UNIVERSITY
OF MANNHEIM
School of Business Informatics
and Mathematics

Chair of Practical Computer Science IV:
Dependable Systems Engineering

Bachelor's Thesis

Secure Voice Authentication in Smart Home Assistants

as part of the degree program Bachelor of Science Business Informatics submitted by

Nils Becker
Matriculation number 1687943

on August 31, 2023.

Supervisor: Prof. Dr. Frederik Armknecht
M. Sc. Christian Müller
M. Sc. Jochen Schäfer

Abstract

With the proliferation of smart home technology, establishing secure and reliable user authentication mechanisms has become imperative. This bachelor's thesis presents a novel voice authentication system tailored for smart home assistants. By leveraging the Support Vector Machine (SVM) as the foundational machine learning tool and Mel Frequency Cepstral Coefficients (MFCCs) for voice feature extraction, this research offers a robust solution to counteract cloned voices while proficiently identifying genuine users. Preliminary results, showcasing the Polynomial kernel's superior efficacy, especially at an MFCC configuration of 128, highlight the potential of this approach. However, observations hint at possible overfitting risks at escalated MFCC configurations, necessitating continuous system refinement. This work underscores the potential of SVM and MFCCs in fortifying smart home assistant security against emerging cyber threats and sets the groundwork for further enhancements in voice-based authentication systems.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

1 Introduction

1.1 Motivation

In the dawn of the Internet of Things (IoT) era, smart home assistants have emerged as one of the most sought-after consumer devices, seamlessly integrating into the fabric of our daily lives. These devices, once seen as a luxury, are now omnipresent, serving users worldwide and offering convenience, control, and information at the mere utterance of a command [JS18]. Yet, with every technological advancement comes an associated risk. The more connected we become, the more exposed we are to potential cyber threats. And herein lies the crux of the current challenge: ensuring that these devices, which have access to vast swaths of personal data, remain impervious to malicious threats.

The voice, being one of the most distinct characteristics of an individual, holds promise as a potent authentication tool. Historically, voice has been used for identification and verification in various spheres, from secure telephonic transactions to access control in restricted areas [WT17]. However, as the digital landscape has evolved, so too have the threats. With the advent of sophisticated voice synthesis tools and deepfake technologies, the voice, as an authentication medium, faces unprecedented challenges. There have been documented instances where cybercriminals, armed with cloned voice recordings, have orchestrated elaborate scams, defrauding individuals and corporations of significant sums [MG19]. In a world where hearing is believing, how do we trust the voices we hear?

Furthermore, as smart home devices become more entrenched in our daily routines, they no longer simply play music or provide weather updates. They are now integrated with security systems, financial transactions, and health monitoring setups. A breach isn't just an inconvenience; it poses tangible risks to personal safety, financial security, and privacy. Given the multitude of functionalities these devices now oversee, ensuring robust, foolproof authentication becomes not just desirable but imperative.

The security of smart home assistants isn't just about protecting the individual user. Consider a scenario where an unauthorized entity gains control over a multitude of devices across a city or even a country. The orchestrated misuse of these devices could lead to widespread disruptions, potentially crippling infrastructures and causing chaos [PJ21]. The larger implications of such vulnerabilities in the IoT ecosystem underscore the urgency of the situation.

This thesis, therefore, isn't just an academic exercise or a pursuit of technological novelty. It's a response to a genuine and pressing challenge in our hyper-connected world. As we inch closer to a future where smart homes are the norm rather than the exception, the need to ensure that these systems are both user-friendly and secure becomes paramount. The objective is clear: to develop an authentication system that's not only robust against present-day threats but is also adaptable, ready to face the challenges of tomorrow.

1.2 Related Work

As smart home assistants are proliferating, the need for robust speaker authentication mechanisms becomes paramount to ensure user security and privacy. Speaker authentication, inherently biometric in nature, offers a unique solution, negating the need for traditional password-based systems [ZZ17]. However, its increasing integration into smart home devices has given rise to novel cybersecurity challenges. Recent studies highlight vulnerabilities where attackers can exploit non-linearities in speaker recognition systems using adversarial examples, enabling unauthorized access [CW18]. Additionally, there are concerns over voice cloning technologies that can generate near-realistic voice imprints of users, posing potential security breaches in speaker-authenticated systems [CB19]. Moreover, the interconnected nature of smart home devices raises data privacy issues, with the possibility of eavesdroppers leveraging compromised devices to extract sensitive user voice data [AP17]. Recognizing these threats, researchers have been exploring defensive techniques, such as deep neural network-based countermeasures against adversarial attacks [GRC19] and continuous voice authentication to detect cloned voices in real-time [PS19]. While significant progress has been made, ensuring airtight cybersecurity in speaker authentication for smart home assistants remains an active area of exploration.

Speaker recognition, an integral discipline within biometrics, focuses on the identification and verification of individuals based on their unique vocal signatures [Rey00]. With the increasing adoption of voice-based systems in various domains, such as smart homes and banking, advancements in speaker recognition have become paramount to address evolving challenges.

Historically, MFCCs have been the cornerstone in voice feature extraction, evolving from the short-term power spectrum of sound. They adeptly capture crucial information emanating from the human vocal tract [DM80]. The power of MFCCs not only lies in capturing speaker-specific characteristics but also in presenting the intricacies of human speech, both physiologically and behaviorally.

However, the quest for perfecting speaker recognition didn't end with MFCCs. Many researchers explored complementary features, with significant ones being Linear Predictive Coding, Perceptual Linear Prediction, and pitch analysis. Such features, when used collectively, can significantly improve the robustness of recognition systems, making them resilient to diverse acoustic environments and degraded audio inputs [MBE10].

From a classifier perspective, Gaussian Mixture Models (GMM) initially enjoyed widespread adoption due to their capability in modeling the distribution of features associated with different speakers [RR95]. But as the landscape of machine learning evolved, so did the techniques for speaker recognition. The emergence of Deep Neural Networks and Recurrent Neural Networks (RNNs) brought forth models that could capture intricate, nonlinear voice patterns with remarkable efficiency. When combined with MFCCs, particularly in hybrid systems merging GMMs with neural architectures, there was a noticeable leap in speaker verification accuracy [Sny+18].

Despite these technological advancements, some traditional models like the SVM retain their importance in the field. SVMs, known for their efficiency in high-dimensional spaces, utilize optimally placed hyperplanes to classify data. Their ability to generalize well, even with limited data, makes them particularly relevant in today's privacy-centric world, where access to massive datasets might be restricted [CSR06].

Lately, a trend in the research community leans towards the combination of various features and classifiers. RNNs, which inherently process time-sequence data, are seen in tandem with MFCCs to capture both time and spectral details. Furthermore, the advent of transformer models, replete with their self-attention mechanisms, has shown potential in identifying long-term dependencies in speech patterns, further expanding the horizons of speaker recognition research.

In the context of the vast and diverse landscape of speaker recognition, it becomes essential for novel systems to make well-informed decisions regarding feature extraction and classification techniques. The resilience of MFCCs and the versatility of SVMs, especially when resources or data availability is a concern, makes their combined use a pragmatic choice for developing a balanced, efficient, and adaptable speaker recognition system.

1.3 Contribution

With the rise of ubiquitous computing and the proliferation of smart home devices, ensuring robust authentication mechanisms has become paramount for user privacy and security. The relevance of a voice authentication system for smart home assistants cannot be understated. Voice, as a biometric modality, offers a natural and seamless method of interaction with smart devices. However, its very convenience can also be a potential point of vulnerability. Recent advancements in deep learning and voice synthesis have made it possible to clone voices with a degree of accuracy that is alarmingly indistinguishable from the original [ZQZ20]. Such cloned voices can potentially be misused to gain unauthorized access to smart home systems, compromising user privacy, and even controlling connected smart devices with malicious intent.

This bachelor thesis, by focusing on implementing a robust voice authentication mechanism that can differentiate between authentic and cloned voices, fills a crucial gap in the current cybersecurity landscape. As smart home ecosystems continue to integrate more deeply into our daily lives, controlling everything from our lighting to our banking, the potential risks associated with unauthorized access become increasingly consequential. Ensuring that only authorized voices can command such systems, and actively blocking cloned or replicated voices, adds an essential layer of protection against potential cyber-attacks.

Moreover, as cyber-attacks become more sophisticated, relying solely on traditional password-based authentication methods or even simplistic voice recognition can leave systems vulnerable. This work, therefore, not only contributes a vital tool in the defense against such threats but also underscores the evolving nature of security challenges in the age of smart homes. By bridging the gap between biometric authentication and the mitigation of voice deepfakes, this thesis emphasizes a proactive approach to cybersecurity, anticipating and addressing next-generation threats in the rapidly evolving landscape of smart home technologies [SZ19].

1.4 Organization of this Thesis

This thesis is organized into five main chapters, each providing insights and details into different aspects of the research conducted on voice authentication for smart home assistants.

1. **Background**
   This chapter provides the foundational knowledge essential for understanding the methodologies used in this research. It begins by introducing the basic principles of sound, followed by an in-depth discussion on MFCCs, detailing their background, the Mel Scale, and the extraction process. The chapter also covers the fundamentals of SVMs, focusing on their motivation, mathematical underpinnings, the importance of hyperparameters,

and tuning techniques. Finally, datasets and libraries, such as Mozilla Common Voice, Tacotron, Scikit Learn, and Librosa, are introduced.

2. **Experimental Design and Methods**
   Here, the specific methodologies employed in the research are discussed. The chapter lays out the system description detailing each step from initialization to error handling. It then elaborates on data gathering processes, both for genuine and cloned voice samples. Feature extraction techniques, model choices, and hyperparameter configurations are also presented. The implementation section explicates the actual process of model training, speaker evaluation, and the integration of the model into the smart home assistant workflow.

3. **Results**
   This chapter presents the results obtained from the experiments. It offers a profound analysis of the performance of various SVM configurations assessing them based on metrics such as accuracy and F1 score. A comprehensive discussion compares the performances and interprets the findings. The chapter culminates with visual aids, including a confusion matrix, to provide a clearer understanding of the results.

4. **Conclusion**
   Concluding the thesis, this chapter encapsulates the research, summarizing the main findings and their implications. It also looks ahead, proposing areas for future research and how the current research could be extended or improved upon.

2 Background

2.1 Sound

Sound, a type of mechanical wave, is conveyed through mediums such as air due to the compressions and rarefactions of particles [RMW07]. Originating from object vibrations, sound propels particles within its surrounding medium. These vibrations manifest distinct characteristics, such as frequency, amplitude, and phase. The human auditory system perceives sounds within a frequency spectrum approximately between 20 Hz and 20,000 Hz [Moo12]. For audio and speech recognition tasks, it's crucial to distill representative features from audio signals. The MFCCs have been acknowledged as a predominant method for this endeavor. MFCCs encapsulate the short-term power spectrum of sounds and are designed to mirror the human ear's frequency-dependent resolution [DM80]. The ensuing section delves into the computation and utility of MFCCs.

2.2 Mel Frequency Cepstral Coefficients

2.2.1 Background

An audio signal undergoes constant changes. For simplification of the analysis, it is assumed that the audio signal remains relatively stable on short time scales. Hence, the signal is divided into frames. Research has indicated that the optimal frame length is between 25 and 64 ms [JY11]. If the frame is too short, there are insufficient samples for a reliable spectral estimate. Conversely, if the frame is too long, the signal changes significantly within the frame.

The next step involves calculating the power spectrum of each frame. This action is motivated by the functioning of the human cochlea, an organ in the ear. The cochlea vibrates at different locations depending on the frequency of incoming sounds. As specific regions of the cochlea vibrate, small hairs get stimulated, and corresponding nerves fire to inform the brain about the presence of certain frequencies. In a similar manner, the periodogram estimate identifies the frequencies present in each frame [Bä+22].

The periodogram spectral estimate contains information that isn't directly necessary for Automatic Speech Recognition. Especially, the cochlea cannot differentiate between closely spaced frequencies, and this effect becomes more pronounced at higher frequencies [Bä+22]. Therefore, periodogram bins are grouped together and summed to determine the energy distribution across different frequency regions. This aggregation is accomplished using a Mel filterbank. The first filter is narrow and measures the energy near 0 Hz. As frequencies increase, the filters become wider, as fine variations become less of a concern. The objective is to capture the approximate energy distribution at each frequency spot. The Mel scale provides guidance in determining the spacing and width of the filterbanks. Equations below should be referred to for the calculation of spacing [Bä+22].

Once the energy values from the filterbank are obtained, their logarithm is taken. This logarithmic transformation is motivated by human hearing. Loudness perception doesn't follow a linear scale. Generally, to double the perceived volume of a sound, the energy has to be increased by a factor of 8. Consequently, large energy variations might not sound significantly different if the sound is already loud. The logarithm operation aligns the features more closely with human auditory perception [Bä+22].

The final step involves the computation of the Discrete Cosine Transform (DCT) of the logarithmic filterbank energies. Two primary reasons drive this transformation. Since the filterbanks overlap, energies from different filterbanks are highly correlated. The DCT is employed to decorrelate these energies [Bä+22].

2.2.2 Mel Scale

The Mel scale quantifies the perceived frequency or pitch of a pure tone relative to its measured frequency. Human perception exhibits greater sensitivity to small changes in pitch at lower frequencies compared to higher ones. By incorporating the Mel scale, features are aligned more closely with human auditory perception [DM80]. To change a frequency from Hz to Mel the following formula can be used:

$$M(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right). \tag{2.1}$$

The inverse formula follows as:

$$M^{-1}(m) = 700\left(10^{\left(\frac{m}{2595}\right)} - 1\right). \tag{2.2}$$

2.2.3 Extraction

The extraction involves the sequential steps of (1) framing, (2) power spectrum computation, (3) filter bank calculation and application, and (4) discrete cosine transformation.

2.2.3.1 Framing

As mentioned in **??** the signal is framed into snippets of a length of 25-64ms. The amount of samples per frame then depends on the sampling rate of the signal. If i. e., a 16 Hz signal is to be framed with a frame length of 25ms, the amount of samples per frame will be in this case 400 samples.
The following steps will focus on extracting MFCCs for an individual frame.

2.2.3.2 Power Spectrum

Using the Discrete Fourier Transform (DFT) in **??** power spectrum for each of the individual audio frames can be calculated.

The (DFT) is an instrumental algorithm in the realm of digital signal processing, enabling the transformation of discrete-time signals from the time domain into the frequency domain [OS99]. Given an input sequence $x[n]$ of length N, the DFT outputs a sequence $X[k]$ of the same length, representing the frequency components of the input. The relation between $x[n]$ and $X[k]$ is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j(2\pi/N)\cdot kn} \tag{2.3}$$

for $0 \leq k < N$. Each $X[k]$ value is a complex number, with its magnitude representing the amplitude of the corresponding frequency component and its phase angle representing the phase shift. The power spectrum, a pivotal concept in spectral analysis, can be derived from the DFT outputs as $|X[k]|^2$. It provides a measure of the energy associated with each frequency component, offering insights into the dominant frequencies present in the signal [LMU18].

2.2.3.3 Computing Mel Filterbanks

Mel filterbanks are designed to approximate the human ear's nonlinear frequency sensitivity, allowing audio features to emphasize perceptually relevant frequency bands, making them more effective for tasks like speech and speaker recognition. The effectiveness of Mel filterbanks in speaker recognition stems from their ability to prioritize the frequency components of speech that humans naturally pay attention to, thereby capturing distinct speaker-specific characteristics rooted in the very nuances that human listeners would subconsciously use to differentiate between speakers [DM80].

Algorithm 1: Mel Filterbank Calculation

```
Data: Lowest Frequency, Highest Frequency, Number of Filter Banks
Result: Mel Filterbanks $H_m(k)$
$mel_{min} \leftarrow$ convertToMel(lowestFrequency)                          // Using ??
$mel_{max} \leftarrow$ convertToMel(highestFrequency)
$num\_freqs \leftarrow$ numOfFilterBanks $+ 2$
$mel\_frequencies \leftarrow$ equallySpacedFrequencies($mel_{min}, mel_{max}, num\_freqs$)
$hz\_frequencies \leftarrow$ convertToHz($mel\_frequencies$)                  // Using ??
for $i = 1$ to $length(hz\_frequencies) - 2$ do
    $H_i(k)$                                                                  // Using ??
end
```

$$H_m(k) = \begin{cases} 0 & \text{if } k < f(m-1) \\ \frac{k-f(m-1)}{f(m)-f(m-1)} & \text{if } f(m-1) \leq k < f(m) \\ 1 & \text{if } f(m) \leq k \leq f(m+1) \\ \frac{f(m+2)-k}{f(m+2)-f(m+1)} & \text{if } f(m+1) < k \leq f(m+2) \\ 0 & \text{if } k > f(m+2) \end{cases} \tag{2.4}$$

In Equation **??**, the calculation of $H_m(k)$, the value of the m^{th} filter in a Mel-frequency filterbank for a given frequency k, is described. The filter value is set to zero when k is less than $f(m-1)$ or greater than $f(m+2)$. For frequencies falling between $f(m-1)$ and $f(m)$, the filter value undergoes a linear increment from 0 to 1 as the frequency advances from $f(m-1)$ to $f(m)$. Conversely, a linear decrement from 1 back to 0 is observed for frequencies lying between $f(m+1)$ and $f(m+2)$. A constant value of 1 is attributed to the filter for frequencies ranging from $f(m)$ to $f(m+1)$. This equation serves to model the contribution of each frequency to the respective filter in the Mel scale [RS78].

2.2.3.4 Applying Mel Filterbanks

The energy of each filterbank is calculated by multiplying every filterbank calcultated in **??** with the power spectrum that was previously computed in **??**. In this process, the energy in each frequency range is weighted with the specific sensitivity of the corresponding filterbank. After all results of the multiplications have been summed up, the total energy carried by each filterbank is obtained; this represents the energy carried by each frequency range.
n values are produced for n filterbanks. The *log* is then used to obtain the n *log* filterbank energies.
?? illustrates the different filterbank energies, given a power spectrum of an audio frame and a set of, in this case, 26 filterbanks. It can be seen that the audio signal carries significantly more power in lower freqencies than in higher ones. Therefore the windowed power spectrum using a low-frequency filter as in c) results in a higher maximum amplitude as seen in d). Part a) also visualizes the logarithmic nature of the Mel scale as the filterbanks grow in size with rising Hz frequency.

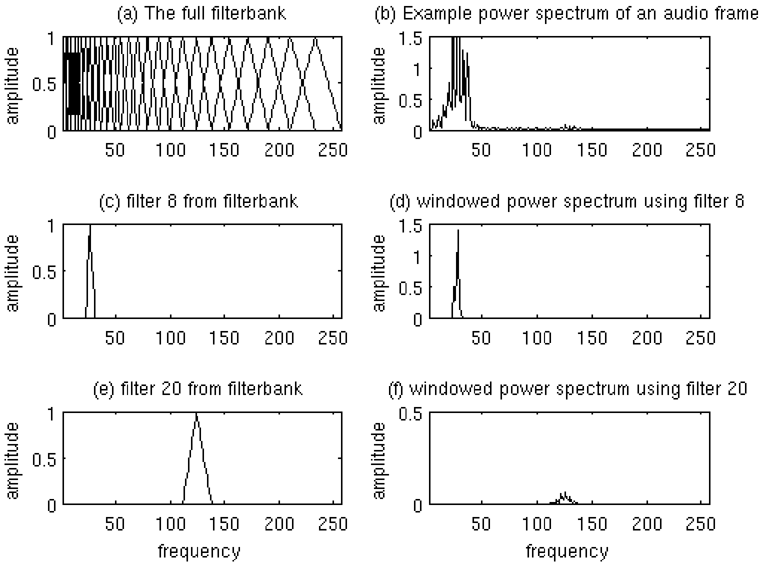


Figure 2.1: Visualization and Application of Mel Filterbanks [GD+17]

2.2.3.5 Discrete Cosine Transform

Once the Mel filterbank energies are derived from a signal, these energies are commonly found to be highly correlated [DM80]. To decorrelate these energies, Discrete Cosine Transformation is employed. By converting the Mel filterbank energies into the cepstral domain using the DCT, the MFCCs encapsulate the envelope of the short-term power spectrum. This transformation to the cepstral domain not only condenses the data dimensionality but also ensures that the ensuing coefficients are largely uncorrelated. This decorrelation becomes particularly beneficial when these MFCCs are used as input features for SVMs, as SVMs often perform better with decorrelated feature vectors [Ben+09].

2.3 Support Vector Machines

2.3.1 Motivation

A Support Vector Machine (SVM) is a supervised machine learning algorithm primarily devised for binary classification tasks, although its application can be extended to multiclass classification and regression problems [CV95]. The core principle underlying SVM is to identify the hyperplane that best separates the data into classes with the maximum margin. The margin, in this context, is defined as the distance between the hyperplane and the closest data points from each class, which are termed as the support vectors. These support vectors are pivotal as they essentially define the position of the separating hyperplane. For non-linearly separable data, SVMs utilize the kernel trick, wherein the data is implicitly mapped to a higher-dimensional space, facilitating the discovery of a separating hyperplane in this transformed space. The choice of kernel, such as linear, polynomial, or radial basis function, plays a crucial role in determining the efficacy of the SVM in complex decision boundaries [RLP06].

2.3.2 Mathematical Foundations

SVMs are powerful supervised machine learning algorithms primarily utilized for classification and regression tasks. Mathematically, the core objective of a SVM is to find the optimal hyperplane that maximizes the margin between two classes in a given feature space [CV95]. Given a dataset $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ represents the feature vectors and $y_i \in \{-1, 1\}$ are the class labels, the hyperplane is defined as:

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{2.5}$$

Here, $\mathbf{w}$ is the weight vector, and b is the bias. The optimal hyperplane is determined by minimizing $||\mathbf{w}||^2$ subject to constraints $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$ for all i. This ensures that the hyperplane correctly classifies all data points while maximizing the margin. For non-linearly separable data, SVMs employ the kernel trick, where a kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$ implicitly computes the dot product in a higher-dimensional space, allowing the algorithm to find a separating hyperplane in that space [RLP06]. The introduction of slack variables $\xi_i \geq 0$ allows SVMs to cope with non-perfectly separable data by introducing a penalty for misclassifications [RLP06].

2.3.3 Hyperparameters and Their Effects

A variety of hyperparameters in SVM affect its performance:

- **Kernel Function**: A SVM uses different kernel functions to transform input data into a higher-dimensional space, enabling classification of non-linear data. Common kernels include linear, polynomial and radial basis function (RBF) [RLP06].

- **Regularization (C)**: The C parameter trades off correct classification of training examples against maximization of the decision function's margin. For larger values of C, a smaller margin will be accepted if the decision function classifies all training points correctly [RLP06].

2.3.4 Tuning Hyperparameters

2.3.4.1 Grid Search

Grid search is commonly described as an exhaustive searching technique. In this method, a predefined subset of the hyperparameter space is systematically traversed to determine the optimal parameter combination for SVMs [HCL03]. Key hyperparameters of SVMs, such as the regularization parameter C and the kernel are often subjected to this search.

2.3.4.2 Cross Validation

Cross-validation, another essential procedure in machine learning, is designed to assess the performance of a model in an unbiased manner. Data is divided into k subsets. For each subset, the model is trained on $k-1$ of them and validated on the remaining one. This process is repeated k times, with each subset serving as the validation set once. A performance metric, often an average over all iterations, is then used to evaluate the model's overall efficacy [HCL03].

2.3.4.3 Combination

In the SVM hyperparameter tuning process, grid search and cross-validation are frequently combined. For every combination of hyperparameters identified by the grid search, performance is evaluated using cross-validation. By doing so, a more comprehensive and unbiased estimate of model performance for each hyperparameter combination is obtained. The set of hyperparameters that yields the best average performance across all cross-validation folds is typically selected as optimal [HCL03].

2.3.4.4 Performace Evaluation

Two of the most widely used metrics for performance evaluation of a SVM are accuracy and the F1 score.

The accuracy is a simple metric which provides a ratio of the correctly predicted instances to the total instances. It is mathematically given by:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{2.6}$$

However, in imbalanced datasets where one class significantly outnumbers the other, accuracy might not provide a clear picture of the model's performance. This is where the F1 score becomes essential.

To understand the F1 score, we first need to define precision and recall. Precision is the ratio of correctly predicted positive observations to the total predicted positives, while recall (also known as sensitivity) is the ratio of correctly predicted positive observations to all the actual positives. Mathematically,

$$\begin{aligned}\text{Precision} &= \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}\\ \text{Recall} &= \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}\end{aligned}$$

The F1 score is the harmonic mean of precision and recall, giving a balance between the two. It is particularly useful in cases where the false positives and false negatives have different costs. The F1 score is defined as:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.7}$$

A model with a higher F1 score is considered better at achieving a balance between precision and recall. By evaluating SVMs using both accuracy and the F1 score, one can have a holistic understanding of the model's capabilities and limitations, especially in diverse and imbalanced scenarios [Bis06].

To visualize these scores and therefore the performance, the confusion matrix, sometimes referred to as an error matrix, is a pivotal tool in machine learning, specifically in classification tasks. This matrix presents a detailed breakdown of true positive, false positive, true negative, and false negative predictions made by the classifier, allowing users to compute various performance metrics, such as accuracy, precision, recall, and the F1 score [Faw06]. By providing insights into the types and frequencies of classification errors, the confusion matrix aids in diagnosing where a classifier is failing and can guide subsequent improvements [Pow11]. As such, it is an indispensable tool in evaluating the effectiveness and reliability of classification algorithms.

2.3.5 Relevance in Speaker Recognition and -Authentication

Speaker recognition is the overarching process of identifying individuals based on their vocal characteristics, while speaker authentication specifically ascertains the claimed identity of a speaker through voice analysis. The unique strength of SVMs in handling high-dimensional data, coupled with their capacity to identify complex, non-linear boundaries, makes them particularly relevant in speaker recognition. They can delineate the subtle differences in voice features, achieving impressive accuracy, especially when combined with robust voice features like MFCCs [CSR06]. The interpretability of SVMs ensures that researchers and practitioners can understand the importance of various voice features in the classification decision, further solidifying SVMs as a valid option in the speaker authentication domain.

2.4 Dataset

2.4.1 Mozilla Common Voice

Mozilla's Common Voice project is a unique and valuable contribution to the world of open-source datasets, primarily aimed at voice and speech recognition research. Launched by the Mozilla Corporation, it aims to open up voice technology by providing a vast, publicly available database of voice samples in numerous languages. The dataset consists of short voice recordings from volunteers around the globe, paired with the corresponding transcriptions. Its diverse collection offers a wide range of accents, tones, and speech patterns, making it an essential resource for training robust voice recognition models [Cor23].

2.4.2 Voice Cloning

Voice cloning, often equated with voice synthesis, delves into the intricate process of generating a synthetic voice that closely emulates a target voice, aiming to achieve near-perfect fidelity. Historically rooted in the realm of speech synthesis, traditional methodologies predominantly revolved around concatenative and parametric strategies [HB96]. Concatenative synthesis, as the name suggests, involves piecing together short speech samples from a large database to form coherent and continuous speech, while parametric synthesis deploys a set of parameters to model vocal tract features, and then synthesizes speech from these parameters. As the field of machine learning burgeoned, particularly with the advent of deep learning, a paradigm shift towards more advanced methods employing neural architectures was observed. Notably, models harnessing the power of Generative Adversarial Networks and Variational Autoencoders for voice synthesis emerged, receiving considerable attention for their efficacy [Hsu+18; VDO+16]. The fundamental hypothesis guiding these models is that, upon being trained on copious amounts of voice data, they possess the capability to engender speech waveforms that bear striking acoustic resemblance to the original voice. In more recent developments, the attention mechanism, epitomized by transformer models, has been infused into speech synthesis frameworks, leading to remarkable enhancements in the naturalness and expressiveness of the synthesized voice [Ren+19]. Nonetheless, despite these monumental strides, challenges persist, both technical and ethical. Of mounting concern is the potential misuse of voice cloning technologies in the creation of deepfake audios, a malicious application that has ignited significant discourse and scrutiny within the academic and research communities [ZQZ20].

2.4.3 Resemble AI

Resemble AI has emerged as a prominent player in the area of voice cloning technologies. Utilizing advanced deep learning algorithms and bespoke neural network architectures, their services can accurately clone human voices with a limited amount of sample data. This capability allows for the generation of synthetic voices that bear a striking resemblance to the original voice [AI23].

2.4.4 Tacotron

Tacotron, an end-to-end generative text-to-speech model, has made significant strides in the field of speech synthesis. Developed by researchers at Google, Tacotron converts text into spectrograms, a visual representation of the frequency content of a signal as it varies with time, which are then transformed into waveforms to generate a synthetic voice. Through its sequence-to-sequence architecture with attention, Tacotron is capable of producing highly natural-sounding speech, capturing intonations and rhythms of human voice effectively [Wan+17a].

2.4.5 Scikit Learn

Scikit-learn is a renowned open-source machine learning library in Python, revered for its simplicity and versatility in facilitating a range of machine learning and statistical modeling tasks [Ped+11]. One of the standout features of Scikit-learn is its implementation of SVMs. The library offers both simple linear SVMs and ones capable of handling non-linear classification using various kernel tricks. The Support Vector Classification SVC class in scikit-learn allows users to customize various parameters such as the type of kernel, penalty parameters, and others, providing a fine-grained control over the SVM behavior [dev23]. With its extensive documentation and an active community, Scikit-learn's SVM module stands as a valuable tool for both novice researchers and seasoned data scientists aiming to harness the power of SVMs in their analyses.

2.4.6 Librosa Library

Librosa is a distinguished Python library that has garnered attention in the audio and music analysis community for its comprehensive suite of tools designed to analyze and process audio signals [McF+15]. Notably, among its vast repertoire of functionalities, is the provision to compute MFCCs. With Librosa's *mfcc* function, users can seamlessly extract these coefficients from an audio time series, benefiting from parameters that allow customization in terms of the number of coefficients, the type of mel filter bank, and more [dev21]. The ease of use, combined with Librosa's efficiency and precision, makes it a contemporary asset for researchers and practitioners aiming to harness MFCCs in their auditory analyses.

3 Experimental Design and Methods

3.1 System Requirements

The task to integrate a voice authentication system within a smart home assistant, specific critical requirements are identified. Firstly, the system must be equipped with an efficient voice feature extraction mechanism. For this purpose, MFCCs are exploited as they encapsulate the primary vocal tract characteristics of speakers and offer a compact representation of voice signals. MFCCs transform the voice signal into a series of coefficients, highlighting the essential features suitable for the authentication task. Once the MFCCs are computed, the next requirement is the integration of an adequate machine learning model. Given the nature of the task, it is necessitated that a model is to be chosen that can distinguish between various speakers efficiently. For this, SVM is leveraged for its remarkable capacity in handling high-dimensional data like MFCCs. The SVM is trained on a labeled dataset, where each voice sample is tagged with its corresponding speaker identity, allowing the model to learn and authenticate future unseen voice features. In essence, the synergy of MFCCs and SVM is aimed to ensure secure and accurate voice-based interactions with the smart home assistant.

3.2 System Description

The smart home assistant system is designed to continuously listen for and process voice commands. The overarching flow of the system can be described as follows:

3.2.1 Initialization

- A configuration is passed to the voice interface which specifies keywords and stop words.
- This configuration is logged for debugging and reference.
- Audio is captured from a microphone. This is done in a continuous loop until a specified stop word is recognized.

3.2.2 Voice Capture and Recognition

- The system adjusts the audio recognizer's sensitivity based on ambient noise.
- Audio is captured and stored for further processing.
- The captured audio is then sent to Google's voice recognition service for transcription.

3.2.3 Keyword Detection and Voice Validation

- If a keyword, as specified in the configuration, is detected in the transcribed text, a further validation step is triggered.

- The validation ensures that the voice is neither cloned nor unauthorized.
- Detected voice samples are checked against a list of authorized voice identities.
- If the voice is either cloned or unauthorized, access is denied and the system exits.

3.2.4 Command Processing

- If the voice is authenticated, the system awaits a voice command from the user.
- Captured voice command is transcribed and then processed.
- The result of this command processing is logged, and the system is ready to process subsequent commands.
- The loop continues until the system recognizes a predefined stop word, signaling it to exit.

3.2.5 Error Handling

- In case of unrecognizable audio, the system logs an error but continues listening.
- If there's a problem in requesting recognition results, it's also logged, and the system resumes its listening state.

3.3 Data Gathering

3.3.1 Genuine Voice Samples

A crucial prerequisite for constructing a robust and representative voice authentication system is an extensive dataset with sufficient variety in the voice samples to train the SVM defined in the requirements in **??**. The cornerstone of the dataset was sourced from the *Mozilla Common Voice* project. This open-source database, recognized for its diverse collection of voice recordings spanning multiple languages and demographics, offered a rich reservoir of voice data. Exploited for its extensive size and variety, the Common Voice dataset provided an invaluable foundation, ensuring that the models had exposure to a myriad of phonetic variations, accents, and speech patterns, essential for generalizability. In detail the utilized dataset for this project was the *Common Voice Delta Segment 13.0* [Moz23]. Speakers who had less than five samples in this dataset were sorted out prior to training, to ensure, that the system would be able to pick up on voice features as good as possible. The modified dataset consists of 272 distinct speakers, providing a total of 14360 samples. **??** provides a detailed view into the distribution of samples regarding the amount of samples available split by gender and age of the speakers. Note that 12 male samples were not labled with a specific age.
To complement the *Mozilla Common Voice* data, individual voice recordings were solicited. Two participants were provided with a set of phrases, which they verbally articulated in different environments, ensuring variations in audio quality. These individual spoken phrases were especially instrumental in capturing localized nuances and idiosyncrasies [Ari+18], potentially overlooked in larger, more generic datasets. The combination of the vast and varied *Mozilla Common Voice* dataset with meticulously curated individual recordings culminated in a comprehensive voice dataset.

Age Group	Female	Male
Teens	134	1041
Twenties	595	2641
Thirties	251	2648
Forties	470	1597
Fifties	1314	3242
Sixties	390	10
Seventies	15	-

Table 3.1: Distribution of Gender Across Age Groups

3.3.2 Cloned Voice Samples

The primary methodology employed for voice cloning was harnessed from Resemble AI's advanced voice generation system. The process can be delineated as:

1. **Sample Collection:** Acquired a set of audio samples from the target voice.
2. **Training:** The refined audio samples were processed by Resemble AI's deep learning platform to create a cloned voice model.
3. **Generation:** Using the trained model, new voice samples were generated based on but not limited to predefined phrases [Ari+18].

Apart from this primary training set, additional test samples were procured from Resemble AI's repository. These samples provided an array of voice variations, enabling a multifaceted assessment of the cloned voice model.
Tacotron-based [Hsu+18; Jia+19; She+18; Wan+17b; Wan+18] voice samples were integrated to offer additional depth to the dataset. Recognized for its nuanced voice and sound analysis methodology, Tractoon samples added further granularity to the dataset, ensuring that the cloned dataset spans across a wider spectrum of voices and sounds.

3.4 Feature Extraction

After identifying MFCCs as the features to be chosen in **??**, the task remains to find a good extraction methodology. In many audio processing tasks, especially in speech and speaker recognition, the first 12 or 13 MFCCs are typically used [Log00]. In contrast to this, 128 MFCCs were extracted from the audio samples in this case. This heightened number, compared to more traditional choices, was chosen to provide a richer spectral envelope representation. By using 128 coefficients, a more comprehensive capture of potential spectral nuances, which might be pivotal in distinguishing between individual speakers, was ensured.

Subsequent to the extraction, the mean of these 128 MFCCs was calculated. By computing this mean, transient fluctuations in the audio sample were effectively mitigated, and a consistent representation, highlighting the dominant spectral characteristics intrinsic to the speaker's voice over the sample's duration, was obtained.

To culminate the feature extraction process, normalization of the mean value was performed. This step was deemed necessary to address potential discrepancies arising from varied recording amplitudes or conditions. Through normalization, any bias induced by the absolute magnitude of the MFCCs was neutralized, ensuring that classifiers and subsequent analytical procedures were solely influenced by the inherent patterns of the speaker's voice, rather than extraneous recording factors.

3.5 Model Choice

When comparing different machine learning models for speaker authentication, SVMs have consistently demonstrated remarkable performance attributes when utilizing GMMs [CSR06]. While neural networks, especially deep learning models, have gained traction for their ability to learn intricate patterns in large datasets, they often require extensive computational resources and larger amounts of data to achieve optimal performance [BCV13]. On the other hand, GMMs have historically been a popular choice in speaker authentication; however, they may suffer from overfitting in scenarios with limited data [RR95]. Decision trees and random forests, despite their interpretability, may not always capture the fine-grained nuances in voice data [Bre01]. SVMs, in contrast, offer a balance between computational efficiency and high performance, even in conditions with fewer training samples, making them particulary usefull for this usecase, as i. e., the previously mentioned dataset may vary a lot terms of samples per speaker.

3.6 Hyperparameters

3.6.1 Kernel

The choice of an optimal kernel is paramount in SVM classification, especially when dealing with high-dimensional feature spaces. Given the 128-dimensional feature set, determining the most suitable kernel became a central focus. To this end, a systematic approach involving grid search coupled with cross-validation was employed. The grid search method, a comprehensive algorithmic strategy, facilitated the exploration of a multitude of kernels, including polynomial, RBF, and linear, amongst others. Each kernel was tuned over its hyperparameter space, ensuring an exhaustive search for the ideal configuration. Concomitantly, *scikit-learn* cross-validation, provided a robust framework for assessing the generalization performance of the SVM under each kernel and its corresponding hyperparameter setting.

3.6.2 C

To optimize the performance of the SVM for speaker recognition, the hyperparameter tuning of the C parameter, representing the trade-off between maximizing the margin and minimizing the classification error, becomes indispensable. Through the systematic grid search approach by *scikit-learn* over potential values of C, specifically $\{0.1, 1, 10, 100\}$, the *scikit-learn* cross-validation strategy was employed to assess the model's performance on unseen data for each C value.

3.7 Implementation

3.7.1 Feature Extraction

In the process of audio feature extraction, the prevalent techniques is the extraction of MFCCs from ??. Using the `librosa` library (??), the audio signal is initially loaded into the memory, where the sampling rate and the audio signal are obtained. Subsequently, the MFCCs of the audio signal are computed, resulting in a matrix representation where the number of coefficients is set to 128. It is of note that these coefficients capture the spectral properties of the audio signal and are therefore indispensable for various audio analysis tasks. To further refine the feature representation, the mean of these coefficients across time frames is computed. This results in a single vector that encapsulates the average spectral content of the audio sample. Finally, to ensure the robustness of the model and to mitigate the potential scale differences across various features, normalization is applied to this mean vector. This normalization maps the values of the MFCC mean vector to a range between 0 and 1. The outcome is a normalized MFCC mean vector, which can be effectively utilized for downstream audio analysis tasks.

3.7.2 Model Training

3.7.2.1 Speaker Classification

The methodology employed for processing voice samples involves an intricate set of steps to prepare and train classifiers shown in the UML activity diagram in ??. While the first part of the diagram exemplifies the process of feature extraction, the second part focuses on the creation of SVM classifiers.
Initially, voice samples are imported from an Excel file located at a designated data path. If the mode is set to save the trained models, a timestamped directory is created to store these models. Following this, a process of feature extraction commences, wherein MFCCs are computed iterative for each voice sample. The features, alongside corresponding speaker IDs, are stored in lists for further computations, which marks the end of the feature extraction process.

The speakers and their respective samples are then counted, and for each unique speaker, a binary label system is formulated. In this system, each speaker's samples are labeled as in a binary fashion. Leveraging the *scikit-learn* library (??), Support Vector Classifiers (SVCs) are trained for each speaker based on these binary labels. Note that, the system also provides an option to fine-tune the hyperparameters of the SVC following ??, utilizing all CPU cores in parallel for efficiency, all other hyperparameters were left at the default proposed by *skikit-learn*, due to computational cost. Once the optimal hyperparameters are determined, the SVC is trained on a split dataset, wherein 20% of the data is reserved for testing. This split ensures validation of the classifier's performance. Each SVC, therefore, becomes adept at distinguishing the presence or absence of a specific speaker based on the audio's MFCCs features. If saving is enabled, the trained classifiers, once successfully trained, are stored on disk with a systematic naming convention to ensure future traceability.

To consolidate the training process, an Excel workbook is generated when in save mode. This workbook captures essential metadata, detailing the speaker ID, the audio path, and the corresponding classifier path for each voice sample which later serves as the speaker-database.

3.7.2.2 Clone Classification

The process for distinguishing between original and cloned voice samples commences with the assembly of the relevant dataset. Initially, voice samples are retrieved from a specified directory, focusing on two essential columns representing the speaker details and their corresponding audio path details.

Two rounds of feature extraction are carried out. In the first round, the spectral properties of each voice sample, epitomized as MFCCs, are extracted. As this is dedicated to original samples, each corresponding speaker label is set to 0.

The second extraction phase is directed towards cloned voice samples, which are accumulated from a predefined directory, with each sample being characterized by its MFCCs. Contrary to the original samples, the associated speaker labels for cloned voices are set to 1.

Once the feature sets from both original and cloned samples are unified, a classifier is trained using these features and their corresponding labels. The essence of this training is to capacitate the model to discern between authentic and cloned voice signatures. Optionally, hyperparameter tuning following **??** can be performed to optimize the classifier's performance. Once again, only the Kernel and C parameter are subjected to optimization due to computational cost.

Post-training, the classifier model is persisted for future use. It is saved in a dedicated directory labeled with the current timestamp to ensure traceability and easy retrieval as the model will later be used to filter out potentially cloned voices (**??**)

This function culminates by returning the trained classifier, offering a systematic means for authenticating voice samples against potential cloning.

3.7.3 Speaker Evaluation

The process of speaker identification is rooted in the analysis of voice sample features and their subsequent classification using specialized models and is displayed as an UML activity diagram in **??**. Initially, the voice sample undergoes feature extraction following **??**.

A critical step, before diving into speaker predictions, is to determine the authenticity of the voice sample. The system checks if the voice is a synthetic clone using a dedicated classifier trained in **??**. If deemed cloned, the function concludes its operations, signifying potential security or authenticity concerns.

For genuine voice samples, the identification task ensues and follows to left branch in **??**. The voice's MFCCs are subjected to an array of classifiers gained by **??**, each designed to identify a specific speaker by making a binary prediction. All classifiers which return a positive prediction are stored alongside their respective desicion function.

After all classifiers have made their prediction, the function evaluates the results. In cases where the voice sample resonates with multiple classifiers, suggesting potential ambiguity, the predictions are systematically ranked. This ranking is primarily based on the classifiers' confidence scores, ensuring the most probable speaker is prioritized. The system then communicates the ranked or solitary prediction, depending on the number of affirmative identifications.

3.7.4 Embedding in Smart Home Assistant Workflow

The Smart Home Assistant used is a modified version of the Smart Home Assistant of the Applied AI Course of the Innopolis Univerity [ang21] renamed to *Pacifier*.

Following **??** and **??**, the smart home assistant, adopts a comprehensive multi-phase operational paradigm.

Initially, it resides in a state of passive vigilance, constantly monitoring the surrounding audio for a user-specified keyword. Detection of this keyword by the *SpeechRecognition* library [Zha23] activates an internal audio evaluation mechanism designed to rigorously assess the audio for authenticity and relevance following **??**. Based on the result from **??** the process divides into two possible directions. If a *False* is read from **??**, a cloned voice has been detected and the system will automatically shut down, denying further access as displayed in thge rightmost branch in **??**. If a voice is deemed genuine, shown by the *True* return value from **??**, all authorized IDs are extracted from the smart home assistants configuration.

In the following process the list of authorized speakers will be compared to the list of recognized speakers, and access will be either denied or granted. Granted access will result in the functionality of the smart home assistant continuing normally, while denied access will result in a system shut down.

?? shows an exemplary output when an authorized voice is trying to access the system.

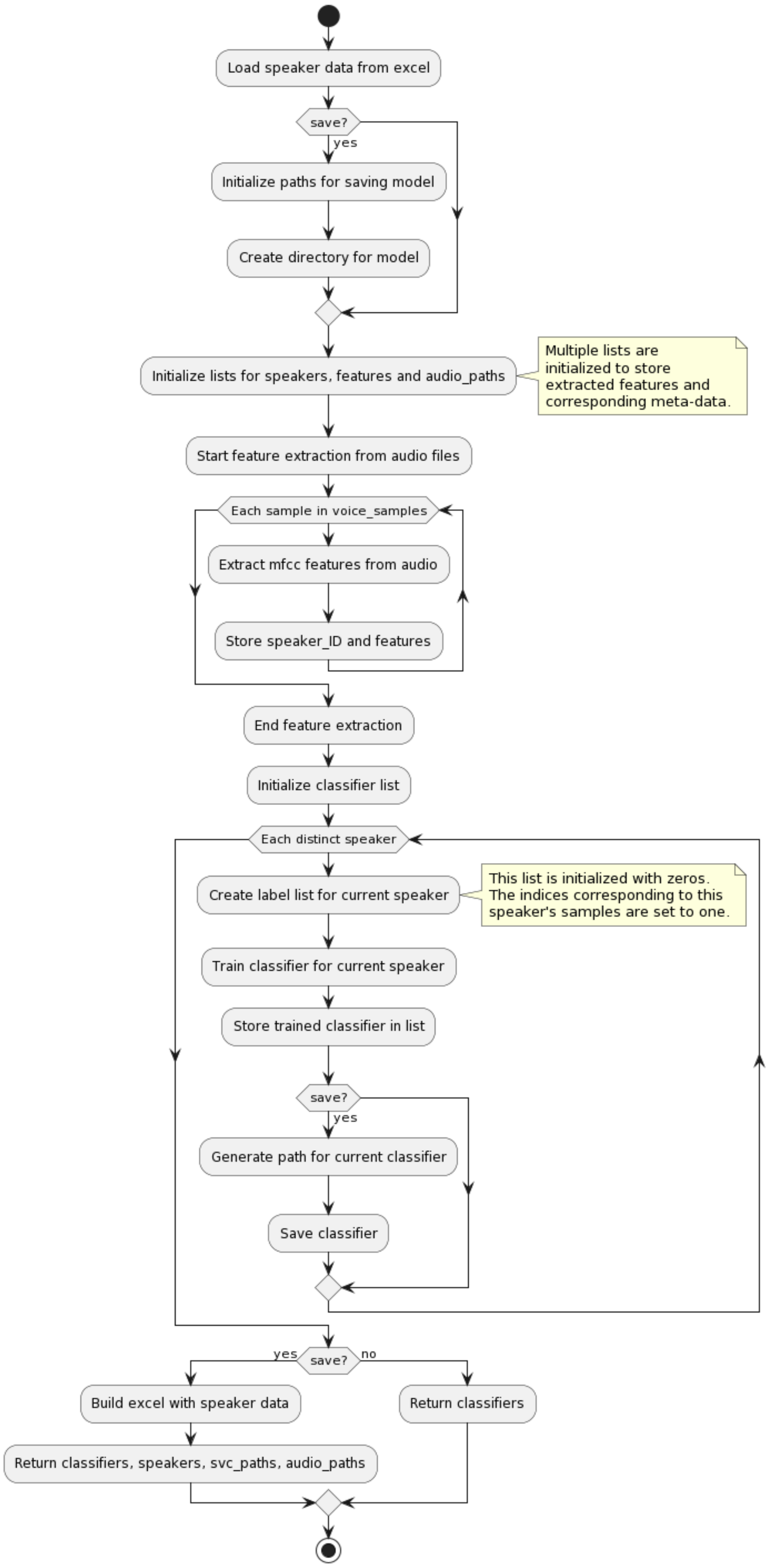

Load speaker data from excel
save?
yes
Initialize paths for saving model
Create directory for model
Initialize lists for speakers, features and audio_paths
Multiple lists are initialized to store extracted features and corresponding meta-data.
Start feature extraction from audio files
Each sample in voice_samples
Extract mfcc features from audio
Store speaker_ID and features
End feature extraction
Initialize classifier list
Each distinct speaker
Create label list for current speaker
This list is initialized with zeros. The indices corresponding to this speaker's samples are set to one.
Train classifier for current speaker
Store trained classifier in list
save?
yes
Generate path for current classifier
Save classifier
yes
save?
no
Build excel with speaker data
Return classifiers
Return classifiers, speakers, svc_paths, audio_paths

Figure 3.1: UML-Activity Diagram of the Speaker Classification Training in ??

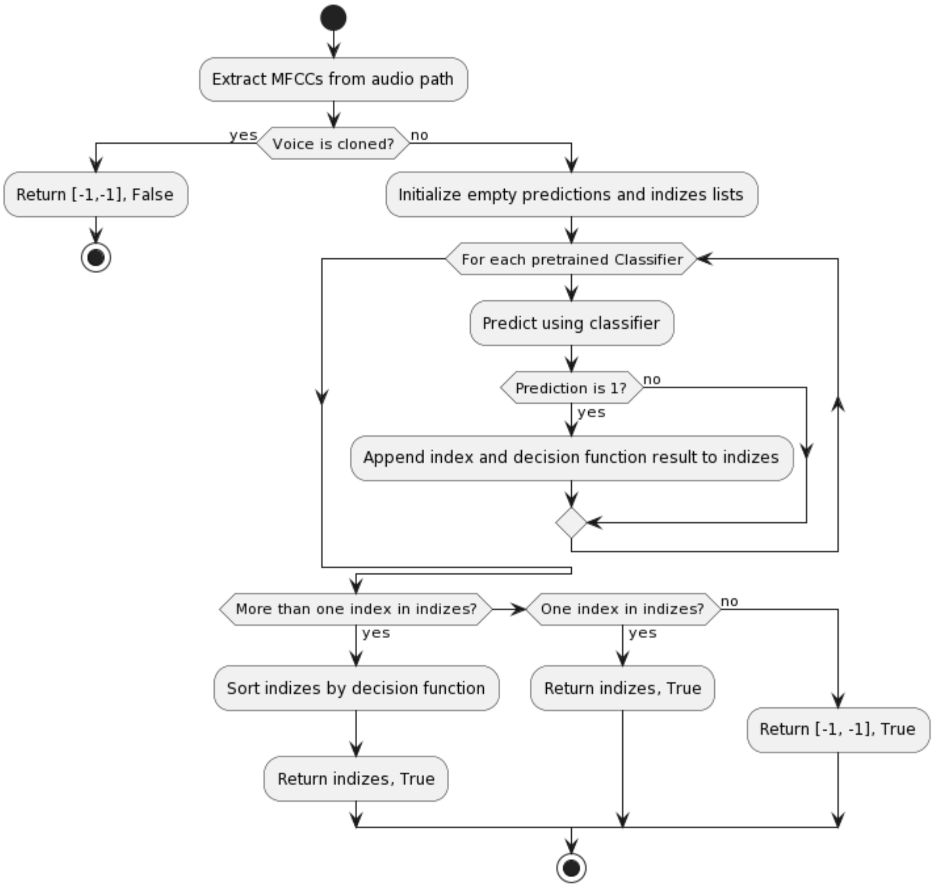
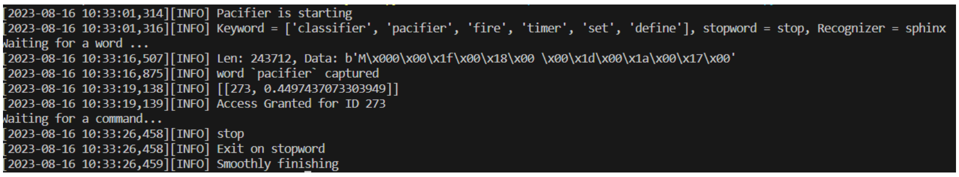

Extract MFCCs from audio path
Voice is cloned?
yes
no
Return [-1,-1], False
Initialize empty predictions and indizes lists
For each pretrained Classifier
Predict using classifier
Prediction is 1?
no
yes
Append index and decision function result to indizes
More than one index in indizes?
One index in indizes?
no
yes
yes
Sort indizes by decision function
Return indizes, True
Return [-1, -1], True
Return indizes, True

Figure 3.2: UML-Activity Diagram of the Speaker Identification Process in ??



```
[2023-08-16 10:33:01,314][INFO] Pacifier is starting
[2023-08-16 10:33:01,316][INFO] Keyword = ['classifier', 'pacifier', 'fire', 'timer', 'set', 'define'], stopword = stop, Recognizer = sphinx
Waiting for a word ...
[2023-08-16 10:33:16,507][INFO] Len: 243712, Data: b'M\x000\x00\x1f\x00\x18\x00 \x00\x1d\x00\x1a\x00\x17\x00'
[2023-08-16 10:33:16,875][INFO] word `pacifier` captured
[2023-08-16 10:33:19,138][INFO] [[273, 0.4497437073303949]]
[2023-08-16 10:33:19,139][INFO] Access Granted for ID 273
Waiting for a command...
[2023-08-16 10:33:26,458][INFO] stop
[2023-08-16 10:33:26,458][INFO] Exit on stopword
[2023-08-16 10:33:26,459][INFO] Smoothly finishing
```

Figure 3.3: Console Output of an Authorized Access to the Smart Home Assistant

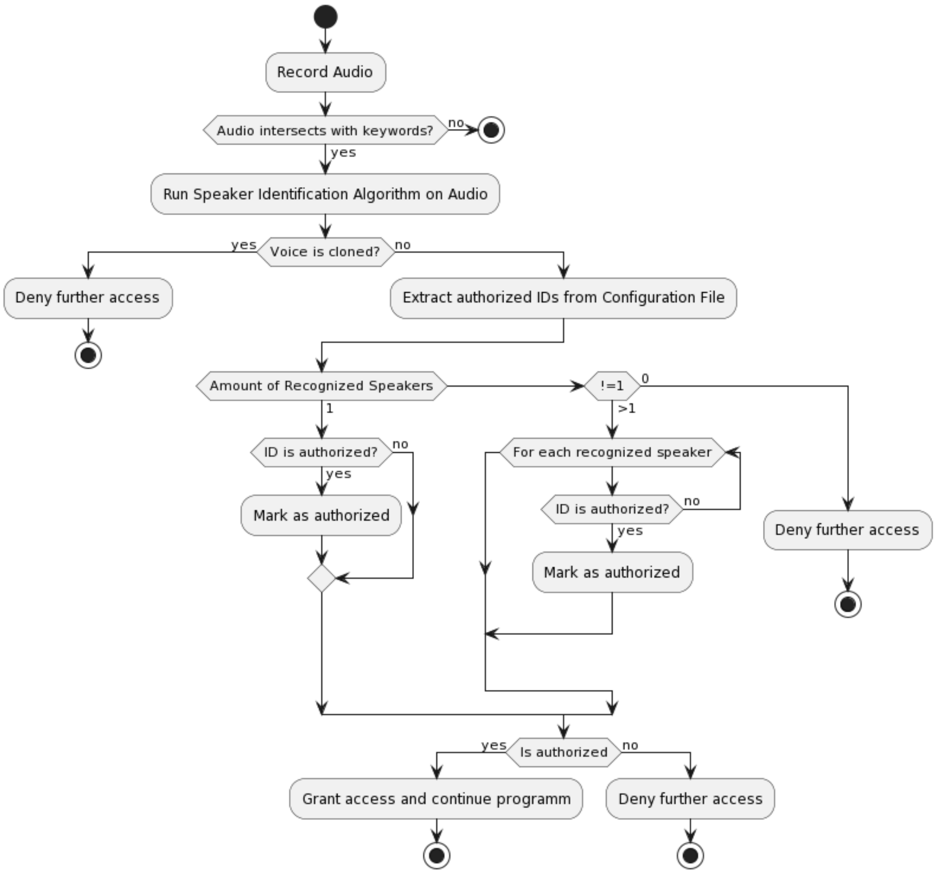

Record Audio
Audio intersects with keywords?
no
yes
Run Speaker Identification Algorithm on Audio
yes
Voice is cloned?
no
Deny further access
Extract authorized IDs from Configuration File
Amount of Recognized Speakers
!=1
0
>1
1
ID is authorized?
no
yes
Mark as authorized
For each recognized speaker
ID is authorized?
no
yes
Mark as authorized
Deny further access
yes
Is authorized
no
Grant access and continue programm
Deny further access

Figure 3.4: UML-Activity Diagram Showing the Embedding of the Speaker Authentication in the Smart Home Assistant Workflow in ??

4 Results

4.1 Analysis

4.1.1 Optimal MFCC Dimensionality and SVM Hyperparameters

For each SVM kernel type - linear, polynomial, and RBF - a detailed investigation was conducted on the impact of different MFCC configurations combined with various regularization parameters C. The parameters have been tested following **??** and implemented via the *scikit-learn* library. The two performance metrices used are the accuracy and the F1 score as explained in detail in **??**. The following values were used to define the optimal Hyperparameters and number of MFCCs.

4.1.2 Linear Kernel

4.1.2.1 Accuracy (%)

MFCCs/C	0.1	1	10	100
16	99.67	99.72	99.88	99.92
32	99.68	99.75	99.84	99.93
64	99.68	99.77	99.84	99.93
128	99.67	99.75	99.85	99.93
256	99.67	99.75	99.84	99.94

Table 4.1: Mean Accuracy for Speaker Identification Using a Linear SVM Kernel Across Different MFCCs Configurations and C Values.

4.1.2.2 F1 Score

MFCCs/C	0.1	1	10	100
16	0.155	0.218	0.201	0.244
32	0.166	0.209	0.187	0.281
64	0.192	0.169	0.199	0.378
128	0.145	0.174	0.214	0.453
256	0.170	0.155	0.234	0.440

Table 4.2: Mean F1 Scores for Speaker Identification Using a Linear SVM Kernel Across Different MFCCs Configurations and C Values.

4.1.3 RBF Kernel

4.1.3.1 Accuracy (%)

MFCCs/C	0.1	1	10	100
16	99.67	99.72	99.88	99.92
32	99.68	99.75	99.84	99.93
64	99.68	99.77	99.84	99.93
128	99.67	99.75	99.85	99.93
256	99.67	99.75	99.84	99.94

Table 4.3: Mean Accuracy for Speaker Identification Using a RBF SVM Kernel Across Different MFCCs Configurations and C Values.

4.1.3.2 F1 Score

MFCCs/C	0.1	1	10	100
16	0.156	0.164	0.237	0.334
32	0.209	0.209	0.255	0.563
64	0.195	0.236	0.267	0.571
128	0.170	0.237	0.226	0.602
256	0.207	0.226	0.234	0.602

Table 4.4: Mean F1 Scores for Speaker Identification Using a RBF SVM Kernel Across Different MFCCs Configurations and C Values.

4.1.4 Polynomial Kernel

4.1.4.1 Accuracy (%)

MFCCs/C	0.1	1	10	100
16	99.78	99.80	99.81	99.84
32	99.87	99.91	99.90	99.90
64	99.94	99.95	99.93	99.93
128	**99.97**	99.97	99.96	99.96
256	99.97	99.97	99.96	99.96

Table 4.5: Mean Accuracy for Speaker Identification Using a Polynomial SVM Kernel Across Different MFCCs Configurations and C Values.

4.1.4.2 F1 Score

MFCCs/C	0.1	1	10	100
16	0.278	0.343	0.373	0.377
32	0.443	0.644	0.661	0.646
64	0.723	0.815	0.800	0.793
128	**0.854**	0.848	0.851	0.830
256	0.842	0.813	0.850	0.825

Table 4.6: Mean F1 Scores for Speaker Identification Using a Polynomial SVM Kernel Across Different MFCCs Configurations and C Values.

4.1.5 Confusion Matrix

?? visualizes a Confusion Matrix (**??**) calculated by the *scikit-learn* library with the choice of hyperparameters and number of MFCCs displaying the performance of the SVM. Each color-intensity of each quater represents the concentration of values in this quater. Here it can be seen that for the optimal choice of hyperparameters a humongous majority of values were predicted correctly as true negatives, which correlates with the relation of the amount of samples in comparison to the amount of speakers.

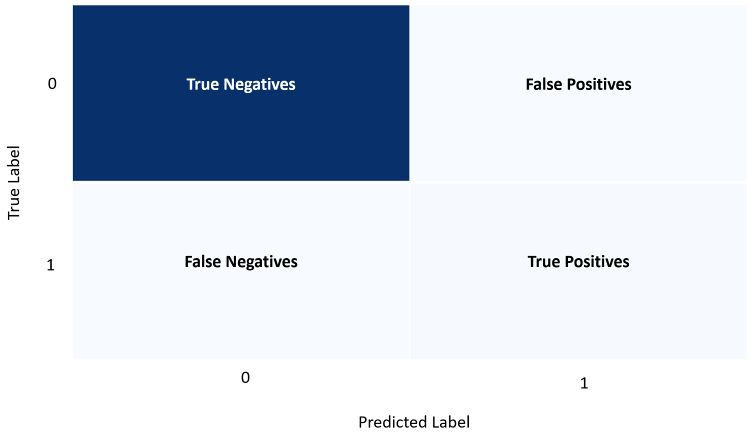


Figure 4.1: Confusion Matrix for the Choice of Hyperparameters and Number of MFCCs (Personal Recreation)

4.2 Discussion

As voice-operated smart home assistants become increasingly embedded within our domestic ecosystems, the assurance of cybersecurity takes center stage. The authentication of legitimate users via voice and the exclusion of imposters becomes paramount. Our exploration into the SVM classifiers with different kernels offers insights into their utility in ensuring the cybersecurity of smart home environments.

4.2.1 Linear Kernel vs RBF Kernel vs Polynomial Kernel

Accuracy: The recorded high accuracy levels are an encouraging sign, especially when considering the strict security requirements of smart home assistants. Notably, the Polynomial kernel stands out, particularly at MFCC = 128, achieving an accuracy of 99.97%. Such superiority might be attributed to the Polynomial kernel's ability to model intricate, higher-degree relationships between features, capturing the nuances and complexities of voice patterns more effectively than the Linear or RBF kernels. However, even with high accuracy levels, the impact of a mere 0.03% error rate in a cybersecurity context cannot be understated - unauthorized access can have cascading repercussions.

F1 Score: In the realm of voice authentication, especially where security is paramount, an F1 score provides a more balanced assessment. The Linear kernel's varied results, despite high accuracy, may be indicative of an imbalance - either too many false positives or too many false negatives. This could mean that the Linear kernel, due to its simplicity, struggles to discern between intricacies of different voices, especially in the presence of background noise or voice modulations typical in a home setting. On the other hand, the Polynomial kernel's high F1 score at MFCC = 128 reaffirms its ability to finely discriminate voices, achieving a good balance between precision and recall.

4.2.2 Impact of MFCC Configuration and Penalty Parameter C

MFCC Configuration: The results suggest that increasing the MFCC value enhances both accuracy and F1 score up to a certain threshold (MFCC = 128). This improvement might be because as the number of mel-frequency cepstral coefficients captured increases, the system gets a more detailed representation of the voice sample, aiding in better differentiation. However, the diminishing returns or potential decline past this threshold might indicate over-complexity, where the system starts to fit to the noise in the data rather than the underlying voice pattern.

Penalty Parameter C: The role of C in SVM is pivotal. It determines the trade-off between achieving a larger margin and ensuring that the training data is correctly classified. Higher values of C for the Polynomial kernel, which leads to better results, suggest that allowing for a stricter boundary - even if it's closer to some data points - results in better generalization to unseen voice samples. This is particularly crucial for cybersecurity, where boundaries need to be tight enough to prevent unauthorized access while ensuring legitimate users are not falsely rejected.

5 Conclusion

5.1 Summary

The evolution of smart home technology has demonstrated a mounting necessity for robust and sophisticated security measures. This thesis focused on enhancing the security framework of a smart home assistant via a voice authentication mechanism, prioritizing the elimination of cloned voices and ensuring seamless identification and authentication of genuine voices.

1. **Choice of SVM:** At the heart of this authentication mechanism is the SVM, a powerful machine learning tool selected for its inherent ability to delineate complex patterns within data sets. SVM's proficiency in handling high-dimensional spaces prooved particularly suited for this voice authentication task, where the differentiation between genuine and cloned voices as well as inbetween speakers may often lie in intricate, nuanced patterns.

2. **Adopting MFCCs for Voice Feature Extraction:** The decision to harness MFCCs as the voice feature extraction technique was pivotal. MFCCs have a storied history in voice and speech processing, distilling voices into representative coefficients that encapsulate their unique characteristics, which deemed very useful in this study. In this context, different configurations of MFCCs were evaluated to strike the right balance between capturing voice intricacies and computational efficiency.

3. **Kernel Insights:** Among the tested SVM kernels, the Polynomial kernel emerged as the most adept at leveraging the features extracted via MFCCs. Especially at an MFCC configuration of 128, its superior performance underscores the synergy between the choice of SVM, the specific kernel, and the feature extraction mechanism.

4. **Potential Vulnerabilities:** Notwithstanding the promising results, certain vulnerabilities were spotlighted. Notably, the Polynomial kernel hinted at potential overfitting at higher MFCC configurations. Such vulnerabilities reiterate the necessity of continuous refinement and vigilance, ensuring that the system remains resilient against evolving cyber threats.

In summary, the choice of a SVM, complemented by the adoption of MFCCs, has set the stage for a robust voice authentication mechanism. While the results obtained are encouraging, they also underscore the relentless pursuit of optimization and adaptability in the face of potential cyber adversaries.

5.2 Future Work

As this research stands as a foundational exploration into voice authentication for smart home assistants using SVMs and MFCC, there are several potential avenues for continued investigation and refinement. First, the system's current configuration is optimized primarily for non-whispering voices. Given the versatility and practical usage of devices like Amazon's Alexa, which can be controlled even with a whisper, future iterations should incorporate models trained and tested on whispered voice commands to ensure comprehensive user authentication in diverse real-world scenarios.

Additionally, while this study primarily focused on optimizing the kernel, penalty parameter C, and the number of MFCCs due to computational cost rising, there's a broad spectrum of other SVM hyperparameters and preprocessing techniques that could further enhance the system's performance. Given adequate computational resources and advancements in optimization algorithms, these parameters should be methodically explored in subsequent research.

Moreover, with the rapid evolution of cyber threats and voice synthesis technologies, the system would benefit from continuous updates to its training data and potential integration of adversarial training methodologies.

Lastly, a comparative study involving other machine learning models or a hybrid ensemble approach can also be explored to ascertain the best-fit model for voice authentication in the dynamic landscape of smart home devices.

Bibliography

[AI23] Resemble AI. *Custom Voice Cloning with AI.* Accessed: 2023-08-01. 2023. URL: https://www.resemble.ai/.

[ang21] str anger. *larisa*. GitHub repository. 2021. URL: https://github.com/str-anger/larisa.

[AP17] E. Alepis and C. Patsakis. “Monkey says, monkey does: Security and privacy on voice assistants.” In: *IEEE Access* 5 (2017), pp. 17841–17851. DOI: 10.1109/ACCESS.2017.2747626.

[Ari+18] Sercan Arik et al. “Neural Voice Cloning with a Few Samples.” In: (Feb. 2018).

[BCV13] Yoshua Bengio, Aaron Courville, and Pascal Vincent. “Representation learning: A review and new perspectives.” In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828. arXiv: 1206.5538 [cs.LG].

[Ben+09] Najim Dehak Ben et al. “Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification.” In: *Interspeech*. 2009, pp. 1553–1556. DOI: 10.21437/Interspeech.2009-385.

[Bis06] Christopher M. Bishop. *Pattern Recognition and Machine Learning.* New York, NY: Springer, 2006.

[Bre01] Leo Breiman. “Random forests.” In: *Machine learning* 45.1 (2001), pp. 5–32. DOI: 10.1023/A:1010933404324.

[Bä+22] Tom Bäckström et al. *Introduction to Speech Processing.* 2nd ed. 2022. Chap. 3.8. DOI: 10.5281/zenodo.6821775. URL: https://speechprocessingbook.aalto.fi.

[CB19] X. Chen and T. Bonaci. “On the feasibility of voice cloning attacks: A preliminary analysis.” In: *2019 International Carnahan Conference on Security Technology (ICCST)*. IEEE. 2019, pp. 1–6.

[Cor23] Mozilla Corporation. *Common Voice by Mozilla.* https://commonvoice.mozilla.org/. Accessed: 2023-08-16. 2023.

[CSR06] William M Campbell, Douglas E Sturim, and Douglas A Reynolds. “Support vector machines using GMM supervectors for speaker verification.” In: *IEEE Signal Processing Society*. Vol. 1. IEEE. 2006, pp. I–I. DOI: 10.1109/LSP.2006.870086.

[CV95] Corinna Cortes and Vladimir Vapnik. “Support-vector networks.” In: *Machine learning* 20.3 (1995), pp. 273–297. DOI: https://doi.org/10.1023/A:1022627411411.

[CW18] N. Carlini and D. Wagner. “Audio adversarial examples: Targeted attacks on speech-to-text.” In: *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE. 2018, pp. 1–7. arXiv: 1801.01944 [cs.LG].

[dev21] Librosa developers. *librosa.feature.mfcc*. Last Accessed: 2023-08-24. 2021. URL: https://librosa.org/doc/main/generated/librosa.feature.mfcc.html.

[dev23] Scikit learn developers. *Support Vector Machines*. Last Accessed: 2023-08-25. 2023. URL: https://scikit-learn.org/stable/modules/svm.html.

[DM80] Steven B Davis and Paul Mermelstein. "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences." In: *IEEE transactions on acoustics, speech, and signal processing* 28.4 (1980), pp. 357–366. DOI: 10.1109/TASSP.1980.1163420.

[Faw06] T. Fawcett. "An introduction to ROC analysis." In: *Pattern recognition letters* 27.8 (2006), pp. 861–874. DOI: 10.1016/j.patrec.2005.10.010.

[GD+17] J. Gómez-Durán et al. "Speech recognition algorithm based on nonlinear techniques." In: 38 (Jan. 2017). URL: https://www.researchgate.net/publication/316253655_Speech_recognition_algorithm_based_on_nonlinear_techniques.

[GRC19] G. Goswami, N. Roy, and B. Chakraborty. "Defending voice biometrics against spoofing attack using non-linear feature extraction technique." In: *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*. IEEE. 2019, pp. 1–4.

[HB96] Andrew Hunt and Alan W Black. "Unit selection in a concatenative speech synthesis system using a large speech database." In: *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on* 1 (1996), pp. 373–376. URL: https://era.ed.ac.uk/bitstream/handle/1842/1082/Hunt%201996.pdf?sequence=1&isAllowed=y.

[HCL03] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. "A practical guide to support vector classification." In: *Department of Computer Science, National Taiwan University* (2003). URL: https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.

[Hsu+18] Wei-Ning Hsu et al. *Hierarchical Generative Modeling for Controllable Speech Synthesis*. 2018. arXiv: 1810.07217 [cs.CL].

[Jia+19] Ye Jia et al. *Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis*. 2019. arXiv: 1806.04558 [cs.CL].

[JS18] L. Johnson and R. Smith. "The Rise of the Internet of Things: Opportunities and Challenges." In: *IoT Journal* 5.2 (2018), pp. 23–40.

[JY11] Bin Jiang and Jun Yang. "Preferred frame length for the short-time magnitude spectrum on speech intelligibility and speech quality." In: *2011 8th International Conference on Information, Communications and Signal Processing* (2011), pp. 1–3. DOI: 10.1109/ICICS.2011.6174266.

[LMU18] Ludwig-Maximilians-Universität. *Spectral Analysis*. https://www.phonetik.uni-muenchen.de/~jmh/lehre/sem/ws1819/emuR/LESSON5/Spectral_analysis_old.html. 2018.

[Log00] Beth Logan. "Mel frequency cepstral coefficients for music modeling." In: *International Symposium on Music Information Retrieval*. 2000. URL: https://ismir2000.ismir.net/papers/logan_paper.pdf.

[MBE10] L. Muda, M. Begam, and I. Elamvazuthi. "Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques." In: *arXiv preprint* arXiv:1003.4083 (2010). arXiv: 1003.4083.

[McF+15] Brian McFee et al. "Librosa: Audio and music signal analysis in python." In: *Proceedings of the 14th python in science conference*. Vol. 8. 2015. URL: https://conference.scipy.org/proceedings/scipy2015/pdfs/brian_mcfee.pdf.

[MG19] F. Martinez and A. Gonzales. "Cloned Voices: The New Age Cyber Threat." In: *CyberSecurity Trends Journal* 6.3 (2019), pp. 47–54.

[Moo12] Brian CJ Moore. *An introduction to the psychology of hearing*. Brill, 2012.

[Moz23] Mozilla. *Common Voice*. Date: 4-24-2023. 2023. URL: https://commonvoice.mozilla.org/de/datasets.

[OS99] Alan V Oppenheim and Ronald W Schafer. *Discrete-Time Signal Processing*. 2nd ed. Prentice Hall, 1999. URL: https://research.iaun.ac.ir/pd/naghsh/pdfs/UploadFile_2230.pdf.

[Ped+11] Fabian Pedregosa et al. "Scikit-learn: Machine learning in Python." In: *Journal of machine learning research* 12.Oct (2011), pp. 2825–2830. arXiv: 1201.0490 [cs.LG].

[PJ21] D. Parker and R. Jain. "IoT Vulnerabilities and Mass Cyber Attacks: A Case Study." In: *Journal of Cyber Threat Intelligence* 4.1 (2021), pp. 15–29.

[Pow11] D. M. Powers. "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation." In: *Journal of Machine Learning Technologies* 2.1 (2011), pp. 37–63. arXiv: 2010.16061 [cs.LG].

[PS19] V. Patel and S. Sanyal. "Continuous voice authentication using deep learning." In: *Procedia computer science* 152 (2019), pp. 103–110.

[Ren+19] Yi Ren et al. "FastSpeech: Fast, robust and controllable text to speech." In: *Advances in Neural Information Processing Systems* 32 (2019). arXiv: 1905.09263 [cs.CL].

[Rey00] D. Reynolds. "Speaker identification and verification using Gaussian mixture speaker models." In: *Speech Communication* 17.1-2 (2000), pp. 91–108. URL: https://doi.org/10.1016/0167-6393(95)00009-D.

[RLP06] S. Raghavan, G. Lazarou, and J. Picone. "Speaker Verification using Support Vector Machines." In: *Proceedings of the IEEE SoutheastCon 2006*. 2006, pp. 188–191. DOI: 10.1109/second.2006.1629347.

[RMW07] Thomas D Rossing, F Richard Moore, and Paul A Wheeler. *The science of sound*. Addison-Wesley, 2007.

[RR95] Douglas A Reynolds and Richard C Rose. "Robust text-independent speaker identification using Gaussian mixture speaker models." In: *IEEE transactions on speech and audio processing* 3.1 (1995), pp. 72–83. DOI: 10.1109/89.365379.

[RS78] L. R. Rabiner and R. W. Schafer. *Digital Processing of Speech Signals*. Prentice-Hall, 1978. URL: https://books.google.de/books?id=YVtTAAAAMAAJ.

[She+18] Jonathan Shen et al. *Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions*. 2018. arXiv: 1712.05884 [cs.CL].

[Sny+18] D. Snyder et al. “X-vectors: Robust DNN embeddings for speaker recognition.” In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018, pp. 5329–5333. URL: https://www.danielpovey.com/files/2018_icassp_xvectors.pdf.

[SZ19] D. Smith and S. Zeadally. “Smart Home Security: Threats and Countermeasures.” In: *Journal of Cyber Security Technology* 3.1 (2019), pp. 17–30.

[VDO+16] Aäron Van Den Oord et al. “WaveNet: A generative model for raw audio.” In: *arXiv preprint arXiv:1609.03499* (2016). URL: https://www.researchgate.net/publication/308026508_WaveNet_A_Generative_Model_for_Raw_Audio.

[Wan+17a] Yuxuan Wang et al. “Tacotron: Towards End-to-End Speech Synthesis.” In: *Proceedings of the Interspeech*. Stockholm, Sweden, 2017, pp. 4006–4010. URL: https://arxiv.org/abs/1703.10135.

[Wan+17b] Yuxuan Wang et al. *Tacotron: Towards End-to-End Speech Synthesis*. 2017. arXiv: 1703.10135 [cs.CL].

[Wan+18] Yuxuan Wang et al. *Style Tokens: Unsupervised Style Modeling, Control and Transfer in End-to-End Speech Synthesis*. 2018. arXiv: 1803.09017 [cs.CL].

[WT17] H. Williams and J. Thompson. “Voice Biometrics: History and Future.” In: *Proceedings of the International Conference on Biometric Technologies*. 2017, pp. 112–119.

[Zha23] Anthony Zhang. *SpeechRecognition*. https://pypi.org/project/SpeechRecognition/. Version: 3.10.0. 2023.

[ZQZ20] Y. Zhang, Y. Qian, and Y. Zhang. “DeepFake Voice Detection.” In: *arXiv preprint* arXiv:2001.00001 (2020).

[ZZ17] S. Zhang and X. Zhu. “Voiceprint recognition technology and its application in security protection.” In: *2017 4th International Conference on Systems and Informatics (ICSAI)*. IEEE. 2017, pp. 1251–1255.

6 Eidesstattliche Erklärung

Hiermit versichere ich, dass diese Abschlussarbeit von mir persönlich verfasst ist und dass ich keinerlei fremde Hilfe in Anspruch genommen habe. Ebenso versichere ich, dass diese Arbeit oder Teile daraus weder von mir selbst noch von anderen als Leistungsnachweise an- dernorts eingereicht wurden. Wörtliche oder sinngemäße Übernahmen aus anderen Schriften und Veröffentlichungen in gedruckter oder elektronischer Form sind gekennzeichnet. Sämtliche Sekundärliteratur und sonstige Quellen sind nachgewiesen und in der Bibliographie auf- geführt. Das Gleiche gilt für graphische Darstellungen und Bilder sowie für alle Internet- Quellen. Ich bin ferner damit einverstanden, dass meine Arbeit zum Zwecke eines Plagiatsabgleichs in elektronischer Form anonymisiert versendet und gespeichert werden kann.

Datum

Nils Becker