Advancing Audio Phylogeny: A Neural Network Approach for Transformation Detection

Milica Gerhardt, Luca Cuccovillo, Patrick Aichroth
Fraunhofer Institute for Digital Media Technology IDMT
Ehrenbergstraße 31, 98693 Ilmenau, Germany
{milica.gerhardt, luca.cuccovillo, patrick.aichroth}@idmt.fraunhofer.de

***Abstract*—In this study we propose a novel approach to audio phylogeny, i.e. the detection of relationships and transformations within a set of near-duplicate audio items, by leveraging a deep neural network for efficiency and extensibility. Unlike existing methods, our approach detects transformations between nodes in one step, and the transformation set can be expanded by retraining the neural network without excessive computational costs. We evaluated our method against the state of the art using a self-created and publicly released dataset, observing a superior performance in reconstructing phylogenetic trees and heightened transformation detection accuracy. Moreover, the ability to detect a wide range of transformations and to extend the transformation set make the approach suitable for various applications.**

***Index Terms*—audio phylogeny, audio provenance, audio transformation detection, audio forensics**

I. Introduction

Audio provenance analysis seeks to trace and authenticate the origin and history of audio content. One crucial subdomain is audio phylogeny, which aims at detecting relationships and transformations within a set of near-duplicate audio items created by derivation and modification, thereby reconstructing a phylogenetic tree as depicted in Figure 1:

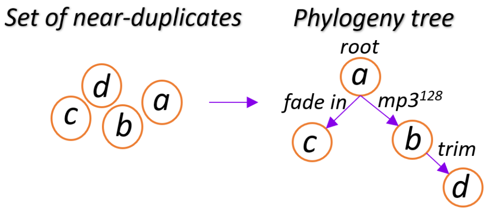

Fig. 1. Audio phylogeny analysis: from a set of near-duplicates to a directed graph with an identified root item

Audio phylogeny can be used within various domains, including e.g. content tracking, metadata propagation and intelligent de-duplication in production archives. However, several key applications lies within the field of media forensics, where it can be used, e.g.,

- to detect the root item within a dataset and trace its derivations, with the purpose of identifying the source and track the propagation of media content within networks;
- to deliver information about specific transformations being performed upon a given item, which can be used to falsify claims about how items were created and modified;
- to decide about which higher-quality, 'upstream' items to focus on during forensics analysis, thereby improving the accuracy of subsequent analysis steps.

The methodological foundations for multimedia phylogeny were provided by Dias *et al.* in [1] and later extended in [2], starting with a focus on images. Since then, these approaches have been adapted and extended to other modalities, leading to the emergence of subdomains of video, text, and, most recently, audio phylogeny. As for the latter, three methods have been proposed so far: Nucci *et al.* [3] proposed an initial "brute force" approach that applied a broad range of transformations from a predefined set $\mathcal{T}$, including respective parameter combinations, to every node pair in the examined near-duplicate set, resulting in a computationally highly demanding system. Maksimović *et al.* [4] and Verde *et al.* [5] then improved the original work by introducing detection functions designed to recognize specific audio transformations, providing comparable detection performance at reduced computational complexity. However, several key requirements for the forensics domain still needed to be addressed:

- *Extensibility*: Being able to extend the predefined set $\mathcal{T}$ to cover additional audio transformations is important, but so far, was difficult and time-consuming to implement with the given approaches.
- *Computational efficiency*: The computational cost of the proposed approaches was still too high for many practical applications, particularly for large datasets or low-latency demands, even more so if the aforementioned requirement related to extending $\mathcal{T}$ comes into play.
- *Transformation detection*: In many situations, it is not only necessary to determine whether a direct parent-child relation exists between file pairs, but also to detect which transformations were applied, e.g. to recover a missing chain of custody.

To address these requirements, we propose a novel approach for audio phylogeny analysis that relies on a Deep Neural Networks (DNN) to detect the most probable transformation that has occurred between every pair of audio items (files) within a near-duplicate set. For this purpose, the following sections will first provide an overview of the current state of

This work was supported by the European Union under the Horizon 2020 AI4Media project, Grant Agreement number 951911, and under the Horizon Europe vera.ai project, Grant Agreement number 101070093.

the art, followed by a detailed description of our proposed approach, including architecture, training process, and inference procedures of our DNN-based classifier. In the evaluation section, we then compare the proposed approach against the state of the art, and demonstrate its extensibility by expanding the set of detected transformations. Finally, we conclude by affirming the approach validity in relation to the discussed requirements, also suggesting potential avenues for future research and development.

II. Literature review

The field of multimedia phylogeny was pioneered with Dias *et al.* [2], where the process of constructing a phylogeny tree from a set of near-duplicate is described along three steps:

1) *Near-duplicate set definition*: This involves identifying a set of documents believed to originate from a common source and/or having experienced analogous modifications, assuming a set of transformations that is denoted as $\mathcal{T}$.
2) *Dissimilarity matrix calculation*: For each near-duplicate set, a matrix represented as $\mathcal{D}$ is determined, capturing the degree of similarity or dissimilarity among document pairs. Typically, this matrix is asymmetric, thereby reflecting the directional aspect of the transformations.
3) *Optimum branching algorithm*: The matrix $\mathcal{D}$ is then fed into an optimum branching algorithm, resulting in a tree-like graph that depicts the evolutionary ties between the documents: Each node represents an individual document, and each connection represents the transformations performed upon parent document to generate its subsequent child document.

As for image phylogeny, building on the aforementioned approach, several optimizations have been proposed over the years, e.g. aiming at the dissimilarity matrix calculation [6], the branching algorithm [7], or approaches that go beyond near-duplicate set analysis, targeting so-called phylogeny forests and multiple parents sets [8]–[11]. Up until the most recent studies from [12] and [13], which dealt with the more complex scenarios of multiple parent phylogeny. The topic of video phylogeny, on the other hand, followed with [14], in which the image phylogeny techniques presented in [2] were applied to frames within near-duplicate videos. In further works, more demanding video phylogeny approaches were presented, addressing videos that are temporally misaligned [15], as well as video phylogeny forests and the possibility of multiple parent videos [16].

As for audio phylogeny (AP) analysis, which is the focus of our work, only three approaches have been published so far. They are described in the following, with their main elements being summarised in Table I.

1) *AP via brute force analysis:* Inspired by the existing work in image phylogeny, Nucci *et al.* [3] proposed a phylogeny approach that analyses a pool of near-duplicate audio tracks. We use the 'brute force' as it applies all possible combinations of transformations and parameters (up to two cycles) to the near-duplicate parent audio file prior to computing its similarity with a potential child audio file. The most time-consuming operation of the approach is the estimation of the transformation τ and parameters β required to obtain the dissimilarity matrix.
2) *AP via detection functions:* Having a focus on encoding detection and exact transformation estimation, Maksimović *et al.* [4] uses multiple detection functions, to identify particular audio transformations, reducing the associated computational cost when estimating the most likely transformation, denoted as τ, and its parameters β.
3) *AP via geometric transforms:* Aiming at an audio phylogeny analysis approach that reduces the computational cost of brute force transformation detection, Verde *et al.* [5] use 2D spectrogram representations of audio tracks in order to find a correlation between audio transformations and detected geometrical transformations via estimated affine transformation parameters. This allows for an estimation of whether transformations like time stretch, pitch shift, or trimming have been applied between two analyzed audio files. Additionally, the presence of dimming in the spectrogram may suggest that fade in/out or mp3 encoding operations have occurred.

While the 'brute force' approach by Nucci *et al.* [3] lacks extensibility and computational efficiency, both Maksimović *et al.* [4] and Verde *et al.* [5] achieved improvements in terms of computational efficiency. Even though the approach by Maksimović *et al.* [4] restricts the number of transformations, there is a number of them to be applied on every pair of audio files. Likewise, Verde *et al.* [5] improves the efficiency of [3] but still uses computationally demanding operations in order to detect parameters of time-/pitch- shifting or trimming which are repeated up to two times. Furthermore, both approaches cannot be considered extensible because every new transformation introduced in the analysis set requires the implementation of a specific detection function.

To overcome the aforementioned lack of extensibility while keeping computational complexity at bay, we propose to leverage the power of neural networks to automatically learn and detect audio transformations without the need for explicit feature engineering. This allows for easy addition of new transformations to the system and improves the extensibility of the approach, without significant effort needed to manually design transformation detection functions. Furthermore, by utilizing neural networks for transformation estimation, our proposed approach reduces the number of transformations that need to be applied to every pair of audio files, thus improving scalability for large datasets and real-time applications.

III. Proposed Method

Let us denote a set of audio files with $\mathcal{A} = \{a_k\}, k \in [1, K]$. In the following, we are going to assume that all files in $\mathcal{A}$ were created starting from a single source audio file, that we refer to as *root*, and that $\mathcal{A}$ is a set of near-duplicate audio

TABLE I
SUMMARY OF THE STATE-OF-THE-ART APPROACHES FOR AUDIO PHYLOGENY (AP)

	AP via brute force [3]	AP via detection functions [4]	AP via geometric transforms [5]
Transformation set $\mathcal{T}$	*Trim / Fade:* $0 \dots 3$ seconds *Mp3 encoding:* 256,192 kbps *Aac encoding:* 256,192,128 kbps	*Trim / Fade:* $0 \dots 3$ seconds *Mp3 encoding:* 320,192,128 kbps *Aac encoding:* 320,192,128 kbps	*Trim / Fade:* $0 \dots 3$ seconds *Time stretch:* $-10\% \dots +10\%$ *Pitch shift:* $-1 \dots +1$ semitones *Mp3 Coding:* quality factor 2,3,4
Transformation detection	none	custom detection functions	time and pitch (affine transform) Mp3 and fade in/out (dimming)
Dissimilarity measure	SNR ratio between time signals	Euclidean distance between features	MSE between spectrograms
Branching algorithm	Oriented Kruskal [2]	Oriented Kruskal [17]	Chu-Liu optimum branching [2]

files related one to another by a *parent-child* relation: Given a pair of files $(a_i, a_j), i \neq j$, a parent-child relation $a_i \xrightarrow[\tau_1,\tau_2,\dots]{} a_j$ exists if and only if a_j was created by applying one or more transformations $\tau_1, \tau_2, \dots$ to a_i, where each transformation is drawn by a finite set $\mathcal{T} = \{\tau_b\}, b \in [1, B]$.

In the following, we are going to propose an algorithm to reconstruct the entire set of parent-child relations and most likely transformations thereof, composed of three steps:

1) *Audio transformation estimation (Section III-A)*: Given a pair of files (a_i, a_j) estimate the probabilities $p^{\mathcal{T}} = \{p_{ij}^{\tau_b}\}$ associated to all possible parent-child relations $a_i \xrightarrow[\tau_b]{} a_j$ due to a single transformation τ_b;
2) *Dissimilarity matrix calculation (Section III-B)*: Given the entire set of files $\mathcal{A}$ and the aforementioned probability vector p, determine a dissimilarity matrix $\mathcal{D} = \{d_{ij}\}$ quantifying the dissimilarities between each pair of files (a_i, a_j);
3) *Phylogeny tree reconstruction (Section III-C)*: Given the dissimilarity matrix $\mathcal{D}$, reconstruct the entire phylogeny tree.

The overall workflow and the interactions between the three steps are also depicted in Figure 2.

A. Audio transformation estimation

The goal of this first step is to determine the most likely transformation τ_b which has occurred between each pair of files (a_i, a_j) in the analysis set $\mathcal{A}$. The seminal work by Nucci *et al.* [3] realized this step by means of an exhaustive search, which, however, is highly demanding and nearly unfeasible for large datasets. This issue was partially addressed by Maksimović *et al.* [4], who proposed a two-step procedure based on a first coarse search followed by a refinement to reduce the required amount of computation.

In this work, instead, we propose to address the transformation estimation in a single step. Given a pair of input audio files (a_i, a_j), we interpret the transformation estimation as a closed-set classification problem, in which each class represents one possible transformation τ_b. The probability of each transformation can thus be computed, e.g., by reading the b-th output of a neural network DNN$(\cdot)$ trained ad-hoc:

$$p_{ij}^{\mathcal{T}} = \{p_{ij}^{t_b}\} = \{p\,(t_b \mid (a_i, a_j))\} = \text{DNN}(a_i, a_j). \quad (1)$$

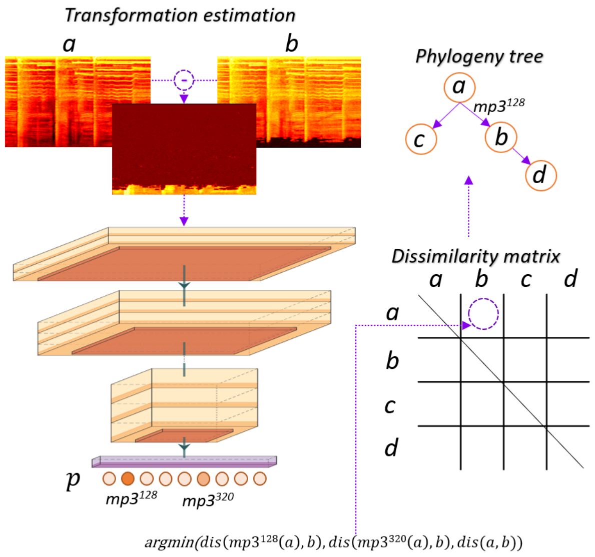
Transformation estimation
a
b
-
Phylogeny tree
a
mp3^128
c
b
d
Dissimilarity matrix
a b c d
a
b
c
d
p
mp3^128
mp3^320
argmin(dis(mp3^128(a),b),dis(mp3^320(a),b),dis(a,b))

Fig. 2. Complete audio phylogeny analysis system with transformation prediction via DNN classifier, dissimilarity calculation, and tree reconstruction

More in detail, we propose to realize the transformation estimation as follows:

1) Map the input pair of audio files (a_i, a_j) to the difference of their mel-spectrograms (M_i, M_j):
$$X_{ij} = (M_i - M_j) = \text{melspec}(a_i) - \text{melspec}(a_j), \quad (2)$$
where $M_i, M_j \in \mathbb{R}^{T \times F}$, with T denoting the amount of frames and F the amount of mel-frequency bins.
2) Extract ResNet50 emmebeddings y_{ij}^{emb} from the input matrix X:
$$y_{ij}^{\text{emb}} = \text{ResNet50}(X_{ij}), \quad (3)$$
where $y_{ij}^{\text{emb}} \in \mathbb{R}^{2048}$.
3) Feed the embeddings y_{ij}^{emb} to a feed-forward classification network FF$(\cdot)$ to compute the class probabilities:
$$p_{ij}^{\mathcal{T}} = \{p_{ij}^{t_b}\} = \text{FF}\left(y_{ij}^{\text{emb}}\right). \quad (4)$$

In order to interpret the output layer as class probabilities, we used one-hot encoding for each transformation in the set, and trained the network using Binary Cross Entroy (BCE) loss. Further details on the structure are reported in Table II.

TABLE II
STRUCTURE OF THE FEED-FORWARD NETWORK FF $(\cdot)$ FOR TRANSFORMATION CLASSIFICATION

Layer	Type	Nb Neurons	Output function	Parameters
1	Dropout	2048-2028	Linear	25%
2	Linear	2028-B	Softmax	

B. Dissimilarity matrix calculation

Given a pair of audio files (a_i, a_j) in the analysis set $\mathcal{A}$, the goal of this step is to compute a dissimilarity score d_{ij} between the two files, and to store it in a global dissimilarity matrix $\mathcal{D} := \{d_{ij}\}$.

We propose to perform this step by leveraging the transformation-probability vector $p_{ij}^{\mathcal{T}}$. Let us denote with τ_1 and τ_2 the two most likely transformations, i.e., the ones corresponding to the first and second largest elements in $p_{ij}^{\mathcal{T}}$. The matrix $\mathcal{D}$ can be built by applying the following procedure to each input pair of audio files (a_i, a_j):

1) Apply the two most likely transformations to the candidate parent file a_i, and retrieve the corresponding mel-spectrograms:

$$\begin{cases} M_i^1 = & \text{melspec}\left(\tau_1\left(a_i\right)\right), \\ M_i^2 = & \text{melspec}\left(\tau_2\left(a_i\right)\right). \end{cases} \tag{5}$$

2) Use the mel-spectrograms to compute possible dissimilarity values between the candidate child file a_j and its possible parent a_i:

$$\begin{cases} d_{ij}^{\varnothing} = & \|M_i - M_j\|_2^2, \\ d_{ij}^{1} = & \|M_i^1 - M_j\|_2^2, \\ d_{ij}^{2} = & \|M_i^2 - M_j\|_2^2, \end{cases} \tag{6}$$

where $\|\cdot\|_2^2$ denotes the squared l^2 norm, and with the notation $d_{ij}^{\varnothing}$ we underline that the dissimilarity is computed without applying any transformation to a_i.

3) Determine the final dissimilarity score d_{ij} for the current pair (a_i, a_j):

$$d_{ij} = \min\left(d_{ij}^{\varnothing}, d_{ij}^{1}, d_{ij}^{2}\right), \tag{7}$$

with the remark that in the general case, we would expect $d_{ij} \neq d_{ji}$, and hence the procedure does not return a distance.

C. Phylogeny tree reconstruction

In the last step, the dissimilarity matrix $\mathcal{D}$ is used to reconstruct the entire phylogeny tree. Similarly to the approaches in [3], [4], we propose to perform this last operation by applying the Oriented Kruskal algorithm, as detailed in [2], to transform $\mathcal{D}$ in a directed graph which corresponds to a phylogeny tree.

As depicted in Figure 2, the nodes of the tree represent files, the connections between the nodes correspond to parent-child relations (and transformation thereof), and it is possible to identify a unique root for the whole near-duplicate input set.

IV. EVALUATION

The algorithm presented in the previous section was evaluated in two phases: In the first phase, we compared the performance and scalability of the proposed approach against state-of-the-art methods, using a base set of transformations the pre-existing algorithms have been designed for. In the second phase, we tested the adaptability of the proposed approach to new demands by extending the set of considered transformations and evaluating the resulting performance.

A. Training setup

The network for transformation detection was trained on a private dataset, composed of 32000 high-quality files, which did not undergo any quality-degrading modification after being recorded.

Given a set of transformations under analysis $\mathcal{T} = \{\tau_b\}$, each file in the training dataset was split into non-overlapping segments of 4 seconds. Each segment was processed by all transformations in the set to produce B variants of the original signal, with B being the amount of transformations. Each variant was also re-processed again by all transformations. Every parent-child relation $a_i \xrightarrow[\tau_b]{} a_j$ created following this two-generations procedure was labeled accordingly to produce the expected values of the output vector $p_{ij}^{\mathcal{T}}$.

We configured the network to process audio signals of 3 seconds length, extracting mel-spectrograms using a window length of $46.4ms$ and hop size of $5.8ms$. Therefore, the input spectrograms have $T = 517$ frames and $F = 256$ frequency bins. The network was trained with stochastic gradient descent using a batch size of 10 samples. The initial learning rate was 1e-3, and was reduced by 10% every 10 epochs, for a total number of 30 epochs.

B. Comparison with the state of the art

To evaluate our method against the existing state of the art, we considered the following base set of transformations:

$$\begin{split} \mathcal{T}_b = \{&\text{none}, \text{mp3}_{320}, \text{mp3}_{192}, \text{mp3}_{128}, \\ &\text{aac}_{320}, \text{aac}_{192}, \text{aac}_{128}, \text{fade}, \text{trim}\}, \end{split} \tag{8}$$

in which trim and fade may occur with variable length between 0.5 and 3 seconds. This set of transformations is equivalent to the ones used in [3] and [4], which are the implementations that we selected for the comparison.

We then collected 6 high-quality source files (3 containing speech signals and 3 containing music), which were not exposed to any quality degrading transformations in the past and could thus be used to create annotated tests of phylogeny trees composed of near-duplicates. These trees were created by selecting one of the transformations from the previously defined set $\mathcal{T}_b \setminus \{\text{none}\}$, to be applied firstly on the root and then on a randomly selected parent in a near-duplicate set until the number of files in the test set reaches 20. These template trees were then applied on every of 6 root audio files, leading to a base test set $\mathcal{S}_b$ composed of 60 phylogeny test trees with 20 nodes each.

TABLE III
COMPUTATION EFFICIENCY ON A SMALL TREE WITH 10 NODES

	Proposed	Approach from [4]	Brute force [3]
time in s	32.7	31	207.3

All calculations performed on the same machine

TABLE IV
TRANSFORMATIONS DETECTION TEST

	1st best transformation	1st and 2nd best transformation
set $\mathcal{S}_b$	87.4%	98.2%
set $\mathcal{S}_e$	83.5%	96.0%

The evaluation metrics used are the ones originally proposed in [2] and then adopted as standard for evaluating a reconstruction of phylogeny trees. The amount of correctly reconstructed roots R, edges E (parent-child links), leaves L (nodes with no children), and ancestry A (lists of all children derived from every node) between ground truth Audio Phylogeny Tree APT_{gr} and reconstructed one APT_r, is measured as follows:

$$\text{Root: } R(APT_{gt}, APT_r) = \begin{cases} 1, & \text{If } Root(APT_{gt}) = Root(APT_r) \\ 0, & \text{Otherwise} \end{cases}$$

$$\text{Edges: } E(APT_{gt}, APT_r) = \frac{|E_{gt} \cap E_r|}{|E_{gt}|}$$

$$\text{Leaves: } L(APT_{gt}, APT_r) = \frac{|L_{gt} \cap L_r|}{|L_{gt} \cup L_r|}$$

$$\text{Ancestry: } A(APT_{gt}, APT_r) = \frac{|A_{gt} \cap A_r|}{|A_{gt} \cup A_r|}$$

Figures 3 to 5 show the results on the evaluation set $\mathcal{S}_b$ for the proposed approach and for the state-of-the-art methods by Maksimović *et al.* [4], and Nucci *et al.* [3].

Unlike the existing state-of-the-art methods, our algorithm was able to identify correctly the root of all phylogeny trees in $\mathcal{S}_b$ independently from the amount of nodes which were pruned. The amount of edges and leaves which were identified correctly is systematically higher than in [4], and decrease at a slower pace than in [3], even though the pre-existing proposal is based on an exhaustive search. Lastly, our method retrieves the highest amount of parent-child relations across generations, as reflected by the ancestry measure being the highest.

In summary, our approach managed to outperform the state of the art in terms of sheer performance. A reason which might explain why the two state-of-the-art approaches performed worse than we would have expected could be that, occasionally, the evaluation set $\mathcal{S}_b$ might present a low dissimilarity score between audio files that do not have a direct link due to the random selection of transformations. Whenever this happens, the low dissimilarity values between one node and many potential parents could lead to ambiguous results in the tree reconstruction of the two pre-existing methods.

In addition, the execution time needed for the reconstruction (as reported in Table III) of a small test tree of 10 nodes is considerably lower than the time needed by the brute force approach [3]. On the contrary, if we compare our execution time to the one achieved in [4], our approach took 1.7 seconds longer: The advantage of our proposal compared to [4] is thus not the sheer execution time, but rather its extensibility and expected efficiency invariance when the set of transformations under analysis $\mathcal{T}$ is augmented.

C. Extensibility and transformation detection

In this section, we test the extensibility of our approach with new transformations, as well as its transformation detection capabilities – requirements that are not satisfied by the existing state of the art. Therefore, we performed additional experiments by considering an extended set of transformations:

$$\mathcal{T}_e = T_b \cap \{\text{pitch}_{\text{up}}, \text{pitch}_{\text{down}}, \text{time}_{\text{up}}, \text{time}_{\text{down}}\}, \quad (9)$$

aiming at the detection of pitch shifting (up or down) and time stretching (speed up or slow down).

This extended set of transformations was used to generate a new set of phylogeny trees S_e, using the same generation procedure described in the previous section. Pitch shifting was applied randomly between -1 and +1 semitones, while for time stretching, we considered values between 0 and 10% of speed up/slow down. Once again, we started from 6 files to produce a total of 60 phylogeny trees of near-duplicates, composed by 20 nodes each. Both datasets we created, S_b and S_e, are publicly available online[1].

We retrained our DNN model while extending our training dataset to cover all 13 transformations as in Equation (9). Figure 6 shows the results of the newly trained model with 13 classes on the dataset $\mathcal{S}_e$. It shows, that the results are at a similar level as the ones we achieved on a set with a lower number of transformations $\mathcal{S}_b$ as presented in Figure 3.

We have further tested if the performance changed when we use a model trained on 13 classes on a dataset $\mathcal{S}_b$ and come to the conclusion that the performances stayed equivalent as shown in Figure 7.

In order to identify the performance of our DNN classifier on predicting the transformations between nodes in phylogeny trees, we have evaluated the amount of correctly detected transformations in both evaluation sets $\mathcal{S}_b$ and $\mathcal{S}_e$. The results of this test are shown in Table IV.

V. CONCLUSION AND FUTURE WORK

We presented a novel approach to audio phylogeny, addressing the challenge of transformation detection between two near-duplicate audio files using a neural network.

The proposed method outperformed the current state of the art while maintaining computational efficiency, and retained its performance after expanding the initial set of transformations, showing that it can be extended at a minimal cost. Thanks to its transformation detection performance, we believe that it can support many media forensic applications.

[1] M. Gerhardt *et al.*, *IDMT audio phylogeny dataset*, version 1.0.0, Zenodo, 2023. DOI: 10.5281/zenodo.8135331 [18]

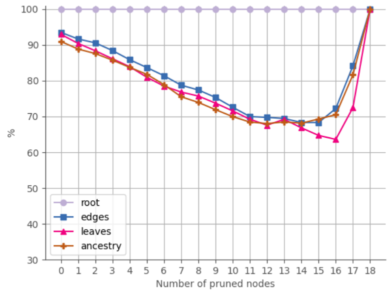

Fig. 3. Reconstructed phylogeny trees results for proposed approach, on the set $\mathcal{S}_b$

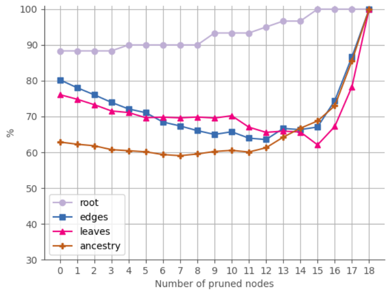

Fig. 4. Reconstructed phylogeny trees results for approach proposed in [4], on the set $\mathcal{S}_b$

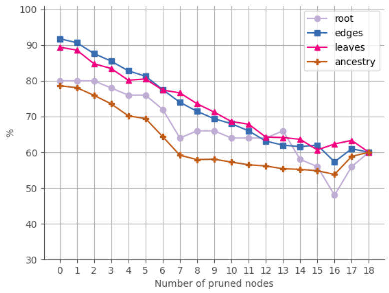

Fig. 5. Reconstructed phylogeny trees results for approach proposed in [3], , on the set $\mathcal{S}_b$

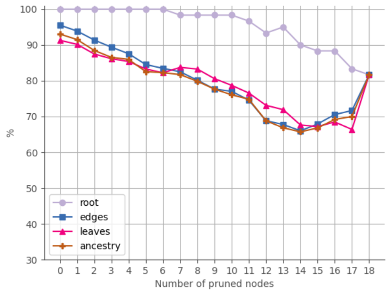

Fig. 6. Reconstructed phylogeny trees results for proposed approach, on the set $\mathcal{S}_e$

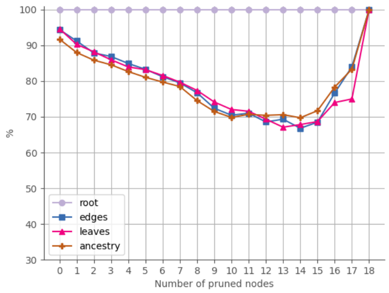

Fig. 7. Reconstructed phylogeny trees results for proposed approach on the set $\mathcal{S}_b$ using the NN model trained on 13 classes

In future research, we want to experiment with neural networks that are more suitable for audio input, and further to explore the complexity of multi-parent audio phylogeny.

References

[1] Z. Dias, A. Rocha, and S. Goldenstein, "First steps toward image phylogeny," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2010, pp. 1–6.

[2] Z. Dias, A. Rocha, and S. Goldenstein, "Image phylogeny by minimal spanning trees," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 7, no. 2, pp. 774–788, 2012.

[3] M. Nucci, M. Tagliasacchi, and S. Tubaro, "A phylogenetic analysis of near-duplicate audio tracks," in *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2013, pp. 99–104.

[4] M. Maksimović, L. Cuccovillo, and P. Aichroth, "Phylogeny analysis for MP3 and AAC coding transformations," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2017, pp. 1165–1170.

[5] S. Verde *et al.*, "Audio phylogenetic analysis using geometric transforms," in *IEEE Workshop on Information Forensics and Security (WIFS)*, 2017, pp. 1–6.

[6] Z. Dias, S. Goldenstein, and A. Rocha, "Large-scale image phylogeny: Tracing back image ancestry relationships," *IEEE Multimedia*, vol. 20, pp. 58–70, 3 2013.

[7] Z. Dias, S. Goldenstein, and A. Rocha, "Exploring heuristic and optimum branching algorithms for image phylogeny," *Journal of Visual Communication and Image Representation (JVCI)*, vol. 27, no. 7, pp. 1124–1134, 2013.

[8] Z. Dias, S. Goldenstein, and A. Rocha, "Toward image phylogeny forests: Automatically recovering semantically-similar image relationships," *Forensic Science International*, vol. 231, no. 1–3, pp. 178–189, 2013.

[9] M. A. Oikawa *et al.*, "Manifold learning and spectral clustering for image phylogeny forests," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 11, no. 1, pp. 5–18, 2016.

[10] F. de O. Costa *et al.*, "Image phylogeny forests reconstruction," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 9, no. 10, pp. 1533–1546, 2014.

[11] A. Oliveira *et al.*, "Multiple parenting identification in image phylogeny," in *IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 5347–5351.

[12] A. Bharati *et al.*, "U-phylogeny: Undirected provenance graph construction in the wild," in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 1517–1521.

[13] X. Zhang *et al.*, "Discovering image manipulation history by pairwise relation and forensics tools," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 5, pp. 1012–1023, 2020.

[14] Z. Dias, A. Rocha, and S. Goldenstein, "Video phylogeny: Recovering near-duplicate video relationships," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2011, pp. 1–6.

[15] F. O. Costa *et al.*, "Phylogeny reconstruction for misaligned and compressed video sequences," in *IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 301–305.

[16] S. Lameri *et al.*, "Who is my parent? Reconstructing video sequences from partially matching shots," in *IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 5342–5346.

[17] Y.-J. Chu and T.-H. Liu, "On the shortest arborescence of a directed graph," *Scientia Sinica*, vol. 14, no. 10, pp. 1396–1400, 1965.

[18] M. Gerhardt and L. Cuccovillo, *IDMT audio phylogeny dataset*, version 1.0.0, Zenodo, 2023. DOI: 10.5281/zenodo.8135331.