TU WIEN Informatics

Schutz des geistigen Eigentums von Sprechererkennungsmodellen durch Audio-Wasserzeichen

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Data Science

eingereicht von

Yelyzaveta Klysa
Matrikelnummer 12208877

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Andreas Rauber
Mitwirkung: Univ.Lektor Dipl.-Ing. Mag.rer.soc.oec. Rudolf Mayer

Wien, 7. September 2025

Yelyzaveta Klysa	Andreas Rauber

TU WIEN Informatics

Protection the Intellectual Property of Speaker Recognition Models via Audio Watermarking

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Data Science

by

Yelyzaveta Klysa
Registration Number 12208877

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Andreas Rauber
Assistance: Univ.Lektor Dipl.-Ing. Mag.rer.soc.oec. Rudolf Mayer

Vienna, September 7, 2025

Yelyzaveta Klysa	Andreas Rauber

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Yelyzaveta Klysa

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.
Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel“ habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, haben ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT- Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 7. September 2025

Yelyzaveta Klysa

Acknowledgements

I would like to express my sincere gratitude to my mother for her unwavering and unconditional love, support, and patience throughout the whole period of my education. I am extremely grateful to you for encouraging me to try everything I was hesitant to start. I would also like to thank my grandparents, who believe in me and the possibility of my accomplishments.

A special thanks to Lesia, who convinced me to pursue the path of Data Science and move to Vienna. You have been a constant motivator during my studies in general, and this writing process specifically.

I am deeply grateful to Rudolf Mayer for his valuable insights, detailed feedback, and incredible support at all stages of this work creation. Thank you for paying attention to all the intricacies of my ideas. I am delighted to have you as my supervisor. I sincerely appreciate the hours of discussions and guidance from Tanja Šarčević, who remained the source of inspiration during this thesis. I could not have done this without you supporting me in shaping my thoughts and bringing structure to them.

I am thankful to my supervisor, Andreas Rauber, for his feedback and all the meetings that flipped my perspective on the problems.

Kurzfassung

Die erheblichen Kosten für die Entwicklung von Machine-Learning-Modellen, zusammen mit ihrer zunehmenden Verbreitung in Anwendungen wie der Sprechererkennung, machen den Schutz des geistigen Eigentums dieser Modelle zu einem wichtigen Forschungsthema. Black-Box-Watermarking-Techniken bieten einen Mechanismus, um die unrechtmäßige Wiederverwendung von Modellen nachzuweisen, beispielsweise wenn diese über einen API-Dienst bereitgestellt werden. Obwohl Watermarking im Bildbereich bereits umfassend untersucht wurde, sind Methoden für Audiomodelle, insbesondere für die Sprechererkennung, nach wie vor rar. Darüber hinaus weisen aktuelle Methoden methodologische Schwächen auf, darunter mangelnde Reproduzierbarkeit und unzureichende Evaluierung; in einigen Fällen basieren sie auf fragwürdigen Designentscheidungen.

Diese Arbeit präsentiert eine systematische Untersuchung von Black-Box-Watermarking-Methoden für Sprechererkennungsmodelle mit dem Ziel, deren Schwächen zu identifizieren und ihre Anwendbarkeit zu verbessern. Wir implementieren bestehende Watermarking-Methoden und replizieren die berichteten Ergebnisse, identifizieren und beheben Reproduzierbarkeitslücken und betonen dabei eine rigorose, reproduzierbare Vorgehensweise in allen zentralen Anforderungen. Darüber hinaus erweitern wir die Evaluierung durch den Einsatz geeigneterer Metriken für die Analyse der Wahrnehmbarkeit der Wasserzeichen sowie durch die Einführung des Konzepts der Angriffskosten, womit wir ein umfassendes Bewertungsframework für Sprechererkennungs-Wasserzeichen vorstellen. Ferner untersuchen wir die Anwendbarkeit der Watermarking-Methoden auf verschiedene Datensätze und Modellarchitekturen (d. h. ihre Allgemeingültigkeit) und bewerten das Risiko, dass Wasserzeichen von einer böswilligen Partei gefälscht werden können, ohne die genaue Methode oder die Einbettungsparameter zu kennen (d. h. ihre Rechtssicherheit). Auf dieser Grundlage schlagen wir anschließend Gegenmaßnahmen vor. Zudem analysieren wir die unterschiedlichen, in Konflikt stehenden Ziele der Unwahrnehmbarkeit und Robustheit der Wasserzeichen, und schlagen Strategien vor, die die Unwahrnehmbarkeit verbessern, ohne die Wirksamkeit des Wasserzeichens oder die Modelltreue zu beeinträchtigen. Abschließend geben wir Richtlinien für das Design und die Evaluierung von Watermarking-Methoden für die Sprechererkennung mit spezifischen Empfehlungen zur Parameterauswahl.

Abstract

The substantial costs of developing machine learning models, alongside their growing adoption in applications such as speaker recognition, make the protection of the intellectual property of these models an important research topic. Black-box watermarking techniques provide a mechanism to verify illicit reuse of models, for instance, when made available through an API service. Although watermarking has been extensively studied for models in the image domain, methods for audio models, specifically speaker recognition, remain scarce. Moreover, current methods exhibit methodological weaknesses, including the lack of reproducibility and insufficient evaluation; in some cases, they rely on questionable design choices.

This thesis presents a systematic study of state-of-the-art black-box watermarking methods for speaker recognition models, with the aim of identifying their weaknesses and improving their applicability. We implement existing watermarking methods and replicate their reported results, identifying and addressing reproducibility gaps, thereby emphasising rigorous, reproducible practice across all key requirements. We further extend their evaluation by employing more suitable metrics for imperceptibility analysis and by incorporating the notion of an attack cost, thereby presenting a comprehensive evaluation framework for speaker recognition watermarks. Furthermore, we examine the applicability of the watermarking methods across different datasets and model architectures (i.e., their generality), and assess the risk of watermarks being forged by a malicious party without knowing the exact watermarking method or embedding parameters (i.e., legality), based on which we subsequently propose mitigation strategies. In addition, we analyse trade-offs between imperceptibility and robustness, and propose strategies that enhance imperceptibility while preserving watermark effectiveness and model fidelity. Finally, we provide guidelines for the design and evaluation of speaker recognition watermarking methods, with specific recommendations on parameter selection.

Contents

CHAPTER 1

Introduction

1.1 Problem Statement

Neural networks require a considerable amount of resources to develop and sustain. Therefore, they have become increasingly valuable and may become the target of theft by malicious parties. Intellectual Property (IP) protection comprises various measures to prevent unauthorised use and distribution of proprietary assets, including machine learning models.

In recent years, research has increased on model protection techniques, especially in the image domain. On the other hand, the audio domain, specifically speaker recognition (SR) models and ways to protect them from being stolen, is still rather unexplored. Speaker recognition models are designed to identify and verify a speaker's identity based on an audio sample. Their primary task lies in determining the identity of a speaker from a list of known speakers. Existing models vary in their architectures and input structure, whether it is a technique based on (i) image processing that works with spectrograms or (ii) raw waveforms of audio signals.

Speaker recognition models are widely used in multiple domains, including biometrics, security, healthcare, customer service, and entertainment. Since these models become more sophisticated and prevalent, protecting them from unauthorised usage is crucial. Unauthorised duplication could lead to privacy violations, which may occur when stolen models are exploited to infer sensitive user data; financial losses; and security breaches, arising from adversarial attacks using stolen models.

One of the most studied approaches for intellectual property protection (IPP) of machine learning models is model watermarking. The idea is to embed a watermark into a model during the training stage, to be able to later prove one's ownership if such a need arises. Two main scenarios are distinguished by the access to the model that is possible during the ownership verification stage: white-box watermarking, when full access to

the internal structure of the model is required for both embedding and verification of the watermark; and black-box watermarking, which relies on memorisation of certain input-output patterns, hence, at the verification stage, the ownership of the model can be proven just via an inference process. The latter approach is especially relevant for real-world scenarios, as it provides greater flexibility in verifying ownership when stolen models are made available, e.g., only behind an API. Therefore, this thesis will focus on the black-box watermarking scheme.

Only a few recent studies [WW22, ZDX+23, LYS+24] discuss model protection techniques for speaker recognition models that were created specifically for the audio domain. Their research is based on convolutional neural network (CNN) models that use the raw waveform as input. However, these works are fragmented and do not follow a unified methodology, and are thus not easily comparable.

We have found several research gaps in the current literature regarding the applicability of these techniques to (i) different state-of-the-art Speaker Recognition model architectures, (ii) different input representations, and (iii) different datasets. So far, the proposed methods have been tested only for one model architecture with one possible input representation. Furthermore, there is an open research question of how the watermarking methods compare in terms of their characteristics due to the lack of a unified analysis among the methods. This thesis aims to fill these gaps and provide guidelines on the selection of the watermarking technique in different settings to Speaker Recognition model owners.

1.2 Research Questions

1. **How does the effectiveness of state-of-the-art watermarking techniques compare across different model architectures and datasets?**
   In order to explore the possibilities of using watermarks for models and datasets with different characteristics, they will be evaluated for scenarios divergent from the originally tested environment in terms of models and datasets. Not only watermark effectiveness, but also fidelity (the ability of the model to maintain its accuracy on the original task) will be considered since it is closely linked to effectiveness and is an essential requirement for model watermarking. This research question is further broken down into two sub-questions.

   a) How effective are the watermarking techniques for other datasets?
   This question aims to assess the applicability of watermarks to datasets that have distinct properties compared to the ones on which the methods were tested. Since speech datasets are created in various environments and have different qualities, it is of interest to explore whether these qualities affect SR watermark effectiveness or fidelity.

   b) To what extent do the watermarking techniques succeed in protecting models with different architectures?

The goal of this question is to investigate the generality of watermarks in terms of speaker recognition model architecture. Hence, the research question aims to analyse the applicability of the same watermarks for other SR models. Effectiveness and fidelity will be evaluated to assess the success rate of the watermarks.

2. **To what extent can we reduce the perceptibility of watermarks while still reaching the same watermark effectiveness?**
   This question aims to analyse the (im)perceptibility of the watermarks – since in a black-box scenario, sending heavily distorted audio to the model may be suspicious to the party that maliciously offers access to the stolen model, and watermark verification requests could thus be blocked by them. We intend to examine the perceptibility of the watermarks in the most effective settings based on the results from RQ1, considering both human and machine perception. Subsequently, we will explore which aspects of the watermark creation impact its imperceptibility and how we can improve it. That will serve as the basis for determining whether we can achieve the same results in terms of effectiveness using a less perceptible watermark.

3. **How robust are state-of-the-art audio watermarking techniques for SR models, and how can their robustness be improved?**
   Since several watermarking methods were not evaluated in terms of robustness, the objective of this research question is to assess their robustness against different watermark removal attacks. This includes both model transformations and audio preprocessing that a malicious party can deploy before running the inference of their stolen model. The goal is, therefore, to determine which attack, and to what extent, can degrade the watermark verification process, and subsequently how the robustness can be improved. Additionally, the degradation of original task accuracy is measured as part of the success evaluation of the attacks. This is crucial, as an attacker will aim to invalidate the watermark verification without suffering a (substantial) accuracy loss of the original task performance. This research question will consider whether the level of imperceptibility or a smaller number of trigger samples can affect the robustness of the watermark and potentially improve it.

1.3 Thesis Structure

The remainder of the thesis is organised as follows. Chapter 2 provides the necessary background information to support understanding of the work. Chapter 3 introduces the threat model that we consider, outlining the roles of the malicious party and the model owner, along with their incentives, knowledge, and capabilities. Chapter 4 discusses the current state of the art, with a focus on black-box watermarking of machine learning models in the audio domain. Chapter 5 describes the experimental design adopted in this work. Chapter 6 presents the evaluation and refinement of state-of-the-art watermarking methods for speaker recognition model protection, along with guidelines derived from

the findings. Chapter 7 outlines additional approaches that we explored to enhance the legality (the inability of a watermark being forged) and robustness of the watermarking methods. Finally, Chapter 8 summarises the main contributions of the thesis and offers a set of practical guidelines based on the overall study.

CHAPTER 2

Background

This chapter outlines the background knowledge relevant to the remainder of the thesis. It begins with an introduction to supervised machine learning, focusing in particular on the task of classification. We then move on to describing neural networks, with an emphasis on those variants used in speaker recognition, and their evaluation. Next, we explain key concepts in audio processing. We finish the chapter with an overview of intellectual property protection in the context of speech models.

2.1 Supervised Machine Learning

Supervised Learning (SL) is a category of Machine Learning (ML) that learns the patterns in the provided dataset. Based on the input variables, a model is trained to predict a target variable. The task of the model can be either classification or regression. The former means a model learns to predict a discrete value from a set of finite options. On the other hand, in a regression task, the target variable is a continuous value. The Speaker Recognition problem is thus a classification task.

2.1.1 Classification

Formally, classification task can be defined as: a target label y_i is assigned to a data sample x_i in dataset D, where $1 \leq i \leq n$, and n is the number of data samples. Each x_i consists of one or multiple features x_{ij}, $1 \leq j \leq m$, where m denotes the number of features. One row X_i of the dataset comprises pairs of data features x_i and predefined class y_i, which represents the ground truth against which the model's predictions are evaluated. The full dataset used for training a machine learning model can thus be

represented as:

$$D = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,m} & y_1 \\ x_{2,1} & x_{2,2} & \dots & x_{2,m} & y_2 \\ \vdots & \vdots & & \vdots & \vdots \\ x_{n,1} & x_{n,2} & \dots & x_{n,m} & y_n \end{bmatrix} \tag{2.1}$$

The goal of training a classification model is to capture the relationships between input features and output classes of the training set and to determine decision boundaries among classes as close to the original unknown function that divides the data as possible. Apart from distinguishing the unique patterns among classes in the training dataset, the model should generalise well to be able to correctly predict the target labels of unseen data based on their features.

The classification task can be further divided into three subcategories: binary classification (where there are only two distinct classes and one data sample can belong only to one of them), multi-class (the number of different classes is larger than two), and multi-label classification (in which one data point can be assigned to more than one class).

During the training phase, the dataset is divided into a training and a test set. A test set emulates previously unseen data and is employed to assess the generalisability of the trained model. Optionally, a validation set may be used to optimise the model during a training phase. It serves the purpose of the model evaluation on unseen data for e.g. choosing the best hyperparameters for the model training.

Regarding speech as input data, multiple tasks can be solved with machine learning models. A full tree overview can be found in [MHIM21]. Figure 2.1 highlights the main differences between the two main tasks in the domain: Speech Recognition and Speaker Recognition. Below, these two tasks are described in more detail.

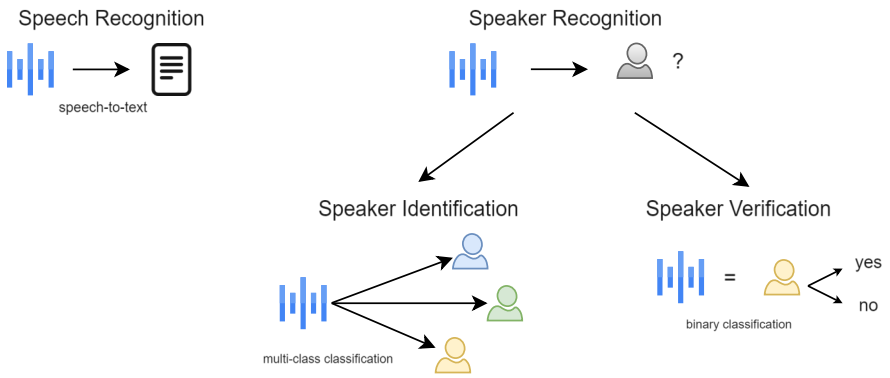


Figure 2.1: Difference between Speech and Speaker Recognition tasks.

Speech Recognition

A speech recognition model aims to identify the pronounced linguistic units: characters, phonemes and words, in order to convert them into text. Systems using speech recognition are widely incorporated in the daily lives of many people, e.g., in voice assistance, automatic transcription, voice-controlled devices such as smart homes, and in the medical domain [MHIM21]. The task deals with acoustic modelling - capturing the relations between raw audio signals and phonetic units - and language modelling - predicting the most probable words in sentences. The language for which the model is created is essential. In terms of training data, the primary requirement is a large corpus to capture language patterns. Nevertheless, diversity in the number of speakers is also important, as it ensures the model learns and generalises across different pronunciations, accents, and dialects. To evaluate the performance of this task, often the Character Error Rate (CER) and Word Error Rate (WER) are measured. The former calculates the difference between recognised and ground-truth characters, and the latter shows the same difference on the word level.

Speaker Recognition

Speaker Recognition models identify and verify a person's identity, based on an audio sample [Dod85]. The primary task of identification lies in determining the identity of a speaker from a set of known voices (i.e., a form of multi-class classification). The speaker verification task determines whether a particular voice from an audio truly belongs to a specific person (i.e., binary classification). This aspect of speaker recognition also concerns whether the model assigns an audio sample to a known speaker or rejects it altogether when the identity does not match any enrolled speaker. Datasets for the speaker recognition task do not require a large amount of recorded speech, as long as the model can capture the distinctive characteristics and differences among the speakers. Nevertheless, the number of speakers has a direct effect on the model's complexity: the more speakers one desires to identify with a model, the more audio samples are needed to differentiate them.

2.1.2 Neural Networks

A Neural Network (NN) is a type of machine learning model used for numerous tasks and goals.Inspired by the functions of biological neurons, of artificial neural computation were introduced by McCulloch and Pitts [MP43], and later extended by Rosenblatt [Ros58]. Rosenblatt's work was among the first to demonstrate a learning algorithm for pattern recognition and is considered a direct predecessor of modern neural networks. A neural network comprises layers of neurons that transform input features through weighted connections, learning patterns of input-output pairs, to produce target values. The general structure of NNs can be divided into three parts:

- The Input Layer, with a size of m features in X, receives the input representations

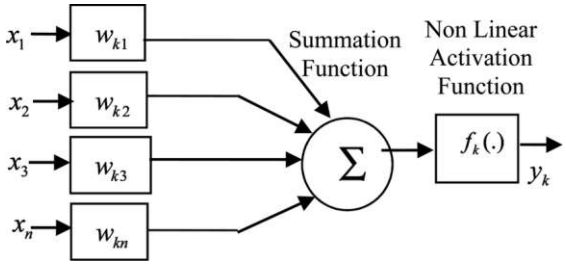


Figure 2.2: Example of a computation process in one neuron [ADND20].

of the data. For SR models, it can be a raw waveform, a spectrogram, or other extracted features of an audio, such as MFCC (for more details on audio features, see Section 2.1.4).

- The Hidden Layers perform the main transformation through the weighted connections between the neurons of different layers. The value of the k-th neuron, $h_k^{(l)}$, in hidden layer l is computed from the inputs. It is obtained as a weighted sum of the input values z_j, multiplied by their corresponding weights w_j, with an added bias term. Finally, an activation function is applied to this sum (Fig. 2.2). Additionally, the hidden layer may include different activation functions, dropouts, and batch normalisation.
- The Output Layer yields a final prediction $\hat{y}_i$ for each x_i. In multi-class classification, the number of output neurons typically corresponds to the number of classes C, whereas in binary classification, a single output neuron with a threshold function is often sufficient.

There are various commonly used activation functions that are applied to enable nonlinear classification of data. The ones relevant for this work are described below, with z being an input value to a neuron:

- *Softmax* produces values in the range between 0 and 1 that sum up to 1. This activation function is widely used in the output layer of a multi-class classification model to transform activations into probabilities of classes.

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \tag{2.2}$$

- *Sigmoid* produces output values in the range $(0, 1)$, which makes it suitable for modelling probabilities. It is used in binary classification tasks; however, it suffers from the vanishing gradient problem, where the derivative becomes very small for large positive or negative inputs. This can slow or even halt learning in deep neural networks.

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{2.3}$$

- *Rectified Linear Unit* produces only non-negative output and is computationally efficient. In addition, it mitigates the vanishing gradient problem, which is why it is often used in different NN architectures. However, ReLU can suffer from the *dying ReLU problem*, where neurons output zero for all inputs and stop learning if their weights are updated into a negative region.

$$\text{ReLU}(z) = \max(0, z) \tag{2.4}$$

- *Leaky ReLU* is a modified ReLU function that incorporates some amount of negative input values based on the value of α. This helps prevent neurons from becoming inactive due to the dying ReLU problem.

$$\text{Leaky ReLU}(z) = \begin{cases} z, & \text{if } z \geq 0 \\ \alpha z, & \text{if } z < 0 \end{cases}, \text{ where } \alpha \text{ is a small positive constant} \tag{2.5}$$

When all the neurons of each layer are connected to all neurons of the subsequent layer, and the connections form an acyclic graph, the network is called a fully connected neural network or Multilayer Perceptron (MLP). A neural network with more than one hidden layer is considered a Deep Neural Network (DNN).

During the training stage, the weights for each neuron in each layer are adjusted to minimise the error between the predicted value $\hat{y}_i$ and the ground truth y_i. The process consists of three stages:

- Feedforward: all input values are passed through the network, and the output is calculated.
- Loss Calculation: compute a loss L that quantifies the error between predicted and true values. A metric that is commonly used for multi-class classification tasks is the Cross-Entropy Loss, denoted as L_{CE}, which is defined as follows.

$$L = -\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c} \log(\hat{y}_{i,c}) \tag{2.6}$$

- Backpropagation is the process of updating the parameters (weights and biases) of a neural network based on the calculated loss. The key aspect of backpropagation is that the error is propagated layer by layer in a *backward* direction, starting from the output layer and moving towards the input layer. These updates are performed using optimisation algorithms like Stochastic Gradient Descent (SGD), Adam [KB17], or RMSprop [Rud17]. In the case of gradient descent, parameters are updated iteratively based on the gradients of a loss function. A *gradient* is a vector of partial derivatives of the loss with respect to the model parameters, indicating the direction of steepest increase of the loss. By moving the parameters in the direction opposite to the gradient, scaled by a learning rate, the loss is reduced and the network learns to improve its predictions.

Commonly, the training dataset is divided into smaller parts, namely *mini-batches*. The gradient is computed for each mini-batch rather than the entire dataset to enforce efficiency and help an optimiser to escape local minima.

Apart from vanishing gradient, exploding gradients may also occur when the product of derivatives across many layers becomes large. This can lead to extremely big weight updates and unstable learning.

It may also happen that a model overly memorises the patterns of a training set and cannot generalise well to unseen data, a problem that is referred to as overfitting. To address this issue, various techniques can be applied. One commonly used method is dropout [SHK+14], where a fraction of neurons is randomly deactivated during training. It is often mentioned in the literature as a regularisation technique [SHK+14], meaning a method that constrains the learning process to prevent the model from fitting the training data too closely.

As model parameters such as weights constantly change during training, the distributions of inputs to each layer (i.e., the activations from the previous layer) can change, a phenomenon known as internal covariate shift [IS15]. There are different approaches to normalisation to mitigate such issues.

Batch normalisation, introduced by [IS15], is performed within each mini-batch, normalising activations of each layer. In addition, it has a positive effect on the efficiency of the training.

Layer normalisation [BKH16], on the other hand, computes normalisation statistics for each individual sample rather than across a mini-batch. This makes it particularly suitable for scenarios with small batch sizes.

Convolutional Neural Network

The Convolutional Neural Network (CNN) [LBD+89] is a widely used type of NN for data that has local spatial structure. Originally designed for images, it works very well also for speech and audio data, having become the most popular choice of architecture [BZ21]. CNNs may contain three main components:

- A Convolutional Layer applies small, learnable *filters* (also called kernels) on the input. In a sliding window approach, these filters are applied over the input space to extract local patterns or structures. Multiple filters can be applied simultaneously to capture different local patterns. The *stride* in the convolutional layer defines the step width when moving the filter over the input, with a bigger value producing a smaller output. *Padding* is a technique in which additional values (typically zeros) are added around the borders of the input. Padding allows the convolution to process edge regions more effectively and can also influence the output size.
- A Pooling Layer aims to reduce the size of its input, performing dimensionality reduction while retaining important local information. It is achieved by applying

an aggregation function over small input windows, such as taking the maximum value (*max pooling*) or the average value (*average pooling*) within each window.

- A Fully Connected (FC) Layer has the same structure as MLP; these layers learn the mapping from the inputs that are processed by applying convolution and pooling layers, to the desired target classes.

Recurrent Neural Network

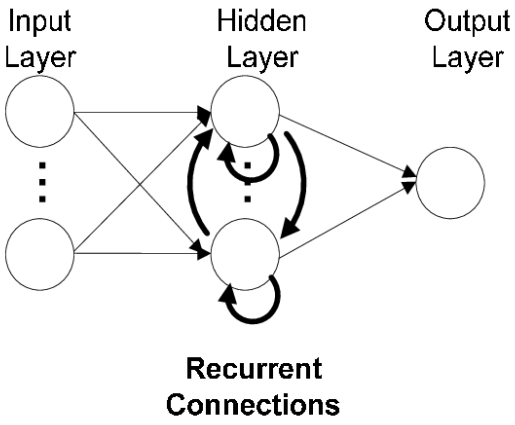


Figure 2.3: Schematic overview of connections in RNN [Car05].

A Recurrent Neural Network (RNN) is a type of neural network designed to process sequential data and memorise past events. The neurons in RNN can be connected in a cycle, making this type of NN especially suitable for time-series data, including audio. The neurons of such architecture can be connected in different ways, for instance, as shown in Figure 2.3: the neurons are connected not only to all neurons of the previous layer, but also recurrently to themselves and to other neurons within the same layer. In the figure, each input neuron is connected to all neurons of the recurrent layer; in turn, these recurrent neurons are interconnected with each other and with themselves; finally, they are also connected to the output neuron.

In RNN architecture, the activation of a neuron at time step t, denoted as h_t, is derived from both the current input value x_t and the previous hidden state h_{t-1}, and can be mathematically represented as follows:

$$h_t = \sigma\left(W_h h_{t-1} + W_x x_t + b\right), \tag{2.7}$$

where W represents weight matrices and σ represents an activation function.

During training, the network is unfolded across time steps, so that the recurrent connections are represented as a sequence of feed-forward layers. The outputs for each time

step are computed, and the loss is calculated over the entire unfolded sequence. Finally, backpropagation through time (BPTT) is applied, which extends the standard backpropagation algorithm to the unfolded network by propagating the gradients backwards through all time steps. In RNNs, the problem of vanishing and exploding gradients is more severe than in CNNs due to the fact that backpropagation is performed across time steps. Gradients are repeatedly multiplied by the same recurrent weights at each step, causing them to shrink or grow exponentially as the sequence length increases [BSF94, PMB13].

Long Short-Term Memory

The Long Short-Term Memory (LSTM) was proposed by Sepp Hochreiter and Juergen Schmidhuber [HS97] to overcome the vanishing gradient problem in RNNs.Specifically, LSTM introduces cells located in the hidden layers with three gates:

1. The Forget Gate f_t is responsible for the previous information, namely, how much of it should be transferred to the cell.
2. The Input Gate i_t determines whether and how much new information is added to the cell.
3. The Output Gate o_t calculates the combination of previous and new information.

In speaker recognition, LSTM networks are used independently in the text-dependent speaker verification task. Furthermore, they can also be integrated into hybrid architectures, where LSTMs are combined with other neural network types, such as CNNs, in order to capture both short-term local patterns and long-term temporal dependencies [BZ21].

2.1.3 Evaluation of ML Classification Performance

The evaluation of machine learning models is performed on the so-called test dataset. How effectively they perform, namely how well a trained model predicts correct target labels, can be measured by several different metrics. The choice of metrics varies depends on the type of model's predictions (e.g., class probabilities, frame-wise labels, or a single label per input sample). A more detailed overview of commonly used metrics for multi-class classification is presented below. We consider that the model M is evaluated on a test set D_{test} defined in the same way as in Equation (2.1). Each y_i corresponds to the predetermined label, i.e. one of the classes c_j with $j \in \{1, ..., C\}$.

The following prediction outcomes can be distinguished for a single instance with respect to its ground-truth label:

- **True Positive** TP_j of class c_j determines how many samples of class j are correctly classified as c_j.
- **False Positive** FP_j of class c_j shows how many samples were incorrectly predicted as c_j, when they belong to another class.

- **False Negative** FN_j counts how many samples of class c_j were misclassified as another class.
- **True Negative** TN_j shows how many instances that do not belong to c_j were correctly classified as not c_j.

Based on these prediction outcomes, commonly used metrics include:

- **Precision** P_j depicts how many predicted positives of class c_j belong to class c_j:

$$P_j = \frac{TP_j}{TP_j + FP_j} \tag{2.8}$$

- **Recall** R_j measures how many predicted positives of class c_j are correctly classified:

$$R_j = \frac{TP_j}{TP_j + FN_j} \tag{2.9}$$

- **F1-score** $F1_j$ provides a balanced measure that combines precision and recall metrics. Since it is easy to improve either precision or recall at the expanse of the other, the F1-score assesses the overall effectiveness of the model in a way that treats both metrics equally. It is computed as follows:

$$F1_j = \frac{2P_jR_j}{P_j + R_j} \tag{2.10}$$

- **Accuracy** Acc measures the overall proportion of correct predictions:

$$Acc = \frac{\sum_{j=1}^{N} TP_j}{\sum_{j=1}^{N}(TP_j + FP_j + FN_j)} \tag{2.11}$$

- **Error Rate** Err is an inverted measure of Accuracy that determines the proportion of incorrectly classified samples:

$$Err = 1 - Acc = \frac{\sum_{j=1}^{N} FP_j + FN_j}{\sum_{j=1}^{N}(TP_j + FP_j + FN_j)} \tag{2.12}$$

In Speaker Recognition models, predictions of the target label may be computed over *sentences* or *frames*. In that case, two more metrics are considered:

- **Sentence Error Rate** SER measures the proportion of sentences that are classified incorrectly by the model. Here, a *sentence* refers to the complete input unit being evaluated, which can be defined as an arbitrary part of an audio signal of a few seconds length, a fully pronounced sentence by linguistic means, an entire audio recording, etc. In some evaluation setups, SER can be computed as the inverse of Accuracy.

$$SER = \frac{Number\ of\ Incorrectly\ Classified\ Sentences}{Total\ Number\ of\ Sentences} \tag{2.13}$$

- **Frame Error Rate** FER measures the proportion of incorrectly classified frames. A frame normally refers to a short segment of an audio signal, during which the signal is assumed to be approximately stationary. It is worth noting that a low FER does not necessarily imply a low SER - if a model operates based on frame-wise prediction, then even if a (substantial) number of the frames are misclassified, the sentence may still be correctly classified, e.g., if the sentence class is determined by the most probable or most frequently recognised class out of all its frames (i.e., a majority voting).

$$FER = \frac{Number\ of\ Incorrectly\ Classified\ Frames}{Total\ Number\ of\ Frames} \tag{2.14}$$

For Speaker Verification tasks, additional metrics for how well the speakers are distinguished [KMS+21] have been proposed:

- The False Acceptance Rate FAR measures how often an unauthorised identity is incorrectly accepted as a known speaker.
- The False Rejection Rate FRR captures how often a legitimate user is incorrectly rejected.
- The Equal Error Rate EER is a point of balance when $FAR = FRR$.

In this work, we will use multiple metrics to measure the effectiveness of the trained models. The choice will be made based on their prediction algorithm and primary metrics used in the original articles that proposed these models. Nevertheless, to make the results comparable, they will be converted, where possible, to Accuracy, Error Rate, or Sentence Error Rate.

2.1.4 Audio processing for Machine Learning

The way audio is processed for model training is highly dependent on the type of audio signal and the task that this model aims to solve.

The audio signals in datasets are usually provided in formats that contain raw waveforms (e.g., *.wav* or *.flac*). The audio waveform can be displayed in time as a change in amplitude, which represents the intensity of the sound, as shown in Figure 2.4.

The original audio signal is an analog signal that needs to be converted into a digital signal, so that it can be processed by a computer. This process is called Analog-Digital-Conversion (ADC), during which a continuous-time audio signal undergoes the following transformations:

- Sampling - the amplitude is measured in discrete time periods determined by the sampling frequency f_s, also called the sampling rate.

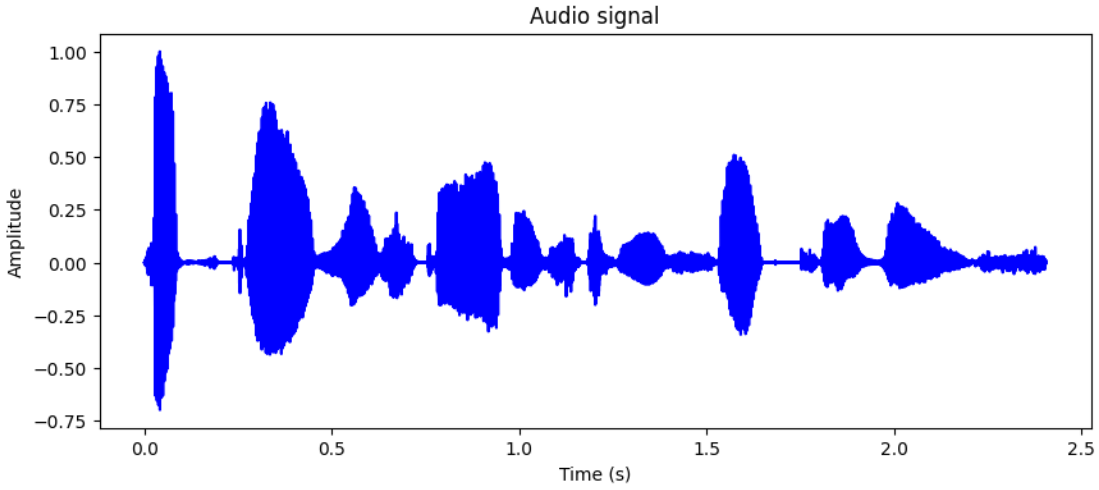


Figure 2.4: Example of a raw waveform of audio signal from TIMIT[GLF+93] dataset after ADC.

- Quantisation - in each resulting sample, the amplitude value is saved as the closest value in a finite set of possible quantisation steps.
- Encoding - the values represented as binary sequences are stored.

The sampling frequency f_s affects the level of detail captured during the sampling process, with a higher value resulting in more detail captured. The standard sample rate for speech processing is 16,000 Hz. Figure 2.4 depicts a speech audio signal after ACD.

The core characteristics of a signal lie in its frequency. To analyse a signal's frequency-related features, the audio must first be converted from time to frequency domain using the Fourier Transform (FT) [Mü15]. For a digital version of the signal, the Discrete Fourier Transform (DFT) is applied to a signal sampled over discrete time intervals. The result consists of the magnitude and phase information corresponding to each frequency bin. While magnitude incorporates the amplitude of the frequency in a signal, phase shows at which point the frequency component starts. The magnitude spectrum represents the contribution of each frequency in the audio signal. It encompasses all the magnitude information received from the output of the DFT.

Before further processing the digital signal to retrieve meaningful features, it is usually divided into smaller overlapping segments, also called frames. This step is performed because the signal is volatile and usually changes over time. Such volatility can arise from variations in pitch, intensity, background noise, or speaker articulation. When we process a smaller segment of the audio signal, the underlying assumption is that it does not change much during this period. To preserve the continuity of the signal, the frames need to overlap, especially in speech audio signals. This is crucial to avoid losing phonetic details and ensure a smooth transition between frames. An example of a segmentation

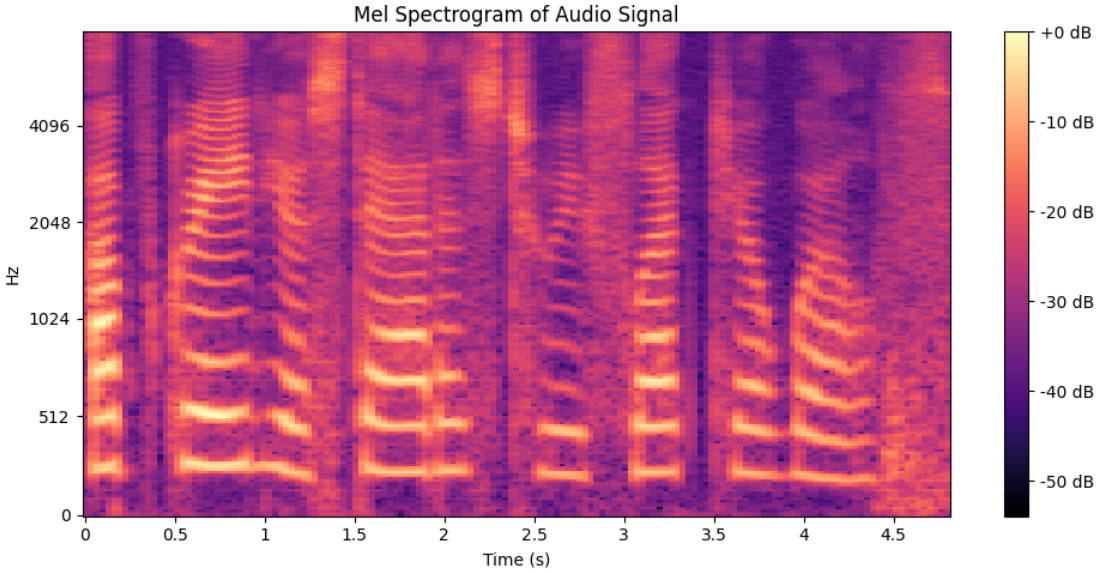


Figure 2.5: Example of a Mel-spectrogram of an audio signal.

technique is silence-based, which divides the audio based on detected silences in the speech.

A Fourier Transform can also be applied to the segments of the audio. In that case, a technique called Short-Time Fourier Transform (STFT) is used. It applies the Fourier Transform to overlapping frames of a signal with the help of a window function w. Specifically, each segment is multiplied by a window function, then the DFT is computed, and all resulting parts can be put back together into a time-frequency representation. The shape of the window function affects how much information from the boundaries of each segment contributes to segment analysis. The general outcome of this step is to get non-zero values for a desired small period of time, and zero elsewhere. The most well-known window functions are rectangular, triangular, Hamming, and Hann windows. The last one is the most common in audio analysis due to the smooth processing of the edges. It follows the cosine shape and can be represented as:

$$w(n) = \frac{1}{2}\left[1 - cos\left(\frac{2\pi n}{N-1}\right)\right], \quad 0 \leq n < N, \tag{2.15}$$

where n is the index of a segment out of N segments. Using STFT, we can retrieve information about changes in the frequency components while preserving the time content.

The output of STFT is a spectrogram, which depicts the energy distribution of a signal across various frequency bands represented over time, where a bandwidth describes a frequency range the signal occupies. Here, energy corresponds to the squared magnitude of a signal, representing its intensity. Applying a Mel filterbank to this spectrogram yields a Mel-spectrogram. In this process, the Mel filterbank comprises triangular overlapping

energy filters that encapsulate specific frequency ranges, also called frequency bins. To approximate the human perception of loudness, the values of the Mel-spectrogram are typically converted to a logarithmic scale, commonly measured in decibels (dB). Figure 2.5 shows an example of a Mel-spectrogram. The Mel-spectrogram is considered a high-dimensional representation containing numerous frequency bins. A more compact representation that can be used to obtain feature information for speaker recognition tasks is the Mel-Frequency Cepstral Coefficients (MFCC). It is computed by applying the Discrete Cosine Transform (DCT) to the logarithmically scaled Mel-spectrogram. MFCC can effectively capture tonal, pitch, and phonetic information while filtering out irrelevant noise. The Discrete Cosine Transform is applied in order to accentuate speaker-identification features. It works similarly to FT, though the output is continuous, resulting in compact feature vectors. Additionally, it performs compression for more efficient data processing and decorrelates the features.

Depending on the environment in which the audio was recorded, an audio signal can be inconsistent over its duration. It can be disturbed by noise, variations in energy levels, or echo. To resolve these issues, an audio signal is analysed in terms of its noise level. If needed, noise reduction techniques or normalisation can be performed, e.g., energy normalisation that converts energy values to a standard scale to ensure consistent loudness over the whole audio signal.

2.2 Intellectual Property Protection

Intellectual property protection (IPP) techniques along the machine learning pipeline address different forms of digital objects, including input data, machine learning models, and their outputs, e.g., the generated content in the case of generative AI models [LMR24]. Approaches proposed in the literature for IP protection of machine learning models can broadly be categorised into *proactive* and *reactive* methods. Proactive strategies aim to prevent unauthorised access to the models, whereas reactive strategies focus on tracing and detecting misuse of the deployed model and proving its ownership, which encompasses model watermarking and model fingerprinting.

Within this broader context, watermarking is one of the most prominent reactive approaches. In general, watermarking refers to embedding an imperceptible identifier into a digital object in order to later prove its ownership. This has been applied in domains such as image watermarking, where ownership marks are embedded into pictures, or data watermarking, where unique patterns are inserted into datasets. Albeit the specific techniques differ across domains, they rely on similar mechanisms of hidden information embedding. Building on these foundations, watermarking has been extended to the protection of machine learning models.

2.2.1 ML Model Watermarking

Intellectual property protection through model watermarking can be realised under different assumptions regarding the level of access to the model. There are distinguished two approaches: *white-box* and *black-box* settings. In the white-box case, the verifier has complete access to the internal parameters and architecture of the model. The watermark can be embedded by modifying weights or by inserting additional structures into the model, and its presence can later be verified through direct inspection of the internal elements of the model.

Alternatively, black-box watermarking relies on the model's input-output behaviour. In this case, the watermark is embedded such that the model memorises specific input–output pairs (often called trigger sets). Verification of the ownership is achieved by querying the deployed model with the triggers and observing whether the returned response corresponds to the designated trigger output.

Certain requirements for the model watermarking technique should be met in order for it to be successful. In the audio domain, it includes [SNKR25, LMR24]:

- Effectiveness - the ownership of the model can be proven anytime.
- Fidelity - adding a watermark should not decrease the performance of the model on its original task.
- Robustness - resilience against modifications that aim to remove a watermark from the model.
- Imperceptibility - the added watermark should result in changes in the original audio sample that are as indistinguishable as possible [HHS+16]. Being indistinguishable comprises two aspects: inaudibility for the human ear and being non-distinctive for a machine.
- Generality - the watermark can be embedded in a different model architecture on a different dataset, ensuring broad applicability beyond the original setup [Boe21].
- Legality - a third party can not create watermarked samples that will trigger the model to identify them as legitimate watermarks if the said model has already been watermarked by the owner [ABPK18, SAMA21].

Backdoor Attacks

The concept of black-box watermarking stems from backdoor attacks on machine learning models. Backdoor attacks aim to manipulate a model so that it behaves normally on standard inputs but produces predefined, abnormal outputs when presented with specific trigger inputs. A common approach to achieving such behaviour is through data poisoning, when a model is trained on a poisoned dataset. However, backdoors can also be introduced through other mechanisms, e.g., by modifying the training procedure or

altering the model architecture. Poisoned data are samples that contain slight, usually imperceptible, alterations or additional patterns, which can be completely unrelated to the original data distribution. When the model is trained on such data alongside the original task, it memorises the injected patterns introduced by an attacker and, subsequently, produces abnormal outputs when presented with the corresponding trigger inputs [GLDGG19]. The output label of the triggers, in this case, can either correspond to a single specific class from the original set of classes or each trigger is assigned to a different class within the same set, distinct from the original ground-truth label of the altered sample.

Black-box ML Model Watermarking

One of the first to use a backdoor in neural networks for protection was Zhang et al. [ZGJ+18], who introduced the concept of black-box watermarking of DNN models. The method was designed to be applicable across different tasks and domains, though it was specifically tested in the image domain.

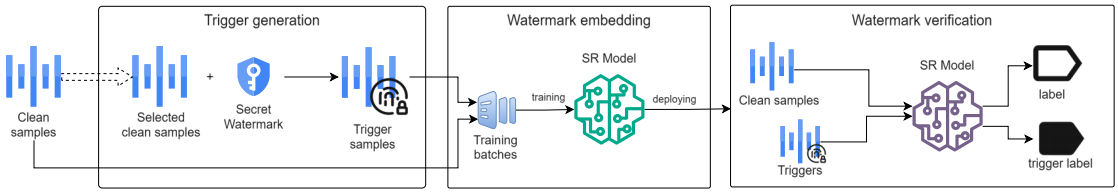


Figure 2.6: Black-box model watermarking system overview.

Black-box watermarking typically involves three key stages as depicted in Figure 2.6: trigger generation (designing the special inputs), watermark embedding (training the model to recognise them), and watermark verification (confirming watermark presence via queries).

The trigger set can be created using various strategies. In general, a subset of original samples from different classes is selected and modified to serve as triggers. The in-distribution triggers are derived from samples that follow the same distribution as the original dataset. Common modification strategies include noise addition, overlaying a secret pattern or key, or slight mislabelling, while ensuring that the resulting trigger still resembles the original data. The out-of-distribution triggers are generated using completely unrelated audio samples. They can be added directly to the dataset or applied on top of existing samples.

In the black-box setting of model watermarking, the goal is for the model to memorise the trigger set alongside the original task, so that the triggers produce the intended outputs while preserving the fidelity of the model.

Furthermore, trigger sets can be created either by modifying the raw input data (e.g., audio waveform) or by altering derived representations (e.g., spectrograms or MFCC). The choice of representation has a direct impact on the verification process: modifying

the input allows straightforward querying of the deployed model, whereas altering derived representations may require access to intermediate layers or feature extractors.

In the context of classification models, in addition to the input pattern of trigger pairs that can be created by various schemes, the method of choosing an output label also varies. Zhong et al. [ZZZ+20] proposed using a completely new class to label trigger samples, making the classification task a $C + 1$ problem. Their results showed that this approach disturbs the classification boundaries of the model less than using an existing one. However, in return, this exhibits a few drawbacks: the method involves adjusting the output layer of the model, which may require modifying the training hyperparameters and can make convergence more challenging [HT23]; the presence of the watermark in the model is more easily detected, namely, if a malicious party knows the originally intended number of classes, or if they also find out the trigger class, the watermark verification process can be prevented [SLH+23].

To address concerns related to the legality of watermark verification, in the event that a malicious party acquires a secret watermark sequence and generation scheme, Zhang et al. [ZJW+20] proposed assigning the triggers to an arbitrary class. The most common approach in the literature so far is to assign an existing class to a newly generated trigger sample [HTXJ24].

Watermark Compromise Techniques

With the discovery of backdoor attacks, including those introduced through data poisoning, numerous defence approaches were also invented. These defence strategies may be applied as attacks against the watermarked machine learning models. These attacks aim to undermine the effectiveness of the watermark, thereby potentially preventing verification of model ownership.

Following the taxonomy proposed by Boenisch [Boe21], we distinguish four main types of watermark compromise techniques that differ in their goals:

- watermark detection - an attacker aims to detect the presence of a watermark in the model;
- watermark overwriting - an attacker embeds another watermark that does not necessarily erase the original one;
- watermark suppression - the goal of the method is to prevent a successful verification of the model's owner;
- and ultimately watermark removal that completely erases the watermark from the model.

Even partial attacks, such as detection or overwriting, can weaken ownership claims and increase the risk of misuse. Thus, it highlights the need for a watermark to be resilient against a broad spectrum of potential threats.

2.2.2 Protection of ML Models in Audio Domain

Even though the primary focus of this thesis is the protection of speaker recognition models, this section discusses approaches introduced for other types of audio-based models. As a main comparison, we first present protection techniques for speech recognition models, and additionally discuss a black-box watermarking approach for audio generation using diffusion models.

Protection of Speech Recognition models

The idea of intellectual property protection of speech recognition models started with a white-box setting in [CDK20]. The authors embedded the watermark in the significant frequency spectrum components of the model using the Discrete Cosine Transform (DCT). Specifically, they select the model weight parameters for watermark embedding and apply the DCT to them. A predetermined number of the largest-magnitude DCT coefficients are then chosen for embedding, since these large coefficients are less sensitive to small modifications and thus preserve model performance.

Jia et al. [JCCCP21] later applied their black-box watermarking scheme to a variety of media, including an audio classification task, namely an RNN model with the Google Speech Commands Dataset [War18] that predicts a command category. They embedded a watermark either in an audio signal overwriting a 1/8 of the signal's length or in a mel-spectrogram representation changing two 10x10-pixel squares. They showed that both approaches are valid options, meaning the model could successfully memorise the triggers and allow ownership verification without significant degradation of the original classification performance. In the case of audio media, however, the choice of representation in which the watermark is embedded affects the verification process, unless a representation can be reliably converted back to a raw audio format. The watermark was incorporated into the model by training it from scratch on clean and trigger samples. Later, Rathi et al. [RSR22] proposed a black-box watermarking scheme for speech recognition tasks using only 10 trigger samples. They built their trigger samples based on work by Carlini and Wagner [CW18] on creating audio adversary examples for automatic speech recognition by adding a slight perturbation, which completely changes the predicted output label.

Chen et al. [CZL+22] introduced another black-box watermark solely for automatic speech recognition (ASR) models. Their approach lies in fine-tuning an existing ASR model on the mix of their created triggers and clean samples. They argue that since the models often predict the outputs frame-wise, in order to ensure successful watermark verification, a watermark should be embedded in each frame. Hence, the authors add a recurring speech clip of the model owner to a clean audio signal to create a watermarked sample. Moreover, they state that the output label should not be a trivial copyright information (unlike [RSR22], who used the phrase of type '*This is a property of* $<$ *author's_name* $>$') since it is vulnerable to evasion attacks based on label detection. The output labels for the trigger set are thus generated linguistic stenography texts [YZH+21].

Protection of Audio Generation Diffusion models

Cao et al. [CLJ+23] presented an application of the black-box watermarking technique to audio generation diffusion models. In this context, the audio diffusion model operates with mel-spectrograms of the given audio signals. The main purpose of a diffusion model is to generate artificial (also referred to as synthetic) audio data. During training, random Gaussian noise is gradually added to an audio signal until it carries only noise, and then the model learns to reverse it back to something that sounds like a real audio signal. The authors embedded their watermark by fine-tuning the diffusion model on a small set of data with 50% triggers, as their experiments in embedding the watermark while training the model from scratch were not successful. They stated it was highly likely due to the high computational demands caused by the large input format size. They experimented with in-distribution and out-of-distribution watermarking schemes, highlighting better watermarking effectiveness and higher model generation quality using an in-distribution approach. The trigger set size was set as 50% of the fine-tuning set. Overall, their two watermarking approaches produced watermarks that were imperceptible to listeners while ensuring successful verification.

CHAPTER 3

Threat Model

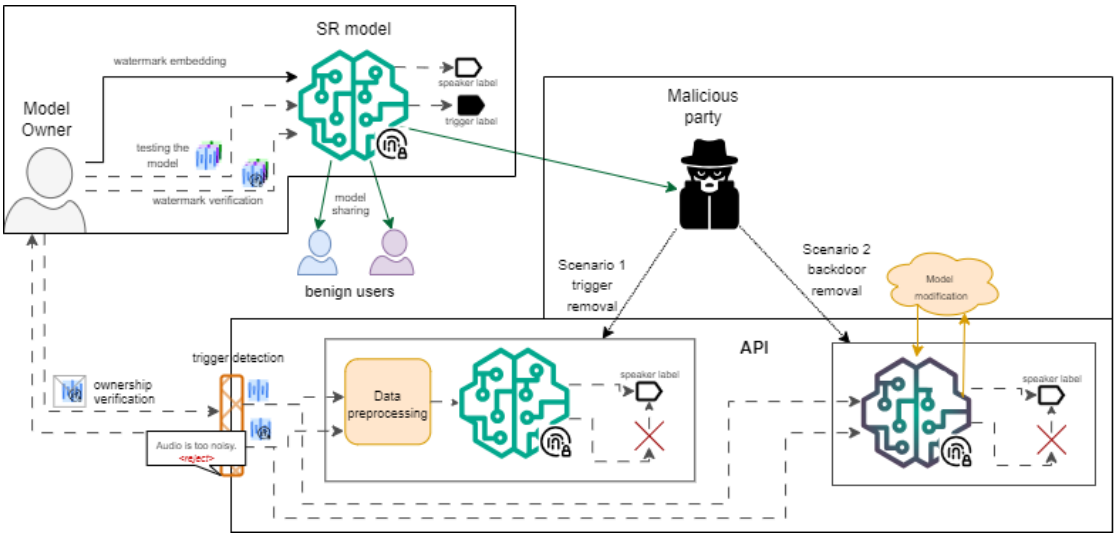


Figure 3.1: Overview of the considered threat model. The model owner trains a speaker recognition model with an embedded watermark and distributes it under controlled conditions. A malicious party may obtain a copy, modify it, and deploy it as an API service with additional processing steps. The model owner can only interact with the stolen model via its API interface.

The threat model describes the setting and assumptions under which the attacker (or malicious party) and defender (or model owner) operate. In this chapter, we describe the threat model for this work, comprising the goals, knowledge, and capabilities of the parties. The overall scheme of interactions and capabilities of the two parties in our considered threat model is depicted in Figure 3.1.

This setting is based on the taxonomy defined in [LMR24]. The model owner is the actor who invested time and resources in training a speaker recognition machine learning model. After that, the model may be deployed for use by external parties under controlled conditions. For instance, the model could be distributed under a specific license that grants the user certain rights to run or access the model, or it could be provided as part of a Machine Learning as a Service (MLaaS), where the model remains hosted in the cloud and users interact with it via an API. In the first case, the user typically receives a legal copy of the model and can execute it locally.

A *legal copy* refers to the instance of the model obtained in a legitimate way: e.g., downloading it from an official source like GitHub, HuggingFace, or AWS. The usage of such a copy is allowed provided that it complies with the license with which it was released.

An *illegal copy* of a model can be created when access to a model was originally provided as a pay-per-query type of service and an attacker stole a model by means of e.g., a model stealing attack [TZJ+16]. However, this scenario will not be considered in our thesis, as watermarking-based protection is not applicable to stolen models that have never been watermarked.

In the following, we describe the roles of the model owner, who seeks to protect their intellectual property, and the attacker, who may attempt to compromise it.

3.1 Attacker

In this threat model, the attacker seeks to compromise the intellectual property of a machine learning model for financial gain by circumventing the employed protective measures.

Incentive: The motivation of a malicious party comes from the possible profits they can make using a stolen model without an investment of resources to train it.

Goal: The attacker aims to utilise a stolen model to obtain monetary gain, e.g., by deploying a stolen model as a paid API service. In doing so, they may attempt to invalidate or ultimately remove any possible embedded watermark to prevent ownership verification through API queries. For this, there are two main factors to take into account:

- Model effectiveness: Regardless of the types of modifications or additional steps an attacker incorporates in their deployment pipeline, the accuracy of model predictions should not deteriorate for the original task. Otherwise, their service may not be used by third parties due to low-quality performance.
- Watermark verification failure: Modifications of the model should invalidate watermark verification. However, this aspect can not be empirically tested by a malicious party due to the lack of access to the training data and trigger set.

Knowledge: An attacker may know or suspect that the stolen model is watermarked. However, they do not know the predefined output label of the triggers. They also do not have access to the watermark trigger set or the original training set to find out more about the watermark embedding process. Hence, there is no certainty about what parts (e.g., frequency bands) of the audio signals are affected by the watermark since they can not compare an original and a watermarked signal.

Capabilities: The first prerequisite of this scenario is that an attacker can obtain a legal copy of the model. Subsequently, they are able to alter the model to their needs before deploying it as a service via a black-box API. These modifications may include fine-tuning the model on additional domain data, adjusting the output layer, or pruning parameters. A malicious party can additionally wrap the execution of the API requests by preprocessing the incoming audio signals or rejecting them to avoid the watermark detection.

3.2 Defender

The defender is the model owner who desires to protect their intellectual property and ensure that their trained model is only used under authorised conditions.

Incentive: The defender's primary motivation is to prevent financial losses that could result from unauthorised copying or redistribution of their trained speaker recognition machine learning model, to the creation of which they devoted their resources. Furthermore, misuse of the model by malicious parties could cause damage to the defender's reputation, especially if the stolen model is deployed with reduced performance, embedded malicious behaviour, or without compliance with licensing terms.

Goal: The model owner aims to achieve multiple goals:

- Watermark effectiveness: Embed a watermark in the model to be able to prove their ownership by querying trigger samples to the API service with a stolen model.
- Fidelity: Achieve a high performance of the watermarked model on the original task.
- Robust watermark: Embed such a watermark so that it is not possible to remove it without suffering a substantial performance drop on the original task, beyond a threshold that would render the model commercially unattractive or impractical to use.
- Imperceptibility: Design a watermark in a way that it is difficult, ideally impossible, for an attacker to detect or distinguish from normal model behaviour.

Knowledge: The model owner has complete knowledge of their model M with the training and trigger datasets used. They also know the schemes with parameters for

creating and embedding the watermark. Additionally, they do not require detailed knowledge of the attacker's identity or strategy but assume that an attacker may obtain a legal copy of the model and attempt to compromise or remove the watermark.

Capabilities: The defender can query a stolen model through API requests and collect the outputs to prove their ownership of the deployed model.

CHAPTER 4

State of the Art

This chapter reviews existing approaches for speaker recognition model protection through black-box watermarking. It begins with an overview of audio watermarking and specific requirements that apply to watermarking methods in audio domain. We then analyse existing work on protecting speaker recognition models in black-box settings, offering a detailed evaluation of the literature and highlighting key research gaps. Finally, we discuss potential techniques that a malicious party might use to remove a watermark.

4.1 Audio Watermarking

Numerous techniques have been developed to watermark audio files. They have been studied in multiple surveys [HHS+16, NT07]. The principle of such watermarks is to embed an extra pattern, inaudible to the human ear, or a random sequence that can be later verified but is hard to detect otherwise. It is worth noting that these techniques have been developed to protect the ownership of the audio data itself rather than a model. Whether or not they can fulfill all the requirements to be applied for model protection is an open research question [Boe21]. Below, we describe frequently considered approaches of audio watermarking.

Time-spread-based. The watermark is added to the signal and distributed over a certain period of time in the audio. A prominent approach is echo hiding [BGML96, GLB96], where the embedding of a repetitive echo may seem like a shortcoming of the recorded audio; even if the modification remains audible, it would be hard to distinguish whether it is placed there deliberately as a watermark. A number of surveys about various echo hiding methods [TA17] and other time-spread watermarking schemes [SNKR25] provide a more detailed overview of available techniques.

Transformation-based techniques. This category applies a watermark to a representation of an audio sample in the frequency domain, obtained by some type of

transformation, e.g., the Discrete Fourier Transform (DFT) or the Discrete Cosine Transform (DCT) [UOHS24]. After embedding, this representation is converted back to the audio waveform, which thus becomes a watermarked sample. One of the first mentions of the DCT application for watermarking in the literature was presented in [SKT98]. Today, a wide range of transform-based watermarks is available, including approaches that apply multiple transformation schemes sequentially.

Spread spectrum based. These watermarks are embedded across a wide spectrum, which masks them to human perception [KM03]. Originally, the method was proposed for images [CKLS97]. For the audio domain, a significant improvement in this watermarking group was by Malvar and Florencio [MF03], who increased imperceptibility by projecting a watermark onto the signal rather than directly inserting it with complete disregard for the original audio. Since then, these techniques have been constantly improved in terms of imperceptibility and robustness, two key aspects of watermarks that we will discuss in more detail below.

4.1.1 Robustness of Audio Watermarks

Audio watermarks must be effective and robust, namely being able to withstand certain transformations [Arn00, SNKR25]. Common robustness checks include the following:

Noise addition: A certain amount of noise is added to the watermarked audio sample to overwrite the watermark. Typically, white Gaussian noise is chosen for this, which refers to random noise drawn from a Gaussian distribution with equal intensity across all frequencies, producing a constant background hiss [UOHS24]. Some works [EHPM23] also consider coloured noise, where the energy is distributed unevenly across the spectrum. For example, pink noise has more energy in lower frequencies, while blue noise emphasises higher frequencies. Such frequency shaping can make the attack less perceptible or more effective against certain watermarking methods.

Bandpass filtering applies a filter that masks or erases a part of the frequency range of an audio signal. If the watermark is concentrated only at the considered frequencies, it will no longer be detectable.

Compression: Applying lossy compression techniques to the audio, such as downsampling the audio to a lower sample rate, then converting it back to its original one, or using lossy codecs (e.g., MP3).

Cropping: Cutting part of a signal. It checks whether the watermark is well distributed throughout the whole length of an audio to be recognised, regardless of the length of the input sample. For pattern-based watermarks, the embedded pattern should be short and memorable enough so that at least one occurrence remains after cropping, allowing successful verification.

Pitch shifting: Altering the frequency content of a signal by changing the semitones of the pitch. It may disturb the watermarks that rely heavily on certain embedded frequency patterns.

Time stretching modifies the length of an audio signal, which leads to a custom tempo.

4.1.2 Imperceptibility of Audio Watermarks

Another crucial aspect of the audio watermarks is their imperceptibility. Ideally, the watermark should be inaudible, or at least unsuspicious, to a human listener, and indistinguishable for a machine [SNKR25]. One of the approaches used for assessing perceptibility is conducting a survey with human listeners. As this is out of scope for this thesis, we focus instead on quantitative measures, which evaluate perceptibility using numerical metrics, such as signal-to-noise ratio and log spectral distortion.

The **Signal-to-Noise Ratio** (SNR)[XHY17] of a watermark is computed by comparing the average powers of the original signal and the embedded watermark, and is computed as

$$SNR_{WM} = 10 \log_{10} \left(\frac{P_{orig}}{P_{WM}} \right), \tag{4.1}$$

where P_{orig} is the average power of the original signal, and P_{WM} is the average power of the difference between watermarked and original samples. The unit of measurement for SNR is decibels (dB). The higher the SNR, the less perceptible the watermark is. The metric represents the overall distortion of the signal based on the energy.

According to the International Federation of the Phonographic Industry (IFPI) [KP16], the Signal-to-Noise Ratio of the watermark should be greater than 20 dB for it to be not disturbing. Building upon this, Hsu et al. [HTY+20] suggested that an SNR greater than 30 dB is preferable for ensuring imperceptibility. These rules has been widely applied in many research studies on audio watermarking [Arn00, HGT15, YDK+23].

The **Log Spectral Distortion** (LSD) measures the difference in log frequency spectrum between the original and watermarked signals [GBGM80]. The LSD is computed as a logarithm of the difference in spectral magnitudes, as follows

$$LSD_{WM} = \frac{1}{N} \sum_{n=1}^{N} \sqrt{\frac{1}{K} \sum_{k=1}^{K} \left[10 \log_{10} \left(\frac{M_{orig}(n,k)^2}{M_{trigger}(n,k)^2} \right) \right]^2}, \tag{4.2}$$

where N is a number of frames, K is a number of frequency bins, M_{signal} is the magnitude of the given signal at frame n and bin k.

The LSD is also measured in dB. Values in the range between 0 and 1 dB represent the most imperceptible watermarks, 1-2 dB a slight but noticeable difference, and a value bigger than 2 dB is a noticeable distortion [GM76, NJC+08].

4.2 Speaker Recognition Model Watermarks

The field of model watermarking primarily originated in the image domain, and thus, the majority of black-box watermarking methods presented so far were designed for

images. Albeit there are works that claim their approaches can be generally applied to other media formats, not many of them explore the speaker recognition models in detail; also, works often do not consider the audio as input, but rather its representation in image format, i.e. treat the whole process as image watermarking. Thus, the speaker recognition model protection is still a rather underexplored area. The methods that are discussed in this section took their ideas from the image domain and adjusted them to fit the SR task for audio files. For a better comparison, the notations were unified to represent the same entities in all the papers.

4.2.1 Segment-based frequency perturbation (Wang and Wu)

Wang and Wu [WW22] propose a black-box watermarking scheme designed specifically for SR models. They add an extra output class (i.e., an additional speaker) to the model by altering the softmax layer, turning the initial C-class task into a $(C+1)$-class problem. All trigger samples are assigned to the new label, and the model is trained from scratch to classify both clean and trigger inputs. Two separate trigger sets are constructed: one used for training, and one for the verification process. Hence, the model learns a trigger *pattern* rather than the specific trigger samples.

Trigger generation: The triggers are first selected at random from the training dataset of audio signals. For each selected signal, a watermark pattern is then embedded in the frequency domain using a discrete cosine transform (DCT). Firstly, a random sequence t of length l is drawn from $\{-1, 0, 1\}$. Then, the audio signal in which the watermark will be embedded is divided into frames, each of length l. Each frame undergoes the DCT, which results in the frequency coefficients of the audio signal. The frame modification can be formalised as follows:

$$X'_n = X_n + \lambda * t, \tag{4.3}$$

where X_n is a frequency representation of an audio fragment, and λ is a hyperparameter that regulates the intensity of the watermark. After embedding the watermark in each fragment of an audio sample, the authors perform an inverse DCT (IDCT) to obtain the watermarked audio sample. They argue that embedding it in each segment helps the model to memorise the pattern in the watermarked audio samples, and since prediction can be done frame-wise, it is crucial that each fragment carries the watermark.

Experiment Setup: The method was evaluated on the TIMIT [GLF+93] dataset using the SincNet [RB18] model [1]. First, the authors follow the split of TIMIT used by Ravanelli [Rav25] for SincNet model: 80% of the data is used for training, and 20% for testing; then, the authors further set aside 10% of the training set for validation. Randomly, 16 speakers are drawn from the 462 available ones, from which 36 trigger samples are created. However, the procedure for deriving 36 samples from the 16 speaker classes is not specified. These 36 trigger audio files are then also divided into three groups: 16 trigger samples are put in the training set, among which 14 are used for training and

[1] The SincNet model and the TIMIT dataset are described in detail in Sections 5.1.1 and 5.2.1, respectively.

2 for validation; the other 20 triggers are used for testing the success rate of watermark verification. The authors state that a validation set is used to choose the best-performing model; however, the specific evaluation criteria are not given.

Evaluated Properties: The authors assessed three key properties: watermark **effectiveness**, measured by the success rate on trigger inputs (WMSR); model **fidelity**, evaluated in terms of the sentence error rate (SER) on the original speaker recognition task. Watermark **robustness** is evaluated by introducing Gaussian noise of various intensities to model inputs and observing the resulting change in WMSR. However, the impact of these attacks on model fidelity is not assessed, leaving the robustness results without proper contextualisation.

4.2.2 Mel mid-frequency band based (Zhang et al.)

Zhang et al. [ZDX+23] propose a method for speaker recognition models that embeds the watermark into the Mel-spectrogram representation of audio inputs. Similar to the approach in [WW22], all trigger samples are assigned to a newly added output class, effectively transforming the task into a $C + 1$-class problem. The model is trained to predict the original and trigger samples from scratch. However, these watermarked samples are not used for the watermark verification stage. As in the approach by Wang and Wu discussed above, the authors create a separate set of triggers for verification, unseen before, where the original audio samples are taken from a test set.

Trigger generation: The watermark pattern is embedded using the least significant bit algorithm (LSB) [vSTO94]. A binary image, referred to as the Key, of size $l \times u$ is first compressed into a binary sequence t of length l. The selected audio signals are divided into frames using a Hann window [Mü15] and processed via Short-Time Fourier Transform to obtain magnitude spectrograms. These are then converted to Mel-spectrograms using a Mel filterbank [DM80]. From each Mel-spectrogram, a matrix A, matching the dimensions of the Key, is randomly selected from a mid-frequency band.

The authors argue that selecting such a matrix from the mid-frequency band with single-pixel values lower than 6.3778 provides an optimal balance between the imperceptibility of the watermark and its robustness. However, the paper does not clarify the unit of this value or explain how it is guaranteed that the pixel values remain below the stated threshold for a selected matrix. The matrix A is then transformed using the Discrete Cosine Transform (DCT), and the watermark sequence t is embedded by modifying the first coefficient of each row:

$$X_1(k)' = X_1(k) + \lambda \times t(k), \tag{4.4}$$

where $k = 1, 2, .., l$ is a row of the matrix A and λ controls the strength of the watermark. Subsequently, the new matrix A' that carries the watermark is converted back to the Mel-spectrogram representation using an inverse DCT. The new magnitude spectrogram is reconstructed. Finally, with the help of `griffin_lim_vocoder` [SKM+20], they recover a waveform of a watermarked audio sample. The goal of the

`griffin_lim_vocoder` is to guess the correct phase of an audio to perform an inverse STFT.

Experiment Setup: The method was evaluated on the TIMIT and LibriSpeech datasets, using two models: SincNet and a CNN with the same architecture as SincNet, except that the *sinc* convolution layer was replaced with a standard convolution layer. The training trigger set for TIMIT was created based on 150 samples randomly selected from the 2,310 samples in the TIMIT training set. The verification triggers are created using an unspecified number of test samples drawn from the TIMIT test set. For evaluation on the LibriSpeech dataset, the authors created a subset of the original full dataset: they randomly retrieved one sample with a length of 12 to 15 seconds from each of 2,484 different speakers for training and 2-6 seconds for testing. For LibriSpeech, 1,500 triggers were created.

Evaluated Properties: The evaluation of watermark **effectiveness** and model **fidelity** follows the procedure of [WW22], using the success rate on trigger inputs (WMSR) and SER on the original as respective measures. Watermark **robustness** is evaluated based on three attack methods: model fine-tuning, model pruning, and watermark overwriting (in the paper called an ambiguity attack that aims to embed another watermark in the stolen model), measured by the resulting change in watermark verification success. **Imperceptibility** is evaluated using the SSIM (Structural Similarity Index Measure) [WBSS04], which measures the perceptual similarity of Mel-spectrograms; and the Cosine Similarity between the original and recovered watermarked waveforms.

4.2.3 Gaussian Noise, Frequency Noise and Unrelated Audio Models (Liao et al.)

Liao et al. [LYS+24] presented three more watermarking methods. Two of them, *Gaussian noise watermark* and *Frequency noise watermark*, add a watermark pattern based on Gaussian noise, while the *Unrelated audio watermark* uses an audio signal unrelated to the original task and dataset. In contrast to Wang and Wu [WW22] and Zhang et al. [ZDX+23], the authors deliberately avoid adding an extra output class, arguing that it could reveal the presence of a watermark if the model exposes an unexpected label during inference. Instead, trigger samples are assigned to an existing class, which is chosen randomly; for their experiments, they used the label *"123"*. They propose to either train the model from scratch or fine-tune it on a mix of original and trigger data. In their experiments, it was shown that the methods are stable enough to generalise at approximately 175 epochs compared to the entire 360 epochs in Wang and Wu, and Zhang et al. approaches. Furthermore, they compared the effectiveness of the watermarks based on the different number of trigger samples, which were calculated by a *trigger set / training set* ratio.

Trigger generation: The *Gaussian noise watermark* involves adding Gaussian noise with a specified Signal-to-Noise Ratio (SNR) to the entire duration of a clean audio sample. The level of desired SNR is a hyperparameter used in the watermark creation

process. The Gaussian noise is generated randomly, and has a certain SNR compared to the original sample, which is controlled by a scaling factor λ, calculated as follows:

$$\lambda = \sqrt{\frac{P_{orig}}{10^{SNR/10} \times P_{noise}}}, \tag{4.5}$$

Then, this scaled noise is applied to the original audio, which results in the watermarked audio sample. Note that Equation (4.5) can be derived from the Equation 4.1.

To reduce perceptibility, the authors propose the *Frequency noise watermark*, a variant that embeds Gaussian noise only in the extreme-low and -high frequency bands of the signal. Specifically, the frequency-limited noise is extracted from the full noise sequence and scaled before embedding. Additionally, Mel-frequency cepstral coefficients (MFCCs) are computed and amplified to preserve speech-relevant features. After these transformations, the modified audio is reconstructed into a waveform.

The third watermarking method presented by Liao et al. is the *Unrelated audio watermark*, where they use an audio unrelated to the task domain as a watermark pattern. The authors conducted their experiments with the "air conditioner sound" as a watermark, but they argue that any sound, especially a common one, will be less suspicious than any type of noise and can be used as a watermark.

Experiment Setup: The experiments were conducted on the TIMIT dataset for the SincNet model. The authors determined the number of triggers with a ratio parameter α = *trigger set / training set*. Most of the experiments utilised the value $\alpha = 1/8$; beyond that, the authors also investigated $1/4, 1/16, 1/32$ as values, showing a decrease in effectiveness with the reduction of α. To the best of our knowledge, the watermark verification is evaluated on the triggers used in the embedding (which is in contrast to the approaches of Wang and Wu, and. Zhang et al., who used a specific verification trigger set).

Evaluated properties: Consistent with [WW22] and [ZDX+23], watermark **effectiveness** is evaluated via watermark success rate (WMSR), while model **fidelity** is assessed using the original task performance (SER). The watermark **robustness** was evaluated by adding Gaussian noise to the input audio files and measuring the watermark success rate, following the methodology in [WW22].

4.3 Limitations of Existing Works

The works for speaker recognition model protection discussed in this chapter are based on rather different assumptions and designs, and use varying methodologies for evaluation. We summarise the most relevant aspects we identified in Table 4.1. While we will further detail relevant shortcomings that impede reproducibility and replicability of the methods in Section 5.3, we give an overview on the most relevant obstacles below.

- **Code availability:** None of the identified works has published their code. Therefore, reproducibility is not possible, and the replicability of their results highly

Table 4.1: Comparative overview of the Speaker Recognition Model Watermarking methods and their limitations.

		Wang and Wu [WW22]	Zhang et al. [ZDX+23]	Liao et al. [LYS+24]
Reproducibility	Datasets	TIMIT	TIMIT; LibriSpeech (subset)	TIMIT
	Models	SincNet	SincNet; SincNet-based CNN	SincNet
	Source code	✗	✗	✗
	Pseudocode	✗	✓	✓(inconsistent)
	WM parameters	? length of sequence t	? *Key* size ? limits of mid-frequency band	✓
	Experiments parameters	✓	? number of triggers	✓
WM	Input format	raw waveform	raw waveform	raw waveform
	Verfication type	pattern (unseeen triggers)	pattern (unseeen triggers)	sample (train triggers)
	Trigger label type	extra label	extra label	existing label
Evaluation	Effectivenes	✓: WMSR	✓: WMSR	✓: WMSR
	Fidelity	✓: SER	✓: SER	✓: SER
	Imperceptibility	✗	✓: SSIM, Cosine Similarity	✗
	Robustness:			
	Model modification	✗	✓Fine-tuning, Pruning: WMSR	✗
	Data preprocessing	✓Gaussian noise: WMSR	✗	✓Gaussian noise: WMSR
	WM overwriting	✗	✓	✗

depends on the detail of the algorithms of trigger generation and embedding, and the specific parameters they used.

- **Code description:** The pseudo-code algorithms presented by Liao et al. [LYS+24], specifically for the *Gaussian noise* and *Extreme frequency* trigger creation, differ from their textual description in their paper that mentions the utilisation of framing, DCT, and MFCC. Due to the lack of published source code, one has to make an assumption about which description is correct.
- **Dataset reproducibility:** Zhang et al. [ZDX+23] used a subset of LibriSpeech without mentioning the exact steps of creating or preprocessing it, or publishing a list of samples belonging to the various subsets. As the subsampling is likely a random process that can also not be estimated, this severely impedes reproducibility.
- **Parametrisation:** Neither [WW22] nor [ZDX+23] provides the parameters for the trigger generation stage. Liao et al. [LYS+24] do not specify which unrelated audio sample of the "air conditioner" class they chose for their evaluation; this affects the reproducibility results. Further, the number of bins used to get MFCC coefficients in Liao et al. [LYS+24] is not specified.

We also identified what we consider flaws in the methods' design or their evaluation by the original author. We discuss these aspects below.

- **Potential Watermark Forgery:** Both [WW22, ZDX+23] evaluate watermark verification using a newly generated "test" trigger set composed of trigger samples unseen during training. This implies that the watermarked model is trained to learn

and generalise the underlying watermark pattern rather than memorising specific triggers, which is an uncommon practice in black-box watermarking [HTXJ24]. While this approach demonstrates pattern-level embedding, it also introduces a potential vulnerability where an adversary might discover a broader set of suitable patterns not explicitly embedded during training and generate new verifiable triggers, thereby increasing the attack surface for ambiguity or overwrite attacks.

- **Potential Watermark Detectability:** Both [WW22] and [ZDX+23] assign an extra label to the triggers, which can potentially expose the presence of the watermark in a model.
- **Potentially Suboptimal Hyperparameter Settings:** The papers do not provide any justification for the choice of the number of trigger samples in the trigger set. Only Liao et al. [LYS+24] experiment with multiple options and present a comparative analysis.
- **Potentially Semantically Weak Imperceptibility Evaluation:** The metrics used to evaluate imperceptibility in [ZDX+23] do not capture audio-specific perceptual characteristics. SSIM [Set21] is a widely used metric for measuring imperceptibility in images, and Cosine Similarity measures the directional similarity of two vectors. As a result, their effectiveness in quantifying the perceptual differences between original and watermarked audio samples is limited and potentially misleading.
- **Potentially Wrong Imperceptibility or Weak Imperceptibility Evaluation:** For the imperceptibility analysis, Zhang et al. [ZDX+23] visualise original and watermarked samples. However, their amplitudes differ extremely in their values: while the amplitude of the original sample varies in the range of -1 and 1 (which is considered a normalised audio), the watermarked sample is in the range of -0.15 and 0.15 only. That indicates either a mistake in the visualisation or a robustness concern of having watermarked audio samples with such low amplitude.

Based on this overview, we outline the following shortcomings in existing literature:

- As shown in Table 4.1, some works provide a more detailed evaluation of the robustness, effectiveness, and fidelity of their proposed methods, while others present insufficient analysis, especially in terms of robustness under different possible attack scenarios.
- The applicability of the methods to other models with the same (raw waveform) or a different (e.g., mel-spectrogram) input structure has not yet been investigated.
- The only architecture that was watermarked is a specific CNN-based model – SincNet [RB18], even though numerous SR models, including ones that improve SincNet results, exist. While Zhang et al. [ZDX+23] also tested a SincNet-based

architecture with a different convolutional layer, no other state-of-the-art speaker recognition model architectures were examined.

- The experiments were conducted on datasets that contain cleanly recorded audio files of people reading sentences that do not consider any real-world disturbances (e.g., background noise from public environments).

We, therefore, observe several research gaps in the current literature regarding the applicability of these techniques to state-of-the-art Speaker Recognition model architectures, various input representations, and diverse datasets. Furthermore, there is an open research question of how the watermarking methods compare in terms of robustness and imperceptibility due to the lack of a unified analysis among the methods.

4.4 Watermark Removal Attacks

Among the potential threats identified in the literature on speaker recognition model protection, watermark suppression and removal attacks are of particular interest, as they directly compromise model protection mechanisms. Accordingly, this section discusses these attack types in detail. The authors in [LJLX24] identified backdoor defence paradigms, some of which can be used to perform these attacks. Furthermore, Boenisch [Boe21] presents more general watermark removal techniques. They include the following.

Data preprocessing attacks. Prior to feeding input data into the model during inference, an additional preprocessing step can be applied to eliminate the watermarked portion of the sample. This prevents the model from encountering a trigger input that would produce a manipulated output. This method does not involve altering the model itself. The attacker's objective is to effectively remove the trigger while preserving the model's original task performance.

The attacker may also deploy a detection mechanism that refuses to process certain inputs, for instance, those with too much noise, unrealistic data format, or abnormal content inside, etc. That means if a watermarked part of a trigger sample is too distinguishable, it may be suspicious enough to be blocked before being passed for inference on the stolen model.

Model modification attacks. This group aims to update the trained model in order to erase its ability to recognise the watermarked samples. Specifically, these approaches include [XZWL22]:

- Fine-tuning: adjusting the model using an additional subset of data, a technique that might be used to improve the performance of a model for a specific task, which in the process might damage a watermark.
- Pruning: cutting some parameters of the model (e.g., with the lowest value of weights, or based on activations on (assumed) clean input).

- Fine-Pruning: a two-step approach, which first prunes the model and then proceeds with training to restore performance of the model, in the process, removing a watermark.
- Retraining: regularly updating the model with new incoming data as it becomes available. This process resembles fine-tuning, but is continuous and its primary goal is not to improve performance on a specific task. Instead, it incrementally incorporates additional data, which may dilute or remove an embedded watermark over time.
- Transfer learning: adapting a pre-trained model to new data. In the process, the final layers of the model are usually modified, which may unintentionally weaken or remove an embedded watermark.

CHAPTER 5

Methods and Experiment Design

This chapter provides a detailed description of the experiments conducted in this thesis. It begins with an overview of the models used, followed by the datasets employed in the experiments. We then describe the creation of trigger sets for each watermarking method, including any modifications made. Details of the watermark embedding process are also provided, specifically, how the models were trained using the trigger sets. The chapter concludes with an explanation of how each aspect of the watermarks is evaluated, specifically effectiveness, imperceptibility, legality, and robustness.

5.1 Speaker Recognition Models

Existing state-of-the-art speaker recognition models vary in their architectures and input structures. The following part describes in detail the state-of-the-art SR ML models examined in this thesis. Overall, three representative speaker recognition models were selected. SincNet serves as the baseline, as it is widely used in prior work on which this thesis builds. AM-MobileNet was selected as a lightweight improvement over SincNet, explicitly designed for deployment on portable devices. Finally, AutoSpeech was included because it utilises a different input representation (i.e., spectrograms) and has been shown to outperform other (i.e., ResNet-based) approaches.

5.1.1 SincNet

SincNet [RB18] is a widely used speaker recognition model with a CNN architecture. The core idea incorporated in the model is parametrised sinc functions in the first convolutional layer that include band-pass filters designed to emphasise low and high frequency components of the audio, which are believed to carry more speaker-relevant information. Such an approach greatly decreases the number of parameters compared to the standard CNN architecture. The raw waveform is split into overlapping segments

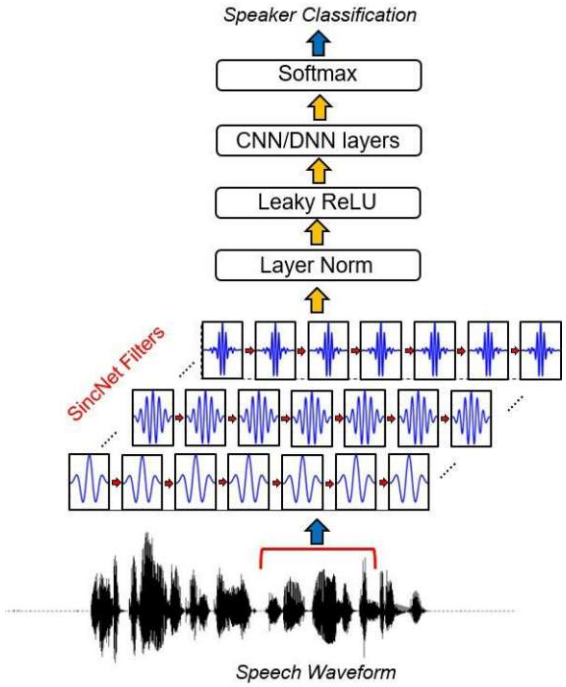


Figure 5.1: SincNet architecture [RB18].

using the Hamming window function, which are then input to the network. However, the authors mention no drastic changes in performance using other window functions. Their empirical results indicate that low and high frequencies carry more significant information regarding a speaker's identity, which is reflected in the learned filters. Hence, the first convolutional layer consists of a filterbank, with each filter focusing on a different frequency band to capture discriminative features. The equation to obtain the output of these filters is the following:

$$y[n] = x[n] * g[n, \theta] \tag{5.1}$$

where $x[n]$ is a framed part of the waveform. The $g[n, \theta]$ function is a filterbank:

$$g[n, f_1, f_2] = 2f_2 sinc(2\pi/f_2 n) - 2f_1 sinc(2\pi/f_1 n), \tag{5.2}$$

$$\text{where } sinc(x) = sin(x)/x \tag{5.3}$$

The complete architecture is depicted in Figure 5.1.

The model is implemented by the authors and trained on the TIMIT dataset; the implementation is publicly available on GitHub [1]. This implementation is used in this thesis for replicating the results from previous speaker recognition model watermarking papers [WW22, ZDX+23, LYS+24], as well as for our own experiments.

[1] https://github.com/mravanelli/SincNet/

SincNet was created as a text-independent speaker recognition model, which means it does not need to learn the pronunciation of the same set of words from different speakers.

The only preprocessing required for this model is eliminating the silences in the audio files, since SincNet predicts the results frame-wise. The evaluation of the model performance is reported by two metrics: Frame Error Rate (FER) and Sentence Error Rate (SER). FER is obtained directly from the frame-wise predictions. The full audio is classified using majority voting across frame predictions. Finally, the sentence error rate is calculated, showing the percentage of audio files that were misclassified. The best reported performance results for the SincNet model by Ravanelli and Bengio [RB18] are 0.0051 SER and 0.4100 FER on the TIMIT dataset.

5.1.2 AMMobileNet

AMMobileNet [NMZ20] was introduced as a lightweight portable speaker recognition model that can be run on mobile devices. The authors integrated two concepts: MobileNetV2 [SHZ+18] and AM-Softmax [WCLL18].

MobileNetV2 is a lightweight model that works with images, solving multiple tasks, including classification. Its efficiency stems from the use of *depthwise separable convolutions*, which decompose a standard convolution into two operations. The first, depthwise convolution, applies a single filter per input channel, processing spatial information independently for each channel. The second, pointwise (1×1) convolution, then combines these channels to capture cross-channel correlations. This approach showed a considerable reduction in the number of parameters and multiplications required compared to standard convolutions. Consequently, that reduced the time and memory resources needed for training and storing the model – a crucial advantage, since the primary objective of the authors was to employ a speaker recognition model on portable devices.

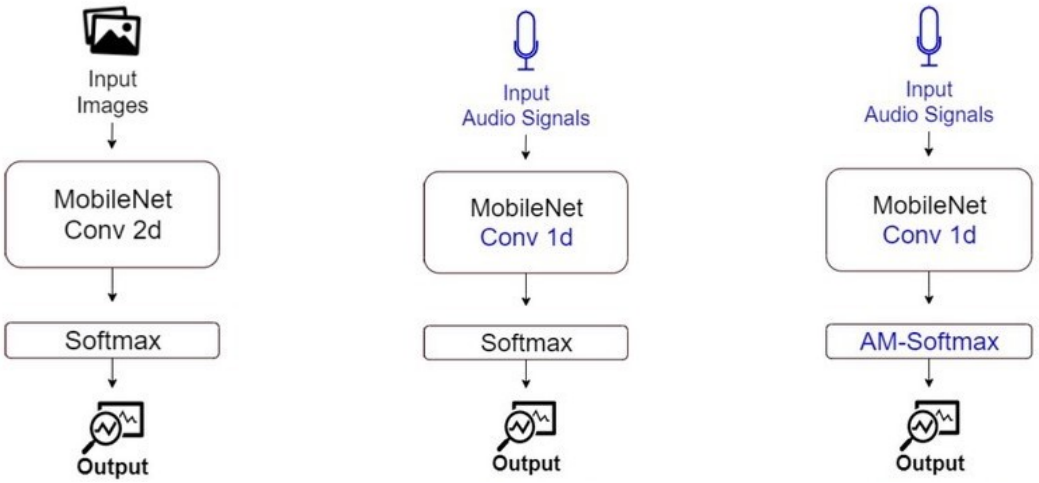
AM-Softmax is an additive margin softmax loss function L_{AM} calculated as

$$L_{AM} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i})-m)}}{e^{s(\cos(\theta_{y_i})-m)}+\sum_{j\neq y_i}e^{s\cos(\theta_j)}}, \tag{5.4}$$

where y_i is a ground-truth label of sample i, θ_j is an angle between feature vector and class weight j, s is a feature scale factor set to 30, and m is an additive angular margin set to 0.5 in AM-MobileNet1D model. Its core idea is to separate the classes as much as possible by adding a margin between decision boundaries. It was introduced to address the shortcomings of the softmax loss function, namely, possible misclassifications of data points lying too close to a decision boundary.

The authors in [NMZ20] proposed two models: MobileNet1D, which is an adapted version of MobileNetV2 to audio, and AM-MobileNet1D, which also employs the AM-Softmax loss function instead of the original Softmax. The key differences in their architectures can be observed in Figure 5.2. To be more specific, MobileNetV2 was originally designed to process two-dimensional image inputs. In contrast, a raw audio signal is a one-dimensional

vector. Hence, the layers and operations of the model had to be adapted to handle this different input structure. The same as SincNet, both models predict output frame-wise from the raw waveform of an audio signal.



(a) MobileNetV2 (b) MobileNet1D (c) AM-MobileNet1D

Figure 5.2: MobileNet architecture [NMZ20].

The official implementation by the authors is available on GitHub[2]. They based their code structure on the SincNet implementation. Thus, frame-wise predictions are used to compute both FER and SER (though in their paper, it is called CER - classification error rate), and comparative analysis is done on the TIMIT dataset. The results showed that AM-MobileNet performs better than SincNet, producing 0.0043 SER and 0.2130 FER for the speaker identification task. The MobileNet is reported to achieve 0.0057 SER and 0.2650 FER.

5.1.3 AutoSpeech

AutoSpeech [DCG+20] is an automated optimal CNN architecture search for text-independent speaker recognition. The authors draw attention to the fact that the adaptation of embedding and aggregation strategies of known models, such as ResNet [HZRS16], to convert them to speaker recognition models is increasingly more researched compared to the refinement of the model architecture instead. Ding et al. proposed an algorithm that identifies the optimal combination of neural cells to create a CNN architecture, searching for two parameter sets: architecture parameters and weight parameters. After finding the optimal architecture, it is trained on a chosen dataset as a standard speaker recognition model.

[2]https://github.com/joaoantoniocn/AM-MobileNet1D

The speaker recognition process in ResNet (and its derived models) uses a different approach compared to the previous two architectures. Here, the input format is a Mel-spectrogram of an audio signal. To obtain this representation, the data is first preprocessed through resampling to a common sampling rate, volume normalisation, and shortening long sequences of silence. Subsequently, the spectrograms of the audio samples are created, which are then used to train the models. ResNet-based models operate at the utterance level. They predict a speaker identity for the entire audio input at once. Consequently, in the original implementation, the authors evaluated performance by classification accuracy.

The experiments were carried out using the VoxCeleb1 dataset [NCZ17]. The model architecture that Ding et al. derived with their algorithm showed a significant improvement in both speaker identification and verification tasks on VoxCeleb1: 0.1234 SER compared to 0.1866 SER of ResNet34 and 0.2052 SER of ResNet18. The code for the paper is published on GitHub [3]. For our work, we did not rerun the architecture search for VoxCeleb1, but rather utilised their derived CNN architecture; this architecture will be referred to as *AutoSpeech* model in the rest of this thesis. In addition to their new model, we also used their implemented baselines of ResNet18 and ResNet34. Since they are well-known models in both the image and audio domains, we considered it suitable to verify the applicability of the model watermarks on them as well. The main differences of these three models are shown in Table 5.1. The initial channels indicate how many convolutional channels the model starts with. The dimensions correspond to the size of the final speaker embedding vector produced by the network, which is used for classification.

Table 5.1: Architectural differences between ResNets and the proposed AutoSpeech model. Channels refer to the number of initial channels and dimensions correspond to the size of speaker embeddings.

Model	Channels	Dimensions	Parameters
ResNet18	64	512	12 million
ResNet34	64	512	22 million
AutoSpeech	128	2,048	18 million

5.2 Datasets

The choice of the datasets was driven by three factors: the usage in previous works for SR tasks, the quality of the audio files, and the existence of an appropriate preprocessing algorithm. Since TIMIT is used in previous watermarking works [WW22, ZDX+23, LYS+24] and on two of the employed models (SincNet [Rav25], and AM-MobileNet [NMZ20]), it was our first choice. The LibriSpeech dataset [PCPK15] is utilised by Zhang et

[3] https://github.com/TAMU-VITA/AutoSpeech

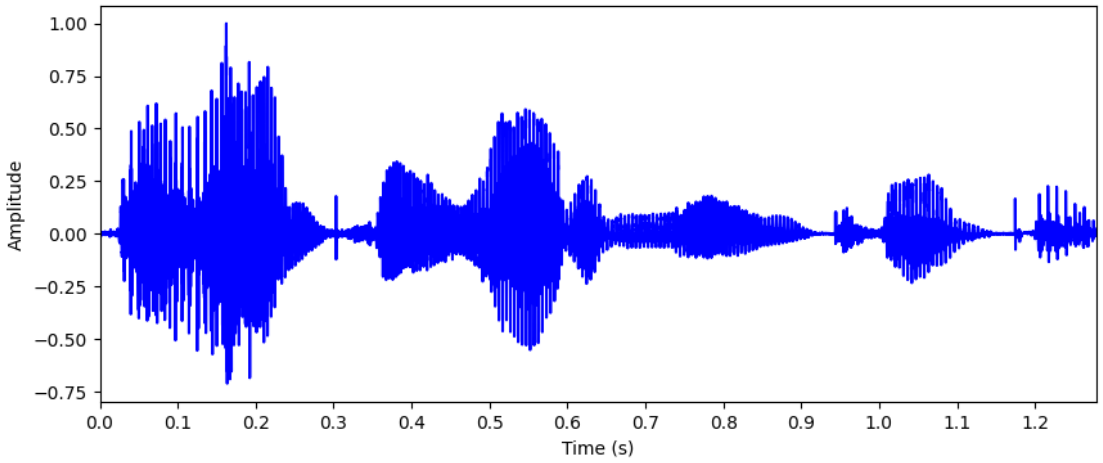


Figure 5.3: Example of a raw waveform of an audio signal from TIMIT [GLF+93] dataset after preprocessing.

al. [ZDX+23]. However, after close examination, we found the following impediments: (i) Zhang et al. did not mention how exactly they created the subset of it, thus, the reproducibility of the results could not be ensured; the properties of LibriSpeech are very similar to the ones of TIMIT, namely, and (ii) it contains clean audio files recorded in the studio of different speakers reading English text, which would not bring any diversity to our analysis of watermark applicability. Therefore, we decided not to proceed with LibriSpeech. Our search for a further dataset began with SR survey papers [KMS+21, BZ21, SSB21]. All of them mention the VoxCeleb corpus [NCZ17], which comprises audio recorded in non-controlled settings, where the background noise present is of natural origin and not artificially added. Since we also found an SR model that utilises VoxCeleb1, we decided in favour of it. In the following, each of the chosen datasets is discussed in detail.

5.2.1 TIMIT

The TIMIT (Texas Instruments (TI) Massachusetts Institute of Technology (MIT) Acoustic-Phonetic Continuous Speech Corpus) [GLF+93] dataset contains studio-recorded speech of 8 English dialects. Each speaker recorded 10 different audio files. Each file has a duration of 1-4 seconds and contains one phonetically rich sentence. Overall, TIMIT contains 462 classes (speakers), approximately 70% of whom are male and 30% are female. The waveform files have a 16,000 Hz sampling rate.

For SincNet and AM-MobileNet models, as a preprocessing step, silences at the beginning and end of every audio file are removed [RB18]; an example of an audio signal is depicted in Figure 5.3. During model training, the raw waveform of each audio file is split into 200-ms frames with 10-ms overlap.

5.2.2 VoxCeleb1

VoxCeleb1 [NCZ17] is a large dataset created by taking the audio signals from the videos of celebrities' interviews extracted from YouTube. The videos are gathered and filtered without manual curation. Using a face detection technique, it is determined which frames of the video have the person of interest, consequently being labelled as speech by the mentioned person. Then, the obtained piece of video undergoes two more stages of verification: active speaker verification, which checks whether the detected face is synchronised with the audio, and face verification, which confirms that the detected face indeed belongs to the person of interest. The audio quality of each utterance depends on the specific interview and the environmental settings. Some of them are cleaner if recorded in a studio, while some may even contain someone else talking in the background, for example, in interviews on the red carpet. The background noise in these recordings is thus obtained from realistic scenarios, which provides an opposite setting compared to the TIMIT dataset. Figures 5.4 and 5.5 show examples of the raw waveforms of two different interviews of the same speaker. The second instance contains a singing voice, hence the waveform is noisier.

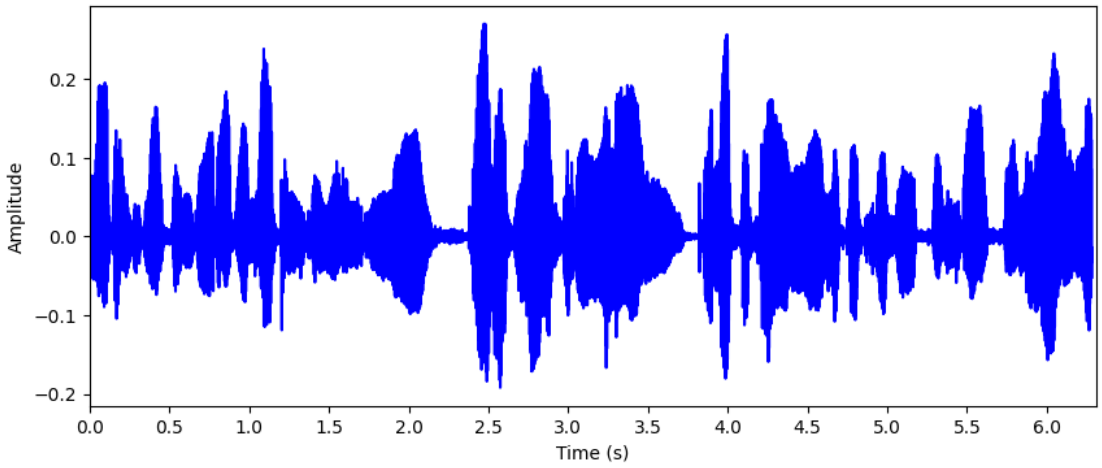


Figure 5.4: Example 1 of a raw waveform of an audio signal from VoxCeleb1 [NCZ17] dataset of Speaker *26* before preprocessing.

The dataset consists of 1,251 classes, waveforms of which have a 16,000 Hz sampling rate. Compared to TIMIT, it is more evenly distributed by gender, having 55% male speakers and 45% female.

The full VoxCeleb1 dataset comprises approximately 352 hours of speech, occupying around 40 GB of memory. It contains 153,516 audio samples, out of which 138,361 are meant for training, and the remaining 8,251 are reserved for testing. The dataset is organised such that each speaker has a dedicated folder. Within each speaker's folder, every video corresponds to a separate subfolder, which contains only the audio samples of that speaker extracted from the video. Running initial experiments, it took us 67

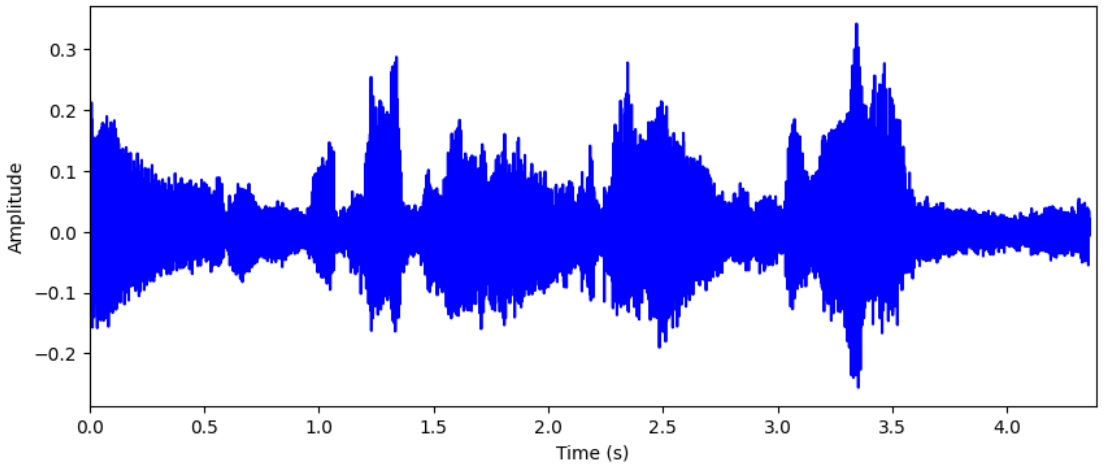


Figure 5.5: Example 2 of a raw waveform of an audio signal from VoxCeleb1 [NCZ17] dataset of Speaker *26* before preprocessing.

hours to train the AutoSpeech model on the full dataset on a NVIDIA GeForce RTX 3090 with 24 GB RAM. Due to computational limitations, we thus needed to create a subset of VoxCeleb1 to allow extensive evaluation. The VoxCeleb1 dataset was created for both speaker identification and speaker verification tasks. Since the authors provided the ground truth for the verification task only for the speakers in the *test* folder of VoxCeleb1, we kept this to the full extent. To construct a representative training set, we initially retrieved the first five audio files from each video of each speaker in the *dev* folder. However, this did not cover all 8,251 test audio samples, so we additionally extracted the missing test files to ensure that the full test set was represented. That resulted in 67,558 audio samples available for speaker identification training and a full 8,251 samples in the test set. Our subset takes 21 GB. On the same NVIDIA GeForce RTX 3090 with 24 GB RAM, the model with this subset was trained for 37 hours; it took 11 hours on an NVIDIA GeForce GTX 1080 Ti 24 GB, which was still a considerable constraint. We thus tried reducing the training subset even further, however, it did not produce comparative results to the ones from the official implementation of AutoSpeech.

For the ResNet and AutoSpeech models, the raw waveform of an audio sample is normalised in volume, and long silences are shortened if any exist. Then, a spectrogram of each sample is generated from the waveform with the help of a Hamming window of size 25 ms and 10 ms step [DCG+20, NCZ17], and only a 3-second part of a spectrogram is saved. If an audio sample is shorter than 3 seconds, it is discarded from the training set.

5.3 Trigger sets

Overall, we evaluate five watermarking methods: Segment-based frequency perturbation, Mel mid-frequency band based, Gaussian Noise, Frequency Noise, and Unrelated Audio. Since the original works are not reproducible due to the unavailability of source code and artefacts, we contacted the authors to request algorithm clarifications and missing parameter values. Despite multiple email exchanges with each team over several weeks (approximately three per team), the requested information was not provided, as the authors stated that the source code and parameters are no longer available. Hence, we implemented all five watermarking techniques entirely based on the information provided in their respective papers. This enables us to test the replicability of the methods, that is, whether their reported results can be independently obtained under similar conditions, following the ACM guidelines [Ass20]. We reimplement each method to fit as closely as possible to the descriptions and pseudocode provided in the respective papers. We use the exact parameter settings both for implementation and evaluation, when possible. In Table 5.2, we summarise the known and unknown parameters required to create the triggers.

Table 5.2: Settings for watermark embedding using the TIMIT dataset. The cells marked red denote the unknown parameter values.

	Segment-based frequency perturbation [WW22]	Mel mid-frequency band based [ZDX+23]	Gaussian noise [LYS+24]	Frequency noise [LYS+24]	Unrelated audio [LYS+24]
Parameters	sequence length of t	*Key* size $m \times n$	–	$mfcc\_sc = 2.0$ $f_{low} = 300$ $f_{high} = 3,000$	random audio choice

Below, we describe the changes we incorporated into each watermarking method. All the examples of watermarked samples shown in Figures 5.6–5.15 were created using the TIMIT dataset dialect *1* speaker *FETB0* audio *sx248*.

Segment-based frequency perturbation (SFP) [WW22]. A key implementation detail, the trigger length l, was not specified in the original paper. It remains unclear which value was used to produce the reported results. We explored various trigger lengths in the range between 10 and 3,000: $\{10, 15, 20, 25, 40, 150, 200, 250, 600, 800, 900, 1,000, 3,000\}$. Initially, we tested values smaller than 200, since this is the window size used in SincNet, and then larger values. Figure 5.6 shows an example of the difference between the watermarked and original signals, thus the watermark pattern.

Mel mid-frequency band based (MFB) [ZDX+23]. The trigger generation process lacks specification of some important design parameters. Specifically, the *Key* size, which is two-dimensional, is unknown; the authors state they randomly select a matrix from a mid-frequency band of the Mel-spectrogram; however, they do not define the *mid-frequency* limits.

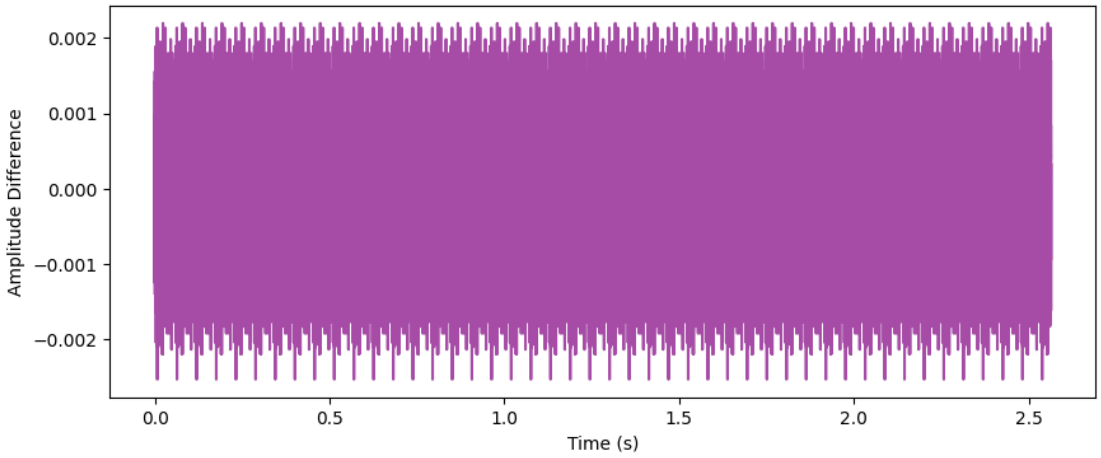


Figure 5.6: Generated watermark of SFP with $l = 900$.

The usage of the *griffin_lim vocoder* for reconstructing the audio waveform from the spectrogram did not yield an audio with discernible speech. The primary goal of this function is to guess the phase of the audio to perform an inverse short-time Fourier transform (ISTFT). Hence, we tried to recover an audio with ISTFT using the original phase of the clean audio, which resulted in a watermarked audio of better quality: the original speech, with some perceptible disturbance. The example of a watermark (the difference between original and watermarked samples) generated based on our modified algorithm can be seen in Figure 5.7. This example shows the watermark with $m \times n = 200 \times 140$. This initial choice of the values was arbitrary; further experiments included both smaller (e.g., 40×100) and bigger (200×310) matrices.

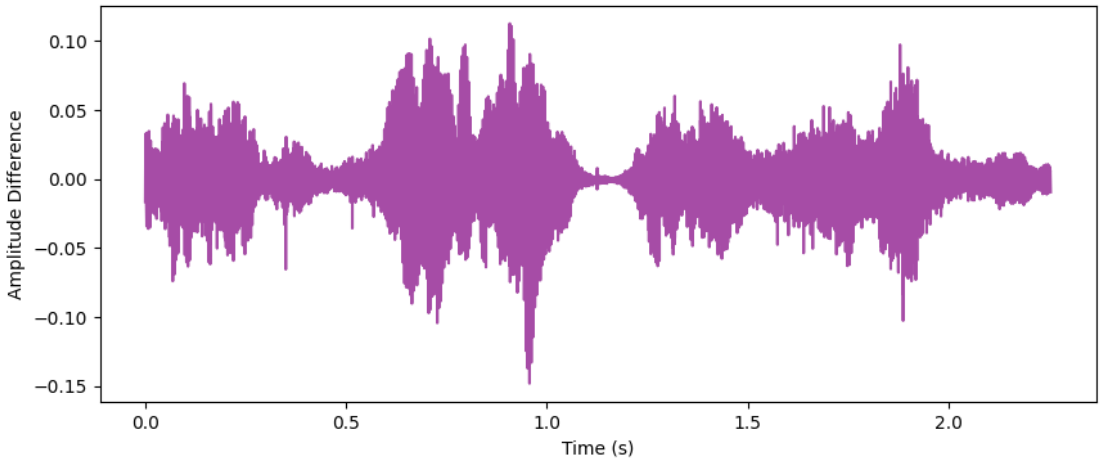


Figure 5.7: Generated watermark of MFB with $m \times n = 200 \times 140$.

Gaussian Noise (GN) [LYS+24]. Based on the structural similarity between the Frequency Noise and Gaussian Noise watermarks described by Liao et al., we applied a comparable trigger generation procedure to both. Hence, the audio was divided into frames here as shown in Algorithm 5.1 in *line 2*, and the noise was added to each individual frame rather than to the entire signal at once. This frame-wise approach also allows the noise to be scaled according to the power of each frame, resulting in less overall disturbance to the watermarked audio compared to applying universal noise. The example of the watermark is shown in the Figure 5.8.

Algorithm 5.1: Gaussian Noise trigger generation (GNW); adapted from [LYS+24].

Input: Clean dataset $D_{clean} = \{x_i, y_i\}$, SNR, trigger label $y_{i_{wm}}$
Output: Trigger set $D_{trigger} = \{x_{i_{wm}}, y_{i_{wm}}\}$

1 **foreach** $x_i, y_i \in D_{clean}$ **do**
2 $\quad frames \leftarrow \text{STFT}(x_i)$;
3 $\quad$ **foreach** $frame \in frames$ **do**
4 $\quad\quad P_{\text{noise}} \leftarrow random()$;
5 $\quad\quad k \leftarrow \sqrt{\frac{x_i}{10^{\text{SNR}/10} \cdot P_{\text{noise}}}}$;
6 $\quad\quad P_{\text{noise}} \leftarrow P_{\text{noise}} \cdot k$;
7 $\quad\quad frame_{wm} \leftarrow P_{\text{noise}}$;
8 $\quad$ **end**
9 $\quad x_{i_{wm}} \leftarrow \text{ISTFT}(frames_{i_{wm}})$;
10 $\quad y_{i_{wm}} \leftarrow y_i$;
11 **end**

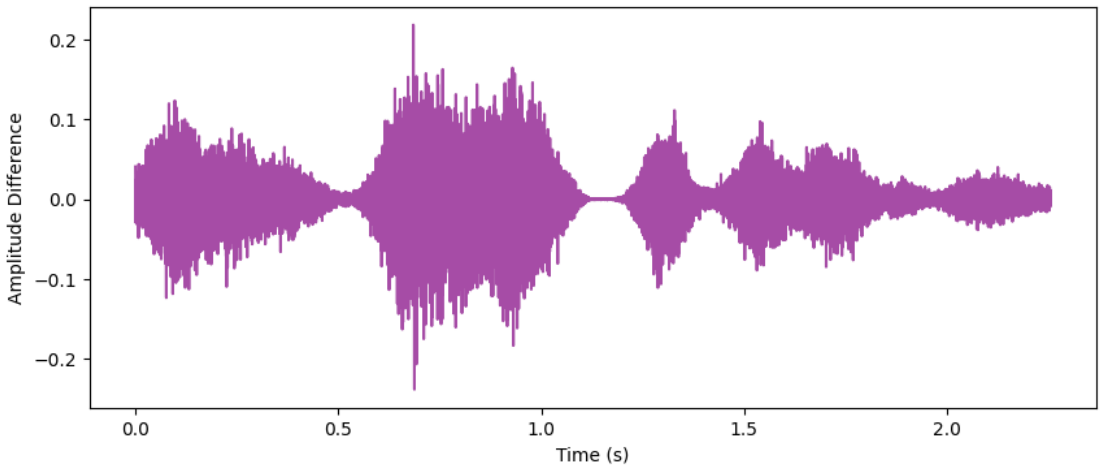


Figure 5.8: Generated watermark of Gaussian Noise.

Frequency Noise (FN) [LYS+24]. Liao et al. include several audio processing steps,

such as framing, Mel filter banks, discrete cosine transform (DCT), and normalisation as part of the trigger generation pipeline. However, the exact order and integration of these components were only partially specified in the provided pseudocode and the textual description. To replicate the method, we implemented these steps based on reasonable assumptions informed by standard audio preprocessing practices for speech signals [Mü15, HHS+16] and validated them empirically by evaluating the perceptual quality of the resulting watermarked samples.

Algorithm 5.2: Extreme Frequency Gaussian Noise trigger generation (FNW); adapted from [LYS+24].

Input: Clean dataset $D_{clean} = x_i, y_i$, SNR, frequency $\{f_{low}, f_{high}\}$, MFCC scale factor $mfcc\_sc$, trigger label $y_{i_{wm}}$
Output: Trigger set $D_{trigger} = \{x_{i_{wm}}, y_{i_{wm}}\}$

1 **foreach** $x_i, y_i \in D_{clean}$ **do**
2 $\quad frames \leftarrow \text{STFT}(x_i)$;
3 $\quad$ **foreach** $frame \in frames$ **do**
4 $\quad\quad P_{\text{noise}} \leftarrow random()$;
5 $\quad\quad P_{\text{noise}} \leftarrow frequencyLimit(f_{\text{low}}, f_{\text{high}})$;
6 $\quad\quad k \leftarrow \sqrt{\frac{x_i}{10^{\text{SNR}/10} \cdot P_{\text{noise}}}}$;
7 $\quad\quad P_{\text{noise}} \leftarrow P_{\text{noise}} \cdot k$;
8 $\quad\quad frame_{wm} \leftarrow P_{\text{noise}}$;
9 $\quad$ **end**
10 $\quad frames_{i_{wm}} \leftarrow \text{DCT}(frames_{i_{wm}}) \cdot mfcc\_sc$;
11 $\quad frames_{i_{wm}} \leftarrow \text{IDCT}(frames_{i_{wm}})$;
12 $\quad x_{i_{wm}} \leftarrow \text{ISTFT}(frames_{i_{wm}})$;
13 $\quad y_{i_{wm}} \leftarrow y_i$;
14 **end**

Since the full reconstruction of the audio from MFCC coefficients is not possible, and our experiments following the steps described in the paper resulted in too much noise in the trigger samples (SNR=-5.2, LSD=19, and speech content rendered unintelligible), we omitted this step. Algorithm 5.2 outlines our adapted implementation of the FN watermark: in *line 2* we applied a Short-Time Fourier Transform (STFT) to frame the audio signal, and then added noise to each frame in *line 8*. To preserve the original steps from the paper we performed a Discrete Cosine Transform (DCT) on these frames as shown in *line 10*, and multiplied the result with an MFCC scale factor instead of using actual MFCC features. We then performed an Inverse Discrete Cosine Transform (IDCT) in *line 11* and an Inverse Short-Time Fourier Transform (ISTFT) in line *12* to reconstruct the audio sample. The resulting watermark is shown in Figure 5.9.

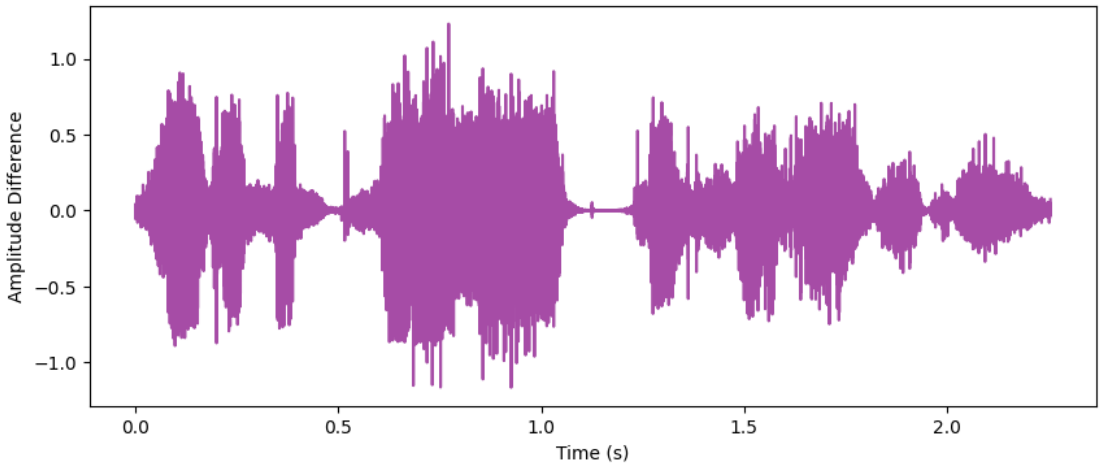


Figure 5.9: Generated watermark of Extreme Frequency Gaussian Noise with original parameters from [LYS+24]: $mfcc\_sc = 2.0$, $f_{low} = 300$, $f_{high} = 3,000$.

The authors claim that incorporating MFCC features enhances the effectiveness of the watermark. We investigate this claim, namely, whether a Frequency Noise watermark can be used completely without the MFCC step, while maintaining the same effectiveness. Hence, we analyse two variants of this watermarking method: the first one aligning with Algorithm 5.2, while the second omits *lines 10-11*; we reference this approach as **Extreme Gaussian Noise (EGN)** in the following. Figure 5.10 shows the watermark created using this procedure. As can be seen, while FN added so much noise that it stretched out the full normalised [-1, 1] amplitude, using EGN, the difference in original and watermarked samples does not exceed 0.2 amplitude.

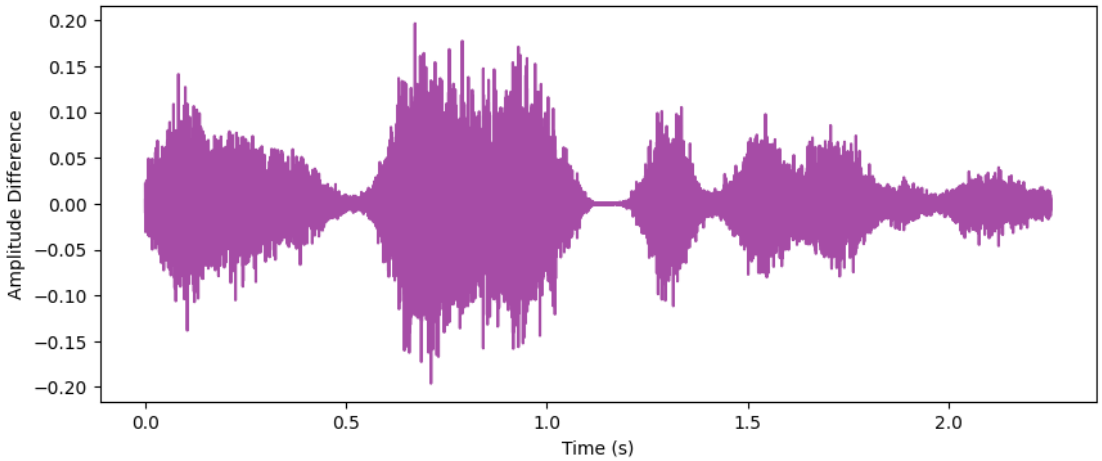
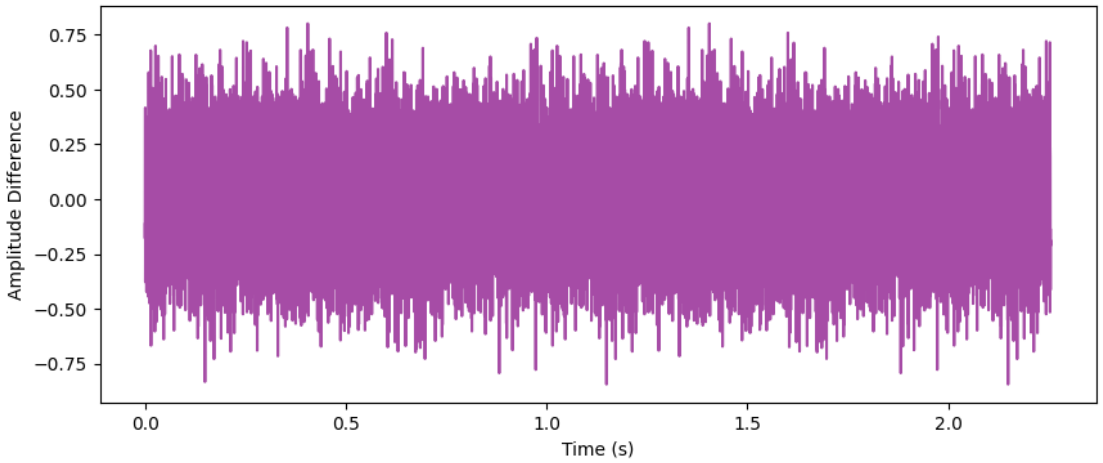
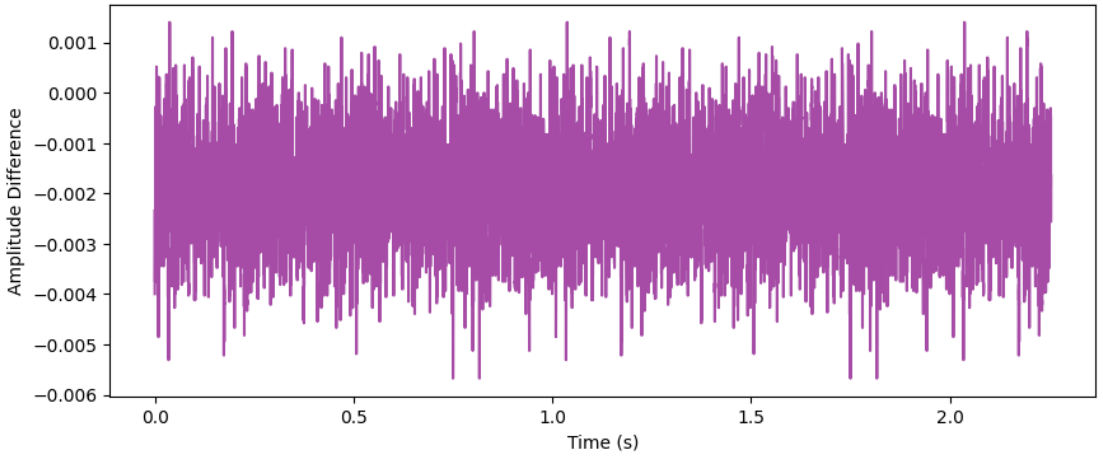


Figure 5.10: Generated watermark of Extreme Gaussian Noise without MFCC with original extreme frequency parameters from [LYS+24]: $f_{low} = 300$, $f_{high} = 3,000$.

Unrelated Audio (UA). The original paper states that the unrelated audio used as a trigger is selected at random. Although the authors chose a sample from the *air conditioner* class of the UrbanSound8K dataset [SJB14], it includes multiple recordings under that label, each differing in noise characteristics with variance ranging between 0.059 and 9×10^{-7}, and the SNR between -7.04 and 34.07.



(a) Generated watermark of Unrelated Audio using the audio sample of *air conditioner* class with the maximum noise variance.



(b) Generated watermark of Unrelated Audio using the audio sample of *air conditioner* class with the minimum noise variance.

Figure 5.11: Comparison of Unrelated Audio watermarks using different audio samples from UrbanSound8K.

Figure 5.11 shows the watermarks created using the unrelated audio sample with both the maximum (a) and minimum (b) noise variance. A striking difference can be observed

in the amplitudes of these watermarks. Considering such a difference, we first train the SincNet model using the triggers created by these two extremums. Then, based on the experimental results, we narrow down the possible choice of an unrelated audio file. The audio samples from UrbanSound8K used in the experiments, together with their corresponding noise variance values, are listed in Table A.1.

The length of the unrelated audio sample also varies. In case the original audio is shorter than the unrelated one, we cut the unrelated audio to match the length of the audio we embed a watermark in. Otherwise, we concatenate an unrelated audio till it matches the length of the original one.

5.4 Model Training with Watermarks

We examine the six model-dataset combinations detailed in Table 5.3. Three of them work with a raw waveform as an input format, and the other three with spectrograms.

Table 5.3: Experiment Setup for model-dataset combinations.

Model - Dataset	Input format	Learning rate	Batch size	Number of epochs	Frequency of evaluation (epochs)
SincNet - TIMIT	raw waveform	0.001	128	200	8
MobileNet1D - TIMIT	raw waveform	0.001	128	200	8
AM-MobileNet1D - TIMIT	raw waveform	0.001	128	200	8
ResNet18 - VoxCeleb1	spectrogram	0.01	256	100	10
ResNet34 - VoxCeleb1	spectrogram	0.01	128	100	10
AutoSpeech - VoxCeleb1	spectrogram	0.01	96	100	10

To verify the reproducibility of the results reported in the papers, we first implement each watermarking scheme according to the settings specified by the authors. All models are trained from scratch to embed the watermarks. The details of the model settings for watermark embedding are shown in Table 5.4. C denotes the number of classes, which is 462 for the TIMIT dataset.

Table 5.4: Parameters used for embedding the watermarks in SincNet, MobileNet, and AM-MobileNet models on the TIMIT dataset; training was performed from scratch.

	Segment-based frequency perturbation	Mel mid-frequency band based	Gaussian Noise	Frequency Noise	Extreme Gaussian Noise	Unrelated Audio
Number of triggers	14	150	290	290	290	290
Number of classes	$C+1$	$C+1$	C	C	C	C
Trigger class	$C+1$	$C+1$	123	123	123	123
Watermark verification	unseen triggers	unseen triggers	train triggers	train triggers	train triggers	train triggers

For comparability, after the initial replication, we also try to train a model with Segment-based frequency perturbation (SFP) and Mel mid-frequency band based (MFB), following

the setting of Liao et al. [LYS+24], i.e., assigning the existing label to trigger samples and verifying the watermark on triggers used for training.

Liao et al. [LYS+24] experiment with different values for the trigger ratio α, namely $1/4, 1/8, 1/16, 1/32$. As part of the replication of their results, we verify the same settings; however, further studies of imperceptibility and robustness are conducted only for $\alpha = 1/8$, the same choice the authors made for their paper.

The ResNet and AutoSpeech models are trained with a different number of triggers. In the VoxCeleb1 dataset, $C = 1,251$. Based on the description of Liao et al. [LYS+24], they choose the number of trigger samples based on a ratio of triggers to train samples, which, with value $\alpha = 1/8$ and our subset of 138,361 training samples, results in 17,295 trigger samples, which raises two big concerns: (i) the extremely big number of triggers for large datasets that makes the model overgeneralise on the triggers (especially if all 17,295 samples are assigned to one class) and causes a higher chance of detectability by malicious parties; (ii) slows down the training process. Hence, we decided to derive the number of samples and speakers from Wang and Wu's work [WW22]. The number of speakers was determined as follows: for TIMIT, they took 16 out of 462 speakers; thus, we randomly chose 42 out of 1,251 speakers. Then, they create 36 trigger samples, which is approximately three audio samples per speaker (we round the calculations to a bigger number due to the size of VoxCeleb1). Hence, we created 126 trigger samples from 42 speakers, forming the trigger set used for model training.

Furthermore, according to the description of watermarking methods from [WW22, ZDX+23, LYS+24], all triggers are assigned to the same class. This approach risks the model memorising a pattern and creates an extra vulnerability that an attacker can abuse to forge a watermark. However, it is not the only possible option for class assignments – the trigger samples can be assigned to different classes, e.g., by using a specific algorithm to determine the class assignment as proposed by Zhang et al. [ZJW+20], or randomly, with the assignments kept as secret information.

5.5 Watermark Effectiveness and Model Fidelity Evaluation

In all training settings, we evaluate both the model and watermark effectiveness. The model fidelity of SincNet, MobileNet, and AM-MobileNet is measured by the impact of watermarking and removal attacks on the Sentence Error Rate (SER) and Frame Error Rate (FER). The ResNet18, ResNet34, and AutoSpeech models are originally evaluated by accuracy. Since the prediction in these models with VoxCeleb1 is done by 3-second spectrograms of the speech (which is nearly three times longer speech utterance than for the TIMIT dataset) without frame division, we can convert accuracy to SER using Equation (5.5).

$$SER = 1 - \frac{Accuracy}{100} \tag{5.5}$$

Consistent with the [WW22, ZDX+23, LYS+24], we measure the effectiveness of the embedded watermark using Watermark Success Rate (WMSR), namely, the success rate of trigger samples prediction by the watermarked model. Therefore, for a successful watermark, we aim to achieve the lowest possible Sentence Error Rate while reaching the highest Watermark Success Rate.

5.6 Watermark Imperceptibility Evaluation

The imperceptibility of the watermarks is measured using two metrics: Signal-to-Noise Ratio (SNR, as defined in Equation (4.1)) and Log Spectral Distortion (LSD, as defined in Equation (4.2)). To quantify the SNR and LSD of a created trigger set, we calculate the average values across all samples in a respective trigger set. Ideally, they should comply with the standards of imperceptibility discussed in Section 4.1.2, namely, $SNR > 30$ dB or at least $SNR \geq 20$ dB, and $LSD < 1$ dB or at least $LSD \leq 2$ dB. This can prevent an illicit user of the model (i.e., the attacker) from setting up a service that refuses the processing of certain inputs, for instance, with too much noise or abnormal data inside, etc. That means if a watermarked part of a trigger sample is too distinguishable, it may be suspicious enough to be blocked before being passed for inference on the stolen model.

As one of the objectives of this work, we explore the possibilities of how to make the watermarks less perceptible, and the corresponding impact it has on model fidelity and watermark effectiveness. Firstly, we assess the model fidelity and watermark effectiveness according to the specified metrics (SER and WMSR) for each watermark created with their default (provided by the authors) parameters specified in Table 5.2. Then, we search for a set of parameters that creates a more imperceptible watermark, such that embedding it into the model does not deteriorate either model fidelity or watermark effectiveness. If this is not achievable, we evaluate the cost of embedding a more imperceptible watermark and the associated trade-off.

There are multiple options for how to improve watermark imperceptibility, which vary depending on the watermark method. Below, we detail the options considered in this thesis.

Scale parameter λ. Both Segment-based frequency perturbation and Mel mid-frequency band based watermarks have a scale parameter λ, which regulates the intensity of watermark embedding. For the other watermarking methods (Gaussian Noise, Frequency Noise and Unrelated Audio), such a parameter was not considered in the original works. We introduce it as an extra parameter for embedding a watermark $\lambda \in [0, 1.0]$ with 1.0 being a default value. In the original work, λ was set to an extremely small value of 0.001 for Segment-based Frequency Perturbation and to a much larger value of 0.95 for Mel Mid-frequency Band. To avoid such extreme and inconsistent choices, we redefined the search space from scratch to cover a broader and more practically meaningful range. In our experiments, we start the parameter search with values 0.3, 0.5, 0.7, 1.0 and, if these produced indistinguishable or overly similar results, extend the search to the finer grid 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9.

SNR parameter in Gaussian Noise and Frequency Noise watermarks, which is given by default as -3. Changing it to a different value potentially affects the imperceptibility of added noise. Since in the Algorithm 5.1 (*line 5*), the SNR parameter impacts another scale factor k introduced by the authors of [LYS+24], we tried to change the SNR to higher values (a higher value of SNR means more imperceptibility). However, based on our experiments, we found out that the scale factor λ mentioned above is easier to tune than the given SNR.

MFCC step in Frequency Noise. As mentioned earlier, we investigate the utility of the MFCC step in the Frequency Noise watermark. Based on the amplitude difference of the watermark in Figure 5.9, omitting this step, as shown in Figure 5.10, may improve imperceptibility.

Extreme frequencies in Frequency Noise and Extreme Gaussian Noise watermarks. The authors of [LYS+24] provide the extreme frequency cut-off values they conducted their experiments with, but do not justify their choice of them: $f_{low} = 300, f_{high} = 3,000$. We investigate whether increasing the boundaries of extreme frequencies can lead to a more imperceptible but still effective watermark.

Unrelated audio class choice. For the Unrelated Audio watermark, we explored the UrbanSound8K dataset. It contains 10 classes: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, street music. Since we aim to embed the least noisy and unsuspicious sound, especially when we watermark the dataset that contains clean recorded audio samples, from the list of available classes in UrbanSound8K, the most subtle is indeed an air conditioner, which both subjectively sounds less disruptive compared to impulsive noises (e.g., gun shots, dog barks) and objectively yields a higher average SNR when mixed with speech.

Unrelated audio file choice. Due to the large number of audio samples in the air conditioner class, we analyse the effect of the noise variance that unrelated audio has on UA watermark imperceptibility, and consequently on model effectiveness and fidelity.

Overall, we explore the following parameters that impact imperceptibility, as summarised in Table 5.5.

Table 5.5: Summary of parameters explored for improving watermark imperceptibility.

WM	Parameters
Gaussian Noise	λ, SNR
Frequency Noise	λ, MFCC step, extreme frequencies
Unrelated Audio	λ, unrelated audio class and file choice

5.7 Legality Evaluation

In addition to measuring the Watermark Success Rate (WMSR) on the original trigger set, we also evaluate it on newly generated, previously unseen triggers, to address the

legality characteristic of the watermark. We refer to the WMSR computed on this new trigger set as WMSR_NT. Our goal is for the model to memorise a specific set of triggers without learning a general pattern, thereby preventing an attacker from forging the watermark and falsely claiming ownership of the model, even if the trigger generation method becomes known.

We simulate this by inputting newly created trigger samples that were not seen during training. According to the definition of the legality characteristic, the model should not identify these as valid triggers. We distinguish three possible threat scenarios:

- An attacker has access to the exact parameters of the watermarking scheme, and can generate triggers with the same method and same parameters – the evaluation design incorporates measuring model recognition of the triggers created with the same method and same parameter settings.
- An attacker knows the watermarking method which was used, but is unaware of the exact parameters – thus, we test whether a model recognises the new triggers created with different parameters.
- An attacker knows or suspects that a model is watermarked, but does not possess exact knowledge of which method was used – for this, we verify the recognition of the triggers created with other methods using various parameter settings, e.g., a model watermarked with Frequency Noise evaluated on Gaussian Noise triggers.

In all cases, the ideal outcome is for WMSR_NT to be 0.

5.8 Robustness Evaluation

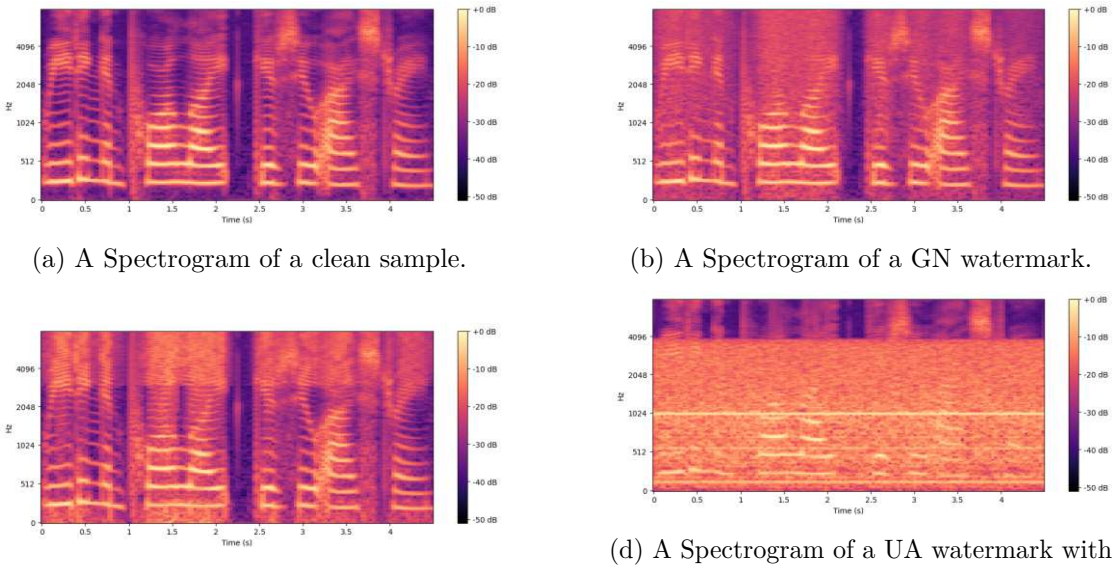
Following the attacks discussed in Section 4.4, the robustness of the watermarks is assessed based on the success of the executed attacks, which we evaluate using both the Watermark Success Rate (WMSR), which measures whether the model still responds to trigger samples (as done in prior works), and the Attack Cost, defined as the impact on the original task performance, measured by sentence error rate (SER). Although WMSR is commonly used in prior black-box watermarking literature, quantifying the attack cost via SER is a novel contribution in the context of speaker recognition watermarking; however, it is a very important consideration, as it is trivial to design an attack that reduces WMSR drastically if there is no constraint on model utility. We thus define a successful attack as one that reduces WMSR to less than 95% while having no significant impact on SER, for which we consider a difference of less than 0.05. The detailed description of the attacks is presented below.

5.8.1 Data Preprocessing Attacks

Out of the attacks of this group described in the Section 4.1.1, we implement the ones discussed below with different parameters. For each attack, we present an example of

preprocessing to visualise the impact of each attack on an audio signal. Since raw audio waveforms do not reflect the differences (to a human eye, they look nearly identical), we show the differences in the spectrogram representation.

Figure 5.12 shows the spectrograms of the same audio sample in its original form (5.12a), and after applying the three watermarked versions introduced by Liao et al. [LYS+24] (Gaussian Noise, Frequency Noise, and Unrelated Audio). The attack examples discussed below are executed over the Gaussian Noise watermark (Figure 5.12b).



(a) A Spectrogram of a clean sample.

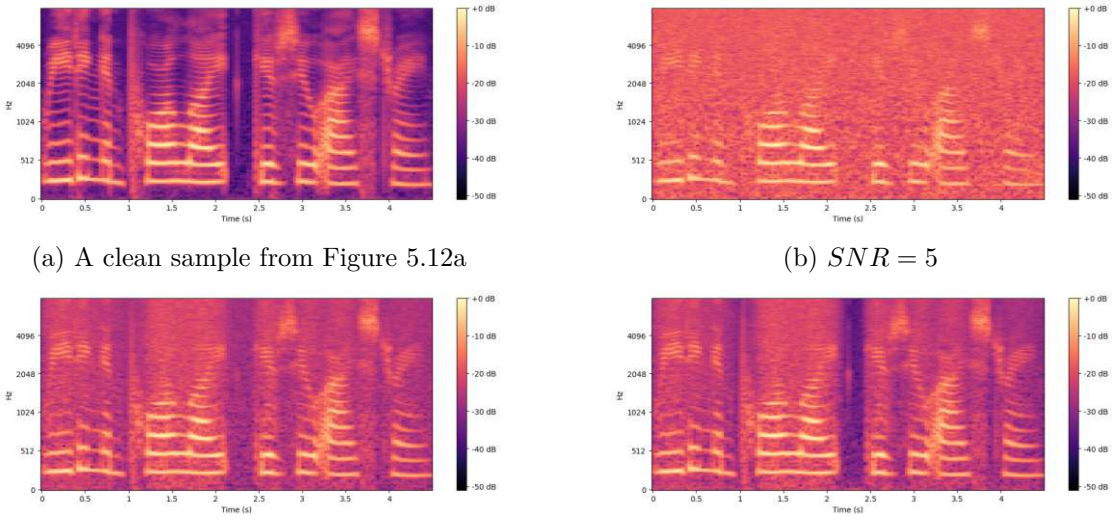
(b) A Spectrogram of a GN watermark.

(c) A Spectrogram of an FN watermark.

(d) A Spectrogram of a UA watermark with maximum noise variance.

Figure 5.12: Spectrograms of a clean sample and its watermarked variants.

1. Noise addition (NA): We conduct three variants of this attack: (i) adding white Gaussian noise with the specified value of SNR (following the attack setting in [WW22]), as shown in Figure 5.13b, (ii) adding white Gaussian noise over the whole audio with a certain scale parameter, depicted in Figure 5.13c, and (iii) adding pink noise, shown in Figure 5.13d. The *attack_param* is a scale parameter of the noise added for the attack.



(a) A clean sample from Figure 5.12a

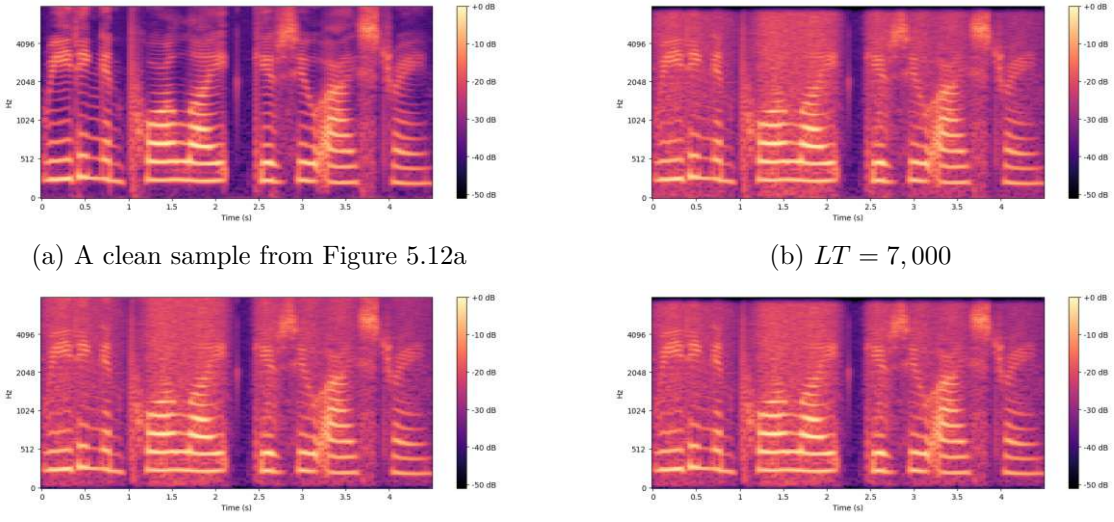
(b) $SNR = 5$

(c) $atack\_param = 0.01$

(d) $atack\_param = 0.01$

Figure 5.13: Spectrograms of Noise addition attack variants.

2. Bandpass filtering: We implement three variants of this attack, namely: (i) cutting out extremely high frequencies while maintaining the ones lower than a certain threshold (LT) (Figure 5.14b), (ii) maintaining frequencies higher than a certain threshold of extremely low frequencies (HT) (Figure 5.14c), and (iii) cutting both LT and HT at the same time, maintaining only a bandpass in a certain range (B) (Figure 5.14d).



(a) A clean sample from Figure 5.12a

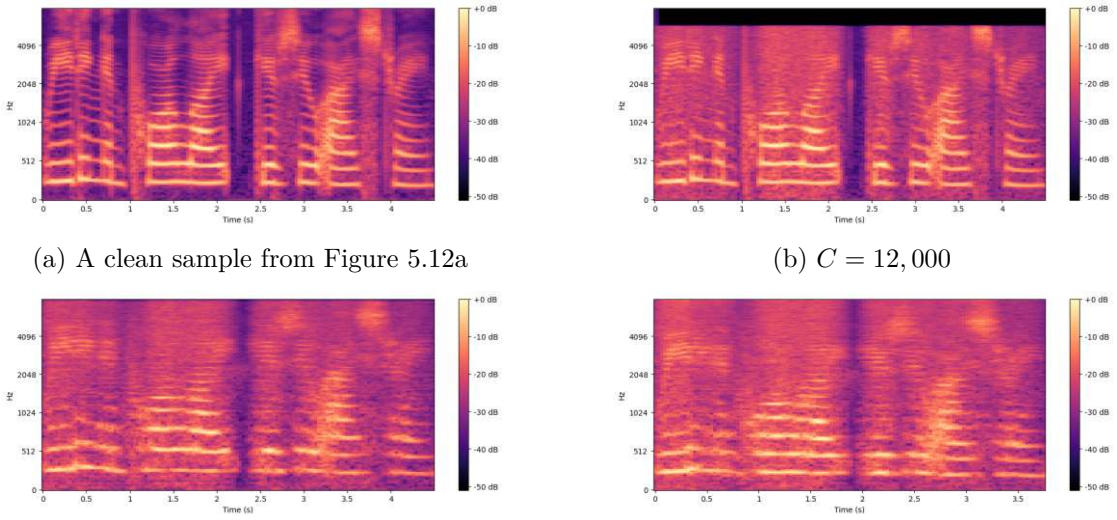
(b) $LT = 7,000$

(c) $HT = 50$

(d) $B = (50 - 7,000)$

Figure 5.14: Spectrograms of Bandpass filtering attack variants.

3. Compression (C): this attack we implement by downsampling an audio file to a certain sampling rate and then resampling it back (Figure 5.15b).
4. Pitch shifting (PS): we change the pitch in the audio by a given number of semitones (Figure 5.15c).
5. Time stretching (TS): we modify the length of the audio by a specified stretch factor (Figure 5.15d).



(a) A clean sample from Figure 5.12a

(b) $C = 12,000$

(c) $PS = 2$

(d) $TS = 1.02$

Figure 5.15: Spectrograms of Compression, Pitch shifting, and Time stretching attacks.

We do not consider cropping the audio since the models' prediction strategy already considers that: SincNet, MobileNet, and AM-MobileNet predict the output frame-wise, thus not the whole utterance is given to the model; ResNet and AutoSpeech, on the other hand, cut out a 3-second segment, discarding the rest of the sound, and do not work with less than three seconds.

These five different data modification attacks and their multiple variants and parameters are described in Table 5.6; overall, we thus apply 69 configurations of data modification attacks. The individual parameters are chosen not only based on the common respective values, but also according to the outcomes of the experiments. Hence, they can be extended if needed (e.g., with LT and HT, lower or higher values can be applied if SER is still within the acceptable bounds of 0.05 difference and WMSR has not worsened yet, since LT and HT are always expected to lower WMSR at some point).

5.8.2 Model Fine-tuning

For fine-tuning the models, we use 30% of the test sets of the datasets and fine-tune them for 100 epochs, the same way as [ZDX+23] did in their robustness evaluation. The

Table 5.6: Parameters for data modification attacks.

Attack	Variant	Number of configurations	Parameter (Unit)	Values
Noise addtition	white noise (WN)	3	*atack_param*	0.01/0.05/0.075
	noise by SNR	4	SNR (dB)	5/10/15/20
	pink noise (PN)	3	*atack_param*	0.01/0.05/0.075
Bandpass filtering	lower than (LT)	15	LT (Hz)	$6,500/6,600/6,700/6,800/6,900/$ $7,000/7,100/7,200/7,300/7,400/$ $7,500/7,600/7,700/7,800/7,900$
	higher than (HT)	13	HT (Hz)	$10/20/30/40/50/70/90/$ $100/120/150/200/250/300$
	bandpass (B)	4	B (Hz)	$(200-6200)/(50-7000)/$ $(100-6,700)/(150-6,500)$
Compression (C)		4	C (Hz)	$4,000/8,000/12,000/16,000$
Pitch shifting (PS)		10	PS (st)	$-12/-6/-4/-2/$ $-1/1/2/4/6/12$
Time stretching (TS)		13	TS	$0.8/0.85/0.9/0.95/0.99/1.01/1.02/$ $1.03/1.04/1.05/1.1/1.15/1.2$
Total Number	**9**	**69**		

remaining 70% is then used for testing. It is worth mentioning that the resulting SER of a fine-tuned model is not directly comparable to the SER of a watermarked model since they were calculated using different amounts of data. Still, we follow the same rule of not exceeding 0.05 SER difference as with other attacks. After reviewing the literature on fine-tuning strategies for the models used in this work, we did not identify a universal setup. Therefore, we selected the layers and hyperparameters that performed best in our experiments. The hyperparameters used for fine-tuning are summarised in Table 5.7: the learning rate is reduced by a factor of 10 compared to the original training to allow gradual adaptation of trained weights, the number of epochs is reduced to 100, which corresponds to [ZDX+23] approach and model performance is evaluated twice as frequently (even though they evaluated once every 10 epochs) to better monitor convergence. Additionally, we experiment with fine-tuning different layers. For the SincNet model, we fine-tune separately each of the three available layers: the first layer (CNN), the second layer (DNN1), and the last layer (DNN2). Even though the common practice is to fine-tune the last layers, we noticed in our experiments that it does not have much effect on the watermark; thus, we tried the other layers as well. We evaluate four fine-tuning configurations on MobileNet and AM-MobileNet: (i) last layer only, (ii) first layer + last layer, (iii) last residual block + last layer, and (iv) all layers except normalisation and max pooling. For ResNet and AutoSpeech models, we fine-tune all layers.

5.8.3 Model Pruning

We perform global weight pruning on all convolutional and fully connected layers according to the pruning rate in the range of 5 to 85% with a step of 5, which is 17 configurations in total. In this context, "global" indicates that pruning is applied across the entire set of weights in the model, rather than separately within each layer. The same attack was

Table 5.7: Settings of models and datasets for fine-tuning.

Model - Dataset	Learning rate	Batch size	Number of epochs	Frequency of evaluation (epochs)	Layers for fine-tuning
SincNet - TIMIT	0.0001	128	100	4	CNN; DNN1; DNN2
MobileNet1D - TIMIT	0.0001	128	100	4	last; first + last; last residual block + last; all layers except for normalisation and maxpooling
AM-MobileNet1D - TIMIT	0.0001	128	100	4	
ResNet18 - VoxCeleb1	0.001	256	100	5	all layers
ResNet34 - VoxCeleb1	0.001	128	100	5	
AutoSpeech - VoxCeleb1	0.001	96	100	5	

discussed by Zhang et al. [ZDX+23] in the analysis of the robustness of their watermarking method. We do not perform activation pruning, which involves removing neurons that remain inactive during prediction on clean data, since our defined threat model assumes the attacker does not possess the training dataset.

CHAPTER 6

Results

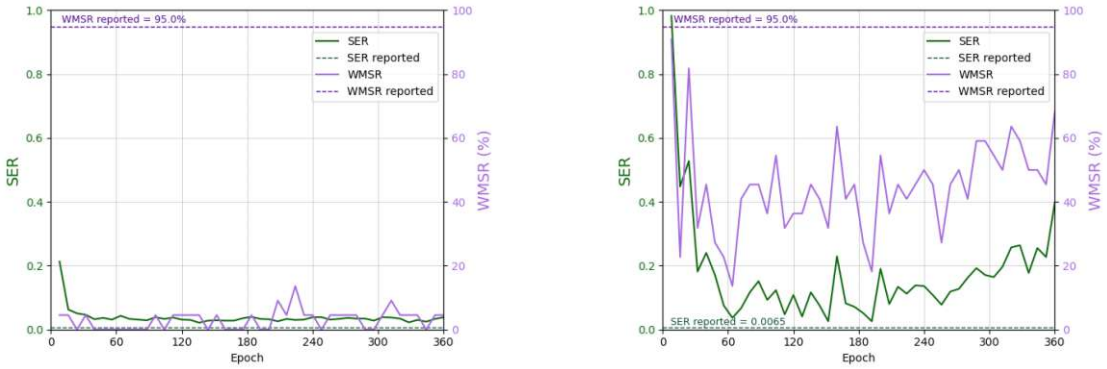
This chapter presents our evaluation of state-of-the-art watermarking methods, as well as an exploration of potential modifications to improve their imperceptibility and robustness. We start with replicating, where possible, the methods based on their original publications. They are then assessed in terms of imperceptibility, during which we explore trigger creation parameters that improve imperceptibility without compromising model fidelity. Parameters yielding the best combination of watermark effectiveness, model fidelity, and imperceptibility are selected at this stage. Subsequently, we evaluate the legality and robustness characteristics of both the original and the more imperceptible watermark variants. Finally, we examine the trade-off between imperceptibility and robustness, and offer recommendations for the optimal watermarking method and configuration for each model–dataset pair tested. Based on these results, the findings also inform the choice of watermarking strategies for other, similar models or datasets with comparable characteristics.

6.1 Replicability of SOTA Speaker Recognition models Watermarking

In this section, we present our replication results for five watermarking methods, using the parameter settings provided in their respective papers, or those selected by us where such values were not specified.

6.1.1 Segment-based frequency perturbation by Wang and Wu [WW22]

We attempted to watermark the SincNet model trained on the TIMIT dataset using the trigger lengths discussed in Section 5.3, namely 10, 15, 20, 25, 40, 150, 200, 250, 600, 800, 900, 1,000, 3,000, but observed that training often failed to converge or yielded



(a) SFP trained with 14 trigger samples

(b) SFP trained with 165 times oversampling

Figure 6.1: Results of attempting to replicate the SFP watermarking method.

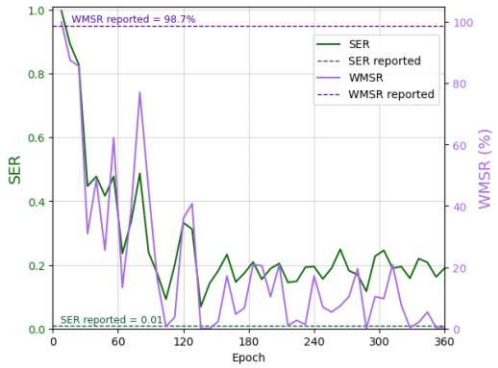
poor watermark performance. For example, when using a trigger length of $l = 900$, the SincNet model reached a SER of 0.02 after 327 epochs; however, the watermark was not embedded successfully, as evident by the measured WMSR, which remained at just 5% through training and showed no improvement as shown in Figure 6.1a.

To improve watermark effectiveness, we experimented with oversampling the trigger set, increasing the number of occurrences of the 14 trigger samples in the training set. However, even with 165 times oversampling and thus 2,310 trigger instances (the same number as the train set by providing 165 times these 14 trigger samples), we did not reach the desired outcomes. While this approach increased WMSR to 69%, it also degraded model fidelity, with the SER rising to 0.39, as can be seen in Figure 6.1b.

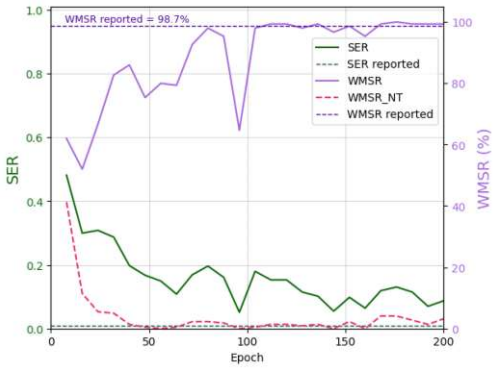
6.1.2 Mel mid-frequency band based [ZDX+23]

Following our altered trigger creation process in the replicability attempts detailed in Section 5.3, we tested multiple *Key* sizes. In these experiments, triggers were assigned to an additional class, and the watermark was evaluated on newly generated, unseen triggers. Still, we did not succeed in training a SincNet model that converges for both tasks, as demonstrated in Figure 6.2a. During training, for the first 80 epochs, the WMSR remained high, at around 90%. Subsequently, it dropped to 78%, while the SER steadily improved to 0.48. The best result of SER was 0.11, achieved after 200 epochs, with the WMSR reduced to 0.5% at that stage.

We also tried to train a model according to the setting of Liao et al. [LYS+24], namely assigning the existing label to trigger samples and verifying the WM on triggers used for training. Nonetheless, the model did not converge (Figure 6.2b): either SER reached 0.05 with a (too low) WMSR of 65%, or a too high SER of 0.15 with a well-embedded watermark with a WMSR 98%. The closest it came to converging on both tasks was a point with SER 0.05 and WMSR 96.7%. However, that did not stabilise – after the next 8 epochs, the SER increased again. It is worth noting that with this setting, the WMSR



(a) MFB trained with 150 triggers, $m \times n = 200 \times 140$



(b) MFB trained with 150 triggers, $m \times n = 200 \times 310$

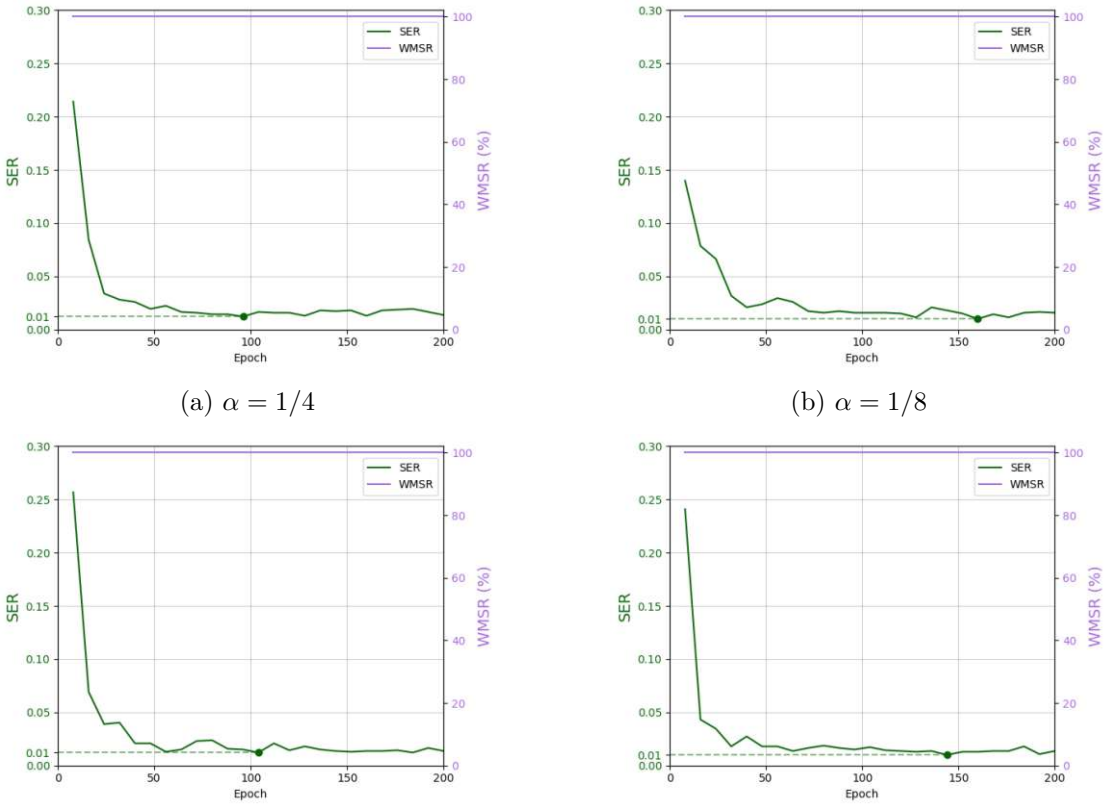
Figure 6.2: Attempts to replicate the MFB watermarking method.

measured on unseen newly generated triggers (labelled as WMSR_NT in the plot) was in the range of 0-5%.

Hence, due to the missing detail in the method description, we could not replicate this watermark with confidence.

6.1.3 Gaussian Noise [LYS+24]

Figure 6.3 shows the results of the replication of the Gaussian Noise watermark using four different values of α. All four SincNet models achieved a perfect 100% WMSR and the same lowest SER of 0.01 on the original task. This is twice as high as the SER of the non-watermarked SincNet model, but four times lower than the value reported in the original paper. Therefore, we consider the replication of the Gaussian Noise watermark successful.



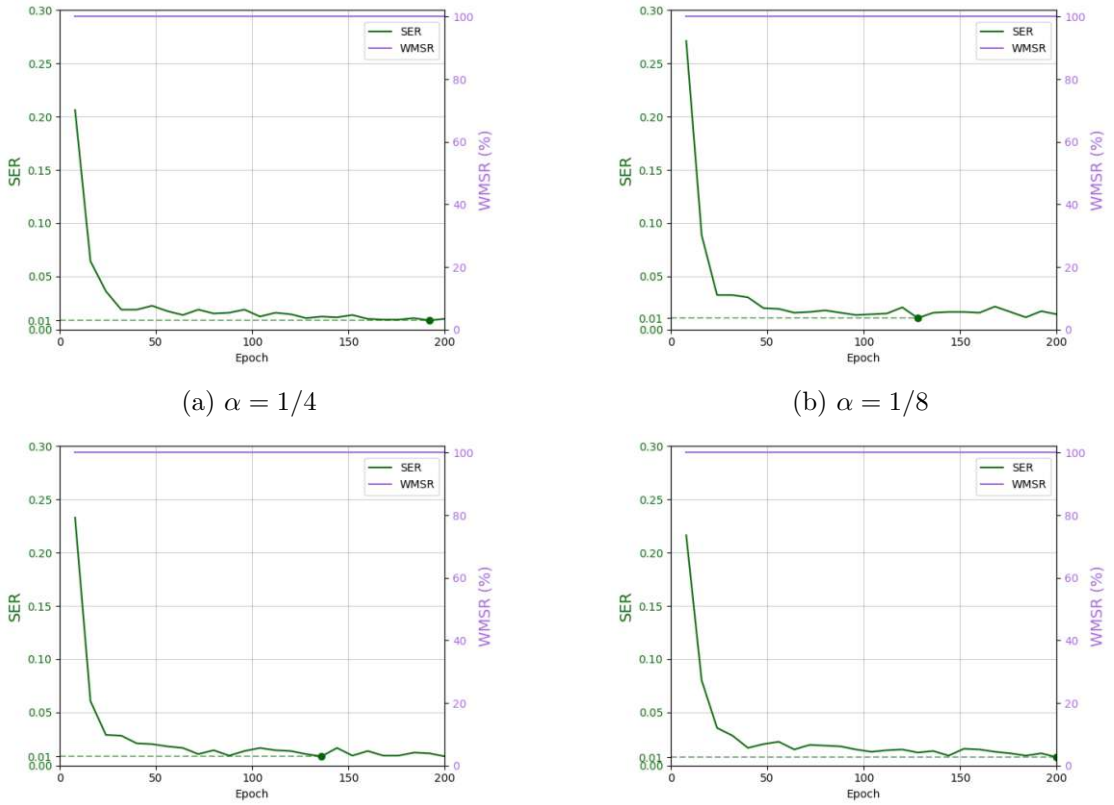
(a) $\alpha = 1/4$ (b) $\alpha = 1/8$

(c) $\alpha = 1/16$ (d) $\alpha = 1/32$

Figure 6.3: Effectiveness (WMSR) and fidelity (SER) of replicated Gaussian Noise watermark on SincNet model with different ratios of trigger set α.

6.1.4 Frequency Noise [LYS+24]

As with the Gaussian Noise watermark, the Frequency Noise watermark also achieves a perfect WMSR of 100%, as shown in Figure 6.4. The SincNet model reaches a SER of 0.01, indicating good model fidelity, comparable to the Gaussian Noise watermark. Thus, we consider the replication of the Frequency Noise watermark on SincNet successful, although it does not noticeably outperform the Gaussian Noise watermark.



(a) $\alpha = 1/4$ (b) $\alpha = 1/8$

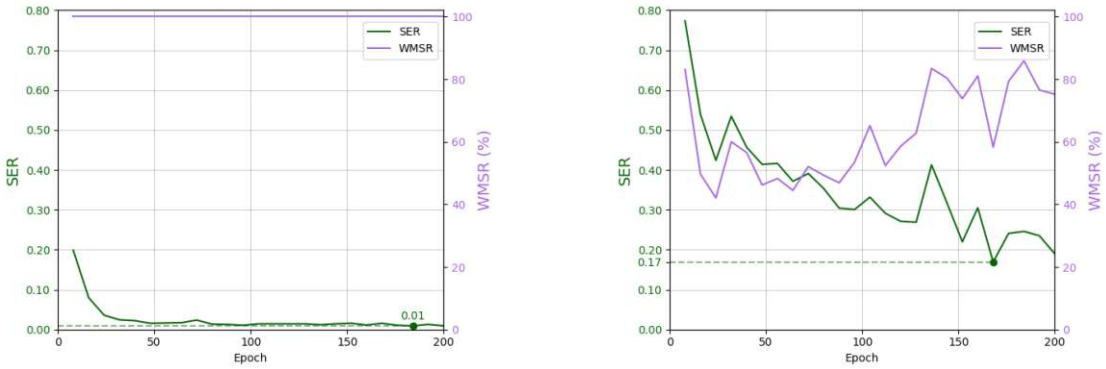
(c) $\alpha = 1/16$ (d) $\alpha = 1/32$

Figure 6.4: Effectiveness (WMSR) and fidelity (SER) of replicated Frequency Noise model with different ratios of trigger set α.

6.1.5 Unrelated Audio [LYS+24]

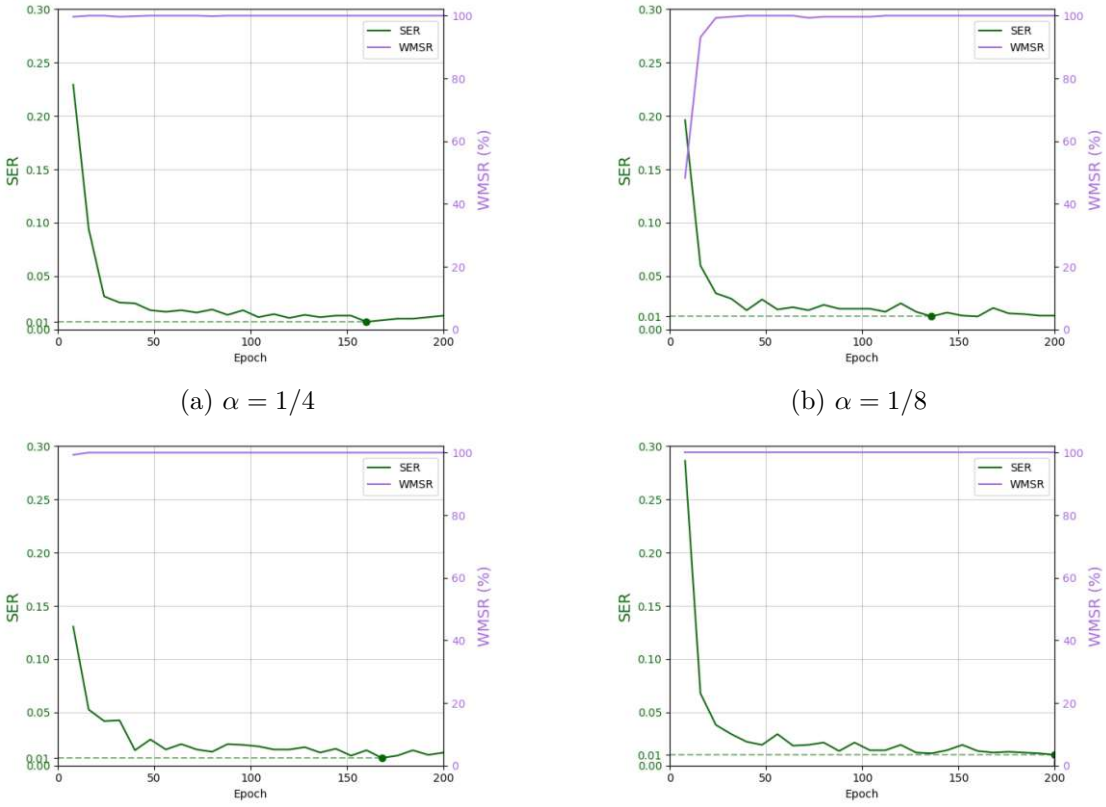
Firstly, we trained the SincNet model with trigger samples created from an unrelated audio with maximum (0.05863584) and minimum (0.0000009) noise variances, the results of which are shown in Figure 6.5. The former model (Figure 6.5a) reaches a perfect WMSR and SER of 0.01, while for the latter (Figure 6.5b), neither the SER nor WMSR reach acceptable results. The lowest point of SER is 0.17, which does not replicate the results reported by Liao et al. [LYS+24].

To follow the experiment design of Liao et al. [LYS+24] for replicability, we then randomly chose an unrelated audio file. The chosen sample has a noise variance of 0.00005785. We conducted experiments with different proportions of trigger samples using this audio sample for trigger creation. The results are shown in Figure 6.6. Regardless of the value of α, all models achieve 100% of WMSR and 0.01 SER at the end of the training, thereby successfully replicating the results.



(a) *noise_var* = 0.05863584 (b) *noise_var* = 0.0000009

Figure 6.5: Unrelated Audio model trained on triggers created from extremums.



(a) $\alpha = 1/4$ (b) $\alpha = 1/8$

(c) $\alpha = 1/16$ (d) $\alpha = 1/32$

Figure 6.6: Effectiveness (WMSR) and fidelity (SER) of replicated Unrelated Audio model with different ratios of trigger set α. Created using a random unrelated audio with noise variance = 0.00005785.

6.1.6 Replicability Findings

This section summarises the replication analysis results of three papers that proposed their black-box speaker recognition model watermarking methods. The comparison with the reported results of the respective papers is presented in Table 6.1.

Table 6.1: Replicability study of state-of-the-art Speaker Recognition model Watermarking methods, including reported results on effectiveness and fidelity from the original works and our replicated results. The results are obtained using the same dataset (TIMIT) and model (SincNet) as the original papers.

		Effectiveness (WMSR↑)		Fidelity (SER↓)	
Method	α	Reported	Replicated	Reported	Replicated
Segment-based frequency perturbation	1/165	95%	5%	0.006	0.002
Mel mid-frequency band based	1/16	100%	0.5%	0.010	0.110
Gaussian Noise	1/4	100%	100%	0.046	0.012
	1/8	100%	100%	0.026	0.010
	1/16	98.9%	100%	0.018	0.012
	1/32	96.6%	100%	0.010	0.010
Frequency Noise	1/4	100%	100%	0.045	0.009
	1/8	100%	100%	0.025	0.011
	1/16	99.8%	100%	0.020	0.009
	1/32	99.4%	100%	0.009	0.008
Unrelated Audio	1/4	100%	100%	0.044	0.009
	1/8	98.3%	100%	0.019	0.010
	1/16	96.2%	100%	0.017	0.008
	1/32	93.7%	100%	0.011	0.010

We were able to achieve a comparable SER with the Segment-based frequency perturbation method. Nonetheless, none of our experiments resulted in WMSR close to the reported value, and the results were thus not replicated.

For the Mel mid-frequency band-based, both WMSR and SER fell short compared to the reported results.

Since we were not able to replicate the results of the first two watermarks (segment-based frequency perturbation and Mel mid-frequency band-based), we do not proceed with them for our further experiments.

All three methods of Liao et al. [LYS+24] achieved perfect watermark success rates (WMSR) across the evaluated settings, exceeding the effectiveness reported by the original authors. In terms of the Unrelated Audio watermark, this can be caused by the choice of unrelated audio. The replicated fidelity closely matches the reported values for smaller trigger set sizes α, especially 1/32. However, our implementations did not exhibit the same rate of fidelity degradation with increasing α as reported. Instead, fidelity

remained relatively stable across trigger sizes, resulting in consistently higher fidelity overall. Interestingly, in most settings, the WMSR is 100% from the very beginning of training. This suggests potential overgeneralisation – the model quickly learns to associate the trigger label with almost any input, even in the absence of the trigger – which is something we would like to avoid; otherwise, the backdoor becomes too obvious and can be easily detected.

6.2 Generality

In this section, we present the results of evaluating the three watermarking schemes on other models and datasets. The default parameter settings used to create trigger sets for the replicated methods are shown in Table 6.2 (this table is an extension of the trigger set parameters presented in Section 5.3). Table 6.3 reports the sentence error rates of non-watermarked models, along with watermark effectiveness and model fidelity for each model-dataset pair we evaluate to assess the generality [Boe21] of the watermarking methods.

Table 6.2: Default parameters of watermarking methods from [LYS+24].

WM	λ	$mfcc\_sc$	ELF	EHF	$noise\_var$
Gaussian Noise	1.0				
Frequency Noise	1.0	2.0	300	3,000	
Unrelated Audio	1.0				0.000058

For TIMIT, the decrease in fidelity of MobileNet and AM-MobileNet relative to the original model performances is in the same order of magnitude as for SincNet (the SERs on the original task of non-watermarked models are noted in the first rows of each respective model in Table 6.3).

In some cases with the VoxCeleb1 dataset, a marginal model performance increase can be observed after applying the watermark, e.g., when using the Gaussian Noise watermark (for all models except ResNet34), the Frequency Noise watermark, or the Unrelated Audio watermark. We attribute this behaviour to the relatively poor baseline performance of the models: the SER exceeds 0.2 for ResNet18 and ResNet34, and is above 0.1 for AutoSpeech, which is significantly worse than the 0.01 SER achieved on TIMIT. Under such conditions, introducing a watermark may lead to unpredictable changes in fidelity, occasionally resulting in performance improvements or degradation. Among all methods, Unrelated Audio watermark is the only one that produced a WMSR below 100%, while the others consistently achieved a 100% watermark success rate.

Overall, all watermarking methods demonstrate applicability across the evaluated models and datasets, achieving high watermark effectiveness and maintaining model fidelity.

Table 6.3: Generality of SOTA watermarking methods: watermarks applied on settings with additional models and datasets. The results of watermarking the SincNet model repeat the ones stated in Table 6.1. Results show watermark effectiveness and model fidelity.

Dataset	Model	WM	Effectiveness WMSR↑	Fidelity SER↓
TIMIT	SincNet	-	-	0.0051
		Gaussian Noise	100%	0.0101
		Frequency Noise	100%	0.0108
		Unrelated Audio	100%	0.0101
	MobileNet	-	-	0.0057
		Gaussian Noise	100%	0.0065
		Frequency Noise	100%	0.0058
		Unrelated Audio	100%	0.0087
	AM-MobileNet	-	-	0.0043
		Gaussian Noise	100%	0.0036
		Frequency Noise	100%	0.0043
		Unrelated Audio	100%	0.0115
VoxCeleb1	ResNet18	-	-	0.2978
		Gaussian Noise	100%	0.2306
		Frequency Noise	100%	0.2485
		Unrelated Audio	100%	0.2859
	ResNet34	-	-	0.2206
		Gaussian Noise	100%	0.2396
		Frequency Noise	100%	0.2016
		Unrelated Audio	98%	0.2071
	AutoSpeech	-	-	0.1691
		Gaussian Noise	100%	0.1476
		Frequency Noise	100%	0.1298
		Unrelated Audio	100%	0.1515

6.3 Imperceptibility

This section presents the analysis of the imperceptibility of state-of-the-art watermarking methods. Furthermore, we investigate how these watermarks can be made more imperceptible in relation to watermark success rate and model fidelity – specifically, whether the imperceptibility requirements can be met without any degradation in WMSR or SER, or what trade-offs are involved otherwise.

Table 6.4 presents the measures of imperceptibility for the generated trigger sets. As

shown, the Unrelated Audio watermark performs best across both datasets, while Frequency Noise produces the most perceptible audio samples. Only the SNR of the Unrelated Audio watermark approaches the acceptable perceptibility threshold of ≥ 20 dB (cf. Section 4.1.2). Therefore, in this section, we present our experiments aimed at improving the imperceptibility of the watermarking methods.

Table 6.4: Imperceptibility of SOTA watermarking methods applied to TIMIT and VoxCeleb1 datasets.

Dataset	WM	Imperceptibility LSD↓	SNR↑
TIMIT	Gaussian Noise	13.0948	16.6514
	Frequency Noise	20.6664	-1.3025
	Unrelated Audio	3.8900	**23.0389**
VoxCeleb1	Gaussian Noise	13.0948	15.9199
	Frequency Noise	22.5900	-2.6448
	Unrelated Audio	4.7516	14.5356

For each dataset-model-watermark setting, we evaluate multiple watermark creation configurations of parameters. Subsequently, we select one configuration for each setting that is used for further analysis of legality and robustness in the next sections. The chosen configurations are highlighted in bold in Tables 6.5-6.22, and their corresponding metrics are restated in Table 6.24 to provide a comparative overview.

6.3.1 TIMIT

Here, we present the imperceptibility evaluation results for the TIMIT dataset carried out according to the setup described in Section 5.6 across three models: SincNet, MobileNet, and AM-MobileNet.

SincNet

Gaussian Noise Figure 6.7 shows the best results regarding SER and WMSR for trigger sets created using various values of the scale parameter λ, together with the corresponding imperceptibility metrics, Log Spectral Distortion (LSD) and Signal-to-Noise Ratio (SNR). It can be observed that none of the λ values result in an LSD below 2 dB. The Gaussian Noise watermark with $\lambda = 0.1$ that corresponds to the highest imperceptibility reaches only 0.075 SER compared to 0.0101 of the original Gaussian Noise (which corresponds to a scale factor of $\lambda = 1.0$). The two models that maintain the same model fidelity as the original one are with $\lambda = 0.6$ and $\lambda = 0.9$, presented in Table 6.5. With $\lambda \leq 0.6$, the SNR becomes higher than 20 dB. Hence, the optimal choice of λ in terms of watermark effectiveness, model fidelity, and watermark imperceptibility is 0.6. The cost of embedding this watermark compared to the clean model is 0.0043.

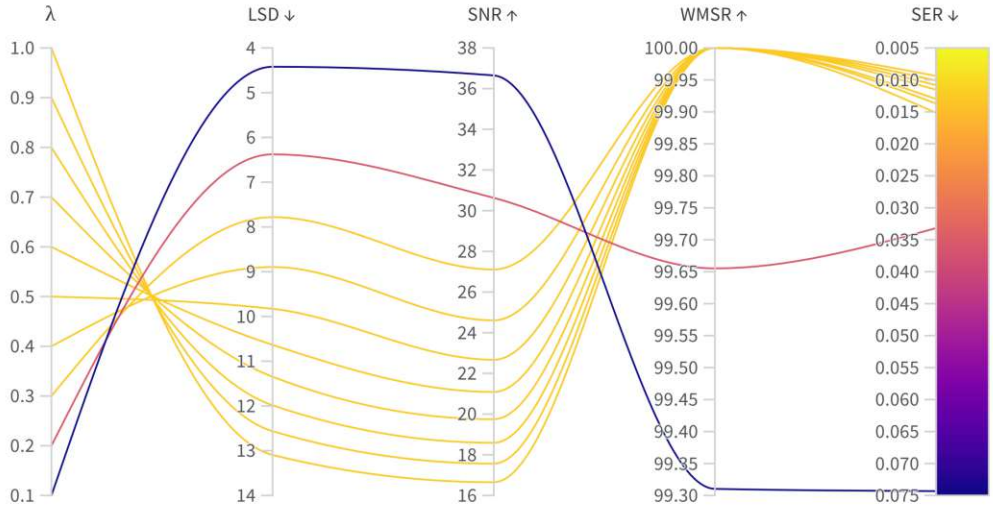


Figure 6.7: Imperceptibility analysis of Gaussian Noise watermark applied to TIMIT SincNet.

Table 6.5: Imperceptibility of Gaussian Noise watermark applied to TIMIT.

WM	λ	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	SNR↑
Gaussian Noise	1.0	100%	0.0101	13.0948	16.6514
	0.9	100%	0.0101	12.5637	17.5663
	0.6	**100%**	**0.0094**	**10.6347**	**21.0876**

Frequency Noise Following the results of the Gaussian Noise watermark, we first explored the different values of extreme frequencies with various MFCC scale factor $mfcc\_sc$ and the embedding scale parameter λ. The corresponding results of training the SincNet model on the TIMIT dataset can be seen in Figure 6.8. The imperceptibility analysis showed that with the same $mfcc\_sc$ and λ, LSD and SNR are almost the same for different extreme frequencies (Extreme Low Frequency (ELF) and Extreme High Frequency (EHF)). While $mfcc\_sc$ does affect the imperceptibility to some extent, λ has the biggest impact, since it regulates the intensity of final watermark embedding in the signal. Even though the watermark effectiveness is almost always 100%, we can observe a visible trade-off between watermark imperceptibility and model fidelity. Still, the SNR does not reach the 20 dB threshold.

Omitting the MFCC step from the trigger creation process leaves us with the following parameters: λ, extreme frequency cut-off values (ELF, EHF). Since our experiments showed that we can reach acceptable SER and WMSR with low λ, e.g., $\lambda \leq 0.2$, we

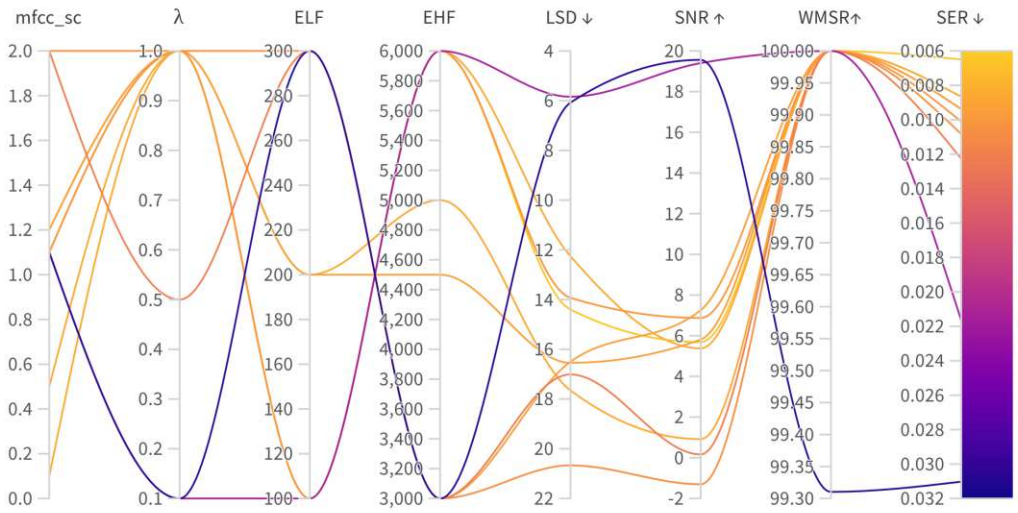


Figure 6.8: Imperceptibility analysis of Frequency Noise applied to TIMIT SincNet.

narrowed our hyperparameter search to this range.

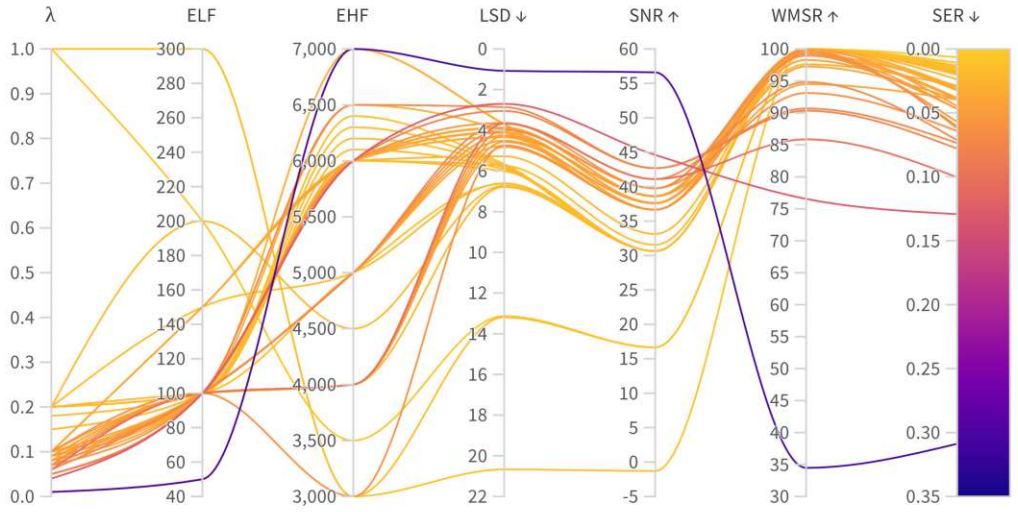


Figure 6.9: Imperceptibility analysis of Extreme Gaussian Noise applied to TIMIT SincNet.

As can be seen in Figure 6.9, most of the considered parameters, except two with $\lambda = 1.0$, result in an $SNR \geq 30$ dB, which fulfils the mathematical definition of watermark

imperceptibility. However, the LSD is still ≤ 7. The only trigger set that has $LSD = 1$ results in 0.3 SER, which is too high for the model that normally achieves a SER of 0.01. Hence, the choice of the trigger creation hyperparameters depends on achieving acceptable values of SER and WMSR.

The experiments depicted in Figure 6.9 that have an $SNR \geq 30$ dB have an $SER \geq 0.012$. The best configurations are presented in Table 6.6. The closest one to the values of the original Frequency Noise watermark is the one marked in bold with SER = 0.0129. Thus, embedding the default version of the Frequency Noise watermark increases the SER by 0.0057 compared to the clean model (SER=0.0051), whereas the more imperceptible watermark costs model fidelity of 0.0078, meaning that improving imperceptibility comes at the cost of additional fidelity loss of 0.0021.

Table 6.6: Imperceptibility of Extreme Gaussian Noise watermark applied to TIMIT.

WM	$mfcc\_sc$	λ	ELF	EHF	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	SNR↑
Frequency Noise	2.0	1.0	300	3000	100%	0.0108	20.6664	-1.3025
Extreme Gaussian Noise	**-**	**0.2**	**100**	**6000**	**100%**	**0.0129**	**6.0068**	**30.6359**
	-	0.2	100	6400	100%	0.0144	5.6605	30.6409

Unrelated Audio For the Unrelated Audio watermark, we focused on the audio samples with noise variance in the middle of the extremums, while also trying different λ values.

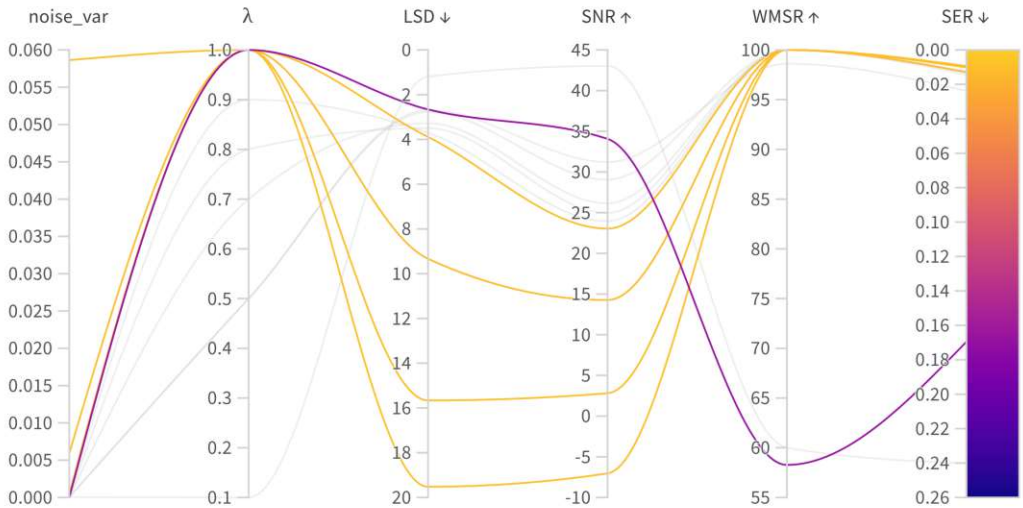


Figure 6.10: Imperceptibility analysis of Unrelated Audio watermark applied to TIMIT SincNet. Evaluation of unrelated audio samples with different noise variance, $\lambda = 1.0$.

Figure 6.10 shows the evaluated trigger sets created using audio samples with different noise variances (indicated by *noise_var*). We can see an extremely big difference in the imperceptibility measurements depending on the audio choice. Most of them achieved a 100% WMSR and SER ≤ 0.02. The specific noise variances that were used are presented in Table 6.7.

Table 6.7: Imperceptibility of Unrelated Audio watermark applied to TIMIT SincNet.

WM	*noise_var*	λ	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	SNR↑
Unrelated Audio	0.0586358	1.0	100%	0.0093	19.5214	-7.0421
	0.0059003		100%	0.0101	15.6635	2.8084
	0.0004316		100%	0.0129	9.3341	14.2619
	0.0000578		**100%**	**0.0101**	**3.8900**	**23.0388**
	0.0000368		98.62%	0.0238	2.6854	31.2193
	0.0000009		58.28%	0.1695	2.6612	34.0712

Among these unrelated audio samples, we chose the one with the highest imperceptibility that still achieves good WMSR and SER: 0.00005785. Its SNR is ≥ 20 dB, but we still investigate various values for λ to see if we can improve the SNR. Figure 6.11 depicts these results.

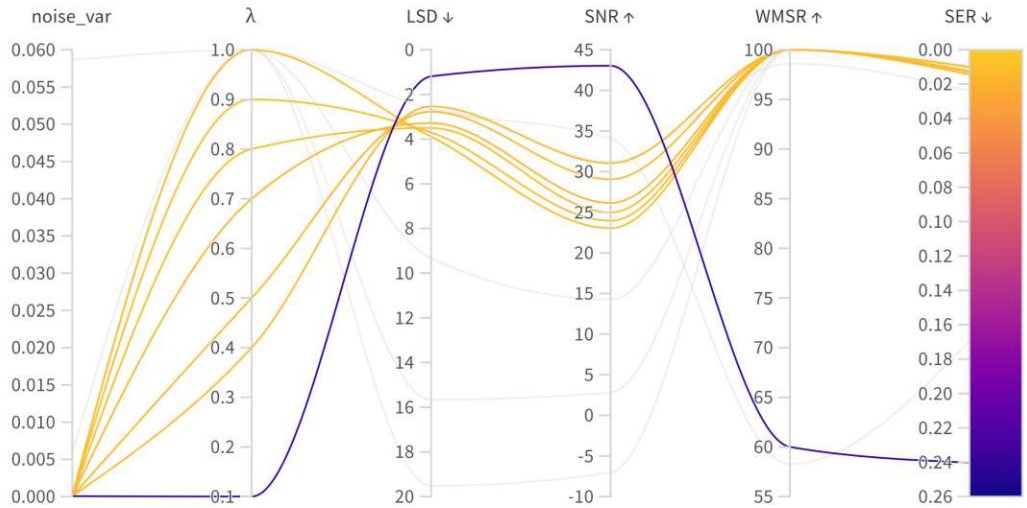


Figure 6.11: Imperceptibility analysis of Unrelated Audio watermark applied to TIMIT SincNet. Unrelated audio sample: *noise_var* = 0.00005785, different values of λ.

Table 6.8 summarises the imperceptibility analysis of the runs with the best achieved SER.

Among them, $\lambda = 0.8$ yields the same SER as $\lambda = 1.0$, but with higher imperceptibility. This suggests that, in this setting, we can use a more imperceptible watermark without loss of either model fidelity or watermark effectiveness.

Table 6.8: Imperceptibility of Unrelated Audio watermark with $noise\_var = 0.0000578$ and different λ applied to TIMIT SincNet.

WM	*noise_var*	λ	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	SNR↑
Unrelated Audio	0.0000578	1.0	100%	0.0101	3.8900	23.0388
		0.9		0.0101	3.6990	23.9539
		0.8		**0.0101**	**3.4947**	**24.9770**
		0.5		0.0123	2.7731	29.05929

MobileNet

Gaussian Noise The lowest value of λ (meaning the highest imperceptibility) that resulted in a SER close to the original Gaussian Noise watermark is 0.2. Since $\lambda = 0.1$ resulted in a twice as big SER, while $\lambda = 0.2$ was within the threshold, we also tried $\lambda = 0.15$, but as can be seen in Figure 6.13, it also fell short in SER. In Table 6.9, one can see the detailed comparison of these runs.

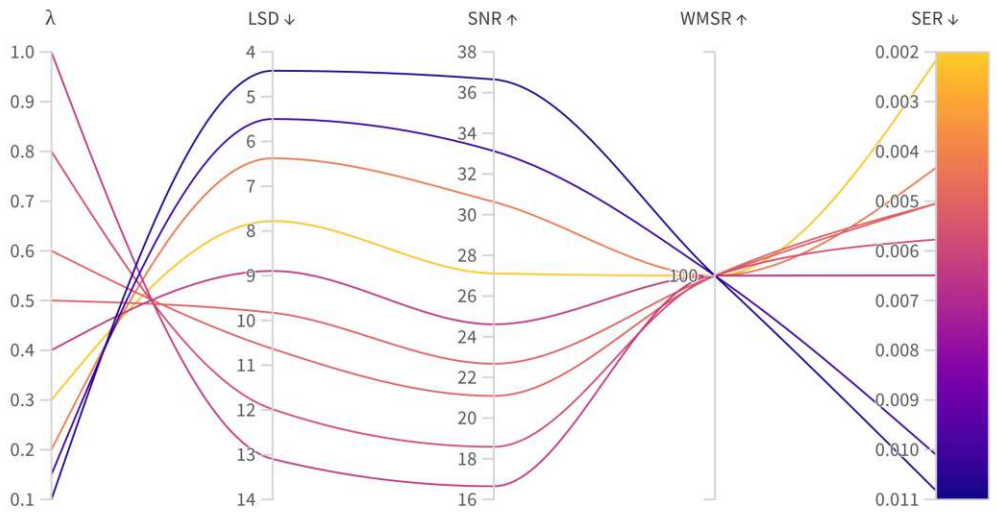


Figure 6.12: Imperceptibility analysis of Gaussian Noise applied to TIMIT MobileNet.

Table 6.9: Imperceptibility of Gaussian Noise watermark applied to TIMIT MobileNet.

WM	λ	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	SNR↑
Gaussian Noise	1.0	100%	0.0065	13.0948	16.6514
	0.3	100%	0.0022	7.7801	27.1078
	0.2	**100%**	**0.0043**	**6.3771**	**30.6293**
	0.15	100%	0.0101	5.4986	33.1279

Frequency Noise Following the insights from the SincNet model evaluation, we analysed only two Frequency Noise watermarks with $mfcc\_sc = 1.1$ and $\lambda = 1.0$ and 0.1. Other experiments were conducted using the Extreme Gaussian Noise watermark (in Figure 6.13, these are the ones that have $mfcc\_sc$ set to 0.0).

As can be seen in Figure 6.13, the original Frequency Noise watermark (with $mfcc\_sc = 2.0$ and $lambda = 1.0$) has the worst imperceptibility – all modifications of the watermark result in better values for LSD and SNR.

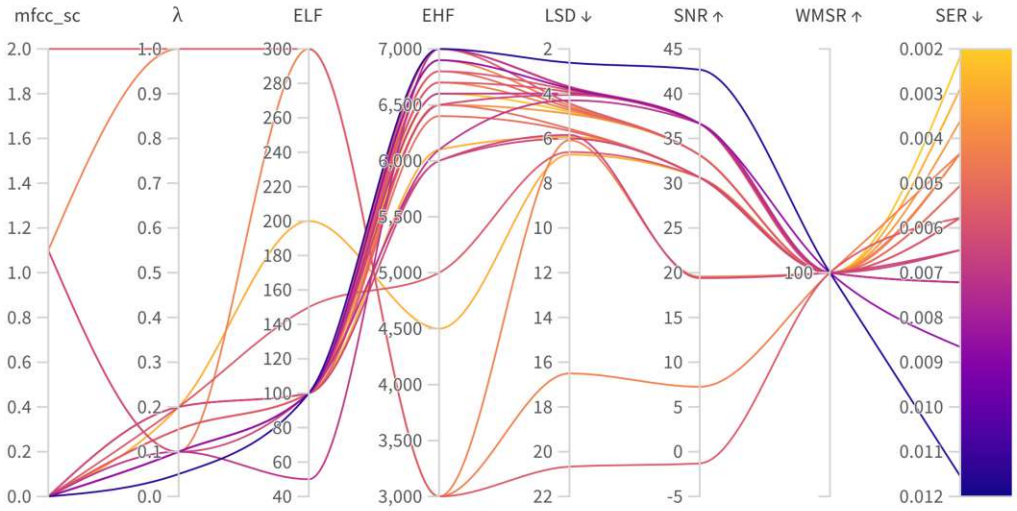


Figure 6.13: Imperceptibility analysis of Frequency and Extreme Gaussian Noise applied to TIMIT MobileNet.

In Table 6.10, we show the five best imperceptibility results. Two of the configurations that have a very similar imperceptibility level yield the same SER as the original Frequency Noise watermark. For further analysis, we chose the one with a better value of LSD, since they both fulfill the SNR requirement.

Table 6.10: Imperceptibility of Frequency and Extreme Gaussian Noise watermark applied to TIMIT MobileNet.

WM	$mfcc\_sc$	λ	ELF	EHF	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	SNR↑
FN	2.0	1.0	300	3,000	100%	0.0058	20.6664	-1.3025
	-	0.05	100	7,000	100%	0.0115	2.6175	42.6875
	-	0.1	100	6,800	100%	0.0058	3.8452	36.6697
EGN	-	0.1	100	6,900	100%	0.0087	3.7658	36.6689
	-	**0.1**	**100**	**7,000**	**100%**	**0.0058**	**3.6815**	**36.6688**
	-	0.1	50	7,000	100%	0.0072	3.7032	36.6682

We observe from the imperceptibility analysis that λ affects the imperceptibility more drastically than the extreme frequency limits. For example, comparing the parameters $ELF = 100$ and $EHF = 7,000$ with $\lambda = 0.1$ and $\lambda = 0.05$, the LSD of the watermark created with $\lambda = 0.05$ is smaller by 1.07 and the SNR is bigger by 6 dB, while lowering ELF to 50 results in nearly the same LSD and SNR values.

Unrelated Audio Following the results of imperceptibility analysis on the SincNet model, we evaluated two unrelated audio samples with noise variances of 0.00003688 and 0.00005785 on different values of λ. Figure 6.14 shows the results of these runs, and Table 6.11 summarises the watermark parameters that lead to the best model fidelity.

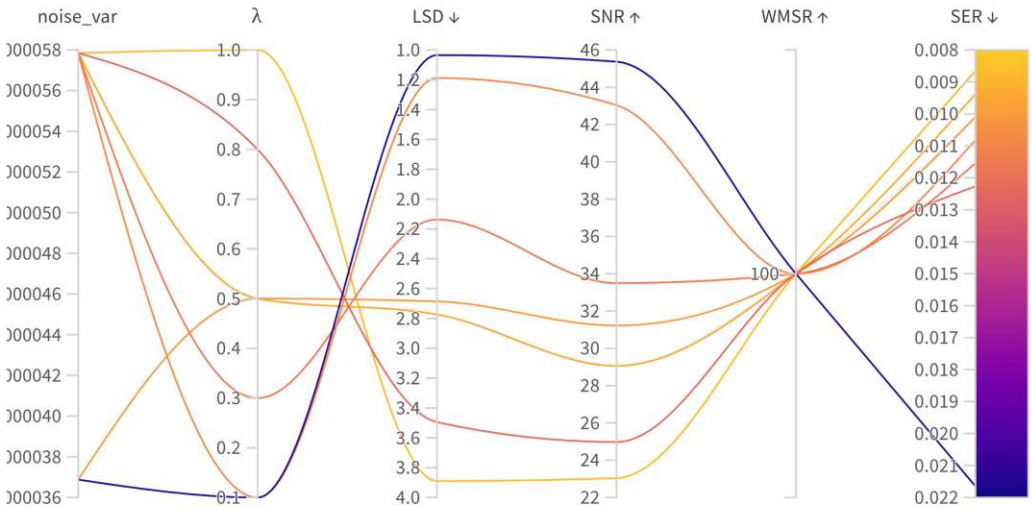


Figure 6.14: Imperceptibility analysis of Unrelated Audio watermark applied to TIMIT MobileNet.

Table 6.11: Imperceptibility of Unrelated Audio watermark applied to TIMIT MobileNet.

WM	*noise_var*	λ	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	Imperceptibility SNR↑
Unrelated Audio	0.0000578	1.0	100%	0.0087	3.8900	23.0388
	0.0000578	**0.5**		**0.0094**	**2.7731**	**29.0592**
	0.0000369	0.5		0.0101	2.6854	31.2192
	0.0000369	*0.1*		*0.0108*	*1.1882*	*43.0357*

All the configurations yielded a SER higher than with the default Unrelated Audio watermark parameters (*noise_var* = 0.00005785 and $\lambda = 1.0$). The cost in fidelity of higher imperceptibility is at least 0.0007, or 0.0019, when aiming to achieve both an LSD and SNR within the defined accepted boundaries. Compared to the non-watermarked MobileNet model (SER=0.0057), the cost of watermarking is 0.0037 and 0.0049, respectively.

AM-MobileNet

Gaussian Noise Figure 6.15 shows the evaluation of AM-MobileNet watermarked with a Gaussian Noise watermark created with different values of parameter λ. The resulting SERs of $\lambda = 1.0$ and $\lambda = 0.1$ are the same and are the best compared to all others. We

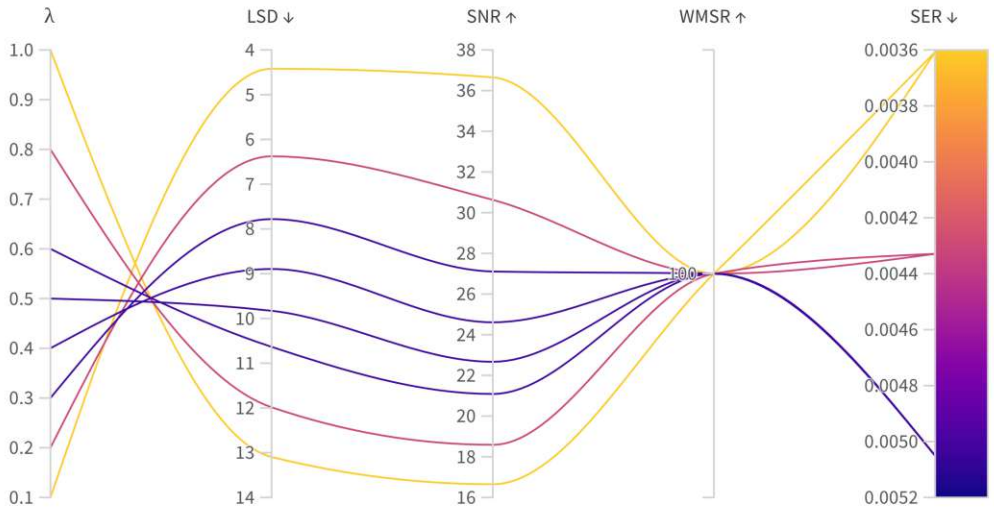


Figure 6.15: Imperceptibility analysis of Gaussian Noise applied to TIMIT AM-MobileNet.

do not measure how statistically true that is, but since it is possible to achieve the same level of fidelity with a more imperceptible watermark, we take the watermark created

with parameter $\lambda = 0.1$ as the most imperceptible watermark with which AM-MobileNet can be trained without losing any model fidelity.

Frequency Noise Based on the results of MobileNet, we first evaluated only the Extreme Gaussian Noise watermark with different hyperparameters (see Figure 6.16). Since none of these configurations preserved the original model fidelity, we also trained a model with a few options of Frequency Noise watermark. These configurations improved SER compared to the Extreme Gaussian Noise watermark, but still did not reach the performance reported for the original Frequency Noise watermark. Furthermore, the Frequency Noise watermark on MobileNet did not yield SNR in acceptable imperceptibility bounds. Therefore, we chose the Extreme Gaussian Noise watermark with the highest SER and SNR values, namely the parameters highlighted in Table 6.12, which results in 0.0008 fidelity cost compared to the clean model (SER=0.0057).

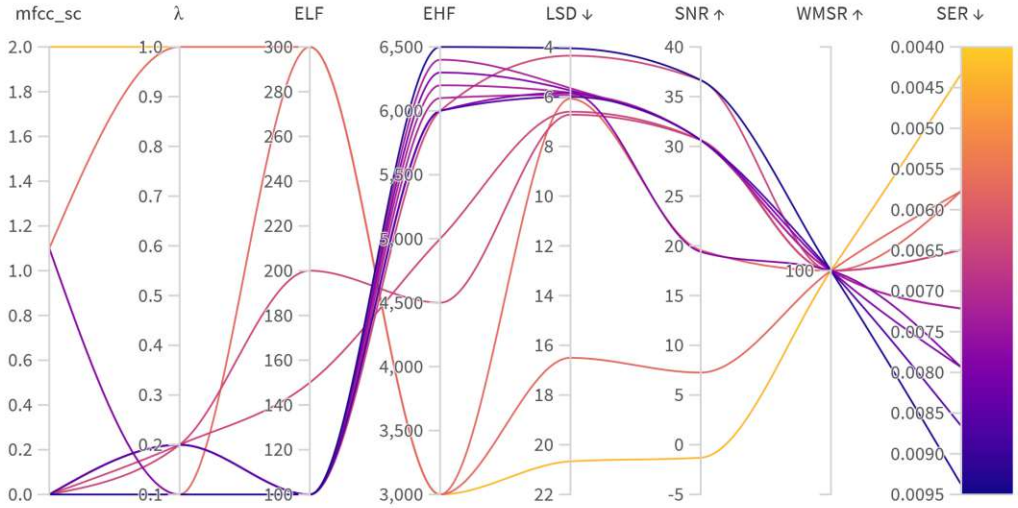


Figure 6.16: Imperceptibility analysis of Frequency and Extreme Gaussian Noise applied to TIMIT AM-MobileNet.

Table 6.12: Imperceptibility of Frequency and Extreme Gaussian Noise watermark applied to TIMIT AM-MobileNet.

WM	$mfcc\_sc$	λ	ELF	EHF	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	SNR↑
FN	2.0	1.0	300	3,000		0.0043	20.6664	-1.3025
FN	1.1	1.0	300	3,000		0.0058	16.5019	7.2560
FN	1.1	0.1	300	3,000	100%	0.0058	6.0815	19.5632
EGN	**-**	**0.1**	**100**	**6,000**		**0.0065**	**4.3513**	**36.6559**
EGN	-	0.2	150	5,000		0.0065	6.6046	30.6361

Unrelated Audio As can be seen in Figure 6.17, way more settings resulted in higher SER on AM-Mobilenet compared to MobileNet watermarked with Unrelated Audio. In Table 6.13, we show the results of individual experiments. None of them excel in model fidelity compared to the *noise_var* = 0.0000578 and $\lambda = 1.0$ setting. However, there is a setting that creates a watermark with SNR $\geq$ 30 dB and LSD $\leq$ 2 dB that costs 0.0022 in SER compared to our original choice of unrelated audio sample and 0.0094 compared to the non-watermarked AM-MobileNet model (SER=0.0043). The watermark leading to the closest value of SER compared to the default watermark is the one created with $\lambda = 0.8$. It results in a cost of model fidelity of 0.0014 and differs in SNR by 1.9382.

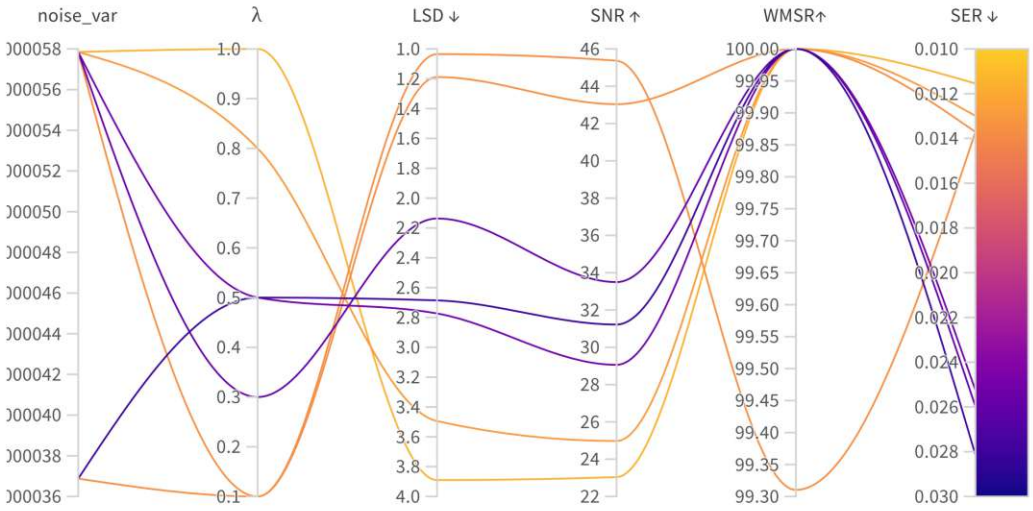


Figure 6.17: Imperceptibility analysis of Unrelated Audio watermark applied to TIMIT AM-MobileNet.

Table 6.13: Imperceptibility of Unrelated Audio watermark applied to TIMIT AM-MobileNet.

WM	*noise_var*	λ	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	SNR↑
Unrelated Audio	0.0000578	1.0	100%	0.0115	3.8900	23.0388
	0.0000578	**0.8**	**100%**	**0.0129**	**3.4947**	**24.9770**
	0.0000369	0.1	99.31%	0.0137	1.0344	45.3719
	0.0000578	*0.1*	*100%*	*0.0137*	*1.1882*	*43.0357*

6.3.2 VoxCeleb1

In the following, we present the imperceptibility evaluation results for the VoxCeleb1 dataset across three models: ResNet18, ResNet34, and AutoSpeech. Since both the ResNet and AutoSpeech models require twice to three times more resources to train compared to SincNet, we narrowed down the hyperparameter search with regard to the results obtained based on the TIMIT dataset.

ResNet18

Gaussian Noise For the Gaussian Noise watermark, we investigated values for $\lambda \leq 0.5$. Figure 6.18 shows these results. With $\lambda = 0.1$, the watermark success rate slightly deteriorates, while other values yield SER close to the results from $\lambda = 1.0$. The triggers created using $\lambda = 0.5$ meet the basic imperceptibility requirement, namely, SNR ≥ 20 dB; and the watermarked ResNet18 results in better model fidelity than the original $\lambda = 1.0$. In Table 6.14 we show the models that achieved the best model fidelity with the Gaussian Noise watermark.

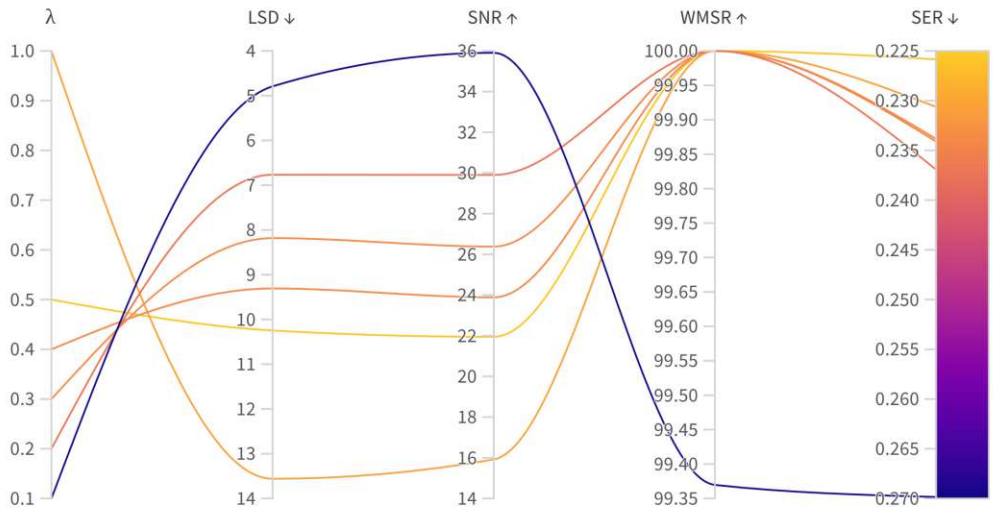


Figure 6.18: Imperceptibility analysis of Gaussian Noise applied to VoxCeleb1 ResNet18.

Table 6.14: Imperceptibility of Gaussian Noise watermark applied to VoxCeleb1 ResNet18.

WM	λ	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	SNR↑
Gaussian Noise	1.0	100%	0.2306	13.55936	15.9199
	0.5	**100%**	**0.2258**	**10.2465**	**21.9467**
	0.3	100%	0.2338	8.1826	26.3834

Frequency Noise With the VoxCeleb1 dataset, we first analysed the triggers created with different parameters of the Frequency Noise watermarking method. As shown in Figure 6.19, some of these models achieved SER lower than the sentence error rate obtained using the original Frequency Noise watermark parameters. However, none of the watermarks reached 20 dB in SNR. Hence, we proceeded with the Extreme Gaussian Noise watermarking method, experiments of which are depicted in Figure 6.20.

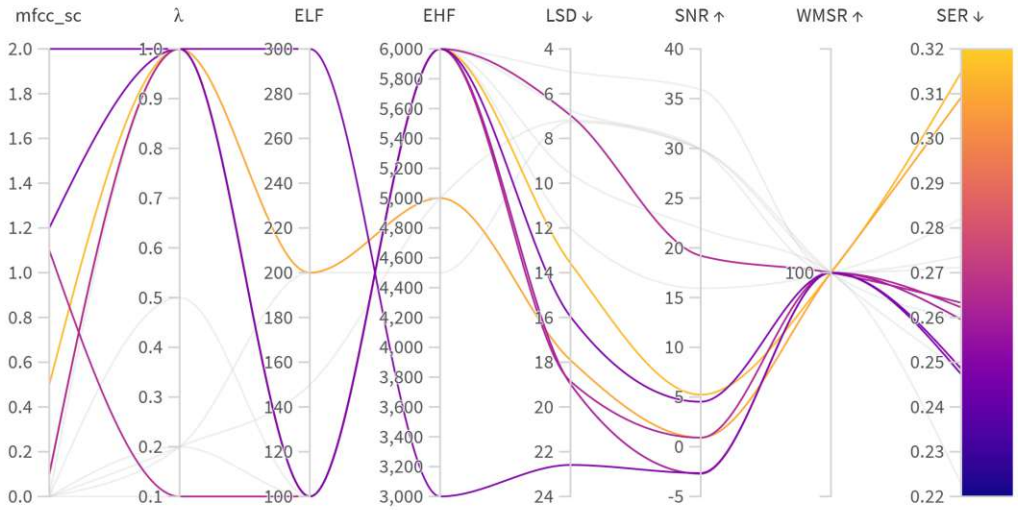


Figure 6.19: Imperceptibility analysis of Frequency Noise applied to VoxCeleb1 ResNet18.

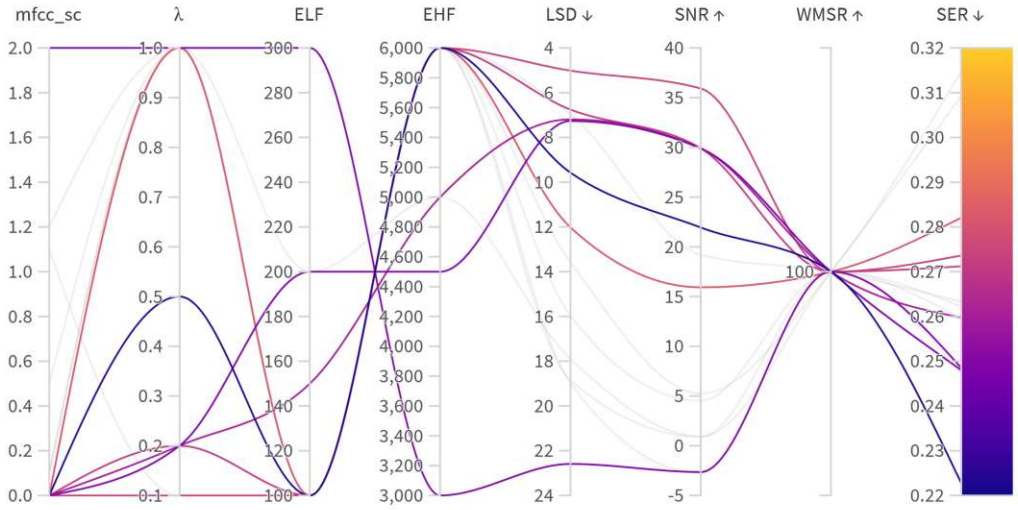


Figure 6.20: Imperceptibility analysis of Extreme Gaussian Noise applied to VoxCeleb1 ResNet18.

We can observe that SERs of Extreme Gaussian Noise models are generally lower than Frequency Noise models, though some still yield satisfactory results while having higher imperceptibility. Table 6.15 summarises the best results in terms of imperceptibility and model fidelity for the models watermarked with Frequency Noise and Extreme Gaussian Noise methods. The optimal parameters are highlighted in bold.

Table 6.15: Imperceptibility of Frequency and Extreme Gaussian Noise watermark applied to VoxCeleb1 ResNet18.

WM	$mfcc\_sc$	λ	ELF	EHF	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	SNR↑
FN	2.0	1.0	300	3,000	100%	0.2485	22.5900	-2.6448
EGN	-	0.5	100	6,000	100%	0.2228	9.5731	21.9465
FN	1.2	1.0	100	6,000	100%	0.2473	15.9588	4.5292
EGN	**-**	**0.2**	**200**	**4,500**	**100%**	**0.2480**	**7.2651**	**29.9050**

Two Extreme Gaussian Noise parameter settings yield a SER better than the original one, and both have an SNR $\geq$ 20 dB. Since here we focus on the level of imperceptibility, we choose the second option.

Unrelated Audio For the Unrelated Audio watermark, we evaluated three different audio samples with different values of λ. As can be seen in Figure 6.21, only three combinations yielded 100% of watermark effectiveness, the exact values of which are

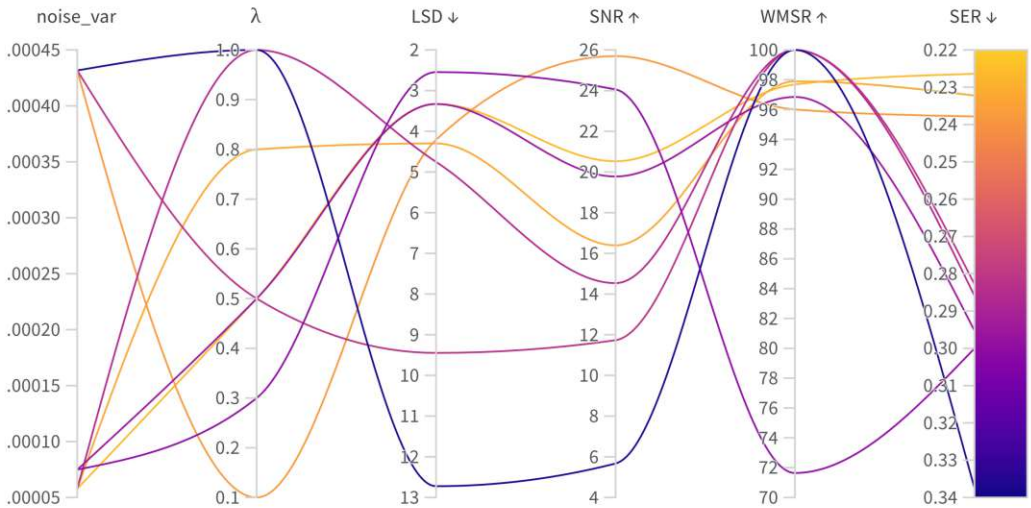


Figure 6.21: Imperceptibility analysis of Unrelated Audio watermark applied to VoxCeleb1 ResNet18.

detailed in Table 6.16.

Table 6.16: Imperceptibility of Unrelated Audio watermark applied to VoxCeleb1 ResNet18.

WM	*noise_var*	λ	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	SNR↑
Unrelated Audio	**0.0000578**	**1.0**	100%	**0.2859**	**4.7516**	**14.5356**
	0.000432	0.5		0.2828	9.4467	11.7309
	0.000432	1.0		0.3374	12.7222	5.6682

The watermark created using the audio sample with noise variance of 0.0000578 and $\lambda = 1.0$ yields the most imperceptible watermark based on our evaluation. However, neither its SNR nor LSD is within the threshold of a good imperceptibility measure when applied to the VoxCeleb1 dataset.

ResNet34

Gaussian Noise Watermarking ResNet34 with a Gaussian Noise watermark using different values of λ resulted in a SER higher than the original one in all cases (Figure 6.22). However, the WMSR with $\lambda = 0.1$ was lower than others (as it was already for ResNet18). Therefore, $\lambda = 0.1$ was excluded, and the choice for the most imperceptible watermark was made in favour of $\lambda = 0.2$. Table 6.17 summarises the imperceptibility measures of these watermarks in more detail.

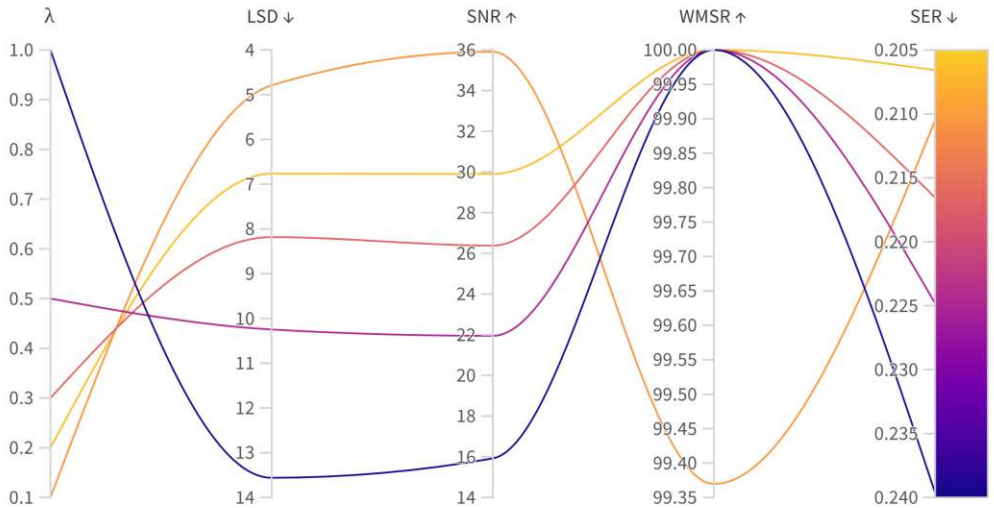


Figure 6.22: Imperceptibility analysis of Gaussian Noise applied to VoxCeleb1 ResNet34.

Table 6.17: Imperceptibility of Gaussian Noise watermark applied to VoxCeleb1 ResNet34.

WM	λ	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	Imperceptibility SNR↑
Gaussian Noise	1.0	100%	0.2396	13.55936	15.9199
	0.2	**100%**	**0.2066**	**6.7661**	**29.9027**
	0.1	99.37%	0.2105	4.7912	35.9091

Frequency Noise For ResNet34 with Frequency Noise watermark, we tried several parameter settings with MFCC. However, none of them resulted in satisfactory SER or imperceptibility measures. Thus, most of the evaluation was done using the Extreme Gaussian Noise watermark.

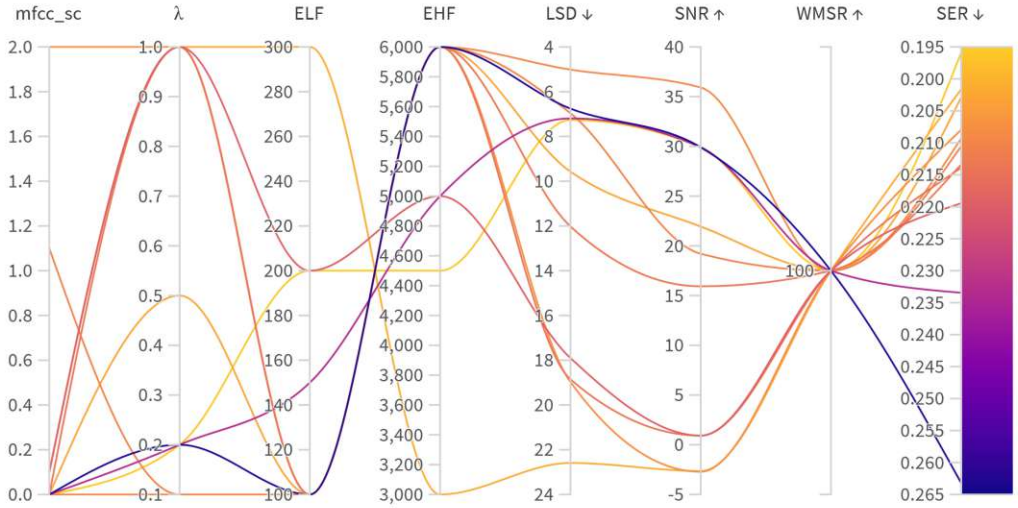


Figure 6.23: Imperceptibility analysis of Frequency and Extreme Gaussian Noise applied to VoxCeleb1 ResNet34.

Table 6.18: Imperceptibility of Frequency and Extreme Gaussian Noise watermark applied to VoxCeleb1 ResNet34.

WM	$mfcc\_sc$	λ	ELF	EHF	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	Imperceptibility SNR↑
FN	2.0	1.0	300	3,000	100%	0.2016	22.5900	-2.6448
EGN	**-**	**0.2**	**200**	**4,500**	**100%**	**0.1959**	**7.2651**	**29.9050**
EGN	-	0.5	100	6,000	100%	0.2029	9.5731	21.9465

Figure 6.23 shows the results of all examined parameter settings of the Frequency and Extreme Gaussian Noise watermarks, whereas Table 6.18 summarises the best results. Only one parameter setting yielded a better SER than the default Frequency Noise watermark while also fulfilling the imperceptibility requirement; this result is highlighted in the table.

Unrelated Audio Since watermarking ResNet34 with the audio sample, which we used to create trigger sets for other model-dataset pairs, did not result in 100% WMSR, we tried to find a combination of an unrelated audio sample and parameter λ that does. As shown in Figure 6.24, only two combinations (the audio sample with noise variance of 0.000432 paired with λ=0.5 and $\lambda = 1.0$) yielded a 100% watermark success rate; however, both of them also worsened the fidelity of the model.

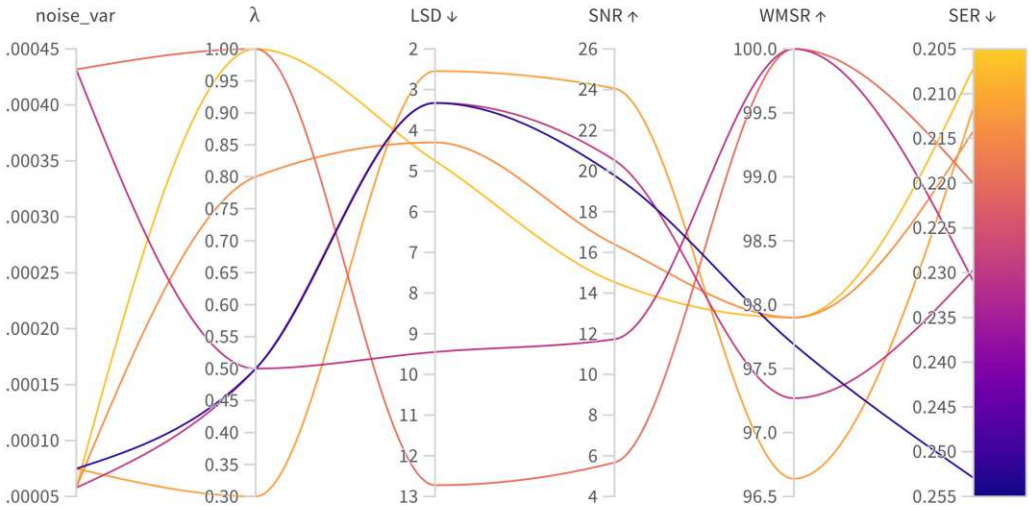


Figure 6.24: Imperceptibility analysis of Unrelated Audio watermark applied to VoxCeleb1 ResNet34.

Table 6.19: Imperceptibility of Unrelated Audio watermark applied to VoxCeleb1 ResNet34.

WM	*noise_var*	λ	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	SNR↑
Unrelated Audio	0.0000578	1.0	97.89%	0.2071	4.7516	14.5356
	0.000432	**1.0**	**100%**	**0.2201**	**12.7222**	**5.6682**
	0.000432	0.5	100%	0.2309	9.4467	11.7309

In Table 6.19 we present the comparative results of ResNet34 watermarked with the Unrelated Audio watermarking method. The setting that achieves a 100% WMSR results in a relative drop in fidelity of 0.013, but it does not exceed the SER of the non-watermarked ResNet34 model. Nonetheless, this setting does not fulfil any of the imperceptibility requirements, with an LSD = 12.7222 and SNR = 5.6682. With a cost of 0.0108 in fidelity, the imperceptibility can improve to 9.4467 and 11.7309, respectively, which still do not fit the required imperceptibility bounds.

AutoSpeech

Gaussian Noise The same as with the ResNet18 and ResNet34, triggers created with $\lambda = 0.1$ resulted in WMSR lower than 100%. Additionally, Figure 6.25 shows that none of the values of λ improved the SER compared to the setting when $\lambda = 1.0$; the closest is with $\lambda = 0.2$. This setting produces a watermark that fulfils one of the requirements of imperceptibility, namely SNR $\geq$ 20 dB, and results in 0.0014 fidelity loss as shown in Table 6.20.

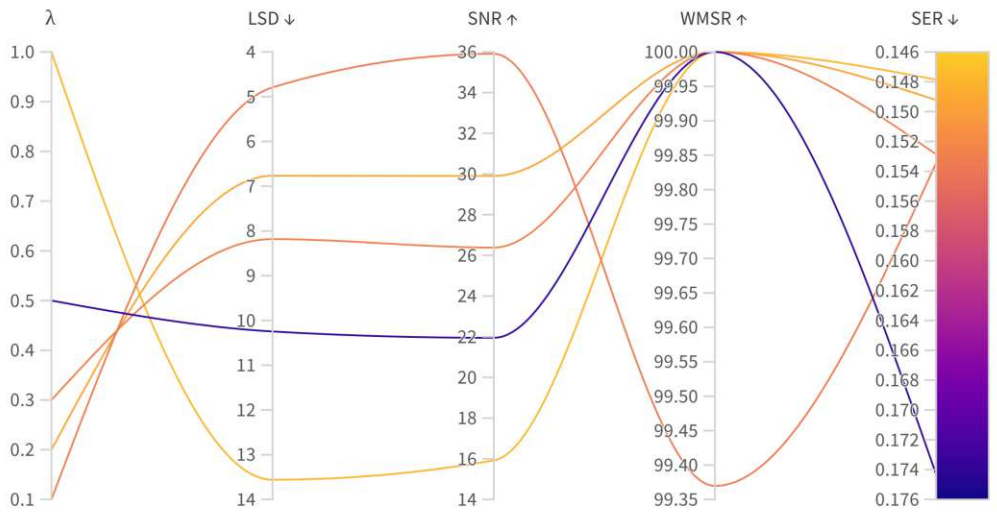


Figure 6.25: Imperceptibility analysis of Gaussian Noise applied to VoxCeleb1 AutoSpeech.

Table 6.20: Imperceptibility of Gaussian Noise watermark applied to VoxCeleb1 AutoSpeech.

WM	λ	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	SNR↑
Gaussian Noise	1.0	100%	0.1478	13.55936	15.9199
	0.2	**100%**	**0.1492**	**6.7661**	**29.9027**

Frequency Noise Following the results from the ResNet18 and ResNet34 models, we evaluated the Extreme Gaussian Noise watermark. The results can be seen in Figure 6.26. Since all of them yielded worse SER values than AutoSpeech watermarked with the default Frequency Noise parameters, we also watermarked the AutoSpeech model with a few settings of the Frequency Noise watermark. However, they also did not exceed the SER of the original Frequency Noise watermark parameters. Therefore, for further evaluation, we selected a model that resulted in the smallest model fidelity loss compared to the Frequency Noise original watermark, namely 0.0102 (Table 6.21).

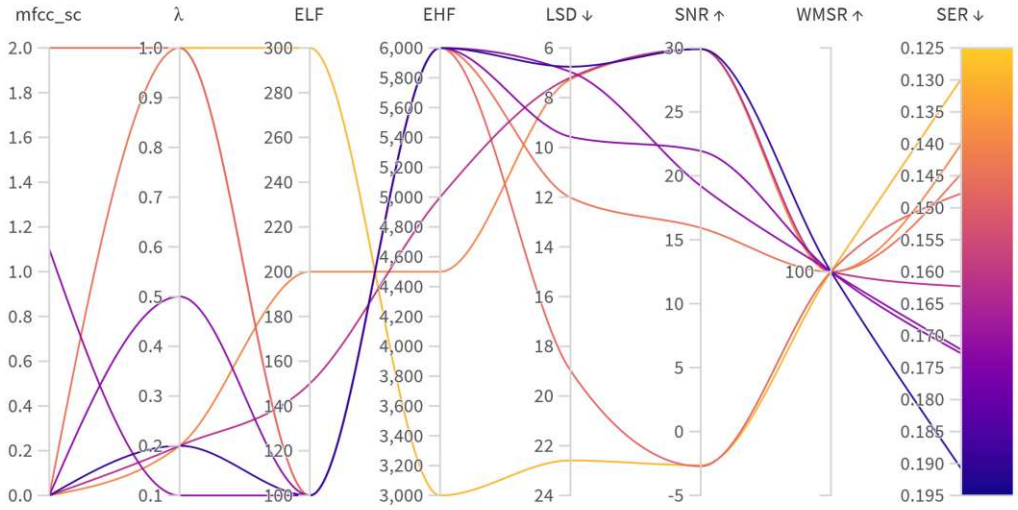


Figure 6.26: Imperceptibility analysis of Frequency Noise applied to VoxCeleb1 AutoSpeech.

Table 6.21: Imperceptibility of Frequency and Extreme Gaussian Noise watermark applied to VoxCeleb1 AutoSpeech.

WM	$mfcc\_sc$	λ	ELF	EHF	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	SNR↑
FN	2.0	1.0	300	3,000	100%	0.1298	22.5900	-2.6448
EGN	**-**	**0.2**	**200**	**4,500**	**100%**	**0.1400**	**7.2651**	**29.9050**
EGN	-	1.0	100	6,000	100%	0.1447	12.0189	15.9296

Unrelated Audio For AutoSpeech, we also see that only a minority of parameter configurations resulted in 100% WMSR; hence, we consider only those. Figure 6.27 visualises the results of our evaluated experiments, and Table 6.22 summarises only the ones that achieved a 100% watermark success rate.

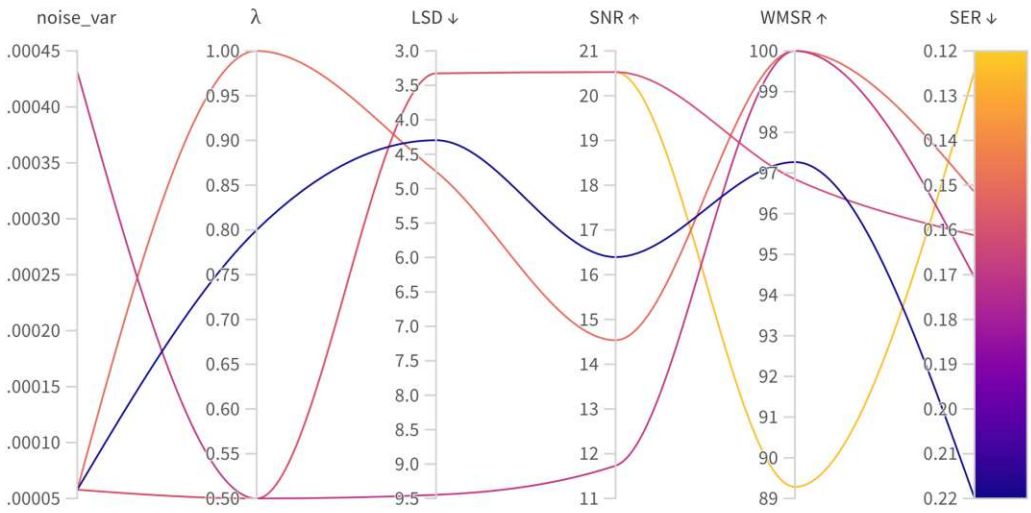


Figure 6.27: Imperceptibility analysis of Unrelated Audio watermark applied to VoxCeleb1 AutoSpeech.

Table 6.22: Imperceptibility of Unrelated Audio watermark applied to VoxCeleb1 AutoSpeech.

WM	*noise_var*	λ	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	SNR↑
Unrelated Audio	**0.0000578**	**1.0**	100%	**0.1515**	**4.7516**	**14.5356**
	0.000432	0.5		0.1706	9.4467	11.7309

6.3.3 Imperceptibility Evaluation Findings

Based on the discussed individual results of each dataset-model-watermark configuration, here, we present the main insights of our imperceptibility analysis.

Table 6.24 summarises the results of the conducted analysis. It contains the selected values described earlier in this section with regard to each dataset-model-watermark configuration. The values underneath the model names are SER of the non-watermarked models. The first row for each watermark contains the performance metrics of default parameter settings provided by Liao et. al. [LYS+24]. In case of the Unrelated Audio watermark, it is our default chosen audio sample with noise variance of 0.00005785 and $\lambda = 1.0$. The second line contains the performance results of the most imperceptible setting identified in this section; its name starts with the prefix 'I' (e.g., Gaussian Noise Model (GNM) versus Imperceptible Gaussian Noise Model (IGNM)). The values in bold are imperceptibility measures that fulfil the minimal requirements of imperceptibility (SNR $\geq$ 20 dB or LSD $\leq$ 2 dB). We highlight in green the settings with which we were

able to successfully improve imperceptibility measures, in red those that resulted in any fidelity loss compared to the clean model, and in orange the settings where the model fidelity drops compared to the default setting of watermark parameters but does not exceed the SER of the non-watermarked model. Additionally, we included, in italics, the setting when a model can fulfil both LSD and SNR requirements of imperceptibility with a higher cost of fidelity. In Table 6.23, we list the trigger creation parameters for the more imperceptible variants of the watermarking methods.

Table 6.23: Parameter settings for a more imperceptible variant of the watermarking methods.

Dataset	Model	WM	λ	ELF	EHF	*noise_var*
TIMIT	SincNet	IGNM	0.6	100	6,000	0.000058
		IEGNM	0.2			
		IUAM	0.8			
	MobileNet	IGNM	0.2	100	7,000	0.0000578
		IEGNM	0.1			
		IUAM	0.5			
	AM-MobileNet	IGNM	0.1	100	6,000	0.0000578
		IEGNM	0.1			
		IUAM	0.8			
VoxCeleb1	ResNet18	IGNM	0.5	200	4,500	
		IEGNM	0.2			
	ResNet34	IGNM	0.2	200	4,500	0.0000432
		IEGNM	0.2			
		IUAM	1.0			
	AutoSpeech	IGNM	0.2	200	4,500	
		IEGNM	0.2			

Based on our imperceptibility analysis, we can conclude the following:

- Overall, no watermark parameters resulted in a trigger set whose LSD is ≤ 2, while also maintaining the same fidelity as the watermark with default parameters.
- For all watermarking methods, we can see that applying the same parameters to different datasets yields different SNR levels. Notably, imperceptibility appears slightly lower in VoxCeleb1. Among all the watermarks, the one that stands out the most is the Unrelated Audio watermark, which has a difference in SNR of 8.5033, even though the difference in LSD is only 0.8616. We hypothesise that it might stem from the fact that the variance of noise in TIMIT is fairly uniform across samples, whereas VoxCeleb1 contains recordings of varying quality (e.g., studio recordings versus red carpet recordings). Consequently, the same watermark can

have different effects on individual samples, with the averaged impact reflected in the overall SNR value.

- In 15 out of 18 dataset-model-watermark combinations, we were able to improve imperceptibility without any loss in the watermark effectiveness. For two combinations, the improvement of imperceptibility came only with the loss of watermark effectiveness. For one case (VC1-ResNet34-UA), we chose the trigger creation parameters that resulted in 100% WMSR with lower imperceptibility, since none of the settings with higher imperceptibility achieved it.

- Out of 15 successful cases of imperceptibility improvement, in four models, trained on the TIMIT dataset, achieved this with a certain cost of fidelity, namely up to 0.0022 difference in SER. Two other models, ResNet34 and AutoSpeech trained on the VoxCeleb1, also showed some SER increase, but their fidelity remained within the range of the respective non-watermarked models' sentence error rates.

- In all combinations of dataset-model-watermark, the best imperceptibility (while maintaining fidelity) was achieved using our altered algorithm for Frequency Noise, namely Extreme Gaussian Noise watermark.

From our imperceptibility analysis of the Frequency and Extreme Gaussian Noise watermarks, we conclude the following:

- No combination of parameters of the Frequency Noise watermark performed better than Extreme Gaussian Noise with regard to the watermark effectiveness, model fidelity, and watermark imperceptibility, as previously noted.

- Our analysis shows that even the Gaussian Noise watermark alone is already highly effective. For the Frequency Noise watermark, it is possible to omit the MFCC step, as it seems not to add in terms of effectiveness and fidelity: our additional experiments demonstrate that using the Extreme Gaussian Noise watermark (which differs from Frequency Noise by omitting lines 10–11 in Algorithm 5.2) still results in a watermark success rate of 100% and a sentence error rate below 0.02 for SincNet, MobileNet, AM-MobileNet; for ResNet18, ResNet34, and AutoSpeech models this method also results in a lower SER than their respective SERs of non-watermarked models.Moreover, the Extreme Gaussian Noise watermark is more imperceptible compared to the Frequency Noise.

- The extreme frequency limits do not affect the SNR value much: e.g., Extreme Gaussian Noise with $\lambda = 0.2$, ELF=100, and EHF=6,000 in one instance and EHF=6,400 in another instance have only 0.005 SNR difference; however, LSD is affected more in this case, showing a 0.3463 difference. With smaller λ, embedding the watermark in more frequency space (e.g., $\lambda = 0.2$, ELF=200, EHF=4,500 compared to $\lambda = 0.5$, ELF=100, EHF=6,000) can still result in higher imperceptibility.

Unrelated Audio:

- For Unrelated Audio watermark, the choice of the audio sample plays a crucial role in terms of the level of perceptibility it adds to the trigger set. However, even a sample with higher noise can be tuned to the required imperceptibility level by a scale factor.
- Generally, the Unrelated Audio watermark requires lower imperceptibility to achieve an acceptable level of fidelity and watermark effectiveness. Moreover, even with lower imperceptibility, in the majority of the cases, the SER of the Unrelated Audio models was the highest among the different watermarking methods that were applied to each dataset-model pair (the exception is ResNet34, though it is still higher than Gaussian Noise and Extreme Gaussian Noise models with the selected parameters).

Table 6.24: Results of the Imperceptibility analysis of the watermarking methods across the datasets and models.

Dataset	Model	WM	Effectiveness WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	Imperceptibility SNR↑
TIMIT	SincNet 0.0051	GNM	100%	0.0101	13.0948	16.6514
		IGNM	100%	0.0094	10.6347	**21.0876**
		FNM	100%	0.0108	20.6664	-1.3025
		IEGNM	100%	0.0129	6.0068	**30.6359**
		UAM	100%	0.0101	3.8900	**23.0389**
		IUAM	100%	0.0101	3.4947	**24.9770**
	MobileNet 0.0057	GNM	100%	0.0065	13.0948	16.6514
		IGNM	100%	0.0043	6.3771	**30.6293**
		FNM	100%	0.0058	20.6664	-1.3025
		IEGNM	100%	0.0058	3.6815	**36.6688**
		UAM	100%	0.0087	3.8900	**23.0389**
		IUAM	100%	0.0094	2.7731	**29.0592**
			100%	*0.0108*	***1.1882***	***43.0357***
	AM-MobileNet 0.0043	GNM	100%	0.0036	13.0948	16.6514
		IGNM	100%	0.0036	4.4207	**36.6494**
		FNM	100%	0.0043	20.6664	-1.3025
		IEGNM	100%	0.0065	4.3513	**36.6559**
		UAM	100%	0.0115	3.8900	**23.0389**
		IUAM	100%	0.0129	3.4947	**24.9770**
			100%	*0.0137*	***1.1882***	***43.0357***
VoxCeleb1	ResNet18 0.2978	GNM	100%	0.2306	13.0948	15.9199
		IGNM	100%	0.2258	10.2465	**21.9467**
		FNM	100%	0.2485	22.5900	-2.6448
		IEGNM	100%	0.2480	7.2651	**29.9050**
		UAM	100%	0.2859	4.7516	14.5356
	ResNet34 0.2206	GNM	100%	0.2396	13.0948	15.9199
		IGNM	100%	0.2066	6.7661	**29.9027**
		FNM	100%	0.2016	22.5900	-2.6448
		IEGNM	100%	0.1959	7.2651	**29.9050**
		UAM	98%	0.2071	4.7516	14.5356
		IUAM	100%	0.2201	12.7222	5.6682
	AutoSpeech 0.1691	GNM	100%	0.1476	13.0948	15.9199
		IGNM	100%	0.1492	6.7661	**29.9027**
		FNM	100%	0.1298	22.5900	-2.6448
		IEGNM	100%	0.1400	7.2651	**29.9050**
		UAM	100%	0.1515	4.7516	14.5356

6.4 Legality Evaluation

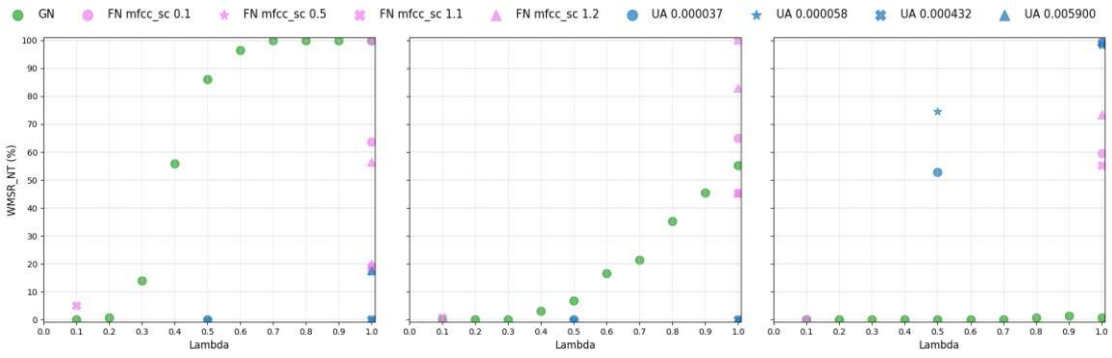
This section evaluates the legality of the three watermarking methods across each dataset–model pair. First, the results obtained using the original trigger creation parameters are presented. Subsequently, the legality of the most imperceptible settings, as defined in the previous section, is examined.

6.4.1 TIMIT

Below, we present the legality evaluation results for the TIMIT dataset carried out according to the setup described in Section 5.7 across three models: SincNet, MobileNet, and AM-MobileNet.

SincNet

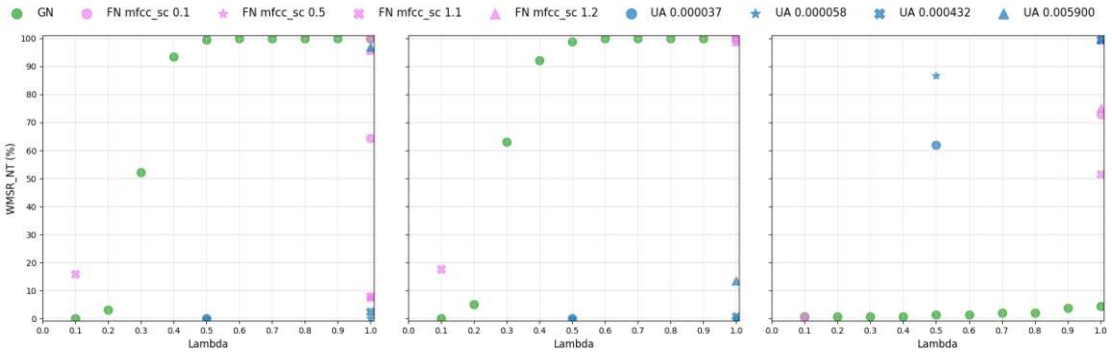
Default WM parameter settings The results of investigating the legality of the replicated SOTA watermarking methods applied to the SincNet model are shown in Figure 6.28 for (a) the Gaussian Noise model (GNM), (b) the Frequency Noise model (FNM), and (c) the Unrelated Audio model (UAM). We can see that the Gaussian Noise model recognises Gaussian Noise triggers, with values for lambda from 1.0 down to 0.7 with 100% effectiveness. With $\lambda = 0.6$, it still yields 96.6% WMSR_NT. Moreover, it also recognises the Frequency Noise watermark that was created with the parameters $\{mfcc\_sc = 1.1, \lambda = 1.0$, ELF=300, EHF=3,000$\}$ with 100%. While the Frequency Noise model only recognises the frequency noise setting with the same parameters as the Gaussian Noise model, the Unrelated Audio model accepts multiple Unrelated Audio and a few Frequency Noise settings as well (all those with $\lambda = 1.0$) with a nearly 100% success rate.



(a) GNM (b) FNM (c) UAM

Figure 6.28: Legality evaluation of the SOTA watermarking methods applied to TIMIT SincNet.

Imperceptible WM parameter settings When considering the imperceptible watermarks obtained in Section 6.3.1, we can see that more watermark settings are now recognised by the Gaussian Noise (Figure 6.29a) and Frequency Noise (Figure 6.29b) models. For the Unrelated Audio model (Figure 6.29c, the recognition is higher in two cases: Unrelated Audio triggers created with the same parameters as the triggers that were used for training, and Frequency Noise triggers created with $mfcc\_sc = 0.1$ improved by 20%.

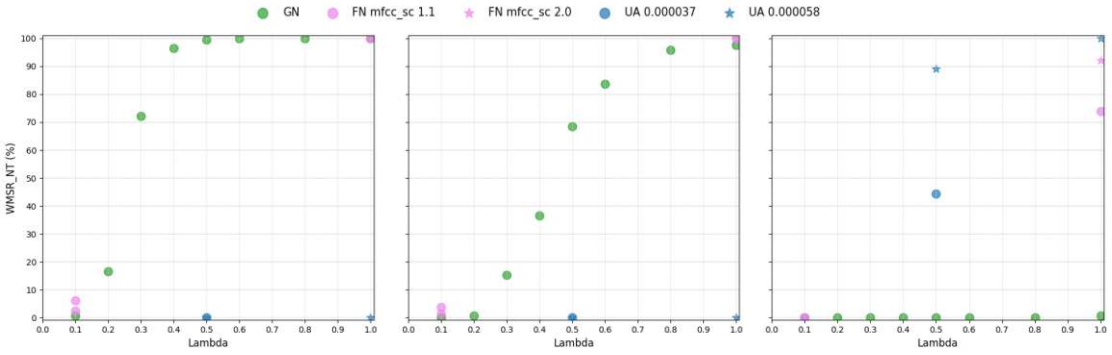


(a) GNM (b) FNM (c) UAM

Figure 6.29: Legality evaluation of the watermarking methods with the most imperceptible parameter settings applied to TIMIT SincNet.

MobileNet

Default WM parameter settings From the results in Figure 6.30, we can see that MobileNet with Gaussian Noise watermark recognises the triggers created with even lower λ (0.5) compared to the respective SincNet model (λ=0.7). Moreover, the Frequency



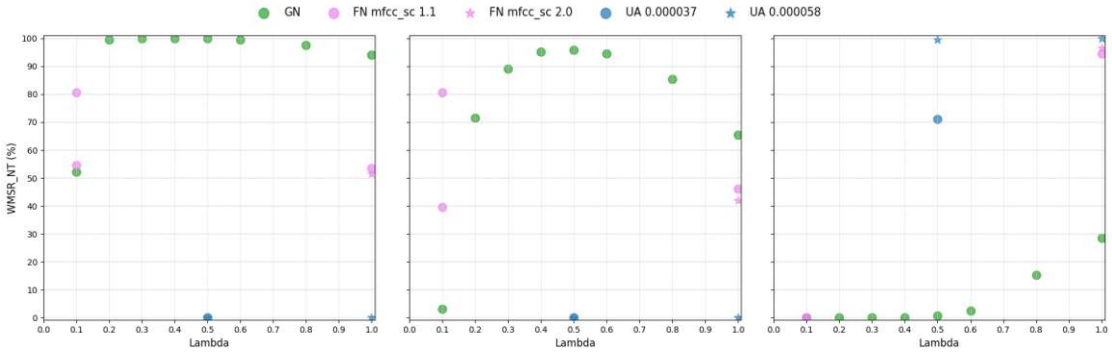
(a) GNM (b) FNM (c) UAM

Figure 6.30: Legality evaluation of the SOTA watermarking methods applied to TIMIT MobileNet.

Noise model is more prone to accepting samples watermarked with Gaussian Noise. The

Unrelated Audio model also shows higher WMSR_NT for Frequency Noise triggers created with $mfcc\_sc = 2.0$.

Imperceptible WM parameter settings The legality evaluation of more imperceptible watermarks (which were created with the selected parameters in Section 6.3.1) indicates slightly worse results, as can be seen in Figure 6.31: the Gaussian Noise model now recognises the triggers created with λ down to 0.2, the Frequency Noise model now does not yield 100% WMSR_NT on any of the tested unseen triggers, though the recognition of the triggers created with the same λ or value close to it still yields higher rates (which are higher than 92%). Lastly, the Unrelated Audio model yields nearly 100% WMSR_NT with $\lambda = 0.5$ of the same unrelated audio as its embedded watermark. Additionally, the Frequency Noise watermark with $mfcc\_sc = 1.1$ is also recognised by the Unrelated Audio model.

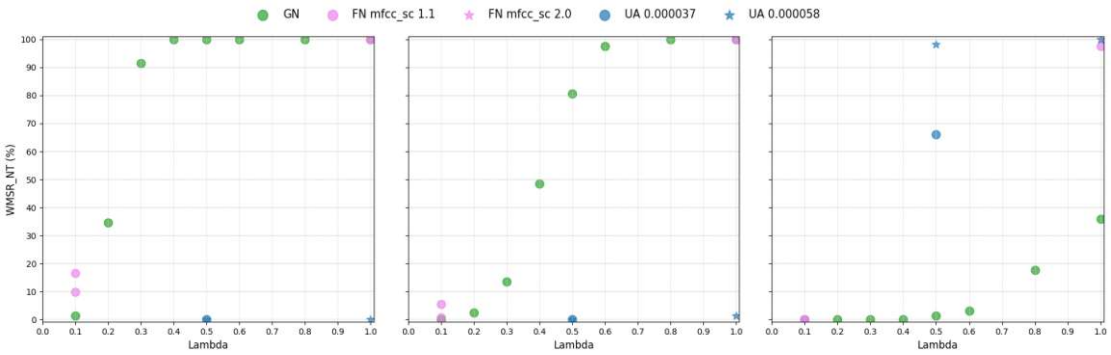


(a) GNM (b) FNM (c) UAM

Figure 6.31: Legality evaluation of the watermarking methods with the most imperceptible parameter settings applied to TIMIT MobileNet.

AM-MobileNet

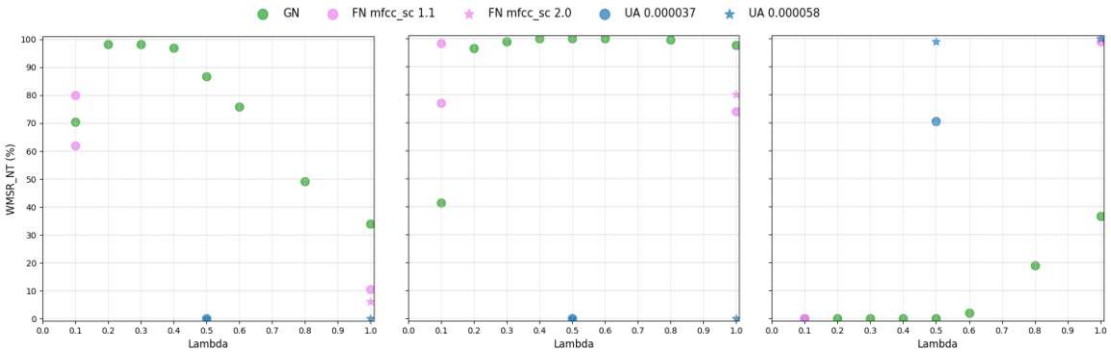
Default WM parameter settings As can be seen in Figure 6.32, the Gaussian Noise model recognises the triggers created with a lower λ of 0.4 compared to the MobileNet model (0.5). The same pattern can be observed for the Frequency Noise model, which now yields WMSR_NT higher than 95% for λ values down to 0.6 compared to MobileNet's 0.8. The recognition pattern of the Unrelated Audio model here closely resembles the MobileNet Unrelated Audio model with the most imperceptible setting (Figure 6.31c), though WMSR_NT values are overall slightly higher for the Gaussian and Frequency Noise watermarks.



(a) GNM (b) FNM (c) UAM

Figure 6.32: Legality evaluation of the SOTA watermarking methods applied to TIMIT AM-MobileNet.

Imperceptible WM parameter settings The creation parameters for the imperceptible watermark settings for the AM-MobileNet model are specified in Section 6.3.1. Regarding the legality characteristic of the watermarked AM-MobileNet models depicted in Figure 6.33, the Gaussian Noise model is more resilient against the newly created triggers compared to both the original Gaussian Noise watermark created with $\lambda = 1.0$ and the most imperceptible Gaussian Noise setting applied to MobileNet, since it recognises only the triggers created with λ values in range of 0.2-0.4. The Frequency Noise model, though, can recognise all settings of Gaussian Noise triggers except for the ones created with $\lambda = 0.1$ and has a higher WMSR_NT rate for Frequency Noise triggers compared to the original Frequency Noise model for AM-MobileNet (Figure 6.32b). Only Unrelated Audio triggers are not at all recognised by the Frequency Noise model. The results on the Unrelated Audio model are almost the same as the original Unrelated Audio watermark settings for AM-MobileNet (Figure 6.32c) and the most imperceptible watermark setting for MobileNet (Figure 6.31c).



(a) GNM (b) FNM (c) UAM

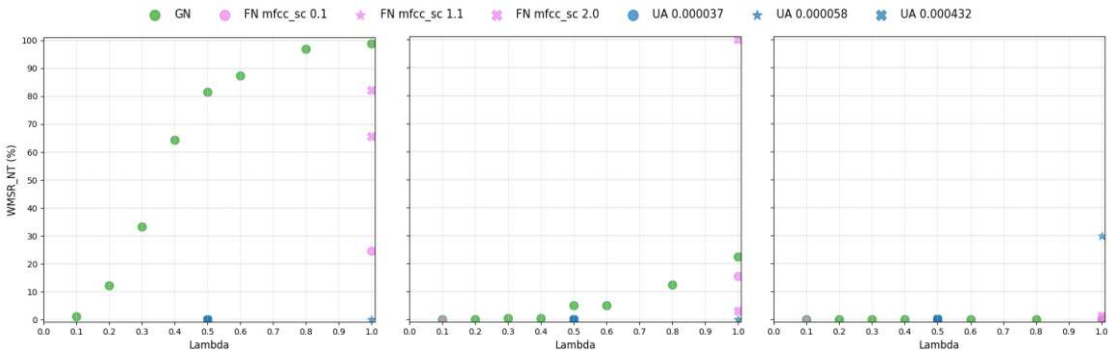
Figure 6.33: Legality evaluation of the watermarking methods with the most imperceptible parameter settings applied to TIMIT AM-MobileNet.

6.4.2 VoxCeleb1

In the following, we present the legality evaluation results for the VoxCeleb1 dataset carried out according to the setup described in Section 5.7 across three models: ResNet18, ResNet34, and AutoSpeech.

ResNet18

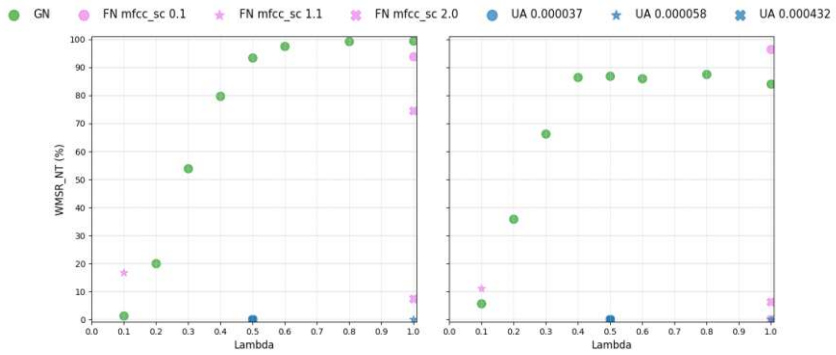
Default WM parameter settings The ResNet18 models in Figure 6.34 show much higher resilience against unseen triggers compared to other models. For the Gaussian Noise model, only two trigger creation parameters result in higher than 95% WMSR_NT: the Gaussian Noise watermarks with λ values 1.0 and 0.9. The Frequency Noise model recognises with 100% only the unseen triggers created with the same parameters as the ones used for training, while the Unrelated Audio model has a maximum of 30% WMSR_NT.



(a) GNM (b) FNM (c) UAM

Figure 6.34: Legality evaluation of the SOTA watermarking methods applied to VoxCeleb1 ResNet18.

Imperceptible WM parameter settings The results shown in Figure 6.35 indicate worse outcomes compared to the original trigger creation settings. In the case of the Gaussian Noise model, the newly created triggers with three different λ values (1.0, 0.8, and 0.6) are now recognised. Moreover, the Frequency Noise model recognises the Gaussian Noise triggers more than the ResNet18 model with default parameters in (Figure 6.34b), though the WMSR_NT is still below 90%. The only newly created triggers that the Frequency Noise model does recognise are Frequency Noise triggers created with $mfcc\_sc = 0.1$, but the WMSR_NT is lower than 100%, which is better than in the case with the original Frequency Noise watermark.

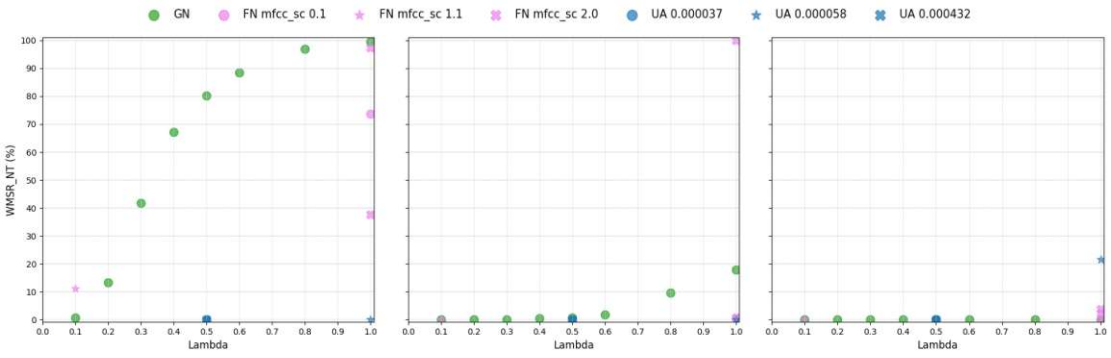


(a) GNM (b) FNM

Figure 6.35: Legality evaluation of the watermarking methods with the most imperceptible parameter settings applied to VoxCeleb1 ResNet18.

ResNet34

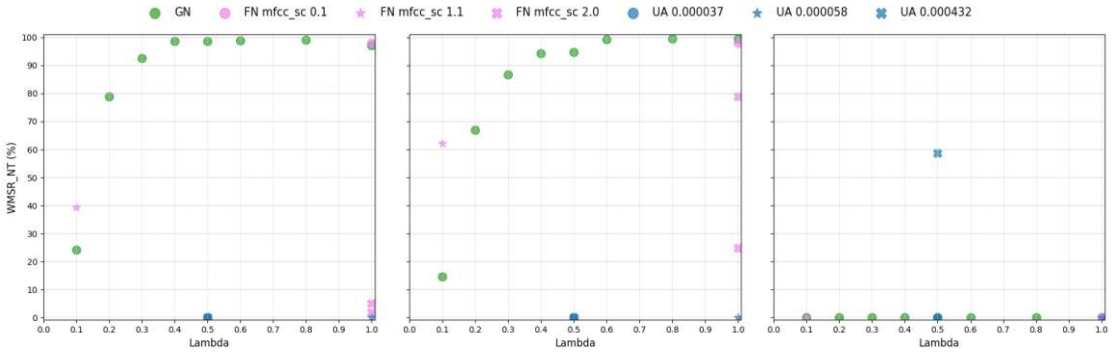
Default WM parameter settings In Figure 6.36, we can see a very similar pattern to that of ResNet18. The only difference is that the Gaussian Noise model also recognises the triggers created with the Frequency Noise watermarking method using $mfcc\_sc = 2.0$.



(a) GNM (b) FNM (c) UAM

Figure 6.36: Legality evaluation of the SOTA watermarking methods applied to VoxCeleb1 ResNet34.

Imperceptible WM parameter settings As with the ResNet18 model, when watermarked with the most imperceptible settings, the WMSR_NT deteriorates with higher imperceptibility for Gaussian Noise and Frequency Noise models, which is shown in Figure 6.37. The difference is that ResNet34 with Frequency Noise watermark recognises fewer trigger creation settings than with the Gaussian Noise watermark, which is the opposite case as it was for ResNet18. With the Unrelated Audio watermark, the model achieves only 60% WMSR_NT when tested on unseen triggers created from the same unrelated audio sample used for watermarking.

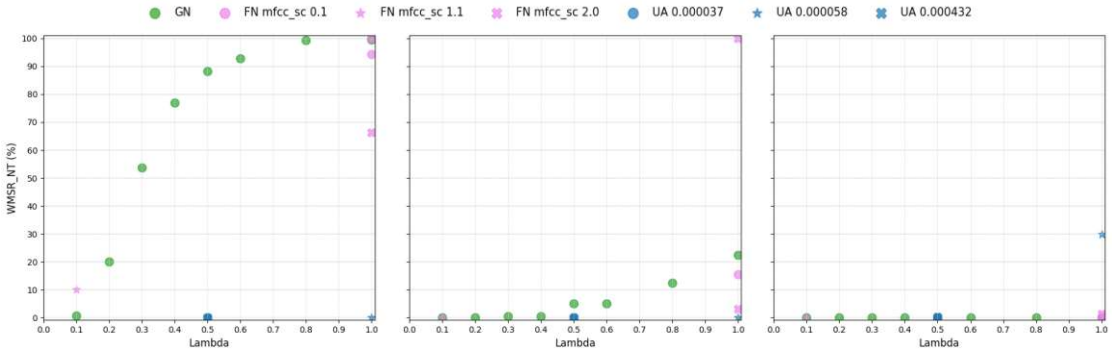


(a) GNM (b) FNM (c) UAM

Figure 6.37: Legality evaluation of the watermarking methods with the most imperceptible parameter settings applied to VoxCeleb1 ResNet34.

AutoSpeech

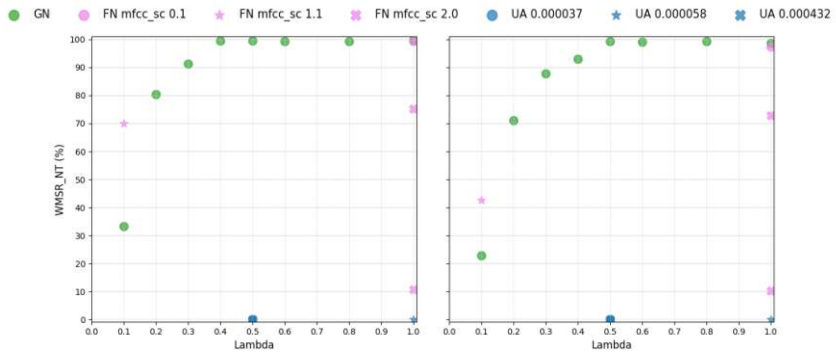
Default WM parameter settings The results in Figure 6.38 show similar patterns as with the ResNet models, where only the Gaussian Noise model recognises the newly generated triggers.



(a) GNM (b) FNM (c) UAM

Figure 6.38: Legality evaluation of the SOTA watermarking methods applied to VoxCeleb1 AutoSpeech.

Imperceptible WM parameter settings The results for the most imperceptible parameter settings are shown in Figure 6.39. We can see the same pattern as with the ResNet models, namely, the more imperceptible watermarks are more prone to cause a model to recognise unseen triggers.



(a) GNM (b) FNM

Figure 6.39: Legality evaluation of the watermarking methods with the most imperceptible parameter settings applied to VoxCeleb1 AutoSpeech.

6.4.3 Legality Evaluation Findings

In Table 6.25, we depict the results of the legality evaluation across datasets and models. GNM denotes the model watermarked by the Gaussian Noise watermarking method, while IGNM denotes the model watermarked using the Gaussian Noise triggers created with imperceptible parameters defined in the previous section. The same applies to FNM and UAM. We highlight in green the imperceptible settings that successfully improve legality results, and in orange those that yield worse legality results compared to their respective default settings.

The legality characteristic shows a fairly distinct effect on the datasets. The Unrelated Audio models trained on VoxCeleb1, in general, do not recognise the newly generated triggers as watermarks. On the other hand, within TIMIT, the Unrelated Audio models achieve comparatively good performance for both MobileNet and AM-MobileNet, recognising the fewest watermark settings compared to GNM and FNM of the respective models. In contrast, SincNet performs much worse, with the Unrelated Audio model recognising 6 out of 22 settings.

The Gaussian Noise models were the worst in legality, compared to the other watermarking methods. This can be explained in general by the fact that Gaussian Noise, Frequency Noise, and Extreme Gaussian Noise watermarks all use a form of Gaussian noise, while Unrelated Audio watermark uses a completely different type of sound. An interesting fact is that in three cases, specifically MobileNet GNM and FNM, and AM-MobileNet GNM, the most imperceptible watermarks showed more resilience in terms of legality compared to the original ones.

Another pattern we can observe in this legality evaluation is that triggers created with the same (or close to) λ values as the real triggers were created are recognised more.

These findings indicate that a malicious party, who does not know the exact watermarking method or embedding parameters, can potentially create trigger samples that will falsely

indicate ownership, even if the model was watermarked with a different method. That is why it is in our interest for the watermark success rate (WMSR) to remain as close to 100% as possible, which we are evaluating in the next section.

Table 6.25: Legality evaluation across models and datasets. The legality characteristic is evaluated based on different triggers: we verify whether a certain model can recognise unseen triggers created by other schemes (Gaussian Noise (GN), Frequency Noise (FN), and Unrelated Audio (UA)), each evaluated with different parameter settings. ✗ denotes a case when a model recognises the unseen triggers with at least 95% success rate, ✓ otherwise.

Dataset	Model	WM	Effectiveness WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	Imperceptibility SNR↑	Legality GN	Legality FN	Legality UA
TIMIT	SincNet 0.0051	GNM	100%	0.0101	13.0948	16.6514	✓5 ✗5	✓6 ✗1	✓5
		IGNM	100%	0.0094	10.6347	21.0876	✓4 ✗6	✓5 ✗2	✓4 ✗1
		FNM	100%	0.0108	20.6664	-1.3025	✓10	✓6 ✗1	✓5
		IEGNM	100%	0.0129	6.0068	30.6359	✓4 ✗6	✓1 ✗6	✓5
		UAM	100%	0.0101	3.8900	23.0389	✓10	✓4 ✗3	✓2 ✗3
		IUAM	100%	0.0101	3.4947	24.9770	✓10	✓4 ✗3	✓2 ✗3
	MNet 0.0057	GNM	100%	0.0065	13.0948	16.6514	✓3 ✗5	✓2 ✗2	✓3
		IGNM	100%	0.0043	6.3771	30.6293	✓2 ✗6	✓4	✓3
		FNM	100%	0.0058	20.6664	-1.3025	✓6 ✗2	✓2 ✗2	✓3
		IEGNM	100%	0.0058	3.6815	36.6688	✓6 ✗2	✓4	✓3
		UAM	100%	0.0087	3.8900	23.0389	✓8	✓4	✓2 ✗1
		IUAM	100%	0.0094	2.7731	29.0592	✓8	✓3 ✗1	✓1 ✗2
	AM-MNet 0.0043	GNM	100%	0.0036	13.0948	16.6514	✓3 ✗5	✓2 ✗2	✓3
		IGNM	100%	0.0036	4.4207	36.6494	✓5 ✗3	✓4	✓3
		FNM	100%	0.0043	20.6664	-1.3025	✓5 ✗3	✓2 ✗2	✓3
		IEGNM	100%	0.0065	4.3513	36.6559	✓1 ✗7	✓3 ✗1	✓3
		UAM	100%	0.0115	3.8900	23.0389	✓8	✓2 ✗2	✓1 ✗2
		IUAM	100%	0.0129	3.4947	24.9770	✓8	✓2 ✗2	✓1 ✗2
VoxCeleb1	ResNet18 0.2978	GNM	100%	0.2306	13.0948	15.9199	✓6 ✗2	✓3 ✗1	✓4
		IGNM	100%	0.2258	10.2465	21.9467	✓5 ✗3	✓4	✓4
		FNM	100%	0.2485	22.5900	-2.6448	✓8	✓3 ✗1	✓4
		IEGNM	100%	0.2480	7.2651	29.9050	✓8	✓3 ✗1	✓4
		UAM	100%	0.2859	4.7516	14.5356	✓8	✓4	✓4
	ResNet34 0.2206	GNM	100%	0.2396	13.0948	15.9199	✓6 ✗2	✓3 ✗1	✓4
		IGNM	100%	0.2066	6.7661	29.9027	✓3 ✗5	✓3 ✗1	✓4
		FNM	100%	0.2016	22.5900	-2.6448	✓8	✓3 ✗1	✓4
		IEGNM	100%	0.1959	7.2651	29.9050	✓5 ✗3	✓3 ✗1	✓4
		UAM	98%	0.2071	4.7516	14.5356	✓8	✓4	✓4
		IUAM	100%	0.2201	12.7222	5.6682	✓8	✓4	✓4
	AutoSp. 0.1691	GNM	100%	0.1476	13.0948	15.9199	✓6 ✗2	✓3 ✗1	✓4
		IGNM	100%	0.1492	6.7661	29.9027	✓3 ✗5	✓3 ✗1	✓4
		FNM	100%	0.1298	22.5900	-2.6448	✓8	✓3 ✗1	✓4
		IEGNM	100%	0.1400	7.2651	29.9050	✓4 ✗4	✓3 ✗1	✓4
		UAM	100%	0.1515	4.7516	14.5356	✓8	✓4	✓4

6.5 Robustness Evaluation

In this section, we begin with the robustness evaluation of the three watermarking methods with regard to each dataset-model pair. Then we evaluate the robustness of the most imperceptible settings.

Due to restrictions on computational resources, we reduced the number of data preprocessing attacks applied to the imperceptible watermark settings of ResNet34 and AutoSpeech. The attacks were selected by the rule that they did not deteriorate the SER of the watermarked models, trained with the original parameter settings of the watermarks, by more than 0.2. The value was chosen to be higher than our 0.05 threshold, with an additional 0.15 buffer to account for variability in the results of individual artefacts.

Additionally, since the SNR-based random noise addition and bandpass filtering data preprocessing attacks always increased the SER of the models beyond the accepted threshold, they were excluded from the analysis in both ResNet and AutoSpeech models.

6.5.1 TIMIT

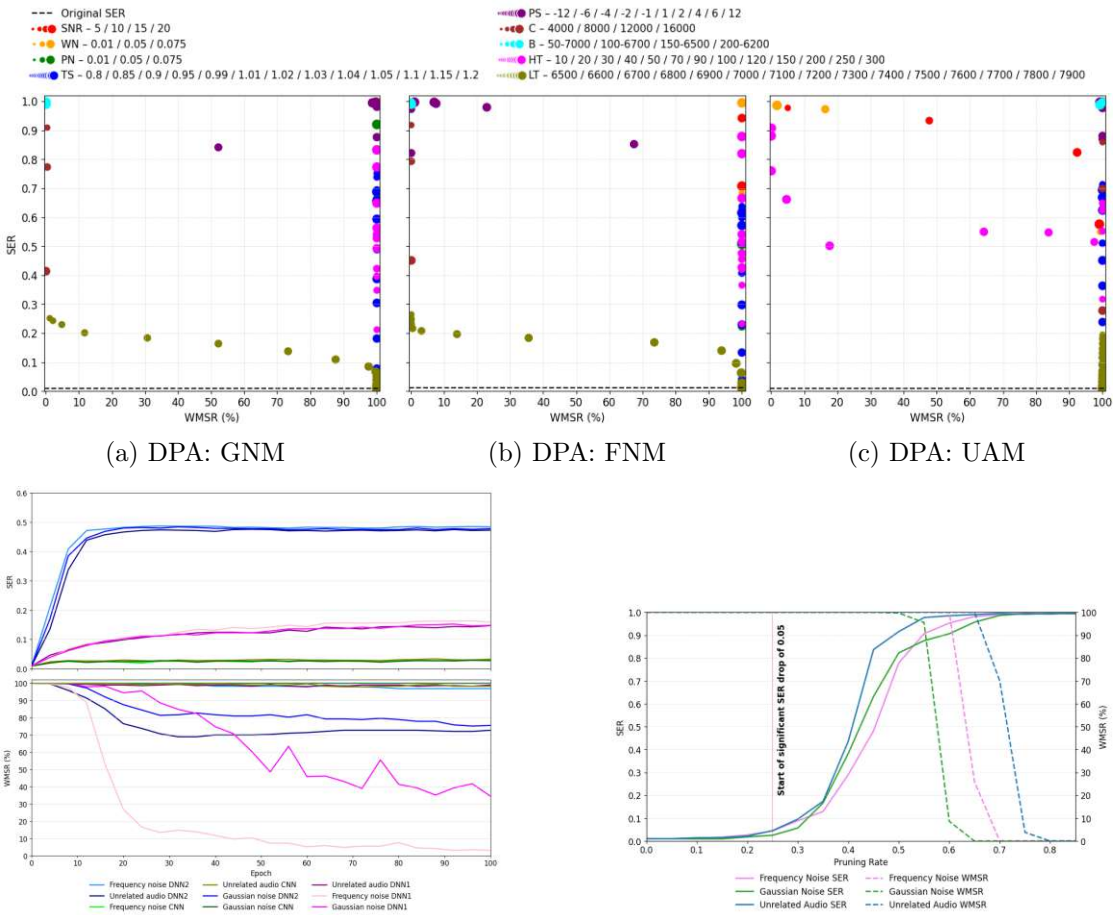
Here, we present the robustness evaluation results for the TIMIT dataset carried out according to the setup described in Section 5.8 across three models: SincNet, MobileNet, and AM-MobileNet.

SincNet

Default WM parameter settings We applied all the mentioned data preprocessing attacks (denoted as DPA in Figures 6.40 to 6.50) across the full range of parameters shown in Figure 6.40a to the Gaussian Noise, Frequency Noise (Figure 6.40b), and Unrelated Audio (Figure 6.40c) models.

Our results show that adding white Gaussian noise, as done by [ZDX+23] using an SNR factor, substantially reduces the effectiveness of the Unrelated Audio watermark, but has a negligible impact on the Gaussian and Frequency Noise models. Nevertheless, the addition of extra Gaussian noise to all audio samples increases the SER to the point that this attack cannot be considered effective, as the attacked model is no longer effective enough. Conversely, maintaining the low-frequency components (LT) on the input audio degrades the performance of the Gaussian and Frequency watermarking models, while leaving the Unrelated Audio watermark unaffected. This can be attributed to the fact that the air conditioning sound used as a trigger in the Unrelated Audio model is predominantly concentrated in lower frequency bands.

While fine-tuning (see Figure 6.40d) the first convolutional layer (CNN) had no observable effect on either task performance or watermark presence, fine-tuning the last layer affected the original task the most, yet had only minimal effect on watermark removal. The most successful attack involved fine-tuning the DNN1 layer, which successfully reduced the WMSR to 3%. However, this came at the cost of a substantial decline in fidelity, as the SER deteriorated faster than WMSR.



(a) DPA: GNM (b) DPA: FNM (c) DPA: UAM

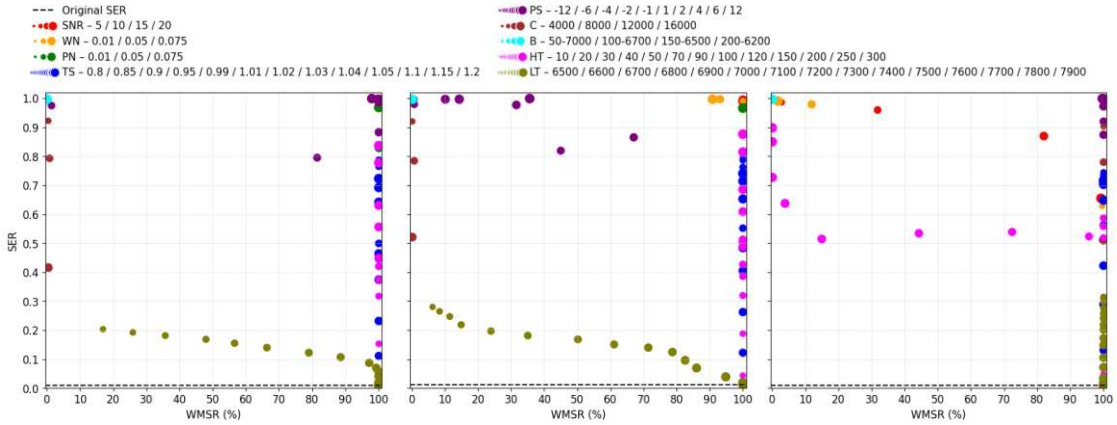
(d) Fine-tuning attacks. (e) Pruning attack.

Figure 6.40: Robustness evaluation of replicated SOTA watermarking methods.

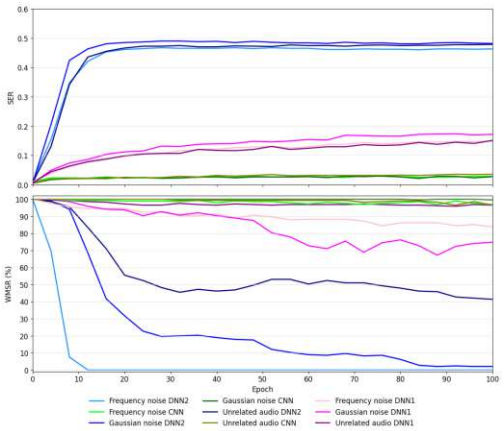
Weight pruning did not result in a successful watermark removal. In Figure 6.40e, we can see that for all WM models, SER starts to deteriorate faster than WMSR. Hence, when the watermark is eventually removed, the model can not perform its original task any longer.

Imperceptible WM parameter settings When applying the same set of attacks to the SincNet models trained with the most imperceptible watermarks, we can see in Figure 6.41 that data preprocessing attacks did not result in much difference for Gaussian Noise (Figure 6.41a) and Unrelated Audio (Figure 6.41c) models. Some of the attacks, such as adding noise by SNR, were more successful in removing a watermark, but the effect on SER stayed the same – with removing the watermark, the model is rendered useless on the original task. However, the Frequency Noise model (Figure 6.41b) was affected more than the other two models: by removing frequencies higher than 7,700 Hz,

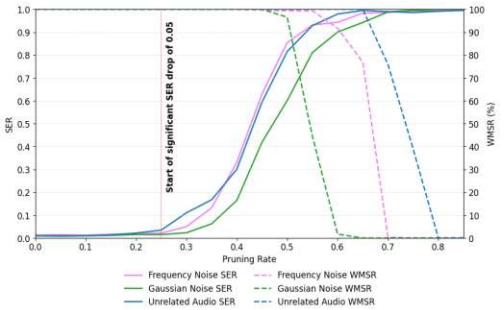
the SER becomes 0.0413 while WMSR drops to 94.9%. Overall, in this setting, we can see that only the Frequency Noise watermark is vulnerable to data preprocessing attacks.



(a) DPA: GNM (b) DPA: FNM (c) DPA: UAM



(d) Fine-tuning attacks.



(e) Pruning attack.

Figure 6.41: Robustness evaluation of the watermarking methods with the most imperceptible parameter settings applied to TIMIT SincNet.

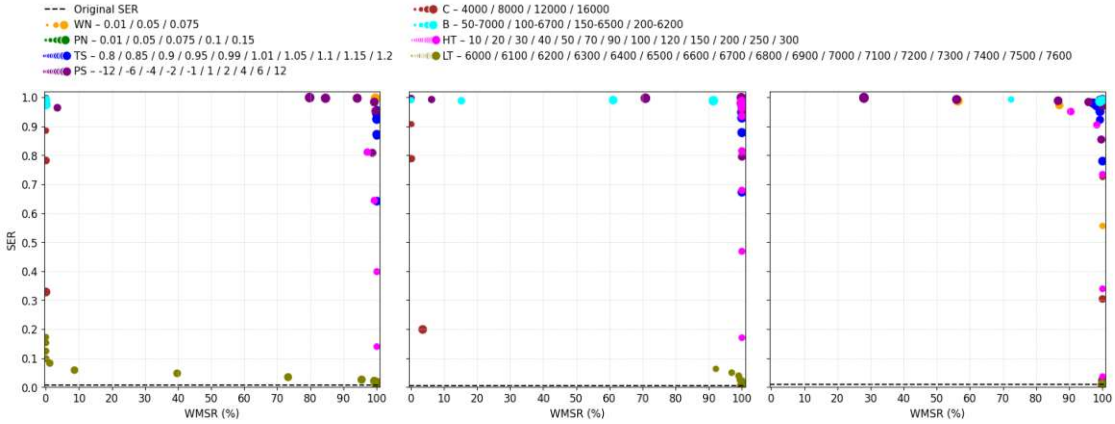
Fine-tuning the last year of the Frequency Noise model (Figure 6.41d) reduced WMSR way faster than in the Frequency Noise default watermark setting (Figure 6.40d); still, since the SER of the model became 0.1545, it cannot be considered a successful attack.

The pruning attack (Figure 6.41e) was unable to remove the watermark in the imperceptible watermark settings without substantially degrading the model performance on the original task.

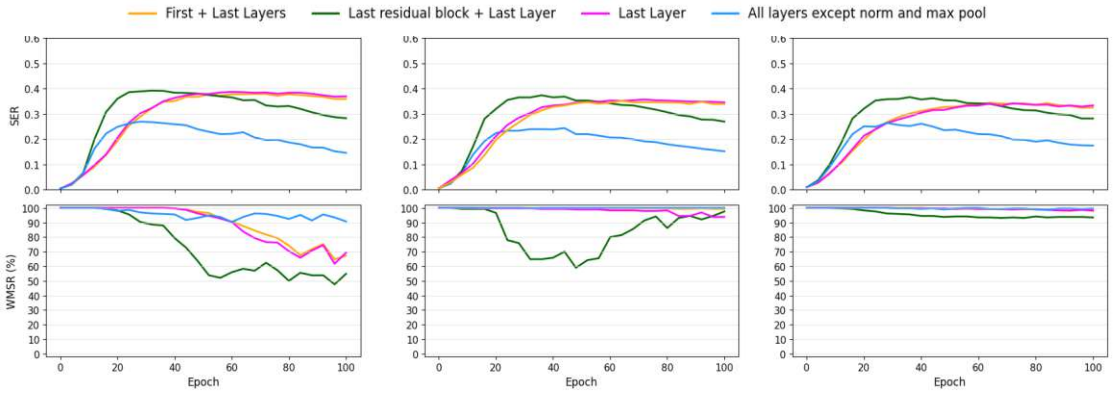
MobileNet

Default WM parameter settings Firstly, we observed that the fidelity of MobileNet is less susceptible to LT data preprocessing attacks than SincNet, which is especially

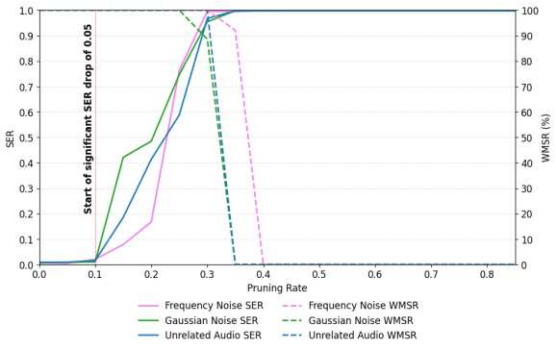
visible for the Frequency Noise model in Figure 6.42b. Therefore, we evaluated the removal of high frequencies starting between 6,000 and 7,600, with a step size of 100 Hz, instead of the 6,500 - 7,900 range considered before. The Gaussian Noise model is highly



(a) DPA: GNM (b) DPA: FNM (c) DPA: UAM



(d) Fine-tuned GNM (e) Fine-tuned FNM (f) Fine-tuned UAM



(g) Pruning attack.

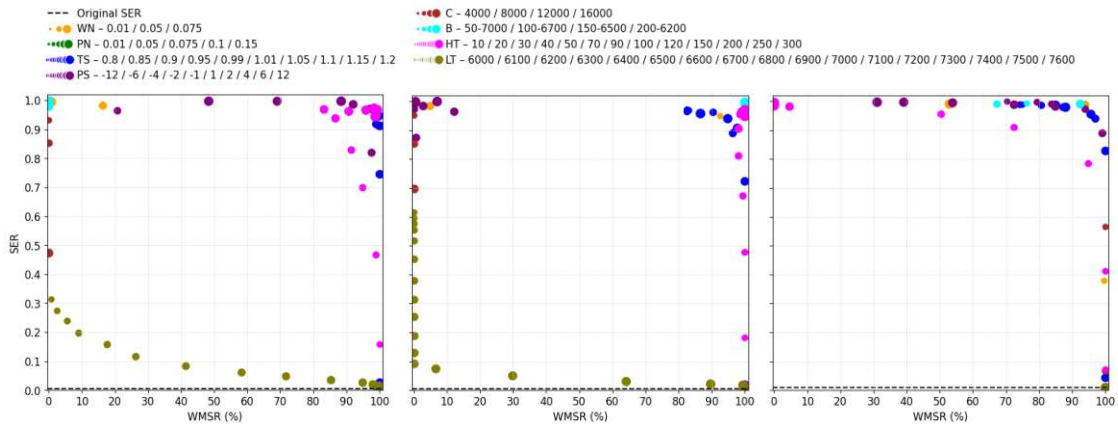
Figure 6.42: Robustness evaluation of the SOTA watermarking methods applied to TIMIT MobileNet.

affected by the removal of high frequencies (Figure 6.42a): when removing frequencies higher than 6,700, SER increases only to 0.036, while the WMSR drops to 74%; and with 6,600 Hz, the SER becomes 0.049, while the WMSR is only 39%. The Frequency Noise and Unrelated Audio models show resilience against the data preprocessing attacks – none of them successfully removed the watermarks.

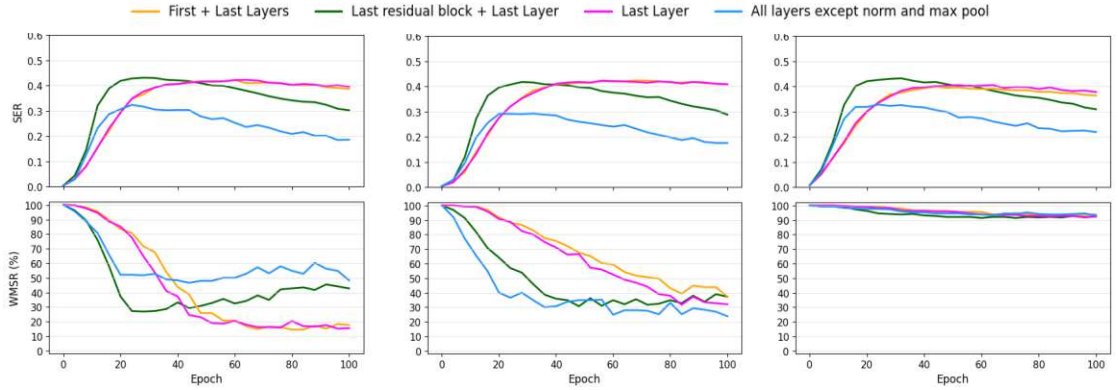
Based on the presented results of fine-tuning and pruning attacks in Figures 6.42d to 6.42g, respectively, they were not able to remove the watermark without suffering a loss of fidelity.

Imperceptible WM parameter settings The most successful data preprocessing attack applied to the Gaussian Noise model was LT with parameter 6,800, which resulted in SER of 0.048 and 71% WMSR. For the Frequency Noise model, the LT attack with 7,300 Hz proved to be the most effective: with a 0.05 SER, the WMSR dropped to 30% rate. No data preprocessing attack was successful with the Unrelated Audio model, as shown in Figure 6.43c.

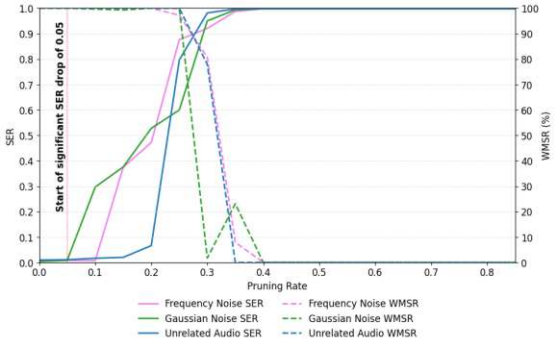
Fine-tuning of all layers except for the normalisation and max pooling layers of the Frequency Noise model (Figure 6.43e) in early stages (namely, after the first four epochs) yields 0.029 SER and 92.1% WMSR, though afterwards the SER grows beyond our difference threshold of 0.05. Both Gaussian Noise and Unrelated Audio models withstand the fine-tuning attacks (Figures 6.43d and 6.43f, respectively). Weight pruning shown in Figure 6.43g does not result in successful watermark removal without compromising the model performance on the original task.



(a) DPA: GNM (b) DPA: FNM (c) DPA: UAM



(d) Fine-tuned GNM (e) Fine-tuned FNM (f) Fine-tuned UAM

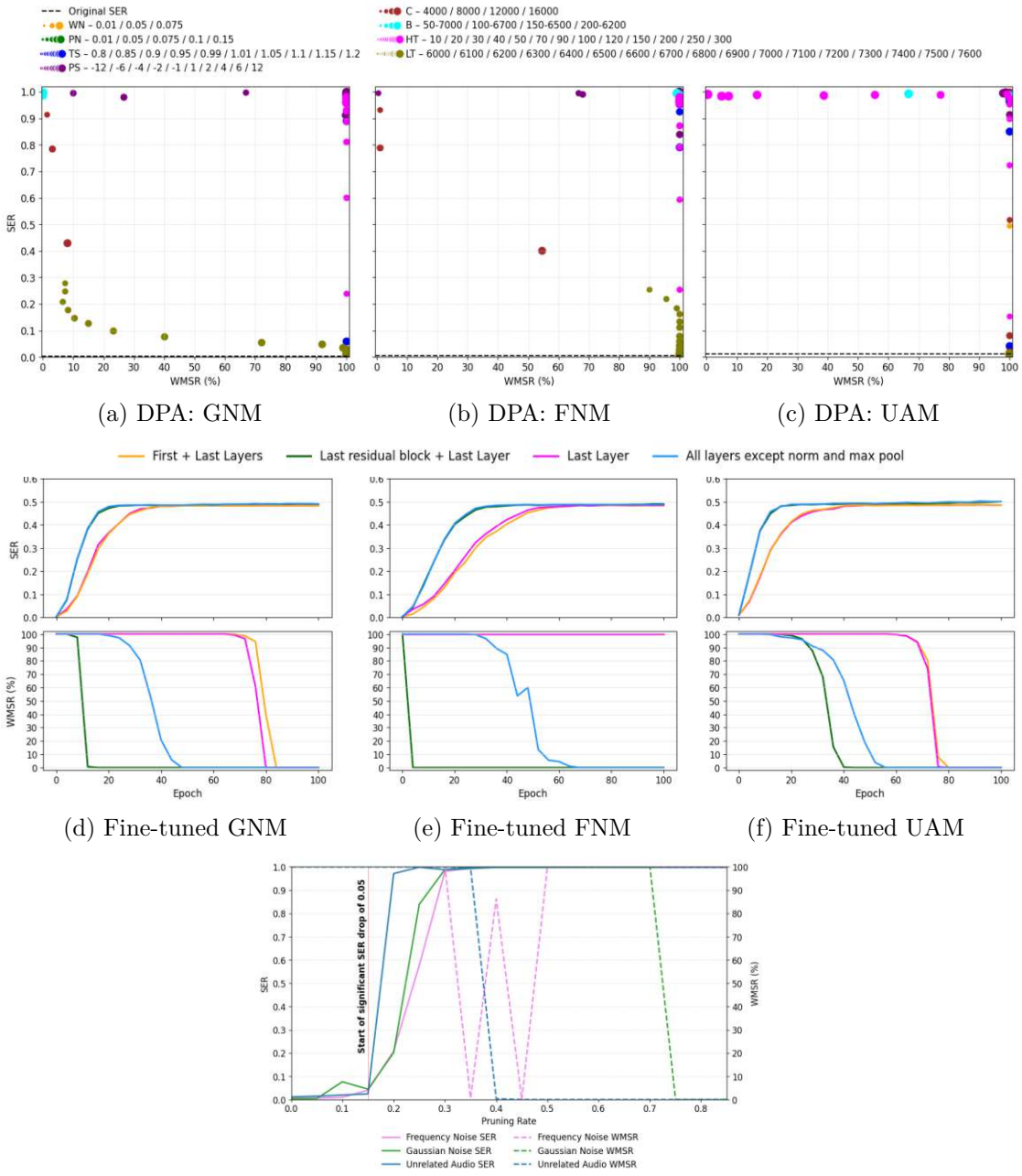


(g) Pruning attack.

Figure 6.43: Robustness evaluation of the watermarking methods with the most imperceptible parameter settings applied to TIMIT MobileNet.

AM-MobileNet

Default WM parameter settings As for MobileNet, the only watermarked model



(a) DPA: GNM (b) DPA: FNM (c) DPA: UAM

(d) Fine-tuned GNM (e) Fine-tuned FNM (f) Fine-tuned UAM

(g) Pruning attack.

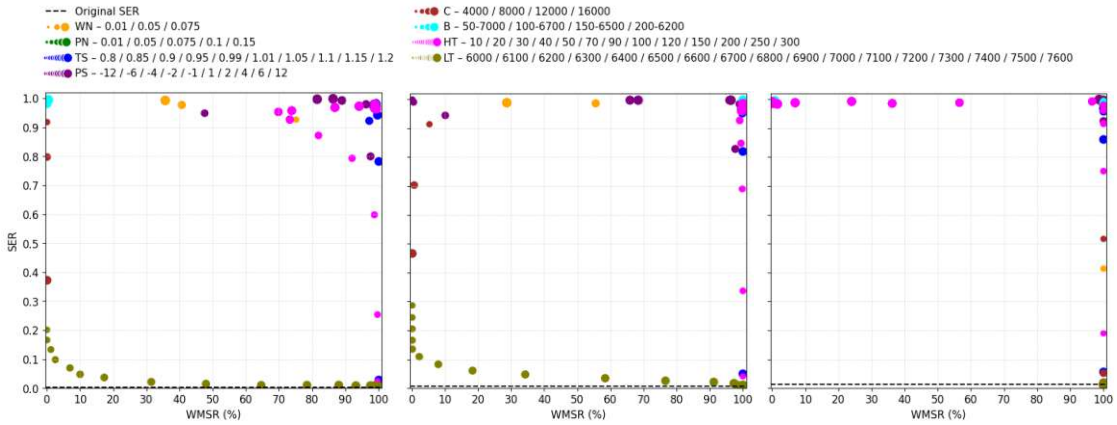
Figure 6.44: Robustness evaluation of the SOTA watermarking methods applied to TIMIT AM-MobileNet.

that suffers from data preprocessing attacks is the Gaussian Noise model (Figure 6.44a). With the LT attack and 6,900 Hz cut-off value, it yields 0.048 SER and 20.7% WMSR. Higher frequencies do not reduce the WMSR by more than 1%, and lower frequencies yield a SER higher than 0.053.

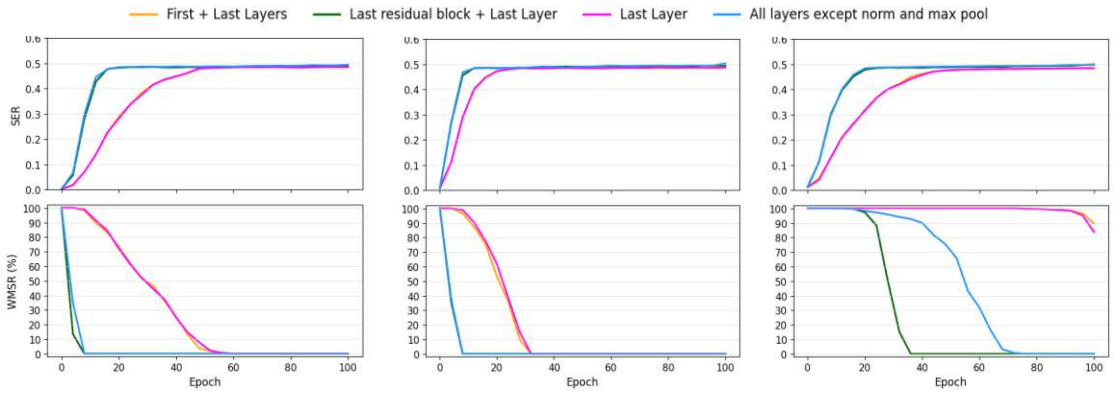
The fine-tuning attack is also only successful for the Frequency Noise Model (Figure 6.44e): after the first four epochs of fine-tuning the last residual block and the last layer, the SER of the model is 0.044, while the WMSR becomes 0%. Conversely, in this case, fine-tuning only the last layer or the first and last layers does not affect the watermark effectiveness at all. Weight pruning does not result in a successful attack, as can be seen in Figure 6.44g.

Imperceptible WM parameter settings One can observe that some data preprocessing attacks, namely LT with different high frequency values, remove watermarks in Gaussian Noise (Figure 6.45a) and Frequency Noise (Figure 6.45b) models more compared to the original, more perceptible settings. The pattern of the LT attack applied to the Frequency Noise model now resembles more the pattern of the Gaussian Noise model, which is expected, since we use an Extreme Gaussian Noise watermark, which resembles the Gaussian Noise watermark. For the Gaussian Noise model, the LT attack with 6,500 Hz parameter yields 0.049 SER while simultaneously removing the watermark, with WMSR dropping to 10%. Moreover, with a SER of only 0.015, which can be achieved by applying LT with 6,800 Hz, the WMSR already drops to 48%. For the Frequency Noise model, the rates are slightly better: the most successful attack within our defined SER bounds is LT with 6,800 Hz, which results in a 0.048 SER and 34% WMSR. The Unrelated Audio model (Figure 6.45c) is again unaffected.

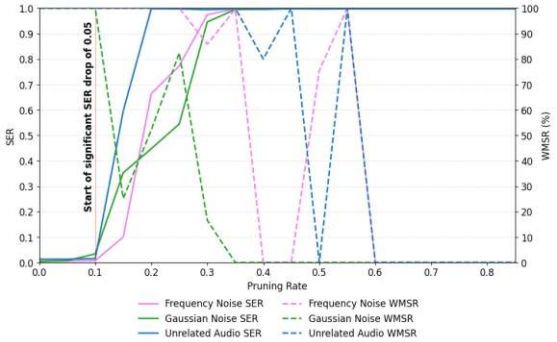
Within the fine-tuning attacks, the Unrelated Audio model (Figure 6.45f) is the most robust one since no combination was able to remove the watermark without deteriorating SER. The fine-tuning of the Frequency Noise model (Figure 6.45e), even though it showed an immediate drop in WMSR, also increased SER beyond the 0.05 difference threshold; thus, it does not successfully remove the watermark. The watermark that was affected the most by a fine-tuning attack is Gaussian Noise (Figure 6.45d). Fine-tuning the last residual block and last layer results in a SER of 0.058 after the first four epochs; even though it is just over the threshold, it yields a 13% WMSR, which means the resulting model is no longer useful. Weight pruning (Figure 6.45g) did not result in successful watermark removal.



(a) DPA: GNM (b) DPA: FNM (c) DPA: UAM



(d) Fine-tuned GNM (e) Fine-tuned FNM (f) Fine-tuned UAM



(g) Pruning attack.

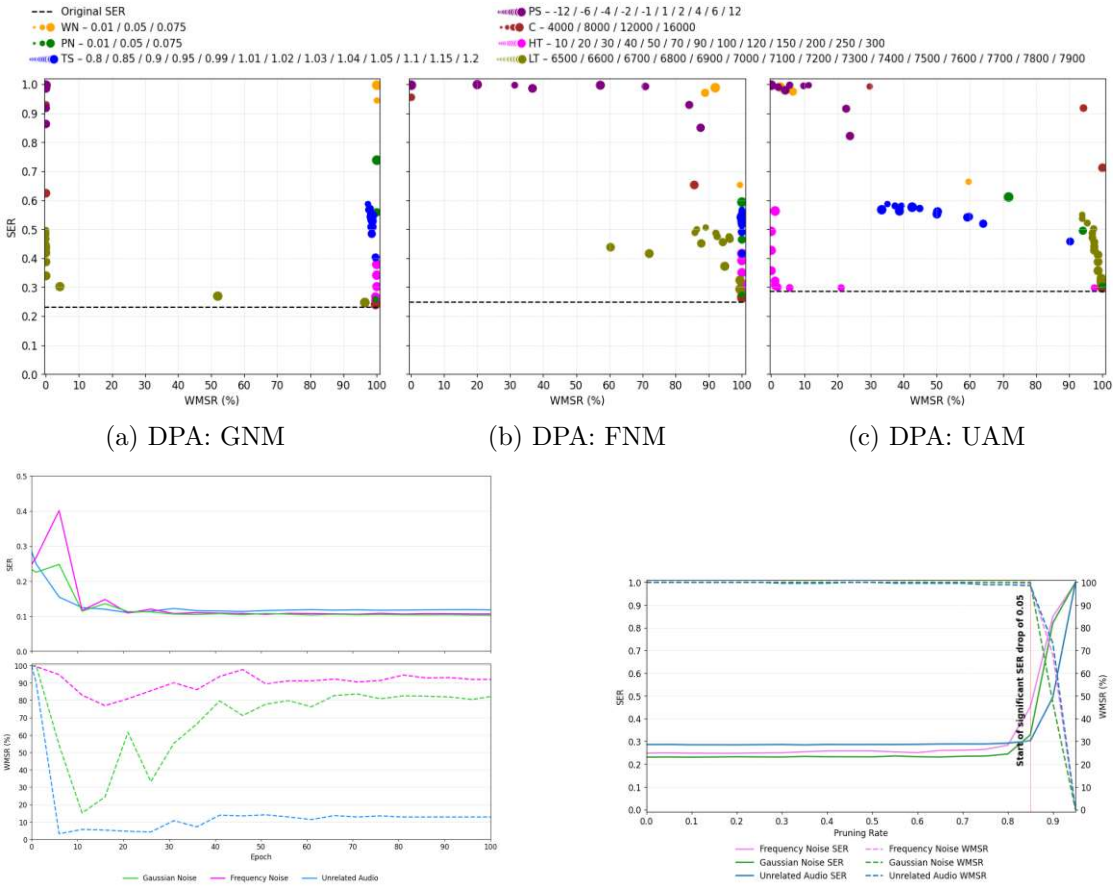
Figure 6.45: Robustness evaluation of the watermarking methods with the most imperceptible parameter settings applied to TIMIT AM-MobileNet.

6.5.2 VoxCeleb1

In the following, we present the robustness evaluation results for the VoxCeleb1 dataset carried out according to the setup described in Section 5.8 across three models: ResNet18, ResNet34, and AutoSpeech.

ResNet18

Default WM parameter settings As we can see in Figure 6.46b, only the Frequency Noise model withstands all data preprocessing attacks. The LT attack applied to the Gaussian Noise model (Figure 6.46a) with a high frequency cut-off parameter of 7,800 Hz can decrease the WMSR to 51.9% while increasing SER to 0.269, compared to the original 0.2306 SER of the Gaussian Noise model. Moreover, we can observe that the Unrelated Audio model (Figure 6.46c) is way less robust against the data preprocessing attacks that maintain high frequencies than it was in the case of the TIMIT dataset.



(a) DPA: GNM (b) DPA: FNM (c) DPA: UAM

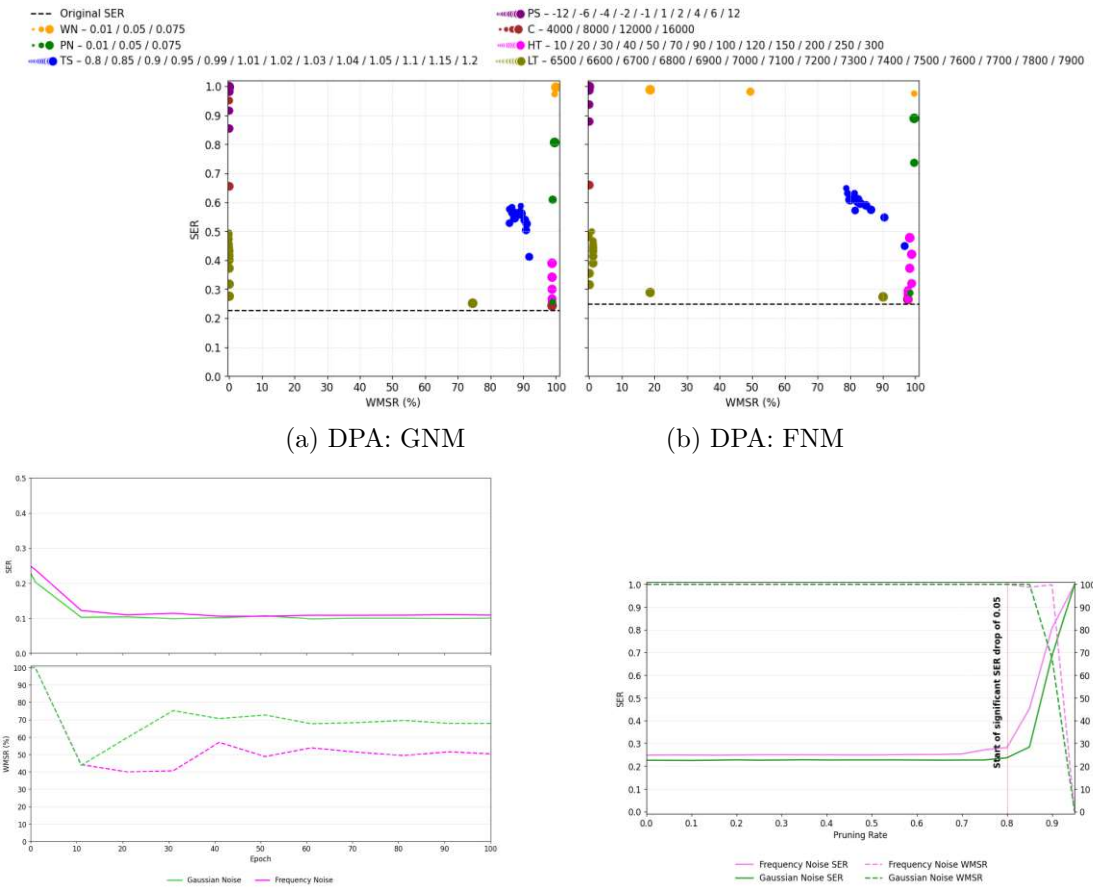
(d) Fine-tuning attacks. (e) Pruning attack.

Figure 6.46: Robustness evaluation of the SOTA watermarking methods applied to VoxCeleb1 ResNet18.

With only a 0.0124 decrease of fidelity, it is possible to reduce the WMSR to 21%, and with a 0.0364 fidelity decrease, the watermark is reduced to a WMSR of only 1.05%.

The fine-tuning attack (Figure 6.46d) is generally more successful for ResNet18 compared to the previously discussed models, since the SER of the models also gets reduced in the process. The watermark that is the least robust is the Unrelated Audio. After stabilising, the WMSR drops to 12.82%. Fine-tuning Gaussian Noise and Frequency Noise models leads to a somewhat decreased watermark performance: the Gaussian Noise model has its WMSR decreased to 82.56%, while the Frequency Noise model shows 92% WMSR. The weight pruning attack in Figure 6.46e does not remove a watermark before a substantial decrease in fidelity.

Imperceptible WM parameter settings The most effective data preprocessing attack applied to the most imperceptible Gaussian Noise and Frequency Noise watermarks



(a) DPA: GNM (b) DPA: FNM

(c) Fine-tuning attacks. (d) Pruning attack.

Figure 6.47: Robustness evaluation of the watermarking methods with the most imperceptible parameter settings applied to VoxCeleb1 ResNet18.

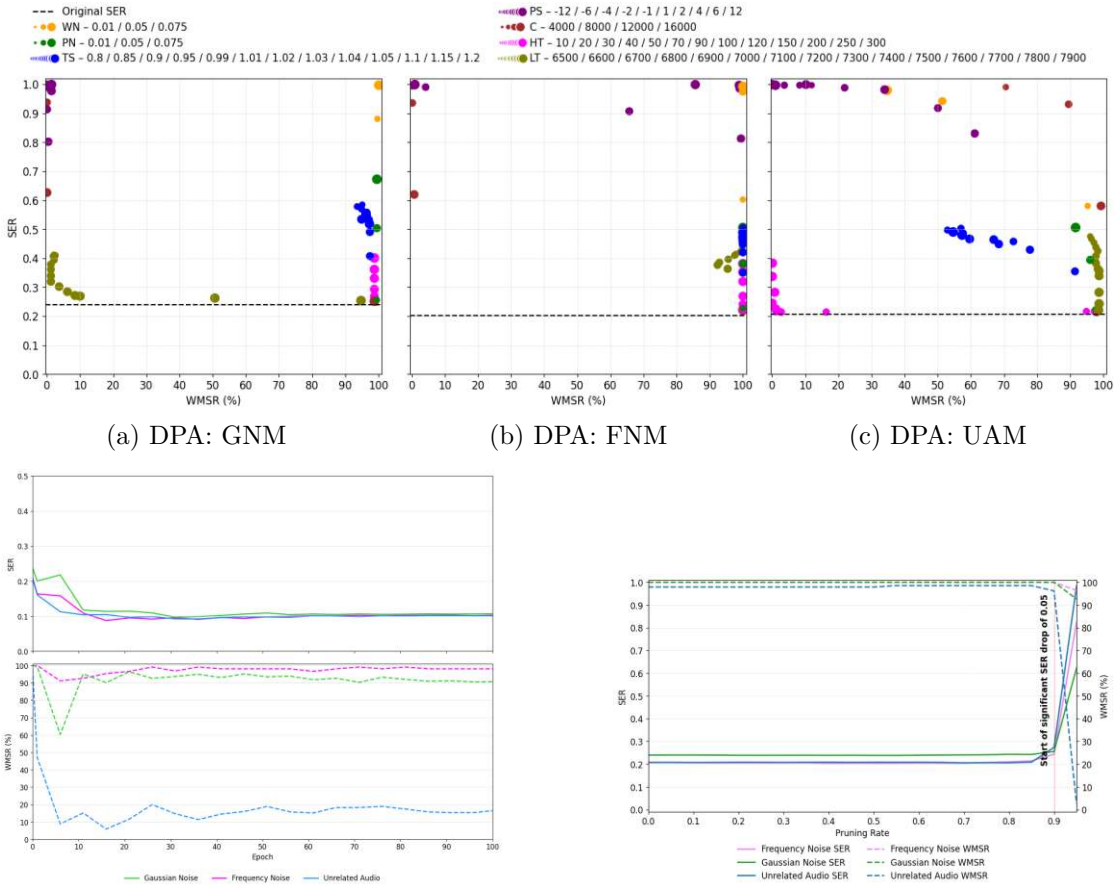
is removing high frequencies, which is shown in Figures 6.47a and 6.47b, respectively. With a 0.0275 increase in SER achieved by LT = 7,900 Hz, the WMSR drops to 74.37%, while removing frequencies higher than 7,800 Hz results in a 0.0506 increase in SER, and the watermark is completely removed (0% WMSR) from the Gaussian Noise model. The LT attack with values of 7,900 Hz and 7,800 Hz also effectively reduces the WMSR of the Frequency Noise model to 89.5% and 18.7%, resulting in a rise of SER by 0.0273 and 0.0432, respectively.

The models during fine-tuning stabilise after approximately 50 epochs, as shown in Figure 6.47c, and the watermark effectiveness drops lower than in the original settings: WMSR of the Gaussian Noise model falls to 68% and of the Frequency Noise model to 50%, which is 14% and 32% lower. The weight pruning in Figure 6.47d does not lead to a successful attack result.

ResNet34

Default WM parameter settings The same way as with ResNet18, the Frequency Noise model is not affected by data preprocessing attacks, as can be seen in Figure 6.48b. In the meantime, the Gaussian Noise (Figure 6.48a) watermark success rate drops to 6.1% with a trade-off of 0.0461 in SER when LT = 7,500 Hz; or with just a 0.0246 SER loss, WMSR falls to 50.63% when LT = 7,800 Hz. For the Unrelated Audio model (Figure 6.48c), the cost of full removal of the watermark is 0.0275 of the model fidelity, which can be achieved by maintaining the frequencies higher than 120 Hz (HT attack). Removing the lowest 40 Hz frequencies leads to a 0.0095 decrease in fidelity and a WMSR drop to 16.18%.

The fine-tuning of the Frequency Noise model does not remove the watermark in its full capacity. In Figure 6.48d we can observe a slight decrease in WMSR of around 3-4% after the first 30 epochs of fine-tuning. The Gaussian Noise model behaves worse, but the WMSR still does not drop below 90%, except for the initial drop to 65% in the first five epochs of fine-tuning. The Unrelated Audio model is the most affected one, the same as in the case of ResNet18. Fine-tuning attack almost fully removes the watermark, resulting in an 18% WMSR. Weight pruning shows to be ineffective for removing a watermark from ResNet34 as well (Figure 6.48e).



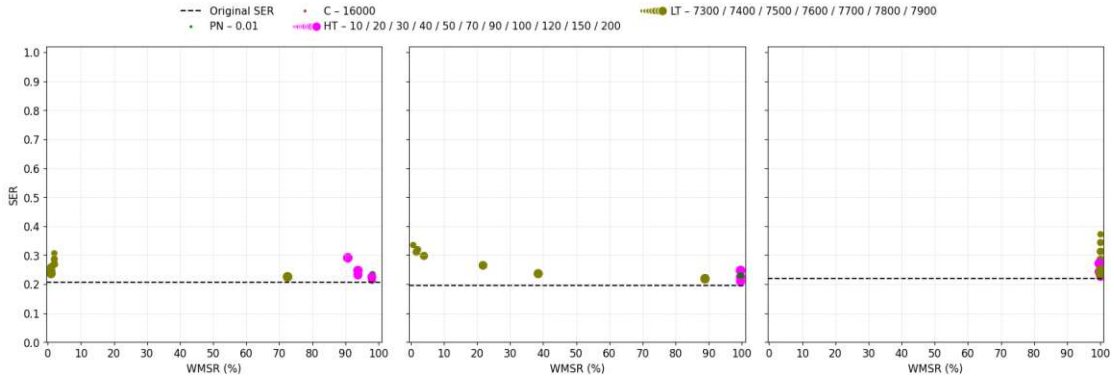
(a) DPA: GNM (b) DPA: FNM (c) DPA: UAM

(d) Fine-tuning attacks. (e) Pruning attack.

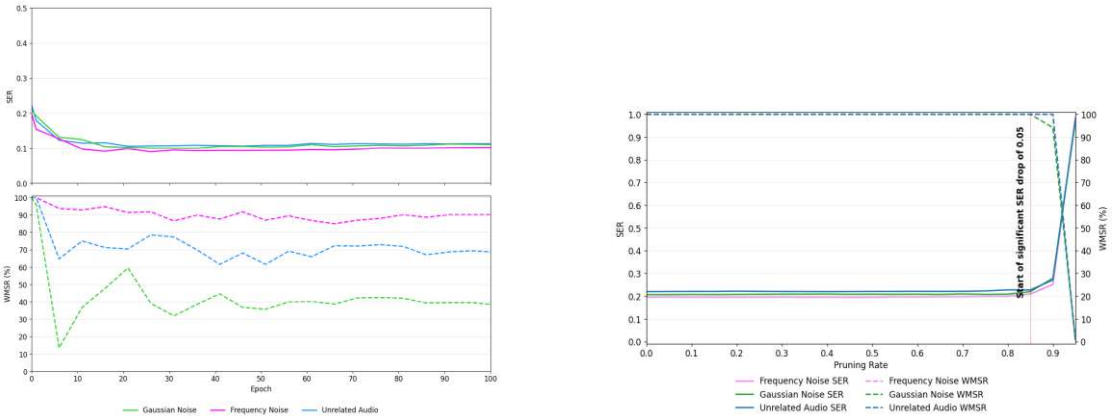
Figure 6.48: Robustness evaluation of the SOTA watermarking methods applied to VoxCeleb1 ResNet34.

Imperceptible WM parameter settings Our choice of an unrelated audio sample for the Unrelated Audio model was more perceptible than the original one, as with the original sample, WMSR of ResNet34 did not yield 100% (reaching 98% instead). Hence, here we can observe better robustness with regard to data preprocessing attacks. No attack successfully removed the Unrelated Audio watermark, as can be seen in Figure 6.49c. The Gaussian Noise watermark (Figure 6.49a) can be completely removed with a 0.0307 loss of fidelity – the WMSR falls to 0.84% when frequencies higher than 7800 Hz are cut out; meanwhile, applying the LT attack with 7900 Hz leads to WMSR 72.5% and 0.0192 SER increase. Furthermore, removing low frequencies such below 120 or 150 Hz also worsens the watermark effectiveness, resulting in 93.7% WMSR, sacrificing 0.026 of model fidelity. The Frequency Noise watermark (Figure 6.49b) also becomes more susceptible to LT attacks, with the best result being achieved by LT = 7,800 Hz, leading to 38.45% WMSR and 0.0414 fidelity decline.

Compared to the ResNet18 models watermarked by original watermarks, we can see that Gaussian Noise and Frequency Noise models are less robust against fine-tuning attacks (Figure 6.49d). Their WMSR falls to 40% and 90%, respectively. On the contrary, the Unrelated Audio model shows better results: its WMSR does not drop below the 70% mark, compared to the previous 18%. The weight pruning does not result in watermark removal based on the findings depicted in Figure 6.49e.



(a) DPA: GNM (b) DPA: FNM (c) DPA: UAM



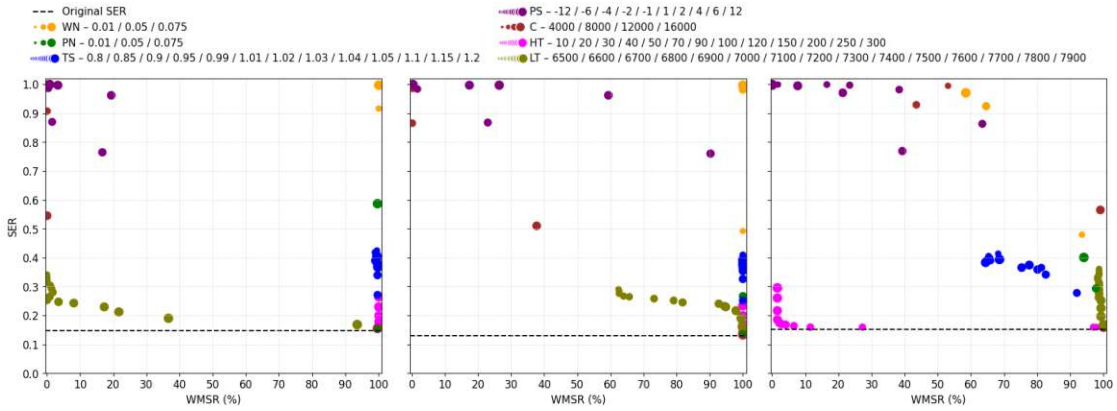
(d) Fine-tuning attacks. (e) Pruning attack.

Figure 6.49: Robustness evaluation of the watermarking methods with the most imperceptible parameter settings applied to VoxCeleb1 ResNet34.

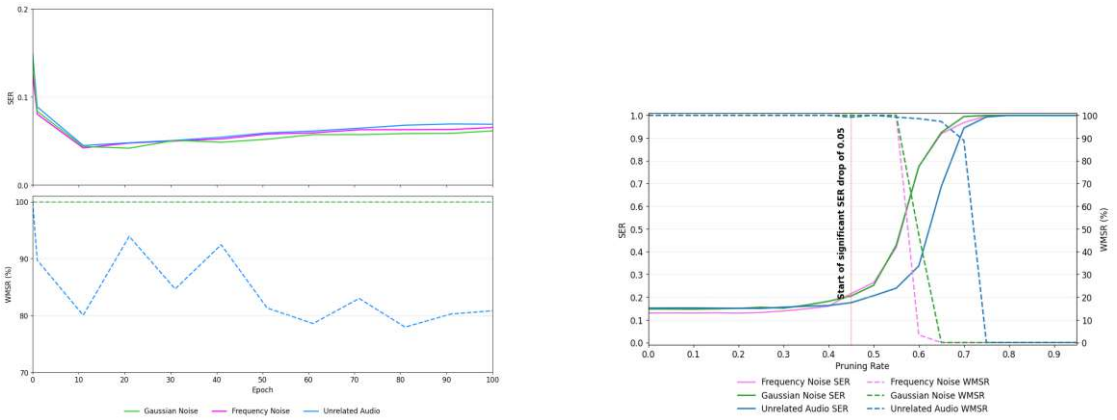
AutoSpeech

Default WM parameter settings The same as with other models trained using VoxCeleb1, the Gaussian Noise model (Figure 6.50a) suffers a WMSR drop when performing LT attacks that remove high frequencies, while the Unrelated Audio model (Figure 6.50c) is not robust against HT attacks that remove low frequencies. To be more specific, the LT attack with 7,800 Hz yields 36.55% WMSR with a 0.044 model fidelity decline. Applying an HT attack with cutting off at 50 Hz leads to a 0.01 increase of SER,

and a WMSR of 11.35%. In order to reduce WMSR to 1.47%, one can apply HT with 150 Hz, which will cost 0.0346 fidelity loss. The Frequency Noise model (Figure 6.50b) successfully withstands the data preprocessing attacks, meaning there is no attack that worsens WMSR while preserving model fidelity.



(a) DPA: GNM (b) DPA: FNM (c) DPA: UAM



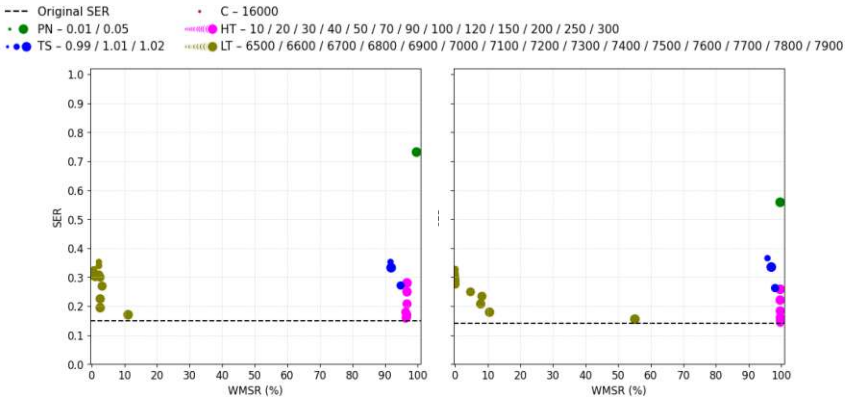
(d) Fine-tuning attacks. (e) Pruning attack.

Figure 6.50: Robustness evaluation of the SOTA watermarking methods applied to VoxCeleb1 AutoSpeech.

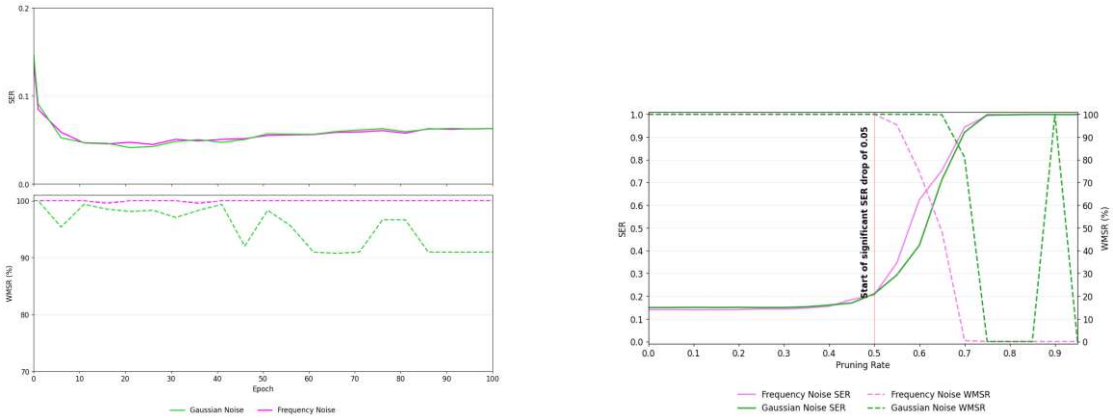
When it comes to fine-tuning, both Gaussian Noise and Frequency Noise models are unaffected in terms of WMSR decrease, which is illustrated in Figure 6.50d. The only watermark that is affected by the fine-tuning attack is the Unrelated Audio model, which WMSR drops to 80%. Weight pruning attack does not lead to the watermark removal as shown in Figure 6.50e.

Imperceptible WM parameter settings Compared to the results of the data preprocessing attacks against AutoSpeech models watermarked with the original parameter settings, the robustness of both Gaussian Noise (Figure 6.51a) and Frequency Noise

(Figure 6.51b) models is reduced. We can see that LT attacks are the most successful for both. Applying the LT attack with 7,900 Hz to the Gaussian Noise model results in 11.14% WMSR with a SER loss of 0.022. The LT attack with 7,800 Hz results in a bigger fidelity loss, specifically 0.0458, but it also almost completely erases the watermark – WMSR 2.5%. The same attacks applied to the Frequency Noise model result in 0.015 and 0.031 SER loss for 7,900 and 7,800 Hz, respectively. At the same time, the WMSR decreases to 63% and 10.5% in these cases.



(a) DPA: GNM (b) DPA: FNM



(c) Fine-tuning attacks. (d) Pruning attack.

Figure 6.51: Robustness evaluation of the watermarking methods with the most imperceptible parameter settings applied to VoxCeleb1 AutoSpeech.

The fine-tuning attack has no effect on the Frequency Noise model; however, it decreases the WMSR of the Gaussian Noise model to 91%, which is shown in Figure 6.51c. Weight pruning results in an extreme decline in fidelity before any watermark is removed (Figure 6.51d).

6.5.3 Robustness Evaluation Findings

In Table 6.26, we present the general overview of the performed attacks on the watermarked models. GNM denotes the model watermarked by the Gaussian Noise watermarking method, while IGNM denotes the model watermarked using the Gaussian Noise triggers created with imperceptible parameters defined in Section 6.3. The same notation applies to FNM and UAM. The results are represented according to the definition of a successful attack:

✓ Watermark is robust: WMSR$\geq$ 95% or SER$\geq$ 0.05

✗ Attack is successful: WMSR$<$95% and SER$<$0.05

Data preprocessing attacks have mixed effects on watermarking methods. Some increase SER too much (>0.05), e.g., all configurations of pitch shifting (PS), most configurations of maintaining a bandpass (B), and time stretching (TS). The latter two, as well as three Gaussian noise attacks (SNR, WN, PN) and compression (C), in fact do not succeed in removing the watermark without degrading model fidelity in any configuration, whereas keeping low (LT) and high frequencies (HT) succeed in some cases. That is why some of the attacks are removed from the view of Table 6.26, namely SNR, three of the configurations of TS, and bandpass filtering (B). We conducted extra pink noise addition (PN) attacks to verify the robustness of MobileNet and AM-MobileNet due to the good results of SER with the originally proposed values.

The LT attack is effective against Gaussian Noise models, as 5 out of 6 dataset-model combinations have cases when WMSR falls below 95%. This is the only data preprocessing attack that was able to remove a watermark from the models trained on the TIMIT dataset. For MobileNet, the cut-off threshold is 6,700 Hz, for AM-MobileNet 6,900 Hz, and for ResNet models and AutoSpeech 7,800 Hz. In case of the most imperceptible watermarks, for the SinceNet model, the threshold becomes 7,700 Hz, 7,300 Hz for MobileNet, 6,800 Hz for AM-MobileNet, and 7,900 Hz for ResNet and AutoSpeech models.

HT with the cut-off at 40 Hz is effective against UAM applied on ResNets and AutoSpeech for the unrelated audio sample with a noise variance of 0.0000578. The same attack on VoxCeleb1 triggers created with the unrelated audio sample with $noise\_var$ =0.00043166 does not affect the watermarked part at all up to 150 Hz, where WMSR becomes 99.6%. Our experiments revealed that the choice of unrelated audio can significantly impact both the imperceptibility of the watermark and its robustness to perturbations. This suggests that the trigger audio selection, while presented as arbitrary by the authors, may function as a critical hyperparameter in this watermarking approach.

The only case where both HT and LT attacks have succeeded in removing a watermark is on ResNet34 with IGNM created using $\lambda = 0.2$. In this case, with HT $\geq$ 120 Hz, the WMSR starts to drop below 95%, namely to 93.7%. The LT cut-off point is at 7,700 Hz.

The only successful fine-tuning attack applied to a model trained on the TIMIT dataset was the one conducted on the AM-MobileNet model watermarked with Gaussian Noise. Moreover, since in general the SER of AM-MobileNet models worsened during fine-tuning, the only point when the attack was successful was the first epochs of fine-tuning, while SER stayed within the defined bounds of 0.05 cost.

For the models trained on the VoxCeleb1 dataset, we observe a different pattern of the fine-tuning attack. Since the SER on the subset of the test set improved during fine-tuning, which can be attributed to a three times higher number of classes and a much higher number of audio samples in the dataset compared to TIMIT, the attack showed successful results more often. The overall pattern in ResNet18, ResNet34, and AutoSpeech first exhibited a high drop in watermark effectiveness and a simultaneous peak in model fidelity, then a gradual stabilisation. Sometimes, it led to WMSR returning to an acceptable level of WMSR ($\geq 95\%$), such as for FNM on ResNet34. The Frequency Noise model on AutoSpeech was not affected by fine-tuning at all: neither in its original state, nor in its more imperceptible Extreme Gaussian Noise alternative. The weight pruning attack proved to be ineffective in every evaluated scenario.

Based on the insights above, each model presented in the table has at least one setting that withstands all robustness checks:

- SincNet: GNM, IGNM, UAM, IUAM
- MobileNet: FNM, UAM, IUAM
- AM-MobileNet: FNM, UAM, IUAM
- ResNet18: FNM
- ResNet34: FNM, IUAM
- AutoSpeech: FNM.

Overall, we can clearly see the trade-off between robustness and imperceptibility of the applied watermarks. Only in the case of Gaussian Noise applied to AM-MobileNet, the model trained with the triggers created using a more imperceptible setting withstands the fine-tuning attack better than the original GNM. In addition to that, one needs to choose the parameters for trigger creation in a way that the triggers are imperceptible enough not to be distinguished as abnormal data, but robust enough to withstand possible removal attacks.

Table 6.26: Robustness evaluation across models and datasets. ✗ denotes a successful attack configuration and ✓a robust watermark under attack configuration.

Dataset	Model	WM	SNR	WN	PN	TS	PS	C	B	HT	LT	FT	P
TIMIT	SincNet	GNM	✓4	✓3	✓3	✓13	✓10	✓4	✓4	✓13	✓15	✓3	✓
		IGNM	✓4	✓3	✓3	✓13	✓10	✓4	✓4	✓13	✓15	✓3	✓
		FNM	✓4	✓3	✓3	✓13	✓10	✓4	✓4	✓13	✓15	✓3	✓
		IEGNM	✓4	✓3	✓3	✓13	✓10	✓4	✓4	✓13	✓14 ✗1	✓3	✓
		UAM	✓4	✓3	✓3	✓13	✓10	✓4	✓4	✓13	✓15	✓3	✓
		IUAM	✓4	✓3	✓3	✓13	✓10	✓4	✓4	✓13	✓15	✓3	✓
	MNet	GNM	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓15 ✗2	✓4	✓
		IGNM	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓14 ✗3	✓4	✓
		FNM	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓17	✓4	✓
		IEGNM	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓14 ✗3	✓4	✓
		UAM	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓17	✓4	✓
		IUAM	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓17	✓4	✓
	AM-MNet	GNM	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓16 ✗1	✓3 ✗1	✓
		IGNM	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓9 ✗8	✓4	✓
		FNM	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓17	✓4	✓
		IEGNM	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓13 ✗4	✓4	✓
		UAM	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓17	✓4	✓
		IUAM	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓17	✓4	✓
VoxCeleb1	ResNet18	GNM	-	✓3	✓3	✓13	✓10	✓4	-	✓13	✓14 ✗1	✗	✓
		IGNM	-	✓3	✓3	✓13	✓10	✓4	-	✓13	✓14 ✗1	✗	✓
		FNM	-	✓3	✓3	✓13	✓10	✓4	-	✓13	✓15	✗	✓
		IEGNM	-	✓3	✓3	✓13	✓10	✓4	-	✓13	✓13 ✗2	✗	✓
		UAM	-	✓3	✓3	✓13	✓10	✓4	-	✓7 ✗6	✓15	✗	✓
	ResNet34	GNM	-	✓3	✓3	✓13	✓10	✓4	-	✓13	✓10 ✗5	✗	✓
		IGNM	-	✓3	✓3	✓13	✓10	✓4	-	✓11 ✗2	✓12 ✗3	✗	✓
		FNM	-	✓3	✓3	✓13	✓10	✓4	-	✓13	✓15	✓	✓
		IEGNM	-	✓3	✓3	✓13	✓10	✓4	-	✓13	✓13 ✗2	✗	✓
		UAM	-	✓3	✓3	✓13	✓10	✓4	-	✓5 ✗8	✓15	✗	✓
		IUAM	-	✓3	✓3	✓13	✓10	✓4	-	✓13	✓15	✗	✓
	AutoSpeech	GNM	-	✓3	✓3	✓13	✓10	✓4	✓1	✓13	✓13 ✗2	✓	✓
		IGNM	-	✓3	✓3	✓13	✓10	✓4	-	✓13	✓13 ✗2	✗	✓
		FNM	-	✓3	✓3	✓13	✓10	✓4	✓1	✓13	✓15	✓	✓
		IEGNM	-	✓3	✓3	✓13	✓10	✓4	-	✓13	✓13 ✗2	✓	✓
		UAM	-	✓3	✓3	✓13	✓10	✓4	✓1	✓6 ✗7	✓15	✗	✓

6.6 Trade-off between Robustness and Imperceptibility

Here, we discuss in detail the trade-off between imperceptibility and robustness for various dataset-model pairs, and afterward, we determine an optimal approach to balance this trade-off. We explore the dataset-model-watermark settings in which the choice of a more imperceptible watermark led to a deterioration of robustness that we discussed above in Section 6.5. Among the models for the VoxCeleb1 dataset, we explore only the parameter configurations for the AutoSpeech model due to the computational constraints. The AutoSpeech model is chosen as the best-performing model for VoxCeleb1.

6.6.1 TIMIT

SincNet

For the SincNet model, only enhancing the imperceptibility of the Frequency Noise watermark led to worse robustness. Thus, it is the only watermarking method evaluated for this model below.

Frequency Noise Since we determined that $\lambda = 0.2$ with the Extreme Gaussian Noise variant of Frequency Noise watermark yields triggers, whose imperceptibility can be measured in SNR ≥ 30 dB, we experimented with the values of extreme low and extreme high frequencies. The idea is that changing them does not drastically influence the LSD and SNR values, but since they regulate the surface of overall watermark embedding, they might have a bigger effect on robustness. Hence, in Table 6.27 we present the trigger creation parameters of the original Frequency Noise watermark, the most imperceptible setting chosen in the previous section, and the ones that we further evaluated in terms of robustness. As can be seen, the ELF and EHF parameters influence the LSD metric, but in terms of SNR, their values differ by 0.001 dB.

Table 6.27: Evaluation of Frequency Noise watermark applied to TIMIT dataset. Results shown on the SincNet model.

WM	$mfcc\_sc$	λ	ELF	EHF	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	Imperceptibility SNR↑
Frequency Noise	2.0	1.0	300	3,000	100%	0.0108	20.6664	-1.3025
	-	0.2	100	6,000	100%	0.0129	6.0068	30.6359
Extreme Gaussian Noise	**-**	**0.2**	**150**	**5,000**	**100%**	**0.0173**	**6.6046**	**30.6361**
	-	0.2	200	4,500	100%	0.0188	6.7246	30.6373

Figure 6.52 depicts the results of the LT attack on the SincNet model watermarked according to the WM settings presented in Table 6.27. The red colour of the points indicates a successful removal according to our definition: SER does not worsen by more than 0.05 while WMSR drops below 95%. We can see that only the attack on the model with creation parameters $\{\lambda = 0.2$, ELF=100, EHF=6,000$\}$ is successful; others

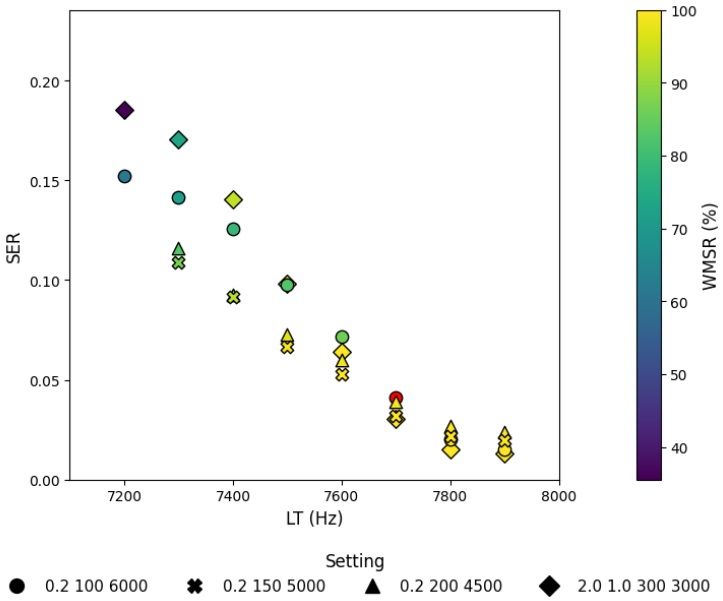
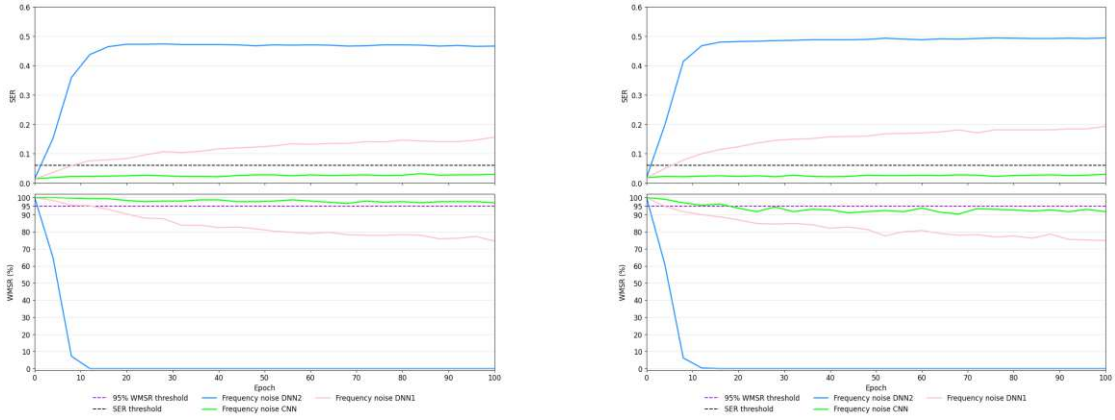


Figure 6.52: The effect of LT data preprocessing attack with various high frequencies on WMSR and SER of different Frequency Noise SincNet models. Successful attacks are marked in red.



(a) Fine-tuning attacks performed on EGNM with $\lambda = 0.2$, ELF=150, EHF=5,000

(b) Fine-tuning attacks performed on EGNM with $\lambda = 0.2$, ELF=200, EHF=4,500

Figure 6.53: Fine-tuning attacks performed on SincNet models watermarked with different variants of Extreme Gaussian Noise watermark.

withstand it well. Hence, we perform a fine-tuning attack on the other two models that have wider band frequency coverage for the watermark. Figure 6.53 shows the results of fine-tuning attacks on the two models of SincNet. We can see that while in Figure 6.53a, the fine-tuning of different layers does not drop the WMSR below the threshold of 95%,

in Figure 6.53b, the fine-tuning of the CNN layer results in WMSR falling to 90%.

Based on the above experiments, we can determine that using a more imperceptible watermark (tuned by λ parameter) but with enough coverage of the frequency band, we can achieve an imperceptible and robust watermark.

MobileNet

For MobileNet, two of the watermarking methods with parameters that create more imperceptible watermarks result in a decrease of robustness: Gaussian Noise and Frequency Noise. These are the ones that we investigate further.

Gaussian Noise For the Gaussian Noise watermark, we take the four values of λ, other than the original one and the most imperceptible one, to compare their robustness. In Table 6.28 are the results of the individual runs of these models. Since the attack to which the Gaussian noise models are most susceptible is LT, in Figure 6.54, we present the comparison of LT attacks applied to these models. We can observe the following: $\lambda = 0.2$ results in three successful attacks, as well as $\lambda = 0.3$; $\lambda = 0.4$ yields two successful attacks; $\lambda = 0.5$ and $\lambda = 0.6$ yield only one successful attack. Taking into consideration the fact that the Gaussian Noise MobileNet model with $\lambda = 0.5$ withstands all four fine-tuning attacks, it makes it more robust than the model with $\lambda = 1.0$.

Table 6.28: Evaluation of Gaussian Noise watermark applied to TIMIT dataset. Results shown on the MobileNet model.

WM	λ	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	SNR↑
Gaussian Noise	1.0	100%	0.0065	13.0948	16.6514
	0.2	100%	0.0043	6.3771	30.6293
	0.3	100%	0.0022	7.7801	27.1078
	0.4	100%	0.0065	8.8952	24.6092
	0.5	**100%**	**0.0050**	**9.8287**	**22.6711**
	0.6	100%	0.0050	10.6348	21.0876

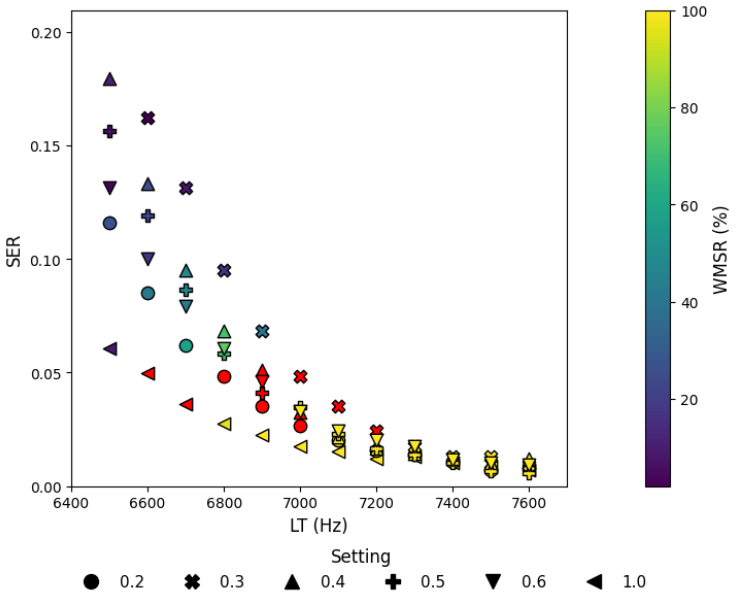


Figure 6.54: The effect of LT data preprocessing attack with various high frequencies on WMSR and SER of different Gaussian Noise MobileNet models. Successful attacks are marked in red.

Frequency Noise As well as for the SincNet model with Frequency Noise watermark, we evaluate the different settings of the Extreme Gaussian Noise variant to see which of them leads to the most robust watermark. We evaluate the same settings as in the SincNet model. As can be seen in Table 6.29, the SERs of the other two variants are the same or even smaller compared to the original watermark settings.

Table 6.29: Evaluation of Frequency Noise watermark applied to TIMIT dataset. Results shown on the MobileNet model.

WM	$mfcc\_sc$	λ	ELF	EHF	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	Imperceptibility SNR↑
Frequency Noise	2.0	1.0	300	3,000	100%	0.0058	20.6664	-1.3025
Extreme Gaussian Noise	-	0.1	100	7,000	100%	0.0058	3.6815	36.6688
	-	**0.2**	**150**	**5,000**	**100%**	**0.0058**	**6.6046**	**30.6361**
	-	0.2	200	4,500	100%	0.0029	6.7246	30.6373

In Figure 6.55, we can clearly see that the LT attack on the original Frequency Noise watermark deteriorates neither SER nor WMSR, while the more imperceptible the watermark is, the easier it is to erase it, though SER worsens faster as well. Based on our defined threshold, the Extreme Gaussian Noise variant with parameters $\{\lambda = 0.2$, ELF=150, EHF=5,000$\}$ results in the least number of successful attacks, namely only one: LT 7,000 Hz. Moreover, verifying the fine-tuning attack, it does not lead to watermark

removal.

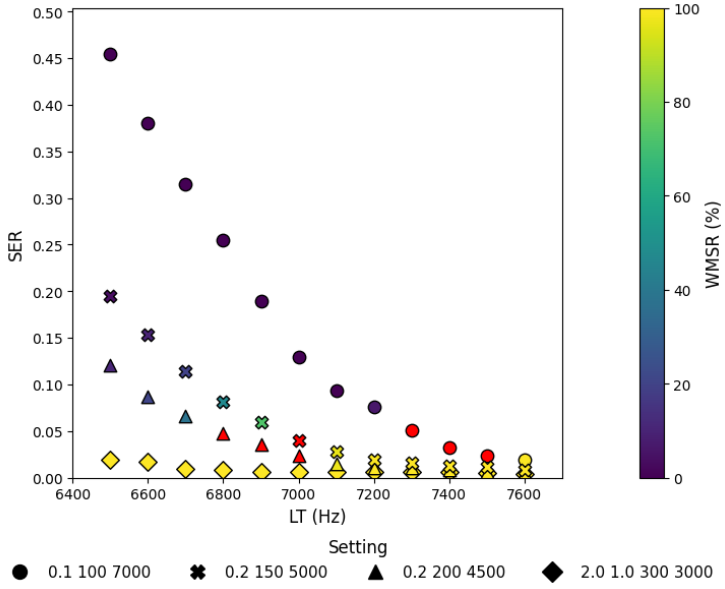


Figure 6.55: The effect of LT data preprocessing attack with various high frequencies on WMSR and SER of different Frequency Noise MobileNet models. Successful attacks are marked in red.

AM-MobileNet

For AM-MobileNet, the situation is similar to MobileNet: only Gaussian Noise and Frequency Noise, when made more imperceptible, suffer from a loss in robustness. These are the methods we investigate further.

Gaussian Noise In Table 6.30, we list the settings that we evaluate. From what we

Table 6.30: Evaluation of Gaussian Noise watermark applied to TIMIT dataset. Results shown on the AM-MobileNet model.

WM	λ	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	SNR↑
Gaussian Noise	1.0	100%	0.0036	13.0948	16.6514
	0.1	100%	0.0036	4.4207	36.6494
	0.2	100%	0.0043	6.3771	30.6293
	0.3	100%	0.0050	7.7801	27.1078
	0.4	100%	0.0050	8.8952	24.6092
	0.5	**100%**	**0.0050**	**9.8287**	**22.6711**

observe in Figure 6.56, the Gaussian Noise model with $\lambda = 0.3$ yields the smallest number of LT attacks that successfully remove the watermark without deteriorating SER too much. Then it is followed by $\lambda = 0.3$ and $\lambda = 0.5$. However, looking also at fine-tuning attack in Figure 6.57, we see that with $\lambda = 0.4$, the WMSR falls to 0% right away. With

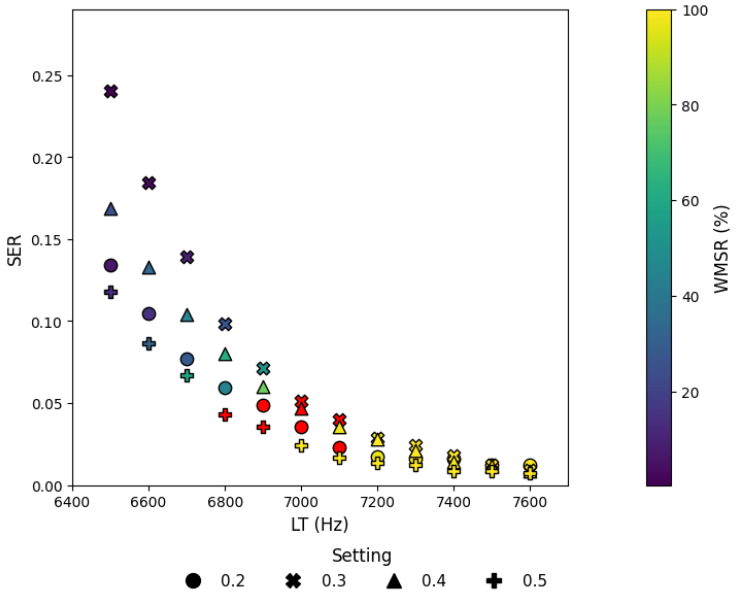
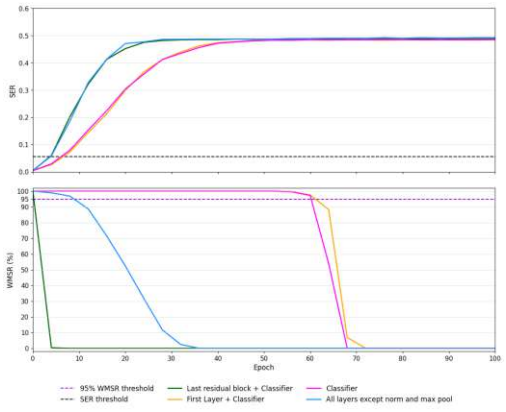
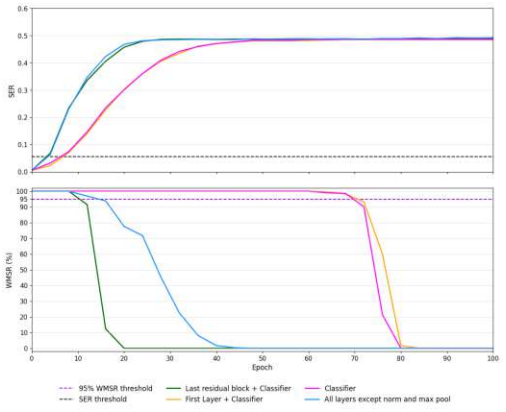


Figure 6.56: The effect of LT data preprocessing attack with various high frequencies on WMSR and SER of different Gaussian Noise AM-MobileNet models. Successful attacks are marked in red.



(a) Fine-tuning attack performed on AM-MobileNet watermarked with Gaussian Noise; $\lambda = 0.4$.



(b) Fine-tuning attack performed on AM-MobileNet watermarked with Gaussian Noise; $\lambda = 0.5$.

Figure 6.57: Fine-tuning attack performed on Gaussian Noise AM-MobileNet.

$\lambda = 0.5$, the drop happens later, when SER increases immensely. Hence, with the fidelity cost of 0.0014, the AM-MobileNet model can be watermarked, so that it withstands all fine-tuning attacks and 64 out of 66 other data preprocessing attacks.

Frequency Noise Following the choices of Frequency Noise settings from the SincNet and MobileNet models, in Table 6.31 are listed the settings we analyse. We additionally evaluated the setting with EHL=100 and EHF=6,100 to compare the results with a larger value of λ, while keeping the frequencies in a relatively close range. We can observe in Figure 6.58 that the closer the frequencies are to the extremums with the same value of λ, the more susceptible the watermark is to the LT attack. As can be observed, LT attacks have only two successful results for the parameter setting of the Extreme Gaussian Noise model with parameters $\{\lambda = 0.2$, ELF=200, EHF=4,500$\}$, compared to the four attacks

Table 6.31: Evaluation of Frequency Noise watermark applied to TIMIT dataset. Results shown on the MobileNet model.

WM	$mfcc\_sc$	λ	ELF	EHF	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	Imperceptibility SNR↑
Frequency Noise	2.0	1.0	300	3,000	100%	0.0043	20.6664	-1.3025
	-	0.1	100	6,000	100%	0.0065	3.6815	36.6688
Extreme Gaussian Noise	-	0.2	100	6,100	100%	0.0072	5.9250	30.6378
	-	0.2	150	5,000	100%	0.0065	6.6046	30.6361
	-	**0.2**	**200**	**4,500**	**100%**	**0.0065**	**6.7246**	**30.6373**

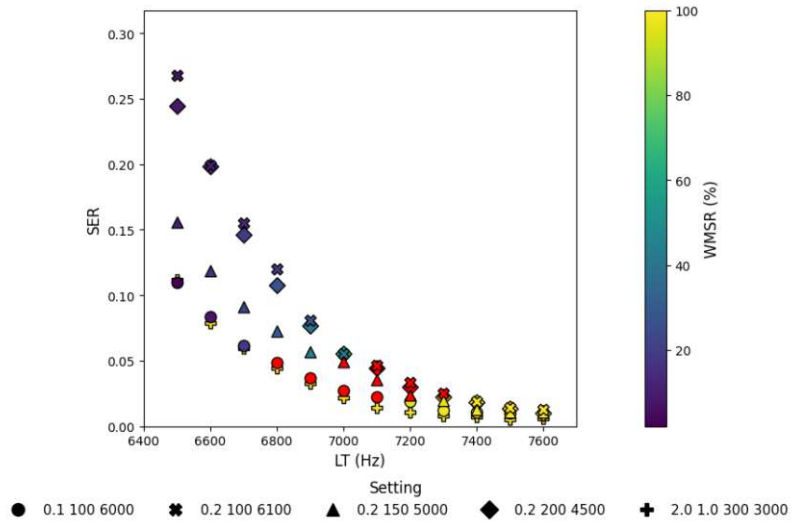


Figure 6.58: The effect of LT data preprocessing attack with various high frequencies on WMSR and SER of different Frequency Noise AM-MobileNet models. Successful attacks are marked in red.

for the parameter setting of the Extreme Gaussian Noise model with {$\lambda = 0.1$, ELF=100, EHF=6,000}. The model is also robust against fine-tuning attacks.

6.6.2 VoxCeleb1

AutoSpeech

Similarly, for the AutoSpeech model, only Gaussian Noise and Frequency Noise watermarks in more imperceptible settings exhibited reduced robustness. These methods are examined further below.

Gaussian Noise Here, we considered other λ values for the Gaussian Noise model that we used in the Imperceptibility evaluation, but only the ones that yielded 100% WMSR. In Table 6.32 are shown these values. From Figure 6.59, it can be seen that all models except one have at least two cases where the watermark can be successfully removed. Only with $\lambda = 0.5$, there is just one successful attack, namely LT with the value 7,900 Hz. The model with $\lambda = 0.5$ also withstands the fine-tuning attack, with the lowest WMSR being 96.22%. With such results, this model is more robust than the original $\lambda = 1.0$ one, but it comes with the cost of fidelity of 0.0264.

Table 6.32: Evaluation of Gaussian Noise watermark applied to VoxCeleb1 dataset. Results shown on the AutoSpeech model.

WM	λ	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	SNR↑
Gaussian Noise	1.0	100%	0.1479	13.5594	15.9199
	0.2	100%	0.1492	6.7661	29.9027
	0.3	100%	0.1529	8.1826	26.3834
	0.5	**100%**	**0.1743**	**10.247**	**21.9467**

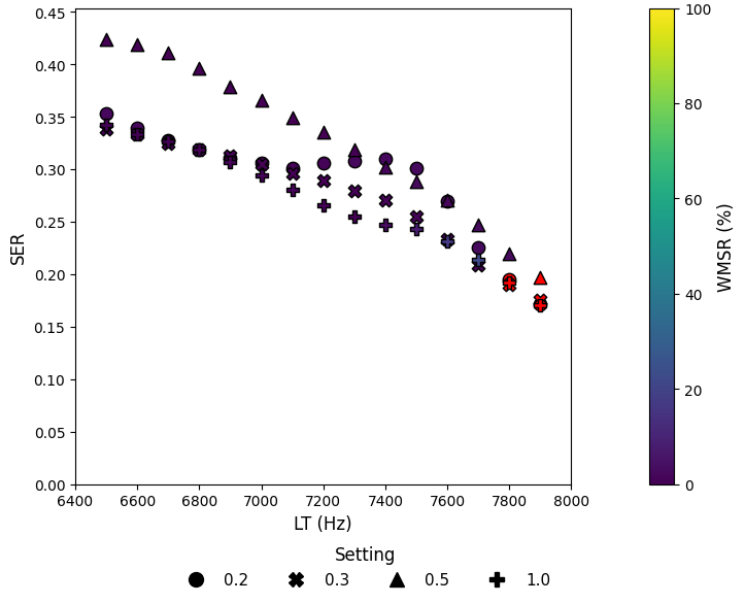


Figure 6.59: The effect of LT data preprocessing attack with various high frequencies on WMSR and SER of different Gaussian Noise AutoSpeech models. Successful attacks are marked in red.

Frequency Noise For the Frequency Noise watermark, we evaluated the robustness of the top-2 models based on the SER (not counting the original and the imperceptible ones), more specifically, the models watermarked by the watermarks shown in Table 6.33. In Figure 6.60, we can see how they perform under the LT data preprocessing attack. Only two models withstand all of them: the original watermark, and the one with parameters $\{\lambda = 1.0$, ELF=100, EHF=6,000$\}$. The resulting SNR of this watermark is 15.93, which is below the acceptable SNR imperceptibility threshold of 20 dB. The fine-tuning attack on this model does not remove the watermark, nor is it successful on other Frequency Noise AutoSpeech models.

Table 6.33: Evaluation of Frequency Noise watermark applied to VoxCeleb1 dataset. Results shown on the AutoSpeech model.

WM	$mfcc\_sc$	λ	ELF	EHF	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	SNR↑
Frequency Noise	2.0	1.0	300	3,000	100%	0.1298	22.5900	-2.6448
	-	0.2	200	4,500	100%	0.1400	7.2651	29.9050
Extreme Gaussian Noise	**-**	**1.0**	**100**	**6,000**	**100%**	**0.1447**	**12.0189**	**15.9296**
	-	0.2	150	5,000	100%	0.1623	7.1944	29.9072

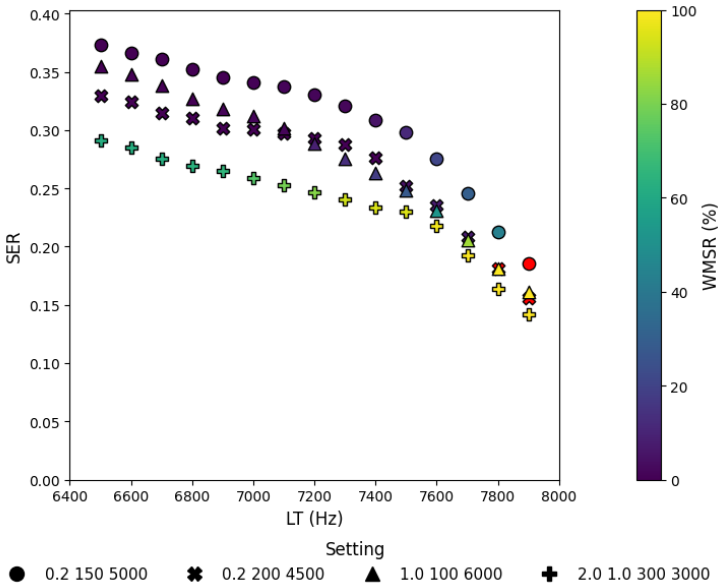


Figure 6.60: The effect of LT data preprocessing attack with various high frequencies on WMSR and SER of different Frequency Noise AutoSpeech models. Successful attacks are marked in red.

6.6.3 Trade-off Evaluation Findings

In this section, by tuning the trigger creation parameters, we improved both the imperceptibility and robustness of two watermarking methods compared to their original settings, namely the MobileNet Gaussian Noise model with $\lambda = 0.5$, and the AutoSpeech Gaussian Noise model, also created with $\lambda = 0.5$.

We achieved the same level of robustness with significantly higher imperceptibility for two models: SincNet with Frequency Noise, specifically Extreme Gaussian Noise using parameters $\{\lambda = 0.2$, ELF=150, EHF=5,000$\}$, and AutoSpeech with Frequency Noise using Extreme Gaussian Noise with parameters $\{\lambda = 1.0$, ELF=100, EHF=6,000$\}$.

In the case of MobileNet REGNM, one attack was still able to successfully remove the watermark. While this represents an improvement over the IEGNM, it remains less robust than in the original setting. Nonetheless, the imperceptibility improved by more than 31 dB relative to the original setting.

For the AM-MobileNet model, two data preprocessing attacks were successful in both Gaussian and Frequency Noise settings. Nevertheless, fine-tuning the AM-MobileNet Gaussian Noise model did not result in a successful removal of the watermark.

Table 6.35 presents the overall results obtained in this section in comparison to the ones from Sections 6.3 and 6.5. As before, the prefix 'I' before the name of a watermarking method refers to the most imperceptible setting selected in Section 6.3, while 'R' denotes

the most robust parameter configuration identified for each dataset–model–watermark combination in this section, which can be found in Table 6.34. Cells are marked green where improvements were achieved, and orange where robustness did not match the original setting or the imperceptibility fell below the required threshold.

Table 6.34: Parameter settings for the most robust variant of the watermarking methods.

Dataset	Model	WM	λ	ELF	EHF
TIMIT	SincNet	REGNM	0.2	150	5,000
	MobileNet	RGNM	0.5		
		REGNM	0.2	150	5,000
	AM-MobileNet	RGNM	0.5		
		REGNM	0.2	200	4,500
VoxCeleb1	AutoSpeech	RGNM	0.5		
		REGNM	1.0	100	6,000

The SincNet model can be watermarked effectively using all three methods, each of which is robust to various attacks. When properly tuned, their imperceptibility satisfies the minimum requirement of SNR $\geq$ 20 dB. Additionally, the Unrelated Audio watermark achieves a low LSD of 3.5 dB. In terms of fidelity, both the Gaussian Noise and Unrelated Audio methods perform best. Considering only imperceptibility and robustness, Extreme Gaussian Noise yields the best results; however, when fidelity is also taken into account, the Unrelated Audio watermark, specifically IUAM, is the preferred method based on the results presented.

For watermarking either the MobileNet or AM-MobileNet model, the best-performing watermarking method in terms of fidelity is Gaussian Noise. Extreme Gaussian Noise achieves the best results for both models, with SER values that are comparable to those of Gaussian Noise. However, in terms of robustness, the Unrelated Audio method again performs best – albeit at the greatest cost to fidelity: 0.0037 and 0.0086 for MobileNet and AM-MobileNet, respectively. All methods meet the minimum imperceptibility requirement of SNR $\geq$ 20 dB. Therefore, the final choice of method depends on which aspect – robustness, imperceptibility, or fidelity – is prioritised. Finally, since both RGNM and REGNM offer the same level of robustness, but REGNM provides higher imperceptibility, it is preferable to choose either REGNM or IUAM.

For the AutoSpeech model, the Unrelated Audio watermark is too imperceptible to withstand robustness checks, even though the SNR is lower than 20 dB. Based on the results, the REGNM model demonstrates greater robustness to attacks and yields a lower SER, while RGNM offers higher imperceptibility. Considering that VoxCeleb1 contains noisy audio recorded in various real-world environments, unlike the high-quality recordings in TIMIT used with SincNet, prioritising robustness is more appropriate in this context. Therefore, REGNM is a preferable choice here.

Table 6.35: Generality of SOTA watermarking methods: Robustness evaluation across models and datasets. ✗ denotes a successful attack configuration and ✓a robust watermark under attack configuration.

Dataset	Model	WM	Fidelity	Imperceptibility		Robustness										
			SER↓	LSD↓	SNR↑	SNR	WN	PN	TS	PS	C	B	HT	LT	FT	P
TIMIT	SN 0.0051	GNM	0.0101	13.095	16.651	✓4	✓3	✓3	✓13	✓10	✓4	✓4	✓13	✓15	✓3	✓
		IGNM	0.0094	10.635	21.087	✓4	✓3	✓3	✓13	✓10	✓4	✓4	✓13	✓15	✓3	✓
		FNM	0.0108	20.666	-1.303	✓4	✓3	✓3	✓13	✓10	✓4	✓4	✓13	✓15	✓3	✓
		IEGNM	0.0129	6.007	30.636	✓4	✓3	✓3	✓13	✓10	✓4	✓4	✓13	✓14 ✗1	✓3	✓
		REGNM	0.0173	6.605	30.636	✓4	✓3	✓3	✓13	✓10	✓4	✓4	✓13	✓15	✓3	✓
		UAM	0.0101	3.890	23.039	✓4	✓3	✓3	✓13	✓10	✓4	✓4	✓13	✓15	✓3	✓
		IUAM	0.0101	3.495	24.977	✓4	✓3	✓3	✓13	✓10	✓4	✓4	✓13	✓15	✓3	✓
	MN 0.0057	GNM	0.0065	13.095	16.651	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓15 ✗2	✓4	✓
		IGNM	0.0043	6.377	30.629	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓14 ✗3	✓4	✓
		RGNM	0.0050	9.829	22.671	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓16 ✗1	✓4	✓
		FNM	0.0058	20.666	-1.303	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓17	✓4	✓
		IEGNM	0.0058	3.682	36.669	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓14 ✗3	✓4	✓
		REGNM	0.0058	6.605	30.636	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓16 ✗1	✓4	✓
		UAM	0.0087	3.890	23.039	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓17	✓4	✓
		IUAM	0.0094	2.773	29.059	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓17	✓4	✓
	AM-MN 0.0043	GNM	0.0036	13.095	16.651	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓16 ✗1	✓3 ✗1	✓
		IGNM	0.0036	4.421	36.649	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓9 ✗8	✓4	✓
		RGNM	0.0050	9.829	22.671	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓15 ✗2	✓4	✓
		FNM	0.0043	20.666	-1.303	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓17	✓4	✓
		IEGNM	0.0065	4.351	36.656	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓13 ✗4	✓4	✓
		REGNM	0.0065	6.725	30.637	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓15 ✗2	✓4	✓
		UAM	0.0115	3.890	23.039	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓17	✓4	✓
		IUAM	0.0129	3.495	24.977	-	✓3	✓5	✓10	✓10	✓4	✓4	✓13	✓17	✓4	✓
VoxCeleb1	RN18 0.2978	GNM	0.2306	13.095	15.920	-	✓3	✓3	✓13	✓10	✓4	-	✓13	✓14 ✗1	✗	✓
		IGNM	0.2258	10.247	21.947	-	✓3	✓3	✓13	✓10	✓4	-	✓13	✓14 ✗1	✗	✓
		FNM	0.2485	22.590	-2.645	-	✓3	✓3	✓13	✓10	✓4	-	✓13	✓15	✗	✓
		IEGNM	0.2480	7.265	29.905	-	✓3	✓3	✓13	✓10	✓4	-	✓13	✓13 ✗2	✗	✓
		UAM	0.2859	4.752	14.536	-	✓3	✓3	✓13	✓10	✓4	-	✓7 ✗6	✓15	✗	✓
	RN34 0.2206	GNM	0.2396	13.095	15.920	-	✓3	✓3	✓13	✓10	✓4	-	✓13	✓10 ✗5	✗	✓
		IGNM	0.2066	6.766	29.903	-	✓3	✓3	✓13	✓10	✓4	-	✓11 ✗2	✓12 ✗3	✗	✓
		FNM	0.2016	22.590	-2.645	-	✓3	✓3	✓13	✓10	✓4	-	✓13	✓15	✓	✓
		IEGNM	0.1959	7.265	29.905	-	✓3	✓3	✓13	✓10	✓4	-	✓13	✓13 ✗2	✗	✓
		UAM	0.2071	4.752	14.536	-	✓3	✓3	✓13	✓10	✓4	-	✓5 ✗8	✓15	✗	✓
		IUAM	0.2201	12.722	5.668	-	✓3	✓3	✓13	✓10	✓4	-	✓13	✓15	✗	✓
	AS 0.1691	GNM	0.1476	13.095	15.920	-	✓3	✓3	✓13	✓10	✓4	✓1	✓13	✓13 ✗2	✓	✓
		IGNM	0.1492	6.766	29.903	-	✓3	✓3	✓13	✓10	✓4	-	✓13	✓13 ✗2	✗	✓
		RGNM	0.1743	10.247	21.9467	-	✓3	✓3	✓13	✓10	✓4	-	✓13	✓14 ✗1	✓	✓
		FNM	0.1298	22.590	-2.645	-	✓3	✓3	✓13	✓10	✓4	✓1	✓13	✓15	✓	✓
		IEGNM	0.1400	7.265	29.905	-	✓3	✓3	✓13	✓10	✓4	-	✓13	✓13 ✗2	✓	✓
		REGNM	0.1447	12.019	15.930	-	✓3	✓3	✓13	✓10	✓4	-	✓13	✓15	✓	✓
		UAM	0.1515	4.752	14.5366	-	✓3	✓3	✓13	✓10	✓4	✓1	✓6 ✗7	✓15	✗	✓

CHAPTER 7

Approaches to improve Legality and Robustness

In the previous chapter, we discussed at length how to balance the imperceptibility and robustness aspects of the watermarks. Since the issue with the legality characteristic persisted throughout all the experiments that we conducted, apart from the Unrelated Audio watermark applied to the VoxCeleb1 dataset, in this chapter, we describe the approaches we implemented to mitigate this problem and to enhance robustness. This chapter consists of four parts: a smaller size of the trigger set, embedding a watermark in a temporal pattern rather than the full audio signal, and using different trigger labels.

7.1 Smaller Trigger Set Size

The main problem with legality that we noticed is that a model overgeneralises on the pattern that is added by the watermark, and then it recognises all the samples that have a similar enough pattern. Here, we experimented with reducing the number of trigger samples that carry the watermark, to rather make the model memorise a specific set of triggers instead of a pattern. The idea originated from the fact that the SincNet model watermarked with Unrelated Audio trained on triggers with $\alpha = 1/32$ had higher WMSR_NT compared to others, which indicated two things: to make a watermarking pattern less generalisable, it should be more imperceptible, and the model should not be exposed to it too often. The subsequent discussion focuses on determining appropriate levels of imperceptibility and exposure.

In Table 7.1 we depict the results of embedding a watermark into SincNet using a smaller number of triggers, which is combined with a smaller value of λ in some of the cases. The robustness was evaluated only for the models that resulted in WMSR $\geq 95\%$. Table 7.1 consists of only those attacks that resulted in successful watermark removal. When a

watermark was able to withstand the attack, we either did not put it in the table (e.g., for pink noise addition (PN)) or marked it as ✓, when this attack was successful for one of the considered models (e.g., compression (C)). Due to computational constraints, the robustness evaluation was conducted in a staged manner: when watermark effectiveness and model fidelity were within acceptable ranges, data preprocessing attacks were applied; if legality and data preprocessing results remained acceptable, model modification attacks were subsequently performed.

Table 7.1: Evaluation of SincNet models watermarked by different approaches that use a smaller number of trigger samples. The legality characteristic is evaluated based on different watermarking methods: we verify whether a certain model can recognise unseen triggers created by another scheme with different parameters: Gaussian Noise (GN), Frequency Noise (FN), and Unrelated Audio (UA).

WM	# of triggers	λ	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	Imperceptibility SNR↑	Legality↓ GN 10	Legality↓ FN 8	Legality↓ UA 5	Robustness C	Robustness LT	Robustness FT
UAM 0.00005785 SER=0.0101	60	1.0	100%	0.0094	4.038	23.105	0	0	1	✓	✓	-
	30	1.0	100%	0.0079	3.874	23.212	0	0	1	✓	✓	-
	10	1.0	100%	0.0101	3.817	23.575	0	0	0	✓	✓	✓3
	60	0.5	100%	0.0108	2.889	29.125	0	0	1	✓	✓	-
	30	0.5	96.67%	0.0123	2.765	29.232	0	0	0	✓	✓	✓1 ✗2
	30	0.1	23.30%	0.0303	1.193	43.209	0	0	0	-	-	-
GNM SER=0.0101	60	0.6	100%	0.0101	10.893	20.862	5	2	0	✓	✓	-
	30	0.6	100%	0.0101	11.020	20.971	3	0	0	✓	✓	-
	10	0.6	100%	0.0087	10.315	21.066	0	0	0	✓	✗2 (7600-7700)	-
	30	0.4	100%	0.0094	9.249	24.492	4	2	0	✓	✓	-
	30	0.2	96.67%	0.0129	6.674	30.513	0	0	0	✗16000	✗5 (7500-7900)	-
	30	0.1	30%	0.0202	4.656	36.533	0	0	0	-	-	-
EGNM ELF=100 EHF=6,000 SER=0.0129	60	0.2	100%	0.0101	6.224	30.415	1	3	0	✓	✗4 (7400-7700)	-
	30	0.2	100%	0.0065	6.274	30.524	1	3	0	✓	✗2 (7600-7700)	-
	10	0.2	100%	0.0081	5.539	30.651	0	0	0	✓	✗3 (7500-7700)	-
	30	0.1	100%	0.0123	4.576	36.544	0	0	1	✓	✗3 (7500-7700)	-

Unrelated Audio Since the legality issue persisted in nearly all settings except for the Unrelated Audio watermarking method, we started the experiments with it.

We chose different sizes of the trigger sets, namely 60, 30, and 10, using $\lambda = 1.0$. All of them resulted in a SER not higher than the SER of the original Unrelated Audio model, which is 0.0101. The legality evaluation of these models shows a substantial improvement compared to Table 6.25: only the triggers from one set of parameters of the Unrelated Audio watermarking method are recognised by the model, which in both cases (with 60 and 30 triggers in the trigger set) are the ones created with $noise\_var = 0.000058$ and $\lambda = 1.0$. Using only 10 trigger samples to embed a watermark into the model results in zero recognition of unseen triggers. The Unrelated Audio model with these parameters also withstands both data preprocessing and fine-tuning attacks.

Subsequently, we tried to reduce λ to make the watermark more imperceptible, namely $\lambda = 0.5$ and $\lambda = 0.1$. Using 60 trigger samples, the WMSR still yielded 100%, while with 30 trigger samples, it started to drop, resulting in 96.67% WMSR. The SER for these models is slightly higher, especially for one trained on fewer triggers, which is 0.0123. Among them, a model watermarked using 60 triggers still recognised the samples from one of the Unrelated Audio watermarking methods, while the one where we used a trigger set of size 30 resulted in zero recognised settings. Data preprocessing attacks were unable to remove these watermarks. We also performed a fine-tuning attack on this model, and discovered that fine-tuning the CNN or first DNN layer of SincNet results in WMSR falling below 95% success rate. With $\lambda = 0.1$, the WMSR drops significantly to 23.30%. Moreover, the SER also increases to 0.0303. Following the previous revelations, we determined that choosing 30 trigger samples provides the fewest drawbacks.

Gaussian Noise Here, we again tried 60, 30, and 10 sizes with $\lambda = 0.6$ (the highest value for which SNR is higher than 20 dB), but looked further into the setting with 30 triggers and various λ values: 0.6, 0.4, 0.2, 0.1. The results for different trigger set sizes in Table 7.1 indicate that the model with 60 triggers recognised five Gaussian Noise settings and two Frequency Noise settings. With smaller trigger sets, legality improves, but this comes at the expense of robustness, as observed in the models with 30 and 10 triggers. Among the different imperceptibility options with 30 triggers, $\lambda = 0.6$ resulted in three parameter settings of Gaussian Noise being recognised, which were $\lambda \in 1.0, 0.9, 0.8$. The Gaussian Noise model with $\lambda = 0.4$ recognised even more settings, while both $\lambda = 0.2$ and $\lambda = 0.1$ resulted in zero recognised settings. Nevertheless, taking into consideration SER, which is higher for $\lambda = 0.2$ and $\lambda = 0.1$, and WMSR, which in the former model still reaches 96.67%, and in the latter drops below the threshold, these parameter settings are not optimal.

We also evaluate the robustness of the first five models of GNM listed in Table 7.1. The majority of the ones that have 100% WMSR can withstand data preprocessing attacks, except for the model with 10 triggers. The watermark in the last Gaussian Noise model, created with 30 triggers and λ=0.1, can be easily removed by multiple approaches since it has only 30% WMSR.

Frequency Noise Applying the same approach with the Frequency Noise watermarking method's Extreme Gaussian Noise variant did not result in a model that fulfils all the requirements of WMSR, SER, legality, and robustness. The smaller the trigger size is, the better the legality is, but the robustness becomes worse. Moreover, the higher λ, specifically 0.2, resulted in a model that recognises four different settings from both Gaussian and Frequency Noise. $\lambda = 0.1$, while better in legality, falls short in robustness.

Findings Overall, this approach demonstrates that it is possible to make the model recognise only designated samples; however, in the majority of cases, the drawback is a significant loss in robustness. Generally speaking, we consider robustness to be a more important characteristic than legality, as it is essential that the ownership of a stolen

model can be reliably proven. In contrast, as long as a malicious party is unaware of the specifics of the watermarking method used to protect the model, it is unlikely they would be able to forge the watermark. One setting of Unrelated Audio watermark leads to no recognition of unseen triggers and is proven to be robust against various attacks, the setting that uses 10 trigger samples. However, here, we face the question of whether 10 triggers are enough to prove the ownership of a model.

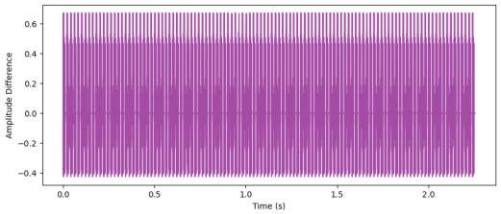
7.2 Temporal Pattern Embedding

Another approach that we tried was embedding a watermark not in the full length of a frequency range, but at a certain time periods, e.g, embedding Gaussian noise in every 20th frame processed with $window = 160$ ms, $step = 20$ ms. That way, a watermark is embedded not continuously, but based on a specific temporal pattern, which may change how much the model generalises on a particular watermark. In Figure 7.1, we show examples of such embeddings with different parameters on the same audio sample. The parameters were for the Gaussian Noise with $\lambda = 1.0$ were selected to explore a variety of scenarios, including overlapping versus non-overlapping frames, sparse versus dense embedding, and different window sizes, rather than being based on any specific algorithm. Then, considering the results, we evaluated two more cases with a smaller λ.

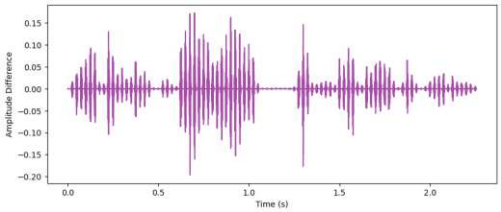
However, we found certain issues with this approach. Table 7.2 presents the results of embedding the created watermarks into the SincNet model. We can observe the following trends:

- When the noise was embedded in non-overlapping frames, the legality was better than in the original Gaussian Noise model, but that led to poor imperceptibility.
- When the noise was embedded in overlapping frames, it improved the perceptual quality of the audio sample, also making the watermark less noticeable; however, the model recognised way more unseen triggers created with various schemes.
- Decreasing the λ improves imperceptibility and legality, but does not completely mitigate the problem of recognising unseen triggers.

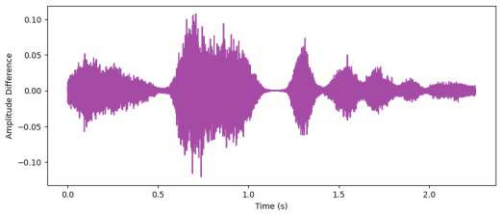
Additionally, the robustness of the discussed seven models was evaluated according to our robustness evaluation design defined in Section 5.8. No attack was successful in removing the watermark without deteriorating the SER beyond the set boundary.



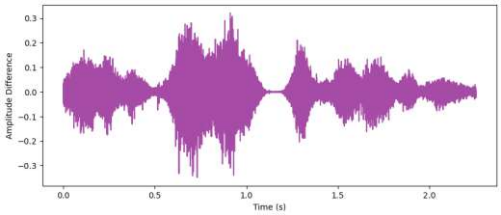
(a) window=160, step=20, each 20th frame, $\lambda = 1.0$



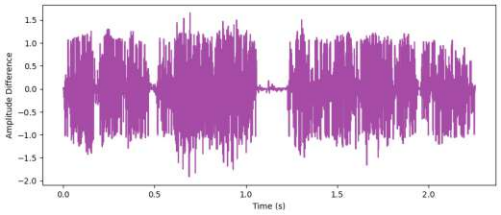
(b) window=1600, step=100, each 10th frame, $\lambda = 1.0$



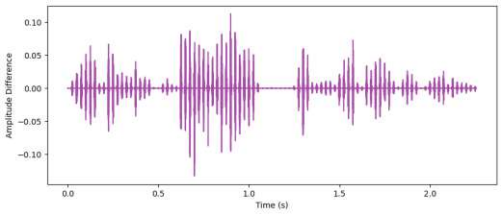
(c) window=3200, step=160, each 5th frame, $\lambda = 1.0$



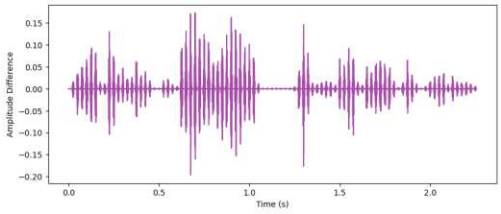
(d) window=200, step=100, each 4th frame, $\lambda = 1.0$



(e) window=20, step=10, each 3th frame, $\lambda = 1.0$



(f) window=160, step=20, each 20th frame, $\lambda = 0.5$



(g) window=200, step=40, each 10th frame, $\lambda = 0.5$

Figure 7.1: Difference between clean and watermarked audio samples in amplitude, taken from the TIMIT dataset, dialect *1* speaker *FETB0* audio *sx248*.

Table 7.2: Evaluation of the SincNet model watermarked by different approaches that use a temporal pattern embedding scheme with Gaussian Noise. The legality characteristic is evaluated based on different triggers. We verify whether a certain model can recognise unseen triggers created by another scheme with different parameters: Gaussian Noise (GN), temporal pattern-based Gaussian Noise (PGN), Frequency Noise (FN), and Unrelated Audio (UA).

WM	λ	window	step	Frame	Effect. WMSR↑	Fidelity SER↓	Imperceptibility LSD↓	SNR↑	Legality↓ GN 10	PGN 7	FN 8	UA 5
GNM	1.0	160	20	20	100	0.0115	26.7440	0.0241	0	2	0	0
	1.0	1,600	100	10	100	0.0202	7.4699	23.6571	7	4	3	0
	1.0	3,200	160	5	100	0.0144	9.1761	23.6821	8	4	3	1
	1.0	200	100	4	100	0.0072	16.0356	0.7541	2	2	2	0
	1.0	20	10	3	100	0.0072	28.1162	-4.3657	0	1	0	0
	0.5	160	20	20	100	0.0339	3.3672	27.6107	4	4	2	0
	0.5	200	40	10	100	0.0209	4.8247	22.6877	3	4	3	0

7.3 Different Labels

Following the idea of assigning different labels to trigger samples introduced in [ZJW+20], we examine whether such unpredictability can support the legality characteristic of watermarking. While we do not adopt the full algorithm proposed by Zhang et al., who assign the labels to triggers according to *Logistic chaotic map*, we randomly assign labels to test whether our chosen speaker recognition models have sufficient capacity to memorise varying label assignments that share the same watermark. This approach ensures that newly generated triggers created by a malicious party would not be recognised by the model – even if the overall watermarking method is known, since the specific label assignments remain secret. For each trigger sample, the assigned label is selected randomly from the set of available speakers, excluding its original identity.

Due to computational resource constraints, we evaluated this approach on three selected models: SincNet, AM-MobileNet, and AutoSpeech. SincNet was used as the baseline model; AM-MobileNet was included as it has been shown to outperform SincNet on the TIMIT dataset; and AutoSpeech was selected as the best-performing model on the VoxCeleb1 dataset.

Table 7.3: Imperceptibility and robustness evaluation of SincNet, AM-MobileNet, and AutoSpeech models watermarked by triggers assigned to distinct speakers.

				Effectiveness	Fidelity	Imperceptibility		Robustness		
Dataset	Model	WM	Parameters	WMSR↑	SER↓	LSD↓	SNR↑	HT	LT	FT
TIMIT	SN 0.0051	GNM	1.0	100%	0.0115	13.0948	16.6514	✓	✗1 (7,400)	✓2 ✗1
		IGNM	0.6	100%	0.0094	10.6347	21.0876	✓	✓	✓2 ✗1
		FNM	2.0 1.0 300 3,000	100%	0.0094	20.6664	-1.3016	✓	✗1 (7,600)	✓2 ✗1
		EGNM	1.0 100 6,000	100%	0.0123	11.0875	16.6521	✓	✓	✓
		REGNM	0.2 150 5,000	100%	0.0144	6.6078	30.6331	✓	✗1 (7,400)	✓2 ✗1
		UAM	1.0 0.000058	100%	0.0123	3.9695	23.0670	✓	✓	✓2 ✗1
	AM-MN 0.0043	GNM	1.0	100%	0.0043	13.0948	16.6514	✓	✗4 (5,900-6,200)	✓
		GNM	0.6	100%	0.0043	10.6321	21.0868	✓	✗4 (6,200-6,500)	✓
		GNM	0.4	100%	0.0036	8.8952	24.6091	✓	✗4 (6,700-7,000)	✓
		GNM	0.2	100%	0.0050	6.3736	30.6287	✓	✗10 (5,800-6,700)	✓
		FNM	2.0 1.0 300 3,000	100%	0.0043	20.6664	-1.3016	✓	✗5 (5,400-5,800)	✓
		REGNM	0.2 200 4,500	100%	0.0058	4.3513	36.6559	✓	✗5 (6,500-6,900)	✓
		UAM	1.0 0.000058	100%	0.0072	3.9695	23.0670	✓	✓	✓
VoxCeleb1	AS	GNM	1.0	96.46%	0.1670	13.4611	16.0410	✗3 (100-150)	✗1 (7,900)	-

The results of this approach are presented in Table 7.3. Overall, the SER of the SincNet model watermarked using different methods is higher than that of the original approach, which can be attributed to the fact that it influences the predictions of more than one class. With regard to data preprocessing attacks, some cases show one successful attack. Given that the watermark cannot be forged using newly generated trigger samples, the case when WMSR does not reach 100% is less critical compared to the original approach. Although fine-tuning the CNN layer of the SincNet model generally leads to a reduction in WMSR, the drop remains relatively small – for instance, down to 94.5% in the case of the Gaussian Noise model with $\lambda = 0.6$, and 93.1% for the Unrelated Audio model.

For AM-MobileNet, we observe that the SER is comparable to the results previously

achieved, though its robustness is considerably weaker, with only the Unrelated Audio model passing all robustness checks.

For the AutoSpeech model, we initially used the same setup with 50 speakers and 126 trigger samples; however, this resulted in a 0% WMSR. After multiple experiments, the results presented in Table 7.3 are based on 300 different speakers and 900 trigger samples. We can thus conclude that this method can be successfully applied to AutoSpeech as well, though it requires a higher number of trigger samples as well as a proportion of speakers to be as successful as the original approach. The robustness and watermark effectiveness (96.46%) do not quite reach our set standard; however, it is comparable to ResNet18 and ResNet34 results.

7.4 Summary

Overall, the approach of using Different Labels (presented in Section 7.3) demonstrates the best performance among those discussed in this chapter. It is applicable across multiple models and achieves desirable outcomes: it improves legality by design, maintains acceptable imperceptibility, and enhances robustness.

The strategy of utilising a Smaller Trigger Set (Section 7.1) shows potential for mitigating the legality problem. However, it requires careful tailoring to each specific dataset-model-watermark instance, which limits its generality.

The Temporal Pattern Embedding approach (Section 7.2) did not completely resolve the legality problem in any of the tried settings, though in some instances it reduced the number of recognised settings.

CHAPTER 8

Conclusion

In this chapter, we summarise our work and reflect on the research questions outlined in Chapter 1. We begin by discussing the main contributions of the thesis, followed by answers to the research questions. Finally, we outline potential directions for future work.

8.1 Insights and Contributions

In this thesis, we investigated existing techniques for black-box watermarking of speaker recognition models. We designed and implemented an evaluation framework to assess watermarking approaches against key requirements: model fidelity, watermark effectiveness, imperceptibility, legality, robustness, and generality. The trade-offs among these characteristics were analysed, and we provided recommendations on suitable watermarking methods for different model and dataset combinations. Furthermore, we proposed approaches aimed at enhancing the robustness and legality of watermarking techniques.

We uncovered critical gaps in the existing works: undocumented parameters, inconsistent method descriptions, and a lack of replicability guidelines. We find that effectiveness and fidelity can vary significantly depending on trigger design and audio preprocessing choices, making the methods difficult to calibrate, highlighting the importance of their transparency.

To advance the field, we introduced audio-specific attack types (i.e., bandpass filtering, compression, time stretching, pitch shifting) and an attack cost metric, which exposes the unrealistic nature of certain attacks that degrade model utility while removing watermarks. We also propose improved, audio-specific imperceptibility metrics (i.e., Log spectrum distortion and Signal-to-Noise ratio) to evaluate watermarking methods.

Our analysis identifies opportunities for improving existing techniques, particularly their robustness against audio preprocessing attacks such as high/low frequency filtering and fine-tuning, both of which successfully removed watermarks in some settings. Additionally,

an inherent trade-off between imperceptibility and robustness can be observed: while higher imperceptibility often reduces robustness to certain attacks, it also lowers the risk of detection by adversaries, thereby enabling stealthier watermarking.

We reveal a crucial vulnerability of the watermarking methods that was not verified earlier, namely, the legality characteristic of the watermarking methods. In several cases, triggers can be forged by a malicious party without knowing the exact parameters or, in particular cases, even the method using which the trigger set was created, which could allow adversaries to falsely claim model ownership. This is possible because the model tends to learn the specific patterns embedded in the triggers rather than the complete input, making it susceptible to imitation. Based on these observations, we propose strategies to mitigate this issue, e.g., assigning triggers to various classes or reducing the trigger set.

8.2 Research Questions

In this section, we present our contributions with regard to the research questions.

1. **How does effectiveness of state-of-the-art watermarking techniques compare in different model architectures and datasets?** This question is addressed in Section 6.2. Overall, we applied the watermarking methods to two datasets and six models. Here, the watermark performance was evaluated by two aspects: the watermark effectiveness WMSR, and the model fidelity. The detailed analysis regarding their specifics is discussed in the sub-questions below.

    a) **How effective are the watermarking techniques for other datasets?** The most notable difference lies in the level of distinction required in a watermarking pattern embedded in the trigger samples. The TIMIT dataset contains clean speech, allowing the model to detect even subtle discrepancies (such as those introduced by the Unrelated Audio watermarking method) without any particular loss in watermark effectiveness, although a gradual loss in fidelity can be observed. In contrast, the VoxCeleb1 dataset comprises audio of varying quality, often accompanied by background noise of different origins. In such conditions, the watermark must be more pronounced for the model to reliably recognise a trigger sample. For instance, comparing the cases where models trained on each dataset achieved a 100% WMSR and the highest possible fidelity, the best SNR for the VoxCeleb1 dataset using the Unrelated Audio watermark was 14.54 dB, while for TIMIT, the SNR reached 24.97 dB, 29.06 dB, and even 43.04 dB, depending on the model. This suggests that for datasets with non-uniform audio quality, such as VoxCeleb1, a more pronounced watermark is necessary for the model to reliably capture it, whereas for more uniform datasets like TIMIT, a more subtle watermark can still achieve the desired fidelity and WMSR.

b) **To what extent do the watermarking techniques succeed in protecting models with different architectures?** In all cases examined, the six models were successfully watermarked using the proposed watermarking methods that we managed to reproduce. Model fidelity varied across watermarking techniques, with the Unrelated Audio method performing the worst in this regard in four out of six cases. Nevertheless, we empirically demonstrated that models with different architectures, feature processing logic, and input formats can be successfully watermarked using three watermarking methods: Gaussian Noise, Frequency Noise, and Unrelated Audio. Furthermore, our proposed Extreme Gaussian Noise algorithm was shown to be applicable across all these models.

2. **To what extent can we reduce the perceptibility of watermarks while still reaching the same watermark effectiveness?** In order to address this question, we proposed using two established audio metrics: Signal-to-Noise Ratio and Log Spectral Distortion – which, to our knowledge, have not previously been applied in the context of watermarking, despite the availability of defined thresholds for these metrics. Next, we assessed the imperceptibility of the reproduced watermarking methods, which had not been systematically evaluated before. This revealed that only the Unrelated Audio watermark came close to the defined threshold of imperceptibility measure, although its performance was highly dependent on the specific unrelated audio sample used. We identified key hyperparameters (both those already existing in the watermarking methods and those newly introduced by us) that significantly influence imperceptibility and can be easily tuned to achieve a desired level of watermark stealthiness. After conducting a hyperparameter search, we proposed a set of parameters that best fulfill this purpose in each case. By further evaluation, we determined the following:

   - For the SincNet model trained on TIMIT, achieving an SNR of ≥ 20 dB (which is considered an acceptable imperceptibility for the watermark) is possible without any loss in model fidelity or watermark effectiveness compared to the original settings. However, using trigger sets with SNR ≥ 30 dB (which is considered an imperceptible watermark) begins to degrade model fidelity, although watermark effectiveness is maintained.
   - Watermarking the MobileNet and AM-MobileNet trained on the TIMIT dataset is possible even with a trigger set, achieving an SNR higher than 30 dB. LSD values below 2 dB (which is considered an acceptable imperceptibility for the watermark) are also achievable with TIMIT, albeit the model's fidelity starts to suffer in this case.
   - For the VoxCeleb1 dataset that contained noisy audio, it is more challenging to achieve higher imperceptibility. In order to preserve both watermark effectiveness and model fidelity, the best results fall within the 20–30 dB SNR range.

- The Unrelated Audio watermark proved difficult to embed effectively in VoxCeleb1 while maintaining imperceptibility, effectiveness, and fidelity. We assume it is due to the nature of Unrelated Audio (namely, air conditioner sound) that is entirely plausible to occur under normal circumstances (i.e., it is in fact not that unrelated to the dataset) and is harder for the model to distinguish reliably.

Overall, except for the Unrelated Audio watermark applied to the VoxCeleb1 dataset, we identified configurations that enable the creation of more imperceptible trigger sets that meet established imperceptibility criteria. Specifically, we improved the imperceptibility in 15 out of 18 dataset–model–watermark combinations.

3. **How robust are the state-of-the-art audio watermarking techniques for SR models and how can their robustness be improved?** To assess the robustness of the watermarking methods, we considered multiple attack scenarios. Our analysis revealed that some attacks reported in previous studies can indeed remove the watermark, but at the cost of severely degrading model fidelity – rendering the model unusable. To address this, we introduced the concept of an attack cost to identify only those attacks that a malicious actor might realistically employ, based on our defined threat model.

   Our robustness evaluation includes both data preprocessing and model modification attacks. The results revealed that the imperceptibility of a watermarking pattern significantly affects its robustness. Overall, this evaluation uncovered the following:

   - Models watermarked with Frequency Noise demonstrated the highest robustness, successfully withstanding all data preprocessing and pruning attacks, as well as fine-tuning in five out of six cases, falling short only for the ResNet18 model.
   - The Unrelated Audio watermark comes second, passing all robustness checks in three out of six cases. Its weaker performance is primarily linked to models trained on the VoxCeleb1 dataset, specifically ResNet18, ResNet34, and AutoSpeech, where the watermark failed to withstand fine-tuning and low-frequency removal preprocessing. These results indicate that the Unrelated Audio watermark is not well-suited for watermarking highly diverse datasets such as VoxCeleb1.
   - Our analysis indicates that state-of-the-art watermarking methods require further refinement, as they exhibit key vulnerabilities by being either too distinguishable for a malicious party to plausibly deny them passing through a model, failing under specific attacks, or being susceptible to forgery.
   - In addition, this research question also led us to evaluate legality, as we discovered during our evaluation that the examined watermarking methods are vulnerable to forgery.

To address these issues, we proposed a refined version of the Frequency Noise watermark that maintains robustness while improving imperceptibility. Specifically, we eliminated the use of MFCC features in the embedding process, which allows for higher-quality reconstruction of the raw audio. At the same time, we retain the injection of Gaussian Noise in extreme frequency bands, now controlled by a newly introduced intensity parameter, enabling finer adjustment of the watermark strength. Moreover, by examining the trade-off between robustness and imperceptibility, and taking into account model fidelity and watermark success rate, we identified improved configurations for the Gaussian Noise and Unrelated Audio methods as well, which more effectively meet the essential audio watermark requirements of fidelity, imperceptibility, and robustness.

Furthermore, this thesis introduces several mitigation strategies to address the legality problem: (i) assigning trigger samples to various speaker classes and (ii) reducing the number of trigger samples, to prevent models from recognising unseen trigger samples, thereby enhancing the legality aspect of watermarking methods.

8.3 Future work

We identify several directions for future work. Firstly, some of our proposed approaches for improving legality and robustness showed promising results and can be analysed further. Specifically, assigning various classes to the triggers. We implemented a lighter version of random assignment to verify the approach; however, it can also be extended to Zhang's idea [ZJW+20] of assigning the labels 'chaotically', meaning from the data input, it can be derived which class is assigned (i.e., by a secret annotation scheme).

Furthermore, the authors of the three watermarking methods we replicated and analysed claim that their techniques support watermark embedding via fine-tuning. However, they do not provide any baselines and do not discuss any results obtained using this approach. Verifying its effectiveness could be a worthwhile direction, provided that robustness is not compromised. If successful, it would significantly improve the efficiency of the watermarking process. Another potential direction would be to explore watermarking in alternative architectures for speaker recognition, such as transformer-based models, which may introduce new opportunities as well as challenges for robustness and imperceptibility.

Considering that black-box watermarking of machine learning models in the audio domain remains relatively underexplored, it would be valuable to examine whether these methods can be extended to Speech Recognition models. This would require the verification of the sensitivity of clean models to modifications, as such models must accurately distinguish each pronounced syllable. Moreover, the threshold for imperceptibility may differ due to stricter perceptual constraints.

APPENDIX A

Appendix

Table A.1: UrbanSound8K audio samples used in the experiments with their corresponding noise variance values.

Noise variance	Audio sample
0.05863584	fold8/80589-0-0-0.wav
0.04734541	fold1/59277-0-0-0.wav
0.00590028	fold9/189989-0-0-1.wav
0.00043166	fold3/13230-0-0-12.wav
0.00017814	fold6/63724-0-0-13.wav
0.00004772	fold5/178686-0-0-60.wav
0.00007485	fold5/178686-0-0-7.wav
0.00005785	fold5/178686-0-0-33.wav
0.00003688	fold9/165454-0-0-12.wav
0.00000090	fold3/159761-0-0-6.wav

Overview of Generative AI Tools Used

Grammarly and Writefull were used in this thesis for grammar checking and style editing.

List of Figures

List of Tables

List of Algorithms

Bibliography

[ABPK18] Yossi Adi, Carsten Baum, Benny Pinkas, and Joseph Keshet. Turning Your Weakness Into a Strength: Watermarking Deep Neural Networks by Backdooring. 2018.

[ADND20] K. Anitha, R. Dhanalakshmi, K. Naresh, and D. Rukmani Devi. Hyperbolic Hopfield neural networks for image classification in content-based image retrieval. *International Journal of Wavelets, Multiresolution and Information Processing*, October 2020.

[Arn00] Michael Arnold. Audio watermarking: features, applications and algorithms. In *2000 IEEE International Conference on Multimedia and Expo. ICME 2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No.00TH8532)*, volume 2, pages 1013–1016, July 2000.

[Ass20] Association for Computing Machinery. Artifact Review and Badging Policy. accessed: 28-07-2025 [Online], 2020.

[BGML96] W Bender, D Gruhl, N Morimoto, and A Lu. Techniques for data hiding. *IBM Systems Journal*, 35(3.4):313–336, December 1996.

[BKH16] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer Normalization, July 2016.

[Boe21] Franziska Boenisch. A Systematic Review on Model Watermarking for Neural Networks. *Frontiers in Big Data*, 4, November 2021.

[BSF94] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, March 1994.

[BZ21] Zhongxin Bai and Xiao-Lei Zhang. Speaker recognition based on deep learning: An overview. *Neural Networks*, 140:65–99, August 2021.

[Car05] Réal Carbonneau. Artificial intelligence for data mining in the context of enterprise systems. August 2005.

[CDK20] Huili Chen, Bita Darvish, and Farinaz Koushanfar. SpecMark: A Spectral Watermarking Framework for IP Protection of Speech Recognition Systems. In *Interspeech 2020*, pages 2312–2316. ISCA, October 2020.

[CKLS97] I.J. Cox, J. Kilian, F.T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687, December 1997.

[CLJ+23] Xirong Cao, Xiang Li, Divyesh Jadav, Yanzhao Wu, Zhehui Chen, Chen Zeng, and Wenqi Wei. Invisible Watermarking for Audio Generation Diffusion Models. In *2023 5th IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, pages 193–202, November 2023.

[CW18] Nicholas Carlini and David Wagner. Audio Adversarial Examples: Targeted Attacks on Speech-to-Text. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 1–7, May 2018.

[CZL+22] Haozhe Chen, Weiming Zhang, Kunlin Liu, Kejiang Chen, Han Fang, and Nenghai Yu. Speech Pattern Based Black-Box Model Watermarking for Automatic Speech Recognition. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3059–3063, May 2022.

[DCG+20] Shaojin Ding, Tianlong Chen, Xinyu Gong, Weiwei Zha, and Zhangyang Wang. AutoSpeech: Neural Architecture Search for Speaker Recognition, August 2020.

[DM80] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, August 1980.

[Dod85] G.R. Doddington. Speaker recognition—Identifying people by their voices. *Proceedings of the IEEE*, 73(11):1651–1664, November 1985.

[EHPM23] Onno Eberhard, Jakob Hollenstein, Cristina Pinneri, and Georg Martius. Pink Noise Is All You Need: Colored Noise Exploration in Deep Reinforcement Learning. March 2023.

[GBGM80] R. Gray, A. Buzo, A. Gray, and Y. Matsuyama. Distortion measures for speech processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):367–376, August 1980.

[GLB96] Daniel Gruhl, Anthony Lu, and Walter Bender. Echo hiding. In Ross Anderson, editor, *Information Hiding*, pages 295–315, Berlin, Heidelberg, January 1996. Springer.

[GLDGG19] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. BadNets: Evaluating Backdooring Attacks on Deep Neural Networks. *IEEE Access*, 7:47230–47244, April 2019.

[GLF+93] John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, David S Pallett, and Nancy L Dahlgren. DARPA TIMIT: acoustic-phonetic continuous speech corpus CD-ROM, NIST speech disc 1-1.1. Technical report, National Institute of Standards and Technology, Gaithersburg, MD, 1993.

[GM76] A. Gray and J. Markel. Distance measures for speech processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(5):380–391, October 1976.

[HGT15] Guang Hua, Jonathan Goh, and Vrizlynn. L. L. Thing. Time-Spread Echo-Based Audio Watermarking With Optimized Imperceptibility and Robustness. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(2):227–239, February 2015.

[HHS+16] Guang Hua, Jiwu Huang, Yun Q. Shi, Jonathan Goh, and Vrizlynn L. L. Thing. Twenty years of digital audio watermarking—a comprehensive review. *Signal Processing*, 128:222–242, November 2016.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, November 1997.

[HT23] Guang Hua and Andrew Beng Jin Teoh. Deep fidelity in DNN watermarking: A study of backdoor watermarking for classification models. *Pattern Recognition*, 144:109844, December 2023.

[HTXJ24] Guang Hua, Andrew Beng Jin Teoh, Yong Xiang, and Hao Jiang. Unambiguous and High-Fidelity Backdoor Watermarking for Deep Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 35(8):11204–11217, August 2024.

[HTY+20] Chih-Yu Hsu, Shu-Yi Tu, Chao-Tung Yang, Ching-Lung Chang, and Shuo-Tsung Chen. Digital audio signal watermarking using minimum-energy scaling optimisation in the wavelet domain. *IET Signal Processing*, 14(10):791–802, December 2020.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE, June 2016.

[IS15] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 448–456. PMLR, June 2015.

[JCCCP21] Hengrui Jia, Christopher A. Choquette-Choo, Varun Chandrasekaran, and Nicolas Papernot. Entangled Watermarks as a Defense against Model Extraction. pages 1937–1954, 2021.

[KB17] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017.

[KM03] D. Kirovski and H.S. Malvar. Spread-spectrum watermarking of audio signals. *IEEE Transactions on Signal Processing*, 51(4):1020–1033, April 2003.

[KMS+21] Muhammad Mohsin Kabir, M. F. Mridha, Jungpil Shin, Israt Jahan, and Abu Quwsar Ohi. A Survey of Speaker Recognition: Fundamental Theories, Recognition Methods and Opportunities. *IEEE Access*, 9:79236–79263, May 2021.

[KP16] Stefan Katzenbeisser and Fabien Petitcolas. *Information Hiding*. Artech House, January 2016.

[LBD+89] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4):541–551, December 1989.

[LJLX24] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor Learning: A Survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1):5–22, January 2024.

[LMR24] Isabell Lederer, Rudolf Mayer, and Andreas Rauber. Identifying Appropriate Intellectual Property Protection Mechanisms for Machine Learning Models: A Systematization of Watermarking, Fingerprinting, Model Access, and Attacks. *IEEE Transactions on Neural Networks and Learning Systems*, 35(10):13082–13100, October 2024.

[LYS+24] Junpei Liao, Liang Yi, Wenxin Shi, Wenyuan Yang, Yanmei Fang, and Xin Yang. Imperceptible backdoor watermarks for speech recognition model copyright protection. *Visual Intelligence*, 2(1):23, July 2024.

[MF03] H.S. Malvar and D.A.F. Florencio. Improved spread spectrum: a new modulation technique for robust watermarking. *IEEE Transactions on Signal Processing*, 51(4):898–905, April 2003.

[MHIM21] Rafizah Mohd Hanifa, Khalid Isa, and Shamsul Mohamad. A review on speaker recognition: Technology and challenges. *Computers & Electrical Engineering*, 90:107005, March 2021.

[MP43] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, December 1943.

[Mü15] Meinard Müller. *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*. Springer International Publishing, Cham, July 2015.

[NCZ17] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. VoxCeleb: a large-scale speaker identification dataset. In *Interspeech 2017*, pages 2616–2620, August 2017.

[NJC+08] Jesper Kjoer Nielsen, Jesper Rindom Jensen, Mads Groesboll Christensen, Soren Holdt Jensen, and Torben Larsen. Waveform approximating residual audio coding with perceptual pre- and post-filtering. In *2008 42nd Asilomar Conference on Signals, Systems and Computers*, pages 1255–1259, Pacific Grove, CA, USA, October 2008. IEEE.

[NMZ20] João Antônio Chagas Nunes, David Macêdo, and Cleber Zanchettin. AM-MobileNet1D: A Portable Model for Speaker Recognition. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, July 2020.

[NT07] Cvejic Nedeljko and Seppanen Tapio. *Digital Audio Watermarking Techniques and Technologies: Applications and Benchmarks: Applications and Benchmarks*. IGI Global, August 2007.

[PCPK15] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210, April 2015.

[PMB13] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training Recurrent Neural Networks, February 2013.

[Rav25] Mirco Ravanelli. mravanelli/SincNet, April 2025.

[RB18] Mirco Ravanelli and Yoshua Bengio. Speaker Recognition from Raw Waveform with SincNet. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 1021–1028, December 2018.

[Ros58] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[RSR22] Pulkit Rathi, Bhadauria Saumya, and Sugandha Rathi. Watermarking of Deep Recurrent Neural Network Using Adversarial Examples to Protect Intellectual Property. *Applied Artificial Intelligence*, 36(1):2008613, December 2022.

[Rud17] Sebastian Ruder. An overview of gradient descent optimization algorithms, June 2017.

[SAMA21] Sebastian Szyller, Buse Gul Atli, Samuel Marchal, and N. Asokan. DAWN: Dynamic Adversarial Watermarking of Neural Networks. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4417–4425, Virtual Event China, October 2021. ACM.

[Set21] De Rosal Igantius Moses Setiadi. PSNR vs SSIM: imperceptibility quality assessment for image steganography. *Multimedia Tools Appl.*, 80(6):8423–8444, March 2021.

[SHK+14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, January 2014.

[SHZ+18] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, Salt Lake City, UT, June 2018. IEEE.

[SJB14] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. A Dataset and Taxonomy for Urban Sound Research. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 1041–1044, Orlando Florida USA, November 2014. ACM.

[SKM+20] Ankit Sharma, Puneet Kumar, Vikas Maddukuri, Nagasai Madamshetti, K. G. Kishore, Sahit Sai Sriram Kavuru, Balasubramanian Raman, and Partha Pratim Roy. Fast Griffin Lim based waveform generation strategy for text-to-speech synthesis. *Multimedia Tools and Applications*, 79(41):30205–30233, November 2020.

[SKT98] M.D. Swanson, M. Kobayashi, and A.H. Tewfik. Multimedia data-embedding and watermarking technologies. *Proceedings of the IEEE*, 86(6):1064–1087, June 1998.

[SLH+23] Yuchen Sun, Tianpeng Liu, Panhe Hu, Qing Liao, Shaojing Fu, Nenghai Yu, Deke Guo, Yongxiang Liu, and Li Liu. Deep Intellectual Property Protection: A Survey, June 2023.

[SNKR25] Euschi Salah, Zermi Narima, Amine Khaldi, and Kafi Med Redouane. Survey of imperceptible and robust digital audio watermarking systems. *Multimedia Tools and Applications*, 84(7):3635–3681, February 2025.

[SSB21] Dávid Sztahó, György Szaszák, and András Beke. Deep Learning Methods in Speaker Recognition: A Review. *Periodica Polytechnica Electrical Engineering and Computer Science*, 65(4):310–328, October 2021.

[TA17] Kadir Tekeli and Rifat Asliyan. A COMPARISON OF ECHO HIDING METHODS. *Engineering and Science*, 2017.

[TZJ+16] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. Stealing Machine Learning Models via Prediction APIs. pages 601–618, 2016.

[UOHS24] Mohammad Shorif Uddin, Ohidujjaman, Mahmudul Hasan, and Tetsuya Shimamura. Audio Watermarking: A Comprehensive Review. May 2024.

[vSTO94] R.G. van Schyndel, A.Z. Tirkel, and C.F. Osborne. A digital watermark. In *Proceedings of 1st International Conference on Image Processing*, volume 2, pages 86–90 vol.2, November 1994.

[War18] Pete Warden. Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition, April 2018.

[WBSS04] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004.

[WCLL18] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive Margin Softmax for Face Verification. *IEEE Signal Processing Letters*, 25(7):926–930, July 2018.

[WW22] Yumin Wang and Hanzhou Wu. Protecting the Intellectual Property of Speaker Recognition Model by Black-Box Watermarking in the Frequency Domain. *Symmetry*, 14(3):619, March 2022.

[XHY17] Yong Xiang, Guang Hua, and Bin Yan. Human Auditory System and Perceptual Quality Measurement. In Yong Xiang, Guang Hua, and Bin Yan, editors, *Digital Audio Watermarking: Fundamentals, Techniques and Challenges*, pages 7–27. Springer, Singapore, March 2017.

[XZWL22] Mingfu Xue, Yushu Zhang, Jian Wang, and Weiqiang Liu. Intellectual Property Protection for Deep Learning Models: Taxonomy, Methods, Attacks, and Evaluations. *IEEE Transactions on Artificial Intelligence*, 3(6):908–923, December 2022.

[YDK+23] Mohamed Yamni, Achraf Daoui, Hicham Karmouni, Mhamed Sayyouri, Hassan Qjidaa, Saad Motahhir, Ouazzani Jamil, Walid El-Shafai, Abeer D. Algarni, Naglaa F. Soliman, and Moustafa H. Aly. An efficient watermarking algorithm for digital audio data in security applications. *Scientific Reports*, 13(1):18432, October 2023.

[YZH+21] Zhong-Liang Yang, Si-Yu Zhang, Yu-Ting Hu, Zhi-Wen Hu, and Yong-Feng Huang. VAE-Stega: Linguistic Steganography Based on Variational Auto-Encoder. *IEEE Transactions on Information Forensics and Security*, 16:880–895, September 2021.

[ZDX+23] Jing Zhang, Long Dai, Liaoran Xu, Jixin Ma, and Xiaoyi Zhou. Black-Box Watermarking and Blockchain for IP Protection of Voiceprint Recognition Model. *Electronics*, 12(17):3697, January 2023.

[ZGJ+18] Jialong Zhang, Zhongshu Gu, Jiyong Jang, Hui Wu, Marc Ph. Stoecklin, Heqing Huang, and Ian Molloy. Protecting Intellectual Property of Deep Neural Networks with Watermarking. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pages 159–172, Incheon Republic of Korea, May 2018. ACM.

[ZJW+20] Ying-Qian Zhang, Yi-Ran Jia, Xingyuan Wang, Qiong Niu, and Nian-Dong Chen. DeepTrigger: A Watermarking Scheme of Deep Learning Models Based on Chaotic Automatic Data Annotation. *IEEE Access*, 8:213296–213305, November 2020.

[ZZZ+20] Qi Zhong, Leo Yu Zhang, Jun Zhang, Longxiang Gao, and Yong Xiang. Protecting IP of Deep Neural Networks with Watermarking: A New Label Helps. In Hady W. Lauw, Raymond Chi-Wing Wong, Alexandros Ntoulas, Ee-Peng Lim, See-Kiong Ng, and Sinno Jialin Pan, editors, *Advances in Knowledge Discovery and Data Mining*, volume 12085, pages 462–474. Springer International Publishing, Cham, 2020.