and *Edit distance* (edit) as described in Müllensiefen et al. (2004) and Müllensiefen and Pendzich (2009). We remark that tversky is the basic version of the Tversky's feature-based method; we have not considered its weighted version described in Müllensiefen and Pendzich (2009).

Since fuzzy, tversky, ukk and edit produce a similarity score between 0 and 1, we need also to establish a cutoff threshold, for each of them, that will be used to produce a final binary value (plagiarism or non-plagiarism). To establish such a threshold we adopted the approach proposed in Müllensiefen and Pendzich (2009), which in turn is similar to those employed to compare binary classifiers (Hand et al. 1997): for each one of the 50 cases, given the two involved melodies m_1 and m_2, for each method $\mathcal{M} \in \{\texttt{fuzzy}, \texttt{tversky}, \texttt{ukk}, \texttt{edit}\}$ we obtain a real value $\mathcal{M}(m_1, m_2) \in [0, 1]$, which must be compared against the known binary court verdict (the *target* of the case, i.e., 1 = plagiarism or 0 = not-plagiarism); to evaluate the relationship between the "continuous classifier" $\mathcal{M}$ and the known binary target of the case, we computed the *optimal cutoff* value of $\mathcal{M}$, named $\mathit{coff}_{\mathcal{M}}$, in the knowledge-base $\mathcal{K}$ used for training the proposed adaptive meta-heuristic; for each method $\mathcal{M}$, we computed $\mathit{coff}_{\mathcal{M}}$ by following the same approach described in Sect. 3.3:

1. Compute the similarity value for each pair of melodies in the knowledge base, using $\mathcal{M}$; let $t_1, t_2, t_3, \ldots$ be all such values.
2. Consider each t_i as a possible threshold for the detection with $\mathcal{M}$.
3. Evaluate the "behavior" of the pair $(\mathcal{M}, t_i)$ with respect to $\mathcal{K}$: take all the *plagiarism* cases (m_1, m_2), represented using PINL that considers both pitch intervals and rhythm, and check whether $\mathcal{M}(m_1, m_2) > t_i$; these are the true positives. Take all *non plagiarism* cases (m_1, m_2) and check whether $\mathcal{M}(m_1, m_2) > t_i$; these are the false positives.
4. Consider the best threshold $\mathit{coff}_{\mathcal{M}}$, that is the value t_i that maximizes the true positives and minimizes the false positives. Then in the Pareto front, select the one that minimizes the number of false positives; if such a solution is not unique, select the one that reaches the highest number of true positives.

We remark that for AM, TM and CM, we have already fixed the best cutoff thresholds. Table 4 shows the similarity threshold (Thr), and the results obtained in terms of *true positives* (*tp*), *false positives* (*fp*), *true negatives* (*tn*), and *false negatives* (*fn*).

Results of the comparison. Once established the thresholds, we proceeded with the comparison. Tables 5 and 6 show the results regarding the 25 plagiarism cases and 25 non-plagiarism cases.

With regard to the 25 plagiarism cases, the adaptive meta-heuristic (AM) showed the best classification capability, i.e., correctly classifying 22 cases of plagiarism (accuracy of 88%). The Tversky's feature-based method also performed well, correctly classify-

Table 4 Optimal cutoff computation for each method used in the experiments

	coff	tp	fp	fn	tn
AM	[0.70, 0.70, 0.90] (CS, DC, OC) + 0.80 (spectral)	52	0	0	52
TM	[0.70, 0.70, 0.90] (CS, DC, OC)	51	2	1	50
CM	0.80 (spectral)	50	8	2	44
fuzzy	0.90	45	12	7	40
tversky	0.34	48	4	4	48
ukk	0.43	34	13	18	39
edit	0.46	31	24	21	28

We have also highlighted the best ensemble (for AM and TM), and best clustering models (for AM and CM)

ing 17 cases of plagiarism (accuracy 68%). The poorest classification is obtained with the edit distance-based method, demonstrating that pure text-based techniques are not always suitable for detecting music plagiarisms.

With regard to the 25 non-plagiarism cases, the adaptive meta-heuristic (AM) showed the best result with 2 false positives (accuracy 92%), while 5 false positives were obtained using the Tversky's feature-based method (accuracy 84%). The poorest classification, once again, has been obtained with the edit distance-based method, further demonstrating that pure text-based techniques are not always suitable for facing music plagiarism detection problems.

In conclusion, the results show that the meta-heuristic presented in this paper is a promising approach.

5.2 The overfitting issue

An adaptive method, such that the one proposed in this work, might suffer of *overfitting*. Ours is not an exception. Overfitting is a critical issue faced in Data Science, that can cause an algorithm to perform poorly when used with unseen data because it was trained too much. To mitigate this problem, and improve the generalization capabilities of the method, we exploited the *k-fold cross-validation* technique for model tuning (i.e., finding the optimal ensemble and clustering technique that yield a satisfying generalization performance).

Specifically, given the knowledge-base $\mathcal{K}$, we carried out the following steps:

1. We randomly split $\mathcal{K}$ into k folds without replacement, where $k - 1$ folds are used for the training, and one fold is used for performance evaluation (test fold); this procedure was repeated k times so that each of the k subsets was used as test fold exactly once.
2. At each step, first, we applied the AM adaptive meta-heuristic, using the $k - 1$ folds chosen as training set, and then, we calculated the performance, in terms of number of true positives and number of false positives, on the current independent test fold.
3. After using all the k folds as test fold, we chose the best adaptive configuration (the adaptive meta-heuristic that showed the best performance, in terms of maximization

Table 5 Comparison of the classification of 25 plagiarism cases

Plagiarism cases	AM	TM	CM	fuzzy	tversky	ukk	edit
Kookaburra (M. Sinclair, 1932) *Down Under* (Men at work, 1980)	✓	✗	✗	✓	✓	✗	✗
Feuilles Mortes (J. Kozma, 1949) *La Maritza* (S. Vartan, 1968)	✓	✗	✓	✓	✓	✓	✓
Crescent City Blues (G. Jenkins, 1953) *Folsom Prison Blues* (J. Cash, 1955)	✓	✗	✗	✓	✗	✗	✗
You Can't Catch Me (C. Berry, 1954) *Come Together* (J. Lennon, 1969)	✓	✗	✗	✗	✓	✗	✗
Pour Toi (L. Gaste, 1956) *Feelings* (M. Albert, 1975)	✓	✓	✗	✓	✓	✓	✗
Sweet Little Sixteen (C. Berry, 1958) *Surfin' U.S.A* (Beach Boys, 1963)	✓	✓	✓	✓	✓	✓	✓
Speedy Gonzales (P. Boone, 1962) *Crocodile Rock* (E. John, 1972)	✓	✓	✓	✓	✓	✓	✗
Bring It On Home (W. Dixon, 1963) *Bring It On Home* (Led Zeppelin, 1969)	✓	✓	✓	✓	✗	✗	✗
He's so fine (The Chiffons, 1963) *My sweet lord*(G. Harrison, 1970)	✓	✗	✗	✗	✗	✗	✗
Love is a wonderful thing (Isley Brothers, 1964) *Love is a wonderful thing* (M. Bolton, 1991)	✓	✓	✓	✓	✓	✓	✗
Ask Any Girl (The Supremes, 1964) *1-2-3* (L. Barry, 1995)	✓	✓	✓	✗	✓	✓	✗
Southwick (M. Parker, 1970) *Auteuil Neuilly Passy* (Les Inconnus, 1991)	✓	✓	✗	✓	✓	✗	✗
Run Through The Jungle (J. Fogerty, 1970) *The Old Man Down The Road* (J. Fogerty, 1985)	✗	✗	✓	✗	✗	✗	✗
Southwick (M. Parker, 1970) *Auteuil Neuilly Passy* (Les Inconnus, 1991)	✓	✓	✗	✓	✓	✗	✗
Theme N.B.C's "Today Show" (R. Ellis, 1971) *Day By Day* (S. Schwartz, 1971)	✗	✗	✗	✓	✓	✗	✗
Alone again (naturally) (R. O'Sullivan, 1971) *Alone again* (B. Markie, 1991)	✓	✗	✗	✓	✓	✗	✗
Life is a Rock (but the radio rolled me) (P. DiFranco & N. Dolph, 1974) *Menu Song* (McDonald's, 1989)	✓	✓	✓	✓	✓	✓	✗
Nelle mie notti (S. Endrigo, 1974) *Il postino* (L. Bacalov, 1996)	✓	✓	✗	✓	✓	✗	✗
Da Ya Think I'm Sexy? (R. Stewart, 1978) *Taj Mahal* (J. Ben Jor, 1972)	✓	✓	✓	✓	✓	✗	✗
Pulsion (J. Loussier, 1979) *Kill You* (Eminem, 2000)	✓	✗	✓	✗	✓	✗	✗
Under Pressure (D. Bowie & Queen, 1981) *Ice Ice Baby* (Vanilla Ice, 1990)	✗	✗	✗	✗	✗	✗	✗
Sugar Don't Bite (S. Harris, 1985) *Papa Don't Preach* (Madonna, 1986)	✓	✗	✗	✓	✗	✗	✗
I cigni di balaka (A. Carrisi, 1987) *Will you be there* (M. Jackson, 1991)	✓	✓	✗	✓	✓	✗	✗
Feel So High (Des'Ree, 1991) *Got 'til It's Gone* (J. Jackson, 1997)	✓	✗	✗	✗	✓	✗	✗
If We Could Start All Over (E. & D. van Passel, 1993) *You are not alone* (M. Jackson, 1995)	✓	✓	✓	✗	✗	✗	✗
One of those Love songs (S. Swirsky & W. Campell, 1998) *Thank God I found you* (M. Carey, 1999)	✓	✓	✗	✗	✓	✗	✗

The first song always represents the *Complaining* song, while the second always represents the *Defending* one. ✓ indicates plagiarism, × indicates no plagiarism

Table 6 Comparison of the classification on 25 non-plagiarism cases

Not-plagiarism cases	AM	TM	CM	fuzzy	tversky	ukk	edit
Wonderful you (R. Overman, 1946) *On a Slow Boat to China* (F. Loesser, 1948)	✓	✓	✓	✓	✓	✓	✗
Boogie Chillen (J. Hooker & B. Besman, 1948) *La Grange* (ZZ Top, 1973)	✗	✗	✗	✗	✗	✓	✓
Jeannie Michele (W. Ferguson, 1953) Theme from *“A Time to Love”* (J. Williams, 1973)	✗	✗	✓	✓	✗	✗	✗
Don’t Cha Know (J. Benson, 1960) *I’d Like to Buy the World a Coke* (Coca-Cola Company, 1971)	✗	✗	✓	✓	✗	✓	✓
Es (E. Chia, 1961) *Hey* (M. Balducci/J. Iglesias, 1980)	✗	✗	✓	✗	✗	✗	✓
Tiny bubbles (L. Pober, 1966) *Hiding the wine* (E. Gold, 2008)	✗	✗	✗	✗	✗	✗	✗
Taurus (M. Skidmore, 1967) *Stairway to Heaven* (Led Zeppelin, 1971)	✗	✗	✓	✓	✗	✗	✓
Hand clapping song (L. Nocentelli et al., 1970) *My love is your love* (W. Houston & C. Raye, 1998)	✗	✓	✓	✓	✓	✓	✓
Till you (R. Repp, 1972) *Phantom song* (A. Lloyd-Webber, 1986)	✗	✗	✗	✗	✗	✓	✓
Let it end (R. Selle, 1975) *How deep is your love* (Bee Gees, 1977)	✗	✗	✓	✗	✗	✗	✓
City of Violets (S. Logarides, 1975) *Chariots of Fire* (E. Papathanasiou, 1981)	✗	✗	✗	✗	✗	✓	✓
Paginas De Mujer (E. Palmieri, 1981) *Oye Mi Canto* (G. Estefan, 1989)	✗	✗	✓	✓	✗	✓	✓
Ooh I Love It (Love Break) (V. Montana & R. Pettibone, 1983) *Vogue* (Madonna & R. Pettibone, 1990)	✗	✓	✗	✓	✗	✓	✓
I Won’t Back Down (T. Petty & J. Lynne, 1989) *Stay With Me* (S. Smith, 2015)	✗	✓	✗	✓	✓	✓	✓
Lay me out by the jukebox when I die (E. Ellis, 1991) *Prop me up beside the jukebox (If I die)* (J. Diffie, 1993)	✗	✗	✗	✗	✗	✓	✓
I think about you (S. McKinley, 1991) *I think about you* (S. Seskin & C. Raye, 1991)	✗	✓	✓	✓	✗	✓	✓
Jenny and I (S. Malmstedt, 1991) *Sleeping in my car* (P. Gessle, 1994)	✓	✓	✓	✓	✓	✓	✓
I’m Frontin (D. Currin/P. Currin, 1993) *Frontin* (P. Williams, 2021)	✗	✗	✗	✗	✗	✗	✗
Sell Your Soul (M. Tisi, 1994) *Take A Picture* (R. Patrick, 1999)	✗	✓	✓	✓	✓	✓	✓
Es (E. Coffey, 1995) *Nothing really matters* (P. Leonard/Madonna, 1998)	✗	✗	✗	✗	✗	✗	✗
Es (E. Chia, 1998) *Hey* (J. Iglesias, 1980)	✗	✓	✗	✓	✗	✓	✓
Sunrise (Yeasayer, 2007) *Pray for Me* (K. Lamar, 2018)	✗	✓	✗	✗	✗	✓	✓
Sunrise (Yeasayer, LLC, 2007) *Pray for me* (A. M. Tesfaye, 2018)	✗	✓	✗	✗	✗	✓	✗
Bright Red Chords (W. Loomis, 2009) *Domino* (J. Cornish, 2011)	✗	✗	✗	✗	✗	✓	✓
Takes All Night (A. Greggs, 2011) *One Last Time* (A. Grande, 2014)	✗	✓	✓	✓	✗	✓	✓

The first song always represents the *Complaining* song while the second always represents the *Defending* one. ✓ indicates plagiarism, × indicates no plagiarism of true positives and minimization of false positives), and validated its performance on the entire set $\mathcal{K}$.

We remark that we tested the k-fold cross validation approach for several values of k, and the best results were observed with $k = 10$. Furthermore, we have implemented an automatic routine in the tool described in Sect. 6, which periodically, given the actual knowledge-base (including new cases added by the administrators during the usage of the tool, if any), applies such a technique to update the meta-heuristic.

6 The web application tool

We have implemented the proposed meta-heuristic into a tool accessible as a Web application.[13] This online service provides visual aids in the form of music scores, probability percentages, and other visually conveyed information, helping the user in identifying a plagiarism case.

The tool allows the management of a database of cases, with detailed information for each case. All information is exposed through the use of suitable and intuitive icons. The overall objective is to offer an ease-to-use interface, guaranteeing a friendly user experience. The web application is graphically illustrated in Fig. 6.

In this section, we first describe the architecture with the functionalities provided by its components and then present an evaluation study about the usability of the tool, in terms of overall user satisfaction.

The tool has been developed exploiting *Flask*, a Python web framework built with a small core and easy-to-extend philosophy.[14] It detects plagiarism through the meta-heuristic described in Sect. 5, implemented by using the python *Scikit-learn* library.[15]

The tool allows to upload a pair of music scores in `musicxml` format (see Fig. 7), through an easy-to-use user interface implemented by using technologies such as *JavaScript* and *Bootstrap*. The scores are received and elaborated by using the JavaScript *AlphaTab* library,[16] that dynamically builds the corresponding music sheets, which are then visualized in a new web page, and can be played back for listening; this enables users to see and hear the music. Then, the user can perform the plagiarism check: the tool sends the uploaded scores to the server-side, which, in turn, uses the proposed meta-heuristic to detect if a plagiarism occurs; the result is finally returned to the client-side and shown to the user (see Fig. 8). Additionally, registered users can add cases to the database (Fig. 9), attaching scores and verdicts (Fig. 10).

6.1 Evaluation study

In this section we describe a preliminary evaluation study, involving 20 people, performed to assess the effectiveness of the tool, in terms of plagiarism detection

[13] http://193.205.161.3:5000.

[14] https://www.fullstackpython.com/flask.html.

[15] https://scikit-learn.org/stable/.

[16] https://www.alphatab.net/.

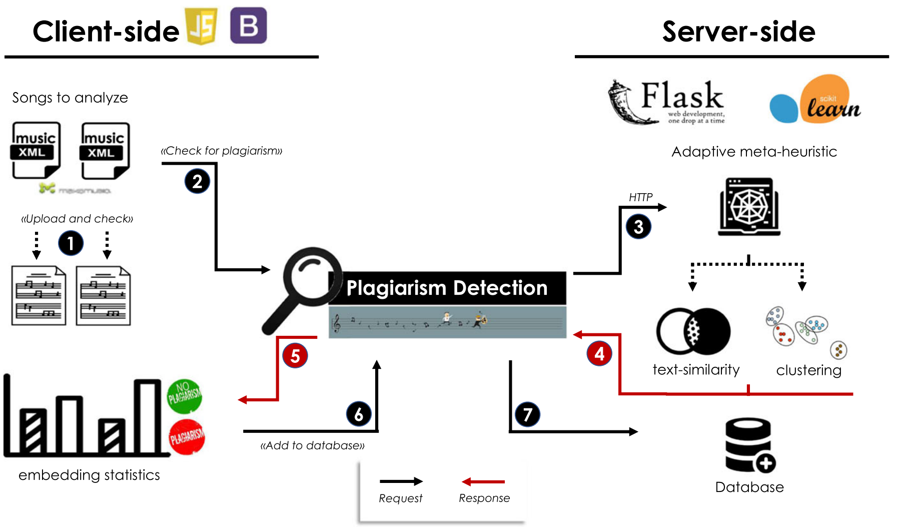
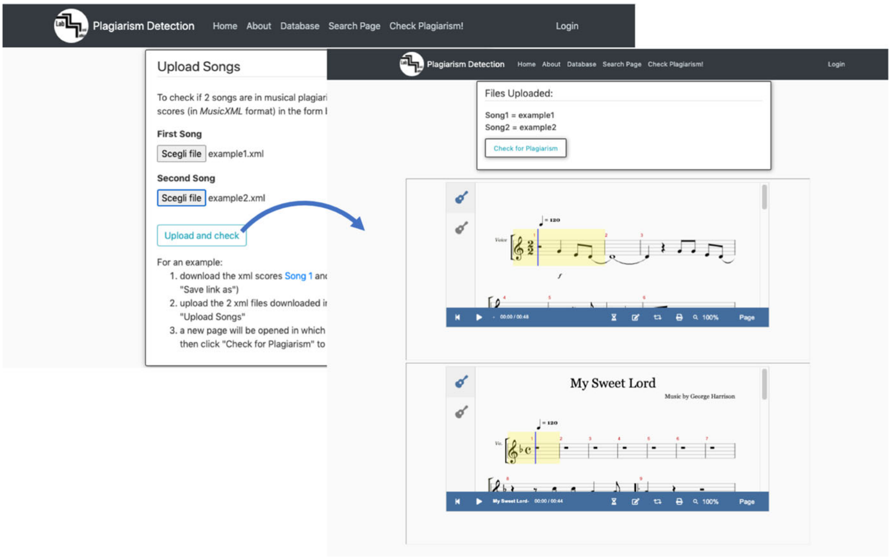


Fig. 6 *Plagiarism Detection* overall architecture and its components: the client-side component accepts users requests in the form of `musicxml` scores, visualizes the corresponding music sheets, and enables the user to start the plagiarism detection utility; the server-side component handles applies the `AM` meta-heuristic. Additionally, registered users can add the analyzed case to the database managed by the platform



Upload Songs

To check if 2 songs are in musical plagiari
scores (in *MusicXML* format) in the form l

First Song

Scegli file example1.xml

Second Song

Scegli file example2.xml

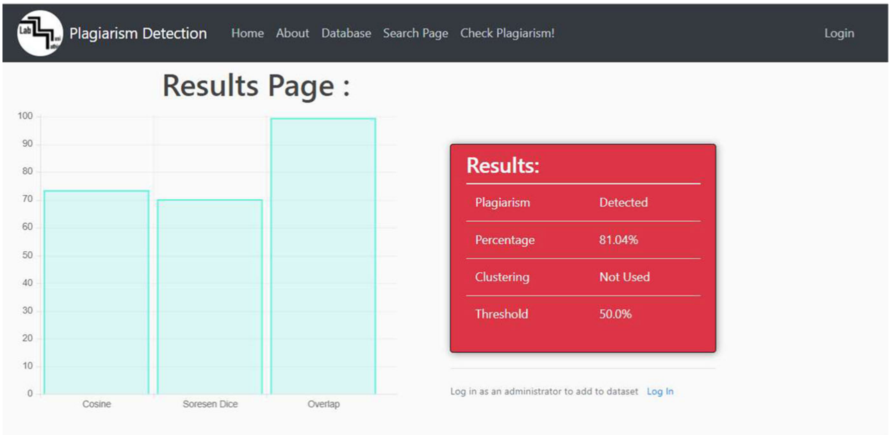
For an example:

1. download the xml scores Song 1 anc
   "Save link as")
2. upload the 2 xml files downloaded ir
   "Upload Songs"
3. a new page will be opened in which
   then click "Check for Plagiarism" to

Files Uploaded:

Song1 = example1
Song2 = example2

Fig. 7 When uploading the songs, the system provides a graphical representation of the scores with playback capabilities



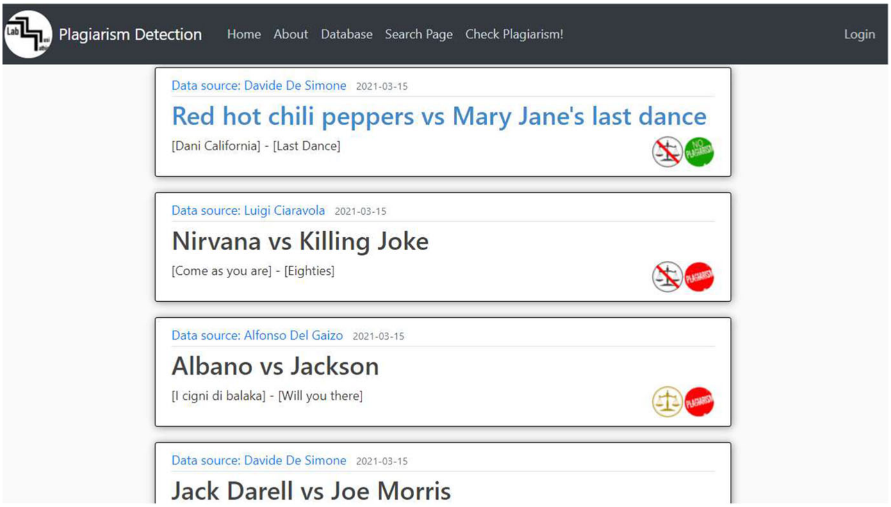
Results Page :

Results:

Plagiarism	Detected
Percentage	81.04%
Clustering	Not Used
Threshold	50.0%

Log in as an administrator to add to dataset Log In

Fig. 8 "Plagiarism Detection - Results" web page of the *Plagiarism Detection* platform. The web page provides all the information regarding the detection task



Data source: Davide De Simone 2021-03-15

Red hot chili peppers vs Mary Jane's last dance

[Dani California] - [Last Dance]

Data source: Luigi Ciaravola 2021-03-15

Nirvana vs Killing Joke

[Come as you are] - [Eighties]

Data source: Alfonso Del Gaizo 2021-03-15

Albano vs Jackson

[I cigni di balaka] - [Will you there]

Data source: Davide De Simone 2021-03-15

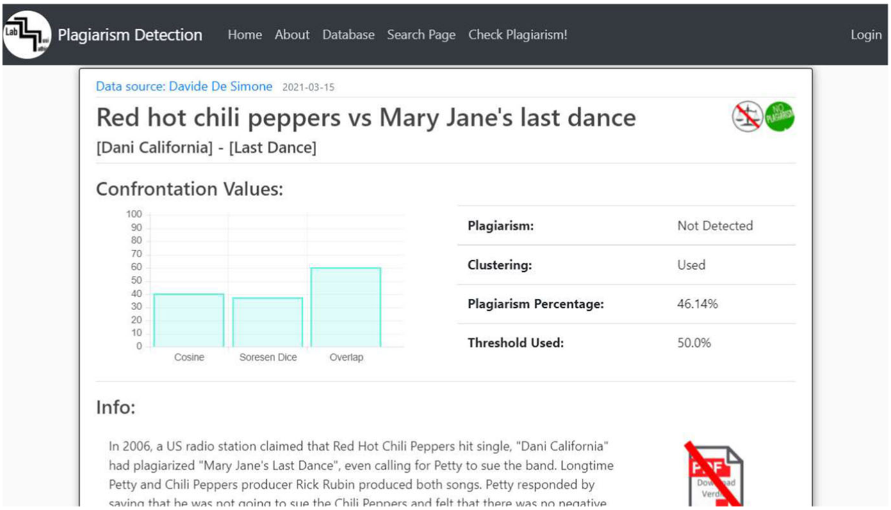
Jack Darell vs Joe Morris

Fig. 9 "All sentences" web page of the *Plagiarism Detection* platform. The web page enable to see all the cases included in the database of the platform

capabilities, as well as the users' perception about its overall acceptance (TAM, Technology Acceptance Model).

The activities required included comparative listening and the study of scores. We are not assuming that sound and notation are equivalent and, therefore, we ask the participants to use both sound material and notations to complete their tasks.

We first describe the employed methodology and then the results obtained. We followed the standard HCI methodology (Lazar et al. 2017), commonly applied in different contexts (Leon et al. 2012; Al-Musawi et al. 2016; De Prisco et al. 2021), as well as for the music plagiarism problem (De Prisco et al. 2016, 2017b).



Data source: Davide De Simone 2021-03-15

Red hot chili peppers vs Mary Jane's last dance

[Dani California] - [Last Dance]

Confrontation Values:

Plagiarism:	Not Detected
Clustering:	Used
Plagiarism Percentage:	46.14%
Threshold Used:	50.0%

Info:

In 2006, a US radio station claimed that Red Hot Chili Peppers hit single, "Dani California" had plagiarized "Mary Jane's Last Dance", even calling for Petty to sue the band. Longtime Petty and Chili Peppers producer Rick Rubin produced both songs. Petty responded by saying that he was not going to sue the Chili Peppers and felt that there was no negative

Fig. 10 “Case detail” web page of the *Plagiarism Detection* platform. The web page enables to insert a new case in the database, or to update and view a previous case

6.1.1 Methodology

The objective of this study was to explore the efficacy of our tool in supporting users when they have to express opinions about alleged plagiarism between given songs. According to the Between-group Design (Lazar et al. 2017), we divided the sample into two groups: 10 participants (standard group) performed the submitted tasks without any support, while the other 10 (enhanced group) could complete the requested task with the support of the tool. It is worth noting that participants of the second group were not forced to accept the decision suggested by the tool, since the tool is intended only to provide an aid for the decision. Additionally, to ensure equal allocation of subgroups of participants to each experimental condition, we stratified the sample both on gender and music skills. We compared these two groups, trying to understand differences between them in terms of efficacy in discovering plagiarism cases.

We also studied the users’ intentions to use the provided tool with selected constructs from the Technology Acceptance Model (Davis 1989), TAM for short. In particular the model considers users’ attitudes, perceived usefulness, perceived ease of use, and finally, perceived playfulness. TAM is a widely used theoretical model to explain and/or predict potential users’ behavioral intentions to accept a new technology/system. In TAM, perceived usefulness refers to “*the degree to which a person believes that using a particular system would enhance his or her job performance*”, while perceived ease of use refers to “*the degree to which a person believes that using a particular system would be free of effort*”. According to what was discussed by Davis (1989), a system that is perceived to be easy to use, is also likely to be widely accepted.

Procedure The study was conducted in the Musimathics research laboratory at the University of Salerno. The personal computer used was equipped with an Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz, and 32.00 GB of main memory. Six pairs of songs to listen to were randomly chosen from the knowledge base. All participants in the study listened to the same songs, available online.[17]

The study has been structured in three different phases in which we carried out: (i) a Preliminary Survey (ii) a Testing Phase and (iii) a Summary Survey, as defined and implemented also in other contexts (e.g., Erra et al. 2018).

In the first phase, we asked participants to fill in a preliminary survey questionnaire in order to collect: *(a)* demographic information (i.e., gender, age, education level), *(b)* information about ICT expertise, *(c)* general attitudes toward music and the problem under investigation. The questions included in this questionnaire were open-ended questions, questions with a "yes" or "no" dichotomous format, questions asking to give a preference up to 8 possible choices, and finally, questions with a rating on a 5-point Likert scale with *strongly agree/strongly disagree* as verbal anchors.

In the Testing Phase, we asked participants to perform 6 tasks; for each task the participant was required to listen to a pair of songs and to express his/her personal opinion about the existence of a plagiarism case. Participants of the enhanced group had the possibility to use our tool in aiding them in their decision. We provided all participants with both audio files (to listen to) and the music scores in PDF format file (for further analysis, if needed). The participants of the enhanced group were also provided with the XML representations of songs (XML-based files) to allow them to use the online service. Therefore, they were free to explore the tool and exploit its functionalities in order to get support for a thoughtful decision.

For both groups, we collected the following information: (a) how many times the songs were listened to; (b) whether participants looked at the music scores; (c) the final user's decision; (d) the tool's decision (for the enhanced group); (e) the time used to complete the task. Participants of the enhanced group were also asked to spend few additional minutes in order to answer the TAM questionnaire.

Final perceptions were collected through a summary survey questionnaire. This questionnaire was composed of questions with preferences among at most 5 choices and questions with a rating on a 5-point Likert scale, with *strongly agree/strongly disagree* as verbal anchors. We also repeated two questions posed in the preliminary survey with the goal of analyzing potential differences at the end of the experiment. Audio files, music scores, surveys and the standard TAM questionnaire are available online. [17]

Users were monitored during the experimentation, and they also could call for assistance if they had doubt about the instructions. The testing was performed in an isolated environment within our research lab in order to avoid distractions. Users were also encouraged to provide informal feedback (such as general comments, suggestions, and so on).

Recruitment Subjects were secondary school students, students of the Music Conservatory of Salerno, students of the University of Salerno, and professionals from various fields (Engineering, Law, Medicine, Education, Music). They were recruited through word-of-mouth advertising. Furthermore, to avoid a biased sample, when recruiting participants, we did not specify the topic of the study, and we only said that we were looking for people, with a music background even if at various levels, interested in

[17] https://bit.ly/3lsZJqJ.

participating in an evaluation study. We were able to recruit: 16 musicians, ranging from students to experts, having, therefore, different skills, 2 producers coming from the music industry, and 2 laws professionals, with a music background. Their participation was voluntary and anonymous, and participants were not compensated for taking part in the interviews. Finally, we obtained the ethical approval to conduct this study, and we asked participants to sign an informed consent in which we explained that data were anonymous and that they would remain confidential. We never shared the acquired data with third parties, but used them only for research purposes.

Data analysis Non-parametric tests were applied to study differences between the groups. The Shapiro-Wilk goodness-of-fit test was used to assess the normality of data (Shapiro and Wilk 1965).

The internal consistency reliability among multi-item scales was examined with the Cronbach alpha (Cronbach 1951). Finally, questionnaire responses were analyzed using SPSS version 20.[18]

6.1.2 Results

In this section, we report the results of each of the three different phases of the evaluation study.

Preliminary Survey Results Results of the Preliminary Survey allowed us to outline the profile of our participants. As shown in Table 7, more than half of the participants were university (bachelor/master) students (65%), the majority were male (i.e., 70%) and the average age was 33. With regard to music skills, half of the respondents spends almost 3 hours per day listening to music, 65% considers themselves "*Competent/Expert*" in the music field, and additionally, a 35% plays a musical instrument. Moreover, only 10% of participants rated as easy the task of discovering similarities between two songs, and 45% rated themselves as competent in respect to this task (50% beginner, only 5% experts). With regard to ICT skill, 65% of participants rated themselves as competent/advanced, and agreed about the usefulness of having a software tool that automatically discover similarities between songs. Six out of nine participants who answered positively to the question: "*Have you ever heard of famous cases of musical plagiarism?*", have indicated the Albano Carrisi–Michael Jackson case, as their known case; this case was not included in the Testing Phase. Finally, none of the participants were hearing impaired.

Testing Phase Results We recall that in this phase we asked participants to perform 6 tasks. In each task the participant had to listen to a pairs of songs and state whether or not the pair was a plagiarism case. Only participants in the enhanced group were given access to our tool.

As we can see in Table 8, the enhanced group successfully identified all plagiarism cases, performing all tasks with no errors and no hesitations. The standard group exhibited uncertainty for some pairs of songs; specifically, 10% was not sure with regard to Task 1, 20% for Task 4, and finally, 20% for Task 5. The majority of mistakes were made when performing Task 2 and Task 5. We also analyzed whether differences existed between the two groups and whether these were statistically significant.

[18] http://www-01.ibm.com/software/analytics/spss/.

Table 7 Participant's demographics

	Number	Percentage (%)
Total participants	20	100
Gender		
Male	14	70
Female	6	30
Age		
< 25 years old	8	40
26–39 years old	3	15
40+ years old	9	45
Education level attained		
Secondary school	7	35
Bachelors/masters	11	55
Other	2	10
Time spent listening to music per week		
1–7 hours	8	40
8–20 hours	7	35
20+ hours	5	25

With regard to this first metric, we found out a statistically significant difference in performing Tasks 2 and 4 ($p < .05$), and the reasons, as explained before, were due to both uncertainty and mistakes. When we asked, as informal feedback, observations to participants, they give two important feedbacks: one of the two songs in Task 1 had a long introduction, while high timbre similarities (mainly with regard to the Electric piano), were present between the songs in Task 4. These situations may have brought participants to have doubts and to make mistakes during the plagiarism identification process.

When analyzing the second metric (whose question was: "*How many times did you listened to the songs*?"), we found a statistical difference only for the Tasks 1, 2, and 3 ($p < .05$, for all of them). Here we can observe that participants in the enhanced group listened to the songs more times (see Songs plays columns), with respect to the other group. When we gathered informal feedback, we discovered that for the songs in these tasks, they changed mind about their final decision, but only after using the tool. In summary, the support of the tool involved a new listening of the songs, helping participants to provide correct decisions. For Tasks 4, 5, and 6 we did not find statistical differences and a possible explanation for which the participants in the standard group increased the number of times they played the songs can be found in the uncertainty clearly showed in these tasks (see Table 8).

When analyzing the third metric (whose question was: "*How many times did you accessed the music scores*?", with results shown in Score accesses column), we found out a difference between the two groups only with regard to Task 2 ($p < .01$) and Task 5 ($p < .05$). Here, as before, the enhanced group made additional accesses after using the tool and achieved 100% of correct answers.

Table 8 Comparison of the tested groups in terms of: (a) errors made (*Errors*), (b) indecision deriving by *I do not know* responses (*Indecision*), (c) number of times a song has been played (*Songs plays*), (d) a number of time a music score has been accessed (*Scores accesses*), (e) the required time to complete the task (*Time*)

	Plagiarism	Group	Errors	Indecision	Songs plays	Scores accesses	Time
Task1 1	✗	Standard	10	10	60/30/10	40/40/20	50/30/20/0
		Enhanced	0	0	10/40/50	60/40/0	10/10/40/40
Task2 2	✓	Standard	40^{*}	0	50/40/10	20/80/0	40/50/10/0
		Enhanced	0	0	0/100/0	60/10/30	0/30/60/10
Task 3	✓	Standard	20	0	40/60/0	70/30/0	40/60/0/0
		Enhanced	0	0	0/50/50	60/10/30	10/30/20/40
Task 4	✗	Standard	10	20	10/90/0	50/40/10	40/30/30/0
		Enhanced	0	0	10/80/10	70/20/10	10/30/60/0
Task 5	✗	Standard	40^{*}	20	40/50/10	30/80/0	40/40/20/0
		Enhanced	0	0	50/50/20	70/20/10	20/30/20/30
Task 6	✓	Standard	20	0	50/60/0	40/50/10	60/30/10/0
		Enhanced	0	0	20/50/30	60/20/20	10/20/50/20

Tasks 2, 3, and 6 were plagiarism cases
1 Values (in percentages) are reported using the following format:
Songs plays are triples ⟨ *only once/2–3 times/more than 3* ⟩
Scores accesses are triples ⟨ *never/only once/more than one* ⟩
Time are quadruples ⟨ *less of 5 mins/between 5 and 10 mins/between 10 and 15 mins/more than of 15 mins* ⟩

Finally, with regard to the last metric (i.e., Time to complete the task, column Time in Table 8), we found out a statistical difference only for the Tasks 1, 2, and 3 (p values $<$ .05 for all tasks). We recall that for these tasks, the participants of the enhanced group listened to the songs more times, and therefore, more time was needed to complete the tasks.

At the end of the testing phase, we asked participants to respond to the TAM questionnaire. Reliability values (Cronbach's alpha) for all subscales are above 0.80, thus above the recommended threshold value of 0.70 (Jum and Nunnally 1994). As shown in Fig. 11, results were highly positive for all subscales, with the highest value obtained with the behavioral intention to use subscale (BI). Specifically, for questions $BI1$ (*"I will use the system on a regular basis in the future"*) and $BI2$ (*"I will strongly recommend others to use the system"*), we have $M = 6.9$, $SD = 0.3$. Relatively to the lowest average value of 5.7, exhibited by the EOU subscale, this is due (as also witnessed by informal feedback) to the low friendliness and flexibility of the tested online service. Specifically the involved questions were $EOU3$ (*"My interaction with the system would be clear and understandable"*), for which $M = 5.5$, $SD = 0.7$, and $EOU4$ (*"I would find the system to be flexible to interact with"*), for which $M = 5.3$, $SD = 0.5$. We recall that our tool is in a prototype form yet, and we are currently working in developing it as a scalable, reliable, and user-friendly framework to assist any type of user (by allowing, for example, to upload for the analysis, more common formats for audio files, in addition to the current XML file format, and removing all jargon that is understandable only by technicians and domain experts).

Summary Survey Phase Results In this section, we report the results of the questions posed in the questionnaire submitted in the third phase. First, for the question "*Which category of users do you think the proposed system could be useful for*?", 65 % of

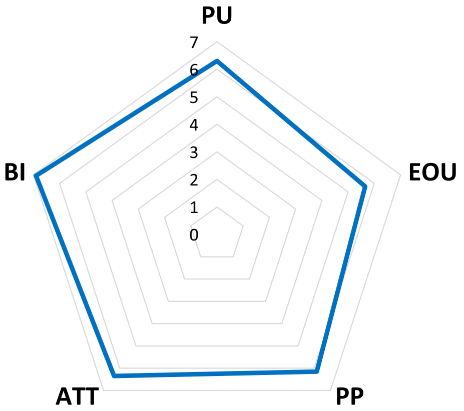


Fig. 11 Results across all subscales of the TAM questionnaire: Perceived Usefulness (PU), Perceived Easy of Use (EOU), Perceived Playfulness (PP), Attitude Toward Using (ATT), Behavioral Intention to Use (BI). Rating on a 7-point Likert scale

the participants stated that the proposed tool could be useful to musicians, music producers, judges, and lawyers, while the remaining stated that it could be useful for all people.

Then, we asked only participants of the enhanced group to rate the overall usefulness and easiness of the tool (rating on a 5-point Likert scale). Generally, all participants rated positive the usefulness of the tool ($M = 5$, $SD = 0$) and interesting the idea behind it ($M = 4.7$, $SD = 0.5$). Confirming results of the TAM (EOU subscale), the lowest value was about the easiness of the tool ($M = 3.9$, $SD = 0.7$).

Finally, in this phase, we also asked participants to express their opinion about how easy it was the task of discovering similarities between two songs and how much they believe a software tool that automatically identifies plagiarism cases can be useful. These same questions were previously asked in the preliminary survey phase. The comparison of the answers, allowed to check whether perceptions and opinions changed after the evaluation process.

With regard to the first question, interestingly, we found out that, in the first place, participants rated difficult the task (M = 2.8, SD = 0.7) but even more after approaching it during the evaluation study (M = 2.3, SD = 0.8). The difference was also statistically significant ($p < .05$). The result is even more interesting if we consider that participants of the enhanced group changed their opinion by lowering their judgement (M = 2.6 before the study, M = 1.8 at the end of the study).

With regard to the second question, we found out that participants, after the evaluation study, agreed more about the usefulness of an automatic tool supporting users in taking decision about the controversial and debated topic of music plagiarism (M = 3.9, SD = 0.8, before and M = 4.6, SD = 0.5 after). We also found out this difference statistically significant ($p < .005$).

7 Conclusion

In this work, we addressed the controversial and debated topic of music plagiarism, characterized, from one side, by a huge amount of money generated in the music industry and, on the other side, by the fact that the actual courts' decisions are primarily based on subjective argumentations about musical material. In this situation, automated music analysis methods that identify music similarities can be of help.

With this aim, we presented an adaptive meta-heuristic for detecting similarities of two melodies. The meta-heuristic combines two different methods; the first one is based on a textual representation of the melodies while the second one uses a vectorial representation of the melodies. We recall that the similarity between melodies is assumed to be a very important factor in a court's decision about whether a new song is an illegitimate version of a pre-existing melody or not.

Beside designing the meta-heuristic, we also deployed it into a Web application, making available the plagiarism detection tool through an online service. We carried out a small-scale evaluation study to evaluate the effectiveness, usefulness, and the overall user acceptance of the tool. The study consisted in having two groups of people, with only one of the group supported by our tool, decide which pair of songs, in a predefined set of pairs, should be considered plagiarism cases and which not. The study showed that the group supported by our tool, conversely to the group with no access to the online service, successfully identified all plagiarism cases, performing all tasks with neither errors nor indecision. Generally, participants rated positively/very positively all questions about the TAM subscales regarding the overall acceptance of the tool. We also found out a high willingness to use the tool in the future; furthermore, the whole sample agreed about the usefulness of a tool that automatically identifies the existence of plagiarism between two pieces of music songs.

As future work it would be interesting to proceed along two different directions. Given the results obtained in the evaluation study with regard to both the idea and to the prototype proposed in this paper, it would be useful to develop a more robust, more scalable and easy to use framework able to encompasses all steps of the whole process, starting from the codification of the music scores, helping with the analysis of the music, and arriving to the final decision about plagiarism. A particular attention could be devoted to provide a friendly-user interface with no jargon terms, in order to allow a wide usage by all interested people and not only by domain experts.

The other direction for future work regards the evaluation study. It would be interesting to carry out a more extensive and representative study with a larger and diversified sample in order to obtain a statistically more significant analysis of the differences between the groups; additionally, a further analysis (i.e., regression analysis) could be performed to identify which variables, among the ones studied with the standard TAM questionnaire, influence the adoption of our tool.

Funding Open access funding provided by Università degli Studi di Salerno within the CRUI-CARE Agreement.

Availability of data and materials The web platform with examples to test is online at: http://193.205.161.3:5000; all questionnaires used in the preliminary evaluation study are available online here: https://bit.ly/3lsZJqJ.

Declarations

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this paper.

Code availability Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Al-Musawi M, Ledesma A, Nieminen H, Korhonen I (2016) Implementation and user testing of a system for visualizing continuous health data and events. In: 2016 IEEE-EMBS international conference on biomedical and health informatics (BHI), pp 156–159

Cason RJ, Müllensiefen D (2012) Singing from the same sheet: computational melodic similarity measurement and copyright law. Int Rev Law Comput Technol 26(1):25–36

Crawford T, Iliopoulos CS, Raman R (1998) String-matching techniques for musical similarity and melodic recognition. Comput Musicol 11:71–100

Cronbach LJ (1951) Coefficient alpha and the internal structure of tests. Psychometrika 16(3):297–334

Davis FD (1989) Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Q 13(3):319–340

De Prisco R, Lettieri N, Malandrino D, Pirozzi D, Zaccagnino G, Zaccagnino R (2016) Visualization of music plagiarism: analysis and evaluation. In: 20th international conference information visualisation, IV 2016, Lisbon, Portugal, July 19–22, 2016, pp 177–182

De Prisco R, Esposito A, Lettieri N, Malandrino D, Pirozzi D, Zaccagnino G, Zaccagnino R (2017a) Music plagiarism at a glance: metrics of similarity and visualizations. In: 2017 21st international conference information visualisation (IV). IEEE, pp 410–415

De Prisco R, Malandrino D, Pirozzi D, Zaccagnino G, Zaccagnino R (2017) Understanding the structure of musical compositions: is visualization an effective approach? Inf Vis 16(2):139–152

De Prisco R, Malandrino D, Zaccagnino G, Zaccagnino R (2017c) Fuzzy vectorial-based similarity detection of music plagiarism. In: 2017 IEEE international conference on fuzzy systems (FUZZ-IEEE). IEEE, pp 1–6

De Prisco R, Guarino A, Lettieri N, Malandrino D, Zaccagnino R (2021) Providing music service in ambient intelligence: experiments with gym users. Expert Syst Appl 177:114951

Downie JS (1999) Evaluating a simple approach to music information retrieval: conceiving melodic n-grams as text

Erra U, Malandrino D, Pepe L (2018) A methodological evaluation of natural user interfaces for immersive 3d graph explorations. J Vis Lang Comput 44:13–27

Flexer A (2014) On inter-rater agreement in audio music similarity. In: ISMIR, Citeseer, pp 245–250

Flexer A, Grill T (2016) The problem of limited inter-rater agreement in modelling music similarity. J New Music Res 45(3):239–251

Flexer A, Lallai T (2019) Can we increase inter-and intra-rater agreement in modeling general music similarity?. In: ISMIR, pp 494–500

Gomaa WH, Fahmy AA et al (2013) A survey of text similarity approaches. Int J Comput Appl 68(13):13–18

Hahn U, Chater N, Richardson LB (2003) Similarity as transformation. Cognition 87(1):1–32

Hand DJ et al (1997) Construction and assessment of classification rules. Wiley, Hoboken

Hussain SAA, Moosavinasab S, Sezgin E, Huang Y, Lin SM (2018) Char2vec: learning the semantic embedding of rare and unseen words in the biomedical literature. In: AMIA

Jum C, Nunnally IHB (1994) Psychometric theory. McGraw-Hill, New York

Lazar J, Feng JH, Hochheiser H (2017) Research methods in human-computer interaction. Morgan Kaufmann, Burlington

Leon P, Ur B, Shay R, Wang Y, Balebako R, Cranor L (2012) Why Johnny can't opt out: a usability evaluation of tools to limit online behavioral advertising. In: Proceedings of the SIGCHI conference on human factors in computing systems, pp 589–598

Mongeaul M, Sankoff D (1990) Comparison of musical sequences. Comput Humanit 24(3):161–175

Müllensiefen D, Pendzich M (2009) Court decisions on music plagiarism and the predictive value of similarity algorithms. Musicae Scientiae 13(1-suppl):257–295

Müllensiefen D, Frieler K et al (2004) Cognitive adequacy in the measurement of melodic similarity: algorithmic vs. human judgments. Comput Musicol 13(2003):147–176

Ó'Maidin D (1998) A geometrical algorithm for melodic difference in melodic similarity. Melodic Similarity Concepts Proced Appl Comput Musicol 11:65–72

Park S, Kwon T, Lee J, Kim J, Nam J (2019) A cross-scape plot representation for visualizing symbolic melodic similarity. In: ISMIR, pp 423–430

Robine M, Hanna P, Ferraro P, Allali J (2007) Adaptation of string matching algorithms for identification of near-duplicate music documents. In: Workshop on plagiarism analysis, authorship identification, and near-duplicate detection (PAN07), pp 37–43

Sculley D (2010) Web-scale k-means clustering. In: Proceedings of the 19th international conference on World wide web, pp 1177–1178

Shapiro SS, Wilk MB (1965) An analysis of variance test for normality (complete samples). Biometrika 52(3/4):591–611

Shepard RN (1957) Stimulus and response generalization: a stochastic model relating generalization to distance in psychological space. Psychometrika 22(4):325–345

Tversky A (1977) Features of similarity. Psychol Rev 84(4):327

Typke R, Wiering F, Veltkamp RC (2007) Transportation distances and human perception of melodic similarity. Musicae Scientiae 11(1-suppl):153–181

Uitdenbogerd AL (2002) Music information retrieval technology. Ph.D. thesis, RMIT University Melbourne Victoria, Australia

Wang S, Song J, Guo R (2018) Char-level neural network for network anomaly behavior detection. In: International conference on human centered computing. Springer, pp 60–68

Wolf A, Müllensiefen D (2011) The perception of similarity in court cases of melodic plagiarism and a review of measures of melodic similarity. In: Int. conf. of students of sustematic musicology, 2011, Citeseer

Xu R, Wunsch D (2005) Survey of clustering algorithms. IEEE Trans Neural Netw 16(3):645–678

Yuan Y, Oishi S, Cronin C, Müllensiefen D, Atkinson Q, Fujii S, Savage PE (2020) Perceptual vs. automated judgments of music copyright infringement

Zhang T, Ramakrishnan R, Livny M (1996) Birch: an efficient data clustering method for very large databases. ACM Sigmod Record 25(2):103–114

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.