ANALYSIS OF SPEECH CONTENT AND VOICE FOR DECEIT DETECTION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

By
Maria Raluca Eskin
September 2024

Analysis of Speech Content and Voice for Deceit Detection
By Maria Raluca Eskin
September 2024

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Hamdi Dibeklioğlu(Advisor)

Uğur Güdükbay

Yusuf Sahillioğlu

Approved for the Graduate School of Engineering and Science:

Orhan Arıkan
Director of the Graduate School

ABSTRACT

ANALYSIS OF SPEECH CONTENT AND VOICE FOR DECEIT DETECTION

Maria Raluca Eskin
M.S. in Computer Engineering
Advisor: Hamdi Dibeklioğlu
September 2024

Deceptive behavior is part of daily life, often without being recognized, leading to severe repercussions. With the recent improvements in machine learning, more reliable detection of deceit appears to be possible. Although current visual and multimodal models can identify deception with adequate precision, the individual use of speech content or voice still performs poorly. Therefore, we systematically analyze such essential communication forms focusing on feature extraction and optimization for deceit detection. To this end, we assess the reliability of employing transformers, spatial and temporal architectures, state-of-the-art pre-trained models, and handcrafted representations to detect deceit patterns. Furthermore, we conduct a thorough analysis to comprehend the distinct properties and discriminative power of the evaluated methods. The results demonstrate that speech content (transcribed text) provides more information than vocal characteristics. In addition, transformer architectures are found to be effective in representation learning and modeling, providing insights into optimal model configurations for deceit detection.

Keywords: automatic deceit detection, behavioral analysis, affective computing, natural language processing, voice processing, deep learning .

ÖZET

ALDATMA TESPİTİ İÇİN KONUŞMA İÇERİĞİ VE SES ANALİZİ

Maria Raluca Eskin
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Danışmanı: Hamdi Dibeklioğlu
Eylül 2024

Günlük yaşamın bir parçası olan aldatıcı davranışlar, genellikle fark edilmeden gerçekleşir ve ciddi sonuçlara yol açabilir. Makine öğrenimindeki son gelişmelerle birlikte, aldatmanın daha güvenilir bir şekilde tespit edilebilmesi mümkün görünmektedir. Mevcut görsel ve çok modlu modeller, aldatmayı yeterli doğrulukla tanımlayabilse de, konuşma içeriği veya sesin bireysel kullanımı hala düşük performans sergilemektedir. Bu nedenle, aldatmanın tespiti için özellik çıkarımı ve optimizasyonuna odaklanarak bu temel iletişim biçimlerinin kullanımını sistematik bir şekilde inceliyoruz. Bu amaçla, aldatma kalıplarını tespit etmek için transformatörlerden, uzamsal ve zamansal mimarilerden, en son teknoloji önceden eğitilmiş modellerden ve el yapımı gösterimlerden yararlanmanın güvenilirliğini değerlendiriyoruz. Ayrıca, değerlendirilen yöntemlerin kendine özgü özelliklerini ve ayrıştırıcı gücünü anlamak için kapsamlı bir inceleme yapıyoruz. Sonuçlar, konuşma içeriğinin (transkripte edilmiş metin) ses özelliklerinden daha fazla bilgi sağladığını göstermektedir. Ek olarak, transformatör mimarilerinin, gösterim öğrenimi ve modellemede etkili olduğu görülmekte ve aldatma tespiti için en uygun model yapılandırmaları hakkında içgörüler sağlanmaktadır.

Anahtar sözcükler: otomatik aldatma tespiti, davranış analizi, duygusal bilişim, doğal dil işleme, ses işleme, derin öğrenme.

Acknowledgement

I wish to extend my profound appreciation to my supervisor, Asst. Prof. Dr. Hamdi Dibeklioğlu, for his exceptional mentorship throughout my academic journey. His meticulous attention to detail and his exceptional ability to clarify complex academic issues, paired with his genuinely supportive and empathetic demeanor, have profoundly influenced my development. From the very start of my studies at Bilkent University, Asst. Prof. Dr. Hamdi Dibeklioğlu has been an outstanding mentor for me. He was the first professor I met and his encouragement was instrumental in motivating me to pursue a master's program at this prestigious institution. His steadfast commitment to high academic standards, remarkable research contributions, and dedication to nurturing student potential have been pivotal in shaping my academic and personal growth. I am deeply grateful for his invaluable guidance. My appreciation also extends to Prof. Dr. Uğur Güdükbay and Prof. Dr. Yusuf Sahillioğlu for their expert advice, constructive critiques, insightful feedback, and encouragement throughout my thesis presentation. Their contributions have enriched the development of my work.

I would like to express my gratitude to my pillar, best friend, and life partner, my husband Esat, who does not cease to inspire me with his intelligence, knowledge, kindness, and empathy, among countless other qualities. He has been by my side through both best and worst of times, offering invaluable support and motivation. His unwavering presence and encouragement have been crucial in empowering me to overcome the most challenging moments. I would like to thank him for being in my life and for consistently helping me to exceed my limits.

I would like to extend my heartfelt admiration to my parents, who have always encouraged me to pursue my dreams and supported all my decisions, even when it meant being apart from them. Their sacrifices for my well-being will eternally resonate within my heart. I will always be grateful to my family, my mother, Georgeta, with her kindness and unmatched positivity; my father, Daniel, with

his wisdom and humility; and my brother, Alex, with his strong character and intelligence, alongside their endless love, have profoundly shaped who I am today, continuously inspiring and motivating me throughout my journey.

I wish to acknowledge my grandparents, to whom I owe a deep sense of gratitude and respect, and to whom I am profoundly thankful to still have in my life. They are the cornerstone of our family and have instilled the most valuable virtues in our characters, which are truly priceless. I would also like to extend my appreciation to my parents-in-law, Ferruhe and Bülent, who embraced me as part of their family, treating me as their own child and doing everything in their power to support me.

I would like to extend my gratitude to Bilkent University for its role as a beacon of excellence in higher education and research, providing an exceptional environment for countless scholars to make a significant impact on the world.

I would like to thank the Turkish Academy of Sciences (TÜBA) for their partial support in our research through the GEBIP award program. Lastly, I want to express my gratitude to The Scientific and Technological Research Council of Türkiye (TÜBİTAK) for their financial support under grant number 122E134.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Deception refers to the intentional act of causing someone to hold false beliefs or misleading them through deliberate manipulation of information, communication, or behavior [1]. It can involve lying, omissions, or presenting information in a way that misrepresents the truth. Deceit appears in studies across multiple disciplines, including philosophy, psychology, and ethics [2] [3], where its definitions often hinge on intentionality and the resulting false belief in the deceived party. Lies are a direct form of deception involving the intentional conveyance of false information. Liars may use tactics such as "staying close to the truth" or offering unverifiable details to avoid detection. Acts of deception commonly involve lies of commission, where false information is provided, and lies of omission, where essential details are withheld. In high-stakes situations, omissions are sometimes judged more severely. Exaggerations, which amplify certain truths to mislead, are prevalent in advertising and social interactions. Misdirection diverts attention from the truth by focusing on irrelevant details. Research shows that liars often combine these tactics, with their effectiveness depending on the communication medium and the nature of the relationship [4].

In various contexts, for instance, personal relationships, workplaces, marketing, politics, legal settings, social media, financial transactions, and education, individuals often engage in deceptive practices to avoid conflict, gain advantage,

influence others, or manipulate outcomes [5]. A range of psychological and social factors shapes this complex behavior. Psychologically, it often stems from motivations such as self-protection, self-enhancement, or cognitive dissonance [6]. Socially, it can be driven by the pursuit of social gain, adherence to norms, power dynamics, or the need for group cohesion [7]. Understanding these factors helps explain why individuals might engage in deceptive behaviors across varied situations.

Participating in deception affects various groups: The deceiver may face psychological stress, guilt, and reputational damage, along with severe professional or legal consequences. The target can experience emotional harm, financial losses, and poor decision-making due to misleading information. Observers might lose trust and change their behavior based on perceptions of integrity. On a broader scale, widespread deception can undermine institutional trust, alter cultural norms, and disrupt social cohesion, affecting personal well-being, professional outcomes, and societal dynamics [8].

Accurate deception detection is crucial for improving legal processes, enhancing security, and building trust in various sectors. In the legal system, it ensures fairer trials, reduces wrongful convictions, and aids investigations. In security, it helps detect threats at borders, in corporations, and counter-terrorism operations. Trust is strengthened in business, government, and personal relationships through reliable deception detection. Advancing technologies, such as Artificial Intelligence and neuroimaging offer benefits like high accuracy, reduced bias, and wider applicability, ultimately serving justice, and protecting society while safeguarding human rights.

Ekman's studies [9] highlight the complexity of detecting deceit and the importance of considering multiple factors, including body language, facial expressions, voice tone, and the consistency or directness of someone's speech, rather than relying on any single indicator. Detecting deceit is inherently challenging due to the complex interaction of several factors. Differences in individual deceptive behaviors, the subtlety and ambiguity of physiological and behavioral signals, cognitive biases, and the limitations of available data, all complicate the accurate

identification of deceit. Although technological advancements and research are improving the accuracy of deceit detection, a significant number of challenges remain. A successful approach to detecting deceit often requires considering various types of cues, for example, visual or electrocardiogram signals, however, in real-life scenarios, they may not always be available. The limited research on voice and speech modalities constitutes another problem for deceit detection, where these data types continue to produce unsatisfactory results.

The primary contributions of this thesis are outlined as follows:

- We systematically analyze the essential communication modalities of voice and speech content for deceit detection.
- We create a stratified 10-fold cross-validation configuration for the Real-Life Trial dataset, considering both its class and gender distributions.
- We develop handcrafted voice representations and employ Wav2Vec, Word2Vec, MPNet and RoBERTa pre-trained models to model deceit patterns.
- We evaluate the performance of Transformer, Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN) and Feed-Forward Networks (NN) for representation optimization.
- We implement the best-performing speech content architecture (TextCNN) [10] to test the validity of the current literature standards for deceit detection.
- We perform extensive experimental analyses on the proposed methods and discuss our results based on the analyzed modality, representation, and modeling type, as well as the dataset's characteristics.
- We assess our findings against state-of-the-art results.

Chapter 2

Related Work

The origins of automatic lie detection can be traced back to the development of the polygraph, over a century ago, which sought to discern deceptive behavior through the measurement of physiological responses, such as heart rate, blood pressure, and respiration, that were hypothesized to correlate with deceit. However, the efficacy of the polygraph was significantly influenced by the proficiency of the examiner and its integration with other evaluative techniques [11]. Moreover, detecting lies with polygraphs is no better than random chance according to a meta-analysis of 253 studies [12].

Ekman and his colleagues, renowned for their work on bodily deception cues, particularly those involving facial expressions, made a pivotal impact on the field of deceit detection with their development of the Facial Action Coding System (FACS) in 1978 [13]. This comprehensive system provides an anatomically grounded framework for characterizing all visually discernible facial movements. It dissects facial expressions into their constituent components, known as Action Units (AUs), which correspond to specific muscle movements. FACS relies on the uniformity of facial muscle anatomy across humans, using AUs to systematically describe and reconstruct facial expressions based on these muscle movements. Building on Ekman's foundational work, a substantial body of research has employed the Facial Action Coding System (FACS) to systematically analyze and

utilize facial expressions as indicators for detecting deception [14] [15] [16].

Numerous researchers who predominantly concentrated their work on non-verbal indicators of deceit also recognized the significance of speech and vocal patterns in detecting deception [9] [17]. Thereby, influencing and shaping subsequent research efforts within the field of deception detection.

The study by Newman et al. [18] was among the first to explore linguistic styles associated with deceptive behavior through computer-based text analysis. Their research identified key linguistic markers of deceitful statements, such as distinct pronoun usage patterns and a higher occurrence of negative emotional language. Another pioneering study that focused on using Linguistics Based Cues (LBC) for detecting deception was conducted by Zhou et al. [19]. This research evaluated the effectiveness of various linguistic constructs, including quantity, diversity, complexity, specificity, expressivity, informality, affect, uncertainty, and non-immediacy, within the context of text-based asynchronous computer-mediated communication (TA-CMC). While Hancook et al. [20] focused their work on the alterations in both liars' and their conversational partners' linguistic patterns during truthful and deceptive interactions in synchronous text-based communication. Their work significantly advanced the understanding of how deception is manifested in written communication.

The Linguistic Inquiry and Word Count (LIWC) [21] tool proposed by Pennebaker et al. underscores the critical role of language analysis in identifying deceit. This technique highlights how linguistic patterns and specific word choices can serve as valuable indicators of deceptive practices, thereby expanding the scope of traditional methods focused solely on physiological or behavioral cues. By encompassing a broad spectrum of social, cognitive, and affective processes, LIWC offers critical insights into underlying psychological states and communication patterns, thereby serving as a significant tool for research in deception detection [22] [23] [24]. The text analysis software compares each word in the text against a comprehensive dictionary, which contains lists of words associated with predefined linguistic, psychological, and topical categories, and then calculates the percentage of words from the text that match each category.

The detection of deceit in speech was initially approached through the analysis of basic acoustic and prosodic features, which were also employed to assess emotional states or improve speech recognition systems. [25] [26] [27] [28]. These types of features—including speech rate, defined as the number of spoken words or syllables per unit of time, and relevant in both spoken and text communication; pitch, which represents the perceived frequency and characteristics of the voice; intensity, referring to the amplitude or loudness of the speech signal; formant frequencies; zero-crossing rate; cepstral coefficients; intonation; and rhythm of speech—encompass various dimensions of the time-frequency domain of communication. The Voice Stress Analysis (VSA) [29] proposed by Horvath et al., examined the impact of stress and emotional states on vocal characteristics, aiming to isolate sub-audible micro-tremors within the vocal spectrum. Nonetheless, research revealed that vocal distortions induced by stress or emotion could significantly impair the accuracy of speech recognition systems. Historically, only a limited number of researchers have succeeded in identifying deception based on speech characteristics.

In contemporary research, this foundational work has evolved into sophisticated methodologies, with cutting-edge deep learning models now being employed to enhance the accuracy and robustness of deceit detection, leveraging advanced data-driven approaches that surpass the limitations of earlier techniques. Numerous researchers have adopted the paradigm outlined by Zhou et al., which delineates the task of deceit classification into two fundamental components: the efficacy of capturing deceptive content and the performance of the classification model in identifying instances of deception [30]. Long Short-Term Memory (LSTM) constitutes one of the most extensively utilized modern architectures for the analysis of temporal data, with notable applications in the field of deceit detection. Randhavane at al. proposed an LSTM-based deep neural network to detect deceptive walking behavior in videos by extracting deep representations from nonverbal cues, such as gaits and gestures [31]. Alternatively, the research conducted by Avola et al. employs a high-dimensional descriptor known as Fisher Vectors (FVs) to extract features from hand skeletons in RGB sequences, in conjunction with the LSTM architecture, enabling real-time detection of deceitful

behavior [32]. In their study of audio-visual deceit detection using eye-tracking signals and log-mel spectrograms as frame-level features, Gallardo et al. integrate Long Short-Term Memory (LSTM) networks with attention mechanisms to develop an automated system for detecting deception. [33]. Besides log-mel spectrograms, many studies focus on combining different types of audio features, such as Mel Frequency Cepstral Coefficients (MFCCs), Chroma and Tonnetz, especially for the emotion recognition task. [34] [35]

Alongside handcrafted algorithms for descriptor extraction, a prevalent approach in current research involves using pre-trained shallow architectures to derive initial embeddings. For instance, neural networks are utilized to generate Word2Vec speech embeddings [36], while Wav2Vec voice embeddings [37] and MCNN speech embeddings [38] employ convolutional neural network (CNN). These embeddings are subsequently integrated with deeper neural networks for further processing which facilitates the acquisition of the necessary informational content for effective deceit detection. Exploring the non-verbal deceit patterns of video data, Venkatesh et al. employ both Deep Recurrent Convolutional Neural Networks and transfer learning [39]. They combine a pre-trained GoogleNet CNN with a Bi-directional LSTM module to capture the spatio-temporal speech information. The rise of transformers [40] which revolutionized the Deep Learning world by changing the way sequenced data, particularly concerning speech content, is seen had a big impact on deceit detection tasks. The Bidirectional Encoder Representations from Transformers (BERT) [41] architecture serves as a fundamental cornerstone for transformer-based language representation models and has subsequently been refined through various enhancements for applications in the domain of deception. One of its advanced variants, FakeBERT, introduced by Kaliyar et al., employs a Convolutional Neural Network (CNN) model following the BERT embedding layer for fake news detection [42]. The study BERTective [43] by Fornaciari et al. illustrates how augmenting the BERT model with attention layers in a hierarchical transformer framework can improve its performance of text-based deceit detection. Furthermore, their research investigates the effectiveness of Feed-Forward Networks (NN) and Convolutional Neural Networks (CNNs) for deceit detection through an extensive hyperparameter optimization

process. In their examination of transformer-based pre-trained models for classifying fraudulent reviews, Gupta et al. evaluate the efficacy of transfer learning using models such as BERT, RoBERTa, ALBERT, and DistilBERT. Their study indicates that RoBERTa [44] demonstrates superior performance when compared to the other assessed models [45]. Finally, Reimers and Gurevych [46] introduced a modified Sentence-BERT model that incorporates siamese and triplet architectures to produce semantically meaningful sentence embeddings which drastically reduces the time required for finding similar sentence pairs, while preserving BERT's accuracy.

Deceit manifests in various aspects of our lives, often without our awareness. Consequently, the development of an effective lie detection system requires a reliable dataset. Current deception-related datasets include EEG-based datasets like EEG-P300 [47], text-based datasets such as Open Domain [48], audio-based datasets like CSC [49] and ReLiDDB [50], as well as multimodal datasets including Real-Life Trials (RLT - video, text, audio) [51] or Bag-of-Lies (video, audio, EEG, gaze) [52]. A crucial feature of a dataset for deceit detection is its ability to simulate realistic deceitful scenarios. In addition, the dataset's size and the number of distinct subjects it evaluates are important factors to consider. As a result, much of the prominent research in the field of deceit detection utilizes the RLT dataset, as it effectively represents real-life scenarios due to the high-stakes implications of trial outcomes and its inclusion of a substantial number of distinct subjects relative to its moderate size. While Krishnamurthy et al. [10] primarily emphasize modality fusion for deceit detection on RLT datset, their work also assesses the individual performance of various feature types, including video, audio, text, and micro-expressions. Their proposed methodology uses pre-trained models, namely Word2Vec [53] and openSMILE [54], as well as Convolutional Neural Networks (CNN) and Feed-Forward Networks (NN) architectures. It incorporates a late fusion mechanism involving a concatenation layer followed by a single fully connected layer. The simplicity of the approach, combined with the results, highlights the essential role of representation learning as an independent factor in attaining optimal performance. Sen et al. [55] explore the complementary modalities of the RLT dataset by extracting representations

for each modality using Facial Action Units (FACS) for visual data, Pitch, Silence, and Speech Histograms for acoustic data, and unigrams, as well as LIWC applied to unigrams, for linguistic data. They employ classifiers such as Support Vector Machines (SVM), Random Forest (RF), and Neural Networks (NN), and test modality fusion through both late and early fusion techniques. Their highest performance is achieved with a late fusion of visual and audio features only, whereas incorporating speech reduces accuracy. Given that facial displays yielded the highest accuracy in unimodal experiments, this suggests that the late combination of voice and speech features, which may contain limited information relevant to deceit detection, negatively affected the fusion outcome. Recent studies on multimodal fusion validate earlier findings by highlighting the greater impact of visual data over text and audio in the RLT dataset. Chebbi et al [56] investigate feature-level and decision-level fusion approaches, employing comprehensive feature extraction and selection techniques to derive 21 features for text, 39 for video, and 72 for audio. Both their unimodal and multimodal evaluations reveal that video features offer more informative content than the other modalities.

Inspired by these findings, our analysis concentrates on unimodal models, specifically investigating the potential of the most dominant communication modalities: voice and speech content. Our approach aligns with the work of Sen et al. [55], emphasizing meticulous feature extraction, and processing using a handcrafted set of voice representations, namely MFCCs, chroma, mel spectrograms, and tonnetz. These spectral representations are employed to effectively capture the emotional content of the voice signal. Additionally, we implement the TextCNN model proposed by Krishnamurthy et al. [10]. To leverage the advantages of transfer learning, we employ pre-trained models, namely, the Wav2Vec [37], Word2Vec [36], Robustly Optimized BERT (RoBERTa) [44] and Sentance-BERT (MPNet) models [46] for representation learning. Furthermore, we implement models of varying complexity, beginning with a simple Feed-Forward Network (NN), followed by the widely used Long Short-Term Memory (LSTM) model and the state-of-the-art Transformer architecture for capturing

significant deceit information allowing us to examine and compare the effectiveness of different levels of architectural complexity in both feature retrieval and refinement.

Chapter 3

Methodology

In our analysis, we explore the potential of using voice and speech content data and the effectiveness of various architectures for deceit detection. We begin by capturing the emotional content of the vocal signal through handcrafting a set of spectral and harmonic voice representations comprising MFCCs, chroma, mel spectrograms, and tonnetz descriptors. To leverage the power of transfer learning in our analysis, we employ the pre-trained Wav2Vec model [37] to create initial voice representations, and Word2Vec [36], RoBERTa [44], and Sentence-BERT (MPNet) [46] pre-trained models for speech content representation. To further optimize the acquired information, we design and analyze different architectures, namely the Feed-Forward Network (NN), Long Short-Term Memory (LSTM), and the state-of-the-art Transformer architecture.

In the subsequent sections, we provide a detailed overview of the various input representations and modeling strategies utilized for analyzing voice and speech content within the context of deceit detection as a classification problem. This discussion includes an examination of the different types of data representations employed, such as handcrafted features, and the modeling approaches applied, ranging from traditional machine learning techniques to advanced deep learning architectures. Our aim is to assess how these methodologies contribute to the effective detection of deceitful behavior based on vocal and speech characteristics.

3.1 Analysis of Voice

3.1.1 Input Representations

For voice analysis, we extract the data of the RLT dataset from its video clips using a sampling rate of 44.1 kHz to ensure high-quality vocal recordings.

We utilize a vocal separation algorithm from the Librosa library to reduce background noise in our data. This algorithm applies cosine similarity to compare frames and aggregates similar frames by calculating their median value on a per-frequency basis. To mitigate bias from local continuity, the algorithm enforces a constraint that requires similar frames to be separated by at least 2 seconds. All preprocessing steps applied to the RLT dataset are detailed in Section 4.2.

3.1.1.1 Handcrafted Descriptors

To capture significant vocal information, we employ a comprehensive set of descriptors, including spectral features such as Mel-Frequency Cepstral Coefficients (MFCCs) and Mel Spectrograms, as well as harmonic features like Chroma and Tonnetz [57].

Mel-Frequency Cepstral Coefficients (MFCCs) are a set of features widely used in speech and voice processing. They provide a compact representation of the power spectrum of the signal, capturing the important characteristics of the sound [58].

A Mel spectrogram is a visual representation of the frequency content of a vocal signal over time, using the Mel scale to emphasize frequencies in a way that approximates human auditory perception. It is commonly used in speech-processing tasks to capture the temporal and spectral features of the signal [59].

Chroma features capture the harmonic and melodic content of vocal signals by representing the twelve pitch classes of the Western musical scale. They are

particularly useful for analyzing and comparing musical content, as they focus on the pitch classes rather than specific frequencies [60].

Tonnetz maps musical pitches onto a six-dimensional space, allowing for the measurement of distances between pitches and chords in a musically meaningful way. This representation is particularly valuable for chord analysis [61].

After extracting these voice features (MFCCs, Mel spectrogram, Chroma features, and Tonnetz), we concatenate them into fixed-size, 274-dimensional, voice representations with a maximum length of 82 for the RLT dataset. This consolidated feature vector is then processed using the network models detailed in Section 3.1.2.

3.1.1.2 Wav2Vec

We also utilize a pre-trained version of Wav2Vec architecture, called Wav2Vec2.0 [62] that combines the benefits of CNN and Transformer architectures to extract contextualized representations from vocal data. The model is trained on 960 hours of LibriSpeech data, sampled at 16 kHz. Consequently, we resample our vocal data to a 16 kHz rate before feature extraction. The Wav2Vec2.0 model generates a set of 768-dimensional representations that vary in number according to the length of the voice sample, with a maximum of 4071.

3.1.2 Modeling

To enhance the input voice representations, we employ two models, LSTM and Transformer, which we use to optimize the initial Handcrafted and Wav2Vec vocal representations. At the end of each optimization model, we use a classification module. Figure 3.1 shows the final architecture containing the input vocal representations, handcrafted or Wav2Vec, the optimization module, and the classification module.

3.1.2.1 LSTM Network

To address the variable length of the input representations, we employ a recurrent architecture, namely the Long Short-Term Memory (LSTM) model, which is renowned for its reliability in capturing temporal information from varying sequence data.

Due to its recurrent nature, the LSTM unit [63] operates for each timestep t. Given an input sequence $\mathbf{x}_t$ at time t, an LSTM unit performs the following transformations:

$$f_t = \sigma(W_f \mathbf{x}_t + U_f \mathbf{h}_{t-1} + b_f) \tag{3.1}$$

$$i_t = \sigma(W_i \mathbf{x}_t + U_i \mathbf{h}_{t-1} + b_i) \tag{3.2}$$

$$o_t = \sigma(W_o \mathbf{x}_t + U_o \mathbf{h}_{t-1} + b_o) \tag{3.3}$$

$$g_t = \tanh(W_g \mathbf{x}_t + U_g \mathbf{h}_{t-1} + b_g) \tag{3.4}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{3.5}$$

$$h_t = o_t \odot \tanh(c_t) \tag{3.6}$$

Where f_t, i_t, and o_t represent the forget, input, and output gates respectively, controlling the flow of information in the LSTM. g_t is the candidate value to be added to the memory cell c_t, while h_t is the LSTM's hidden state at time t. W and U represent the weight matrices, and b represents the bias terms for the respective gates. The symbol $\odot$ denotes element-wise multiplication, σ represents the sigmoid activation function, and tanh stands for the hyperbolic tangent activation. In words, the forget gate decides which information from the previous cell state c_{t-1} to discard, the input gate determines new information to be added to the cell state, and the output gate controls the information to be exposed in the hidden state h_t. The candidate value g_t is computed based on the current input and the previous hidden state, which, after gating, contributes to the updated cell state c_t.

In our model implementation, we utilize a single LSTM layer. This layer is followed by a classification module, which consists of one normalization layer and one fully connected layer.

3.1.2.2 Transformer Network

The Transformer architecture represents another employed model in our research for deceit detection applied to the analysis of input voice representations, for which we follow the implementation of Vaswani et al. [40]. We consider the input voice representations as embeddings and we use masking to tackle the loss of information that can be caused by padding, as the Transformer Encoder requires a fixed-sized input. Figure 3.1 shows the diagram of our Transformer architecture.

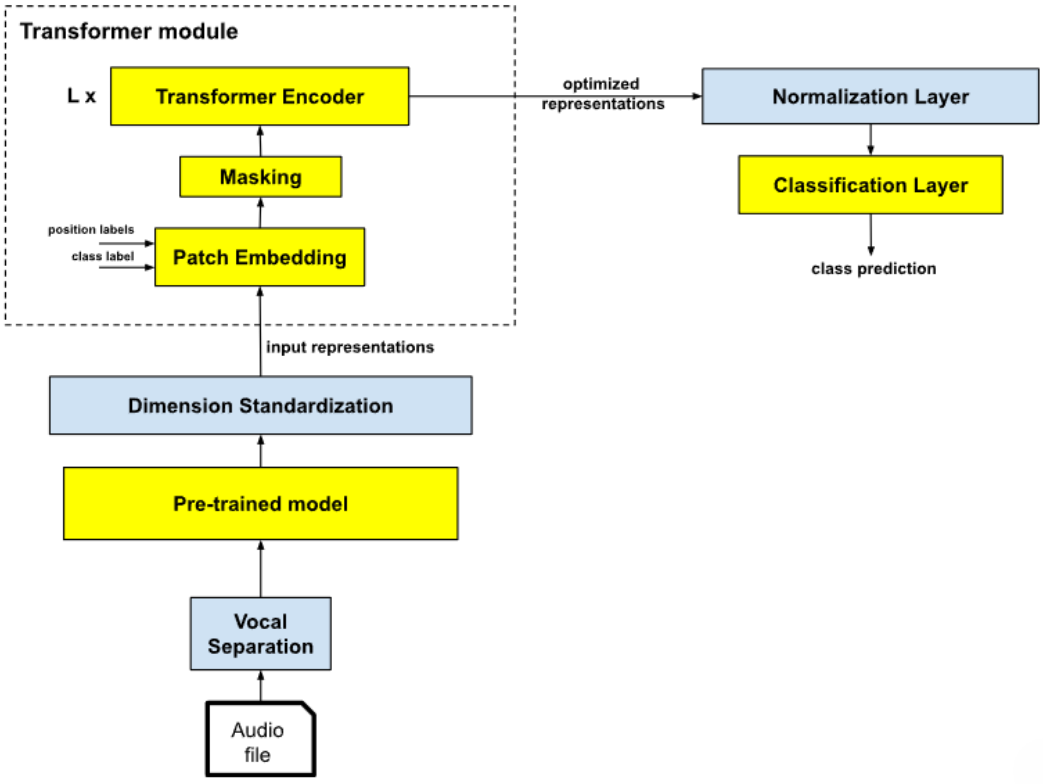

Figure 3.1: Transformer Architecture

Before adding the position labels and class label to the embeddings, we zero-pad them in the Dimension Standardization module. Then, we create a mask

M for the padded regions of the embeddings which we use in the Transformer Encoder to discard the information from these zero-padded regions.

The Transformer architecture contains a number of L cascaded encoders. We set $L = 1$ to reduce the complexity of the model as our input representations are modality-specific descriptors or preprocessed with a pre-trained model.

We begin by passing the input voice representations, X_l, where l represents the transformer encoder number, to a Normalization Layer followed by the Multi-Headed Self-Attention module. In this module, we divide the input X_l into H concatenated heads and apply the self-attention mechanism. This mechanism is a Scaled-Dot Product Attention variation, where the query, key, and value matrices are obtained through linear projections of the same input. For each head, the query ($Q_{h,l}$), key ($K_{h,l}$), and value ($V_{h,l}$) matrices are computed using distinct linear projections of X_l, described by the following equations:

$$Q_{h,l} = X_l W_{\text{query},h,l} \tag{3.7}$$

$$K_{h,l} = X_{m,l} W_{\text{key},h,l} \tag{3.8}$$

$$V_{h,l} = X_{m,l} W_{\text{value},h,l} \tag{3.9}$$

where $W_{\text{query},h,l} \in \mathbb{R}^{D \times D_h}$, $W_{\text{key},h,l} \in \mathbb{R}^{D \times D_h}$, and $W_{\text{value},h,l} \in \mathbb{R}^{D \times D_h}$ are the weight matrices in of the Multi-Headed Self-Attention module and D_h is the same for all heads and is calculated as the dimension of the input$D$ divided by the number of heads, $D_h = \frac{D}{H}$.

To discard the contribution of the padding to the context information found in the attention scores, we set the value of the padded regions in the attention mask M to $-\infty$. Moreover, we compute the attention scores for each head h using the mask of the padded regions in the embeddings M as shown in the below equation:

$$A_{h,l} = \text{softmax}\left(\frac{Q_{h,l} K_{h,l}^T}{\sqrt{D_h}}\right) M \tag{3.10}$$

The resulting attention scores reflect the contribution of each embedding in the voice input representations to the overall context information.

For the next step, we compute the result of the Scaled-Dot Product Attention for the h-th head by multiplying the attention scores with the value matrix, as shown below:

$$\text{head}_{h,l} = A_{h,l}V_{h,l} \tag{3.11}$$

We concatenate the obtained results for each head and project them through a fully-connected layer to obtain the final results O_l of the Masked Multi-Head Self-Attention module:

$$O_l = [\text{head}_{1,l}, \ldots, \text{head}_{H,l}]W_{\text{output},l} \tag{3.12}$$

where $W_{\text{output},l} \in \mathbb{R}^{D\times D}$ is the weight matrix.

Next, we pass this output matrix through a residual connection followed by a Layer Normalization layer:

$$X'_l = \text{LN}(O_l + X_l) \tag{3.13}$$

where LN stands for Layer Normalization.

The final component of the Transformer Encoder is a Feed-Forward Network, FFNl, which consists of two fully connected layers with a Gaussian Error Linear Unit (GELU) activation function in between. This is combined with a residual connection of the normalized output from the Masked Multi-Head Self-Attention module. The output of the l-th Transformer encoder then serves as the input for the next Transformer layer, denoted as X_{l+1}:

$$X_{l+1} = \text{LN}(\text{FFN}_l(X'_l) + X'_l) \tag{3.14}$$

The first fully-connected layer of the FFN projects the hidden representations from D-dimensional space to $4D$-dimensional space, and the second one projects back to D-dimensional space.

All the hyperparameters used for the Transformer model are detailed in Section 4. In the end, we use the same classification module as the one used with the LSTM model.

3.2 Analysis of Transcribed Speech

3.2.1 Input Representations

The RLT dataset includes transcripts of the raw speech content data. As a preliminary step in processing these transcripts, we manually correct them to address any missing phrases and correct typographical errors. Furthermore, we remove punctuation, stop words (e.g., "a," "the," "is," "are"), filler words (e.g., "um," "uh," "okay," "like"), and all problematic samples. All preprocessing steps applied to the RLT dataset are detailed in Section 4.2.

3.2.1.1 Word2Vec

For speech data, we use the Word2Vec [36] model, which is, like the Wav2Vec, a shallow feed-forward network that employs the Continuous Bag of Words (CBOW) architecture to predict a target word based on its surrounding context. The pre-trained version of this model has been trained on a subset of the Google News dataset, which contains approximately 100 billion words, producing 300-dimensional vectors for over 3 million words and phrases. For the RLT dataset, we obtain a maximum of 182 speech content representations, 300-dimensional vectors, for a single sample.

Through Word2Vec's prediction of words from their neighboring terms, it effectively captures both semantic relationships and word similarities. The dense vector representations generated by this pre-trained model are instrumental in a wide range of natural language processing tasks, as they encode important syntactic and semantic word features.

3.2.1.2 MPNet

MPNet (Sentence-BERT), an enhancement of BERT specifically designed for sentence embeddings [46] utilizes Siamese and triplet network architectures to generate high-dimensional (768-dimensional) embeddings, which vary depending on the length of the input sample. We obtain a maximum of 18 MPNet speech content representations for the RLT dataset. The objective of MPNet is to train sentence embedding models using extensive datasets of sentence pairs, applying a self-supervised contrastive learning approach.

MPNet generates rich, meaningful sentence embeddings efficiently, reducing computational resources needed for similarity tasks while preserving the accuracy of BERT's embeddings. This makes it highly effective for tasks requiring detailed sentence-level semantic understanding. Therefore, we employed the pretrained MPNet base model, which was fine-tuned on a concatenation from multiple datasets consisting of a total of 1 billion sentence pairs.

3.2.1.3 RoBERTa

We leverage RoBERTa, a variant of the BERT architecture introduced by Liu et al. [44], which enhances BERT's performance by modifying its training process. RoBERTa is pretrained on a vast English corpus using a self-supervised method with the Masked Language modeling (MLM) objective. Unlike traditional recurrent neural networks (RNNs) or autoregressive models such as GPT, RoBERTa learns bidirectional representations by predicting masked words within sentences. This model employs larger batch sizes, extended training durations, and more data, resulting in improved language representation and contextual understanding. The output of RoBERTa includes feature vectors of 768 dimensions, consistent with those produced by MPNet and Wav2Vec2.0, as all three models leverage Transformer-based architectures, which reach a maximum length of 738 for the RLT dataset.

3.2.2 Modeling

To enhance the input speech content representations, we employ 4 models, Feed-Forward Network, CNN, and LSTM and Transformer, which we use to optimize the input Word2Vec, MPNet, and RoBERTa speech content representations. At the end of each optimization model, we use a classification module. Figure 3.2 shows the final architecture containing the input speech content representations, Word2Vec, MPNet, or RoBERTa, the optimization module, and the classification module.

3.2.2.1 Feed-Forward Network

To capture the spatial and temporal information from the input speech content representations we design a simple Feed-Forward Network. As the Feed-Forward Networks require a fixed-sized input, we zero-pad our initial speech content representations before refining them. The architecture for representation optimization contains 2 identical encoders, spatial and temporal. Each of the encoders uses two fully-connected layers (FC), two normalization layers, a two activation layers. As shown in Figure 3.2, we use the encoders to extract more meaningful speech information by consecutively reducing the dimension of the speech content representations and then, their number.

We adapt our Feed-Forward Architecture to each speech content representation size, by downsizing in each encoder with a fixed factor of 0.25 of the initial dimension of the input. To avoid increasing the number of parameters too much and to make the Feed-Forward Network adaptable, we also apply the downsizing factor in the classification module, only for the configurations using the Feed-Forward Network. The classification module is similar to the encoders, the main difference being the last activation and normalization layers we remove. For both the classification module and the Feed-Forward Network we use LeakyRELU as activation.

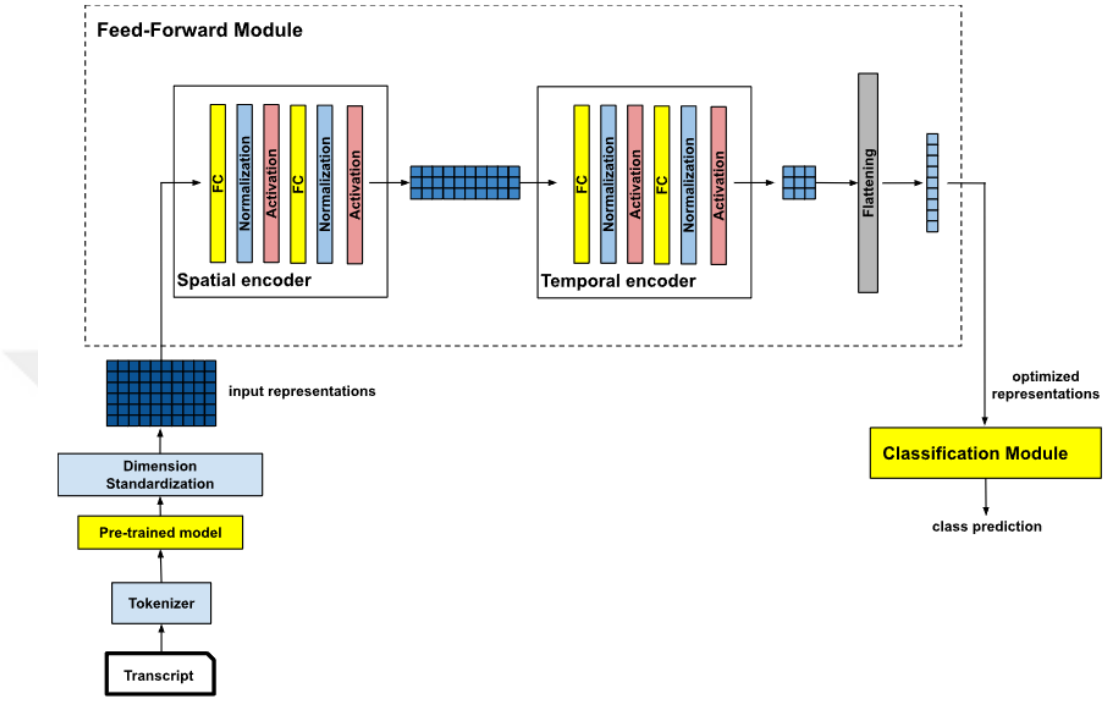

Figure 3.2: Feed-Forward Architecture

The classification module consists of two fully connected layers, a normalization layer, and a LeakyReLU activation function, as shown in Figure 3.3.

3.2.2.2 Convolutional Neural Network

Inspired by the TextCNN model introduced by Krishnamurthy et al. [10], we apply a similar approach to model speech content representation. We zero-pad our initial speech content representations before further processing them, as we did for the Feed-Forward Network. We employ the TextCNN architecture comprised of a single convolutional layer and a max-pooling layer to produce the sentence representation. We use the same architectural parameters as the original paper, a filter of size 3 with 20 feature maps, and apply a max-pooling window size of 2. Following this, we use a fully-connected layer with 300 neurons and a Rectified Linear Unit (ReLU) activation function. Consequently, we obtain a representation of dimension 300 for the input transcribed speech, which we feed

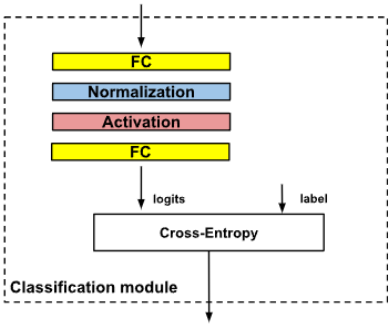

Figure 3.3: Classification module for the Feed-Forward and CNN architectures.

to our classification module, Figure 3.3, consisting of two fully connected layers, a normalization layer, and a LeakyReLU activation function.

3.2.2.3 LSTM and Transformer Networks

We utilize the LSTM and the Transformer Networks, as described in Section 3.1.2, for modeling speech content, similarly to how they are applied to vocal data. The primary distinction lies in the embedding size and the length of the transcribed speech, which we adjust to align with the characteristics of the speech data.

Chapter 4

Experimental Protocol

This chapter outlines the experimental protocol employed in our investigation of deceit detection through voice and speech analysis. The section begins with a description of the dataset utilized, specifically the Real-Life Trial (RLT) dataset [51], which comprises a diverse collection of high-stakes court trial videos. We then detail the preprocessing techniques applied to both voice and speech content data to ensure high-quality inputs for subsequent modeling. This includes the use of a vocal separation algorithm to mitigate background noise and manual corrections to the transcripts.

Following data preparation, we present the hyperparameter tuning process, emphasizing our method for optimizing model performance through a comprehensive grid search approach. The chapter concludes with an overview of the evaluation setup, including the use of cross-entropy loss as the objective function and the application of cross-validation to ensure robust and generalizable results. We detail our approach for addressing gender imbalance and evaluating model performance through Correct Classification Rate (CCR), setting the stage for a thorough examination of the experimental results in subsequent chapters.

4.1 Dataset

The Real-Life Trial (RLT) dataset [51] is a medium-sized, multimodal collection comprising 121 videos featuring 56 distinct subjects from significant high-stakes court trials in U.S. criminal justice history. These high-stakes trials involve cases where the defendants are at risk of severe consequences, such as extended imprisonment. The dataset includes corresponding transcripts for the videos, with ground truth labels determined by the trial verdicts.

In addition to the transcripts provided in the RLT dataset, we extract the voice data from the video clips at a sampling rate of 44.1 kHz to ensure high-quality vocal recordings. As illustrated in Figure 4.1, the lengths of vocal samples in the RLT dataset vary significantly. This variability in sample length presents a common challenge for machine learning models that require fixed-size inputs. We address this issue through our model designs described in Section 3 for both speech and voice data

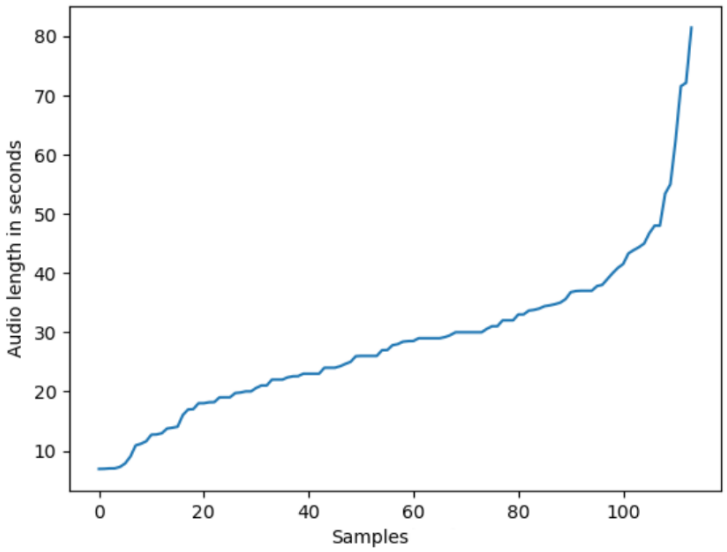

Figure 4.1: The length distribution of voice samples for RLT dataset.

4.2 Preprocessing

We employ a vocal separation algorithm available in the Librosa library for music and sound analysis to eliminate potential background noise in our voice data. This algorithm uses cosine similarity to compare frames and aggregates similar frames by computing their median value on a per-frequency basis. To minimize bias arising from local continuity, the algorithm enforces a constraint that requires similar frames to be spaced at least 2 seconds apart. This method effectively diminishes the influence of sporadic or non-repetitive deviations from the average spectrum, thereby facilitating the removal of unwanted vocal elements.

As an initial step in processing the RLT dataset transcripts, we manually correct the transcripts to handle missing phrases and rectify typographical errors. Additionally, we remove punctuation, stop words (such as "a," "the," "is," "are"), and filler words (such as "um," "uh," "okay," "like"). In our experiments, described in Section 5, we test the effect of tokenization, converting text to lowercase, removing whitespace, excluding numerical characters, eliminating punctuation, filtering out stop words, removing sparse and specific terms, stemming, and lemmatization.

Given that the vocal separation method requires a minimum length and that both voice and speech modalities perform better with longer samples, we exclude the samples with a duration of less than 5 seconds. Table 4.3 provides a detailed distribution of the remaining samples and highlights the gender imbalance present in the RLT dataset, which we addressed in our experiments.

Table 4.1: Gender distribution across classes in the RLT dataset.

	Female	**Male**	**Total**
Deceitful samples	47	10	57
Truthful samples	26	31	57
Total	73	41	114
Percentage	64.04%	35.96%	100%

4.3 Hyperparameters

We perform a comprehensive hyperparameter tuning using a grid search algorithm for all our models, detailed in Section 5. Although grid search can be computationally demanding when dealing with a broad range of values, we leverage existing literature to identify an effective range, optimizing the search process. The range of hyperparameter values considered is outlined in Table 4.2 together with the other optimization algorithms considered during our training process. To the remaining parameters, we assign fixed values, as detailed in Section 3.

Table 4.2: Evaluated hyperparameters and optimization algorithms.

Hyperparameter	**Values**
Embedding size	{1e^{-4}, 5e^{-5}, 1e^{-5}}
Number of Self-Attention Heads	{1e^{-4}, 5e^{-5}, 1e^{-5}}
CNN kernel size	{ 3, 5, 8}
Initial Learning Rate	{1e^{-4}, 5e^{-5}, 1e^{-5}}
Weight Decay	{5e^{-3}, 1e^{-4}, 5e^{-4}}
Optimization Algorithm	{SGD, AdamW}
Learning Rate Decay	{StepLR, LambdaLR}
Dropout probability	{0, 1e^{-2}, 1e^{-3}}
Number of Epochs	{50, 100}

In Section 5, we present the results for the most meaningful hyperparameter values. We chose the best-performing model based on the highest validation score achieved during training.

4.4 Evaluation Setup

We are addressing deceit detection as a classification task, therefore, we choose cross-entropy loss to optimize our model. In our training, we aim to minimize the cross-entropy loss given by Equation 4.1

$$Q_v^* = \arg\min_{Q_v} -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{2}\mathcal{Y}_{i,c}\log(p_{i,c,v}) \tag{4.1}$$

Let Q_v denote the parameters of the visual model. We use N to represent the number of training samples, Y to indicate whether the truth label c is correct for sample i, and $p_{i,c;v}$ to denote the predicted probability that observation i is assigned to the truth label c based on visual information.

Table 4.3: 10-fold configuration for the RLT dataset.

Dataset split (10 folds)	**Subjects' IDs (Genders)**	**Deceitful samples**	**Truthful samples**	**Female subjects**	**Male subjects**
Fold 1	1, 6, 7, 27, 41 (F, F, M, M, M)	7 (7F, 0M)	7 (3F, 4M)	2	3
Fold 2	2, 8, 22, 28, 39 (F, M, M, M, F)	8 (7F, 1M)	8 (6F, 2M)	2	3
Fold 3	3, 4, 23, 24, 42 (F, M, F, M, M)	18 (18F, 0M)	17 (5F, 12M)	2	3
Fold 4	5, 10, 11, 13, 14 (F, F, M, M, M)	4 (4F, 0M)	4 (1F, 3M)	2	3
Fold 5	9, 30, 40, 43, 44 (F, M, F, M, M)	3 (1F, 2M)	3 (1F, 2M)	2	3
Fold 6	12, 16, 17, 19, 45 (F, M, M, F, M)	5 (5F, 0M)	6 (3F, 3M)	2	3
Fold 7	15, 34, 35, 36, 46 (F, M, F, M, M)	3 (1F, 2M)	2 (1F, 1M)	2	3
Fold 8	18, 20, 25, 26, 29 (M, F, M, M, F)	4 (4F, 0M)	4 (1F, 3M)	2	3
Fold 9	21, 31, 32, 33, 47 (F, M, M, M, F)	3 (0F, 3M)	3 (3F, 0M)	2	3
Fold 10	37, 38, 48, 49, 50 (M, M, M, F, F)	2 (0F, 2M)	3 (2F, 1M)	2	3
Total	50 unique subjects	57	57	20	30

To evaluate the effectiveness of our methods in real-world scenarios by preventing information leakage caused by overlapping subjects' data across folds,

we utilize the cross-validation technique. We design a well-balanced 10-fold configuration, as presented in Table 4.3. To tackle the gender imbalance in the RLT dataset, shown in Table 4.1, our fold setup considers not only the number of unique subjects in each fold but also the gender of the subjects. All of our experiments are conducted using this setup.

To further mitigate any potential imbalances in our RLT dataset, we use the Correct Classification Rate (CCR) as our evaluation metric. The CCR reflects the proportion of correctly classified samples, providing an overall measure of accuracy and enabling straightforward comparison between different models.

Chapter 5

Experiments & Results

We present our experimental analysis of voice and speech data on the RLT dataset for deceit detection using all the architectures described in Section 3, we finetune the hyperparameters mentioned in Section 4 and discuss the outcomes of each experiment.

5.1 Analysis of Voice

To analyze the performance of vocal data for deceit detection, we employ the Long Short-Term Memory (LSTM) and Transformer architectures for the Handcrafted and Wav2Vec input representations, detailed in Section 3, following the experimental protocols from Section 4.

5.1.1 Handcrafted representations

As seen in Table 5.1, the LSTM model does not easily overfit on the training set. According to the validation result for the best-performing configuration, it doesn't generalize on the dataset despite our balanced cross-fold validation setup. The

observed outcome may be attributed to the sparsity of the RLT dataset, as the LSTM model is specifically designed to capture temporal dependencies. Given the relatively limited number of representations, as indicated in Table 5.17, this characteristic could influence the model's performance.

Table 5.1: Performance of the LSTM model on the RLT dataset using Handcrafted representations.

Learning Rate	**Embedding Size**	**Number Parameters**	**CCR**		
			Train	**Validation**	**Test**
1e-03	16	6706	0.6202	0.4678	0.4994
	32	17506	0.7132	0.4457	0.4116
	64	51394	0.8211	0.4859	0.4334
	128	**168322**	**0.9198**	**0.4354**	**0.5972**
	256	598786	0.986	0.5029	0.5138
	512	2246146	0.9984	0.4798	0.5098
1e-04	16	6706	0.5989	0.5478	0.4678
	32	17506	0.6462	0.4497	0.5006
	64	51394	0.6863	0.4865	0.3984
	128	168322	0.7856	0.5214	0.4483
	256	598786	0.8705	0.5135	0.4784
	512	2246146	0.9333	0.4662	0.4648
1e-05	16	6706	0.5281	0.5467	0.5331
	32	17506	0.5403	0.497	0.4544
	64	51394	0.5323	0.4591	0.3957
	128	168322	0.6053	0.397	0.4956
	256	598786	0.6904	0.4769	0.5271
	512	2246146	0.743	0.4435	0.4448

The Transformer model has more consistent results, as presented in Table 5.2, despite the slightly lower test accuracy for the best-performing hyperparameter configuration. However, the validation CCR values are still lower in some cases. This can be caused by the poor ability of the Handcrafted representations to capture enough information from the vocal input. Moreover, the results indicate that the Transformer model achieves the highest test CCR value with a lower embedding size of 64, whereas the LSTM model necessitates a larger embedding size of 128 to attain comparable performance, which directly affects the number of parameters.

Table 5.2: Performance of the Transformer model on the RLT dataset using Handcrafted representations.

Learning Rate	Embedding Size	Number Attention Heads	Number Parameters	CCR		
				Train	Validation	Test
1e-03	32	8	24258	0.9639	0.5392	0.4361
	64	8	73090	0.9949	0.553	0.4706
	64	16	73090	1	0.4525	0.5028
	128	16	244482	1	0.4573	0.4302
	128	32	244482	1	0.519	0.4923
	256	16	882178	1	0.5073	0.4289
1e-04	32	8	24258	0.6036	0.5512	0.4727
	64	8	73090	0.7013	0.5293	0.5513
	64	**16**	**73090**	**0.7146**	**0.5108**	**0.5807**
	128	16	244482	0.8547	0.4713	0.5759
	128	32	244482	0.8023	0.5661	0.4545
	256	32	882178	0.954	0.3786	0.5136
	256	16	882178	0.9308	0.6814	0.4768
1e-05	32	8	24258	0.4979	0.4458	0.5308
	64	8	73090	0.5339	0.545	0.5686
	64	16	73090	0.5174	0.5139	0.4982
	128	16	244482	0.5616	0.4343	0.5372
	128	32	244482	0.5109	0.3753	0.5074
	256	32	882178	0.6164	0.451	0.3968
	256	16	882178	0.611	0.4623	0.5507

5.1.2 Wav2Vec representations

Similar to the experiments for the Handcrafted representations, we employ the Transformer model for the Wav2Vec representations. The CCR results for this experiment presented in Table 5.3 show that combining two transformer-based architectures fails to detect deceit due to the increased complexity. The optimal hyperparameter configuration employs an embedding size of 256, significantly expanding the model's dimensions. This requirement for a larger embedding size is likely influenced by the high dimensionality and number of Wav2Vec voice representations, shown in Table 5.17.

We did not include the results for the LSTM model and Wav2Vec voice representations, as the model failed to learn, under any configuration, the training

Table 5.3: Performance of the Transformer model on the RLT dataset using Wav2Vec representations.

Learning Rate	Embedding Size	Number Attention Heads	Number Parameters	CCR		
				Train	Validation	Test
1e-03	32	8	167714	1	0.4653	0.491
	64	8	360002	1	0.4785	0.5167
	64	16	360002	1	0.6052	0.5018
	128	16	818306	1	0.4527	0.4812
	128	32	818306	1	0.4954	0.5111
	256	32	2029826	1	0.4464	0.4665
	256	16	2029826	1	0.4722	0.4766
1e-04	32	8	167714	0.7825	0.4202	0.5419
	64	8	360002	0.8772	0.5232	0.8772
	64	16	360002	0.8859	0.5127	0.5006
	128	16	818306	0.956	0.4592	0.4733
	128	32	818306	0.9331	0.3275	0.4069
	256	32	2029826	0.993	0.3868	0.5187
	256	**16**	**2029826**	**0.9929**	**0.5034**	**0.5385**
1e-05	32	8	167714	0.5369	0.484	0.4965
	64	8	360002	0.6072	0.5751	0.5115
	64	16	360002	0.5576	0.4932	0.5051
	128	16	818306	0.6371	0.5554	0.4523
	128	32	818306	0.6532	0.5282	0.4729
	256	32	2029826	0.692	0.4651	0.5085
	256	16	2029826	0.6966	0.4526	0.5293

data. This issue arises from the vanishing gradients in the LSTM model when processing very long data sequences, with Wav2Vec generating up to 4.071 voice representations per sample.

5.2 Analysis of Transcribed Speech

For the assessment of speech content data for deceit detection, we employ the Feed-Forward, CNN, LSTM, and Transformer architectures for the Word2Vec, MPNet, and RoBERTa representations, detailed in the previous sections.

5.2.1 Word2Vec representations

As shown in Table 5.4, the training CCR values indicate that the model is prone to overfitting. Given the low complexity of both Word2Vec and Feed-Forward models, it was anticipated that this configuration would yield suboptimal results. However, despite the model's simplicity, it performs better with a smaller embedding size, surpassing the performance of the Wav2Vec voice representations.

Table 5.4: Performance of the Feed-Forward model on the RLT dataset using Word2Vec representations.

Learning Rate	**Embedding Size**	**Number Parameters**	**CCR**		
			Train	**Validation**	**Test**
1e-03	16	8298	0.9279	0.4387	0.5064
	32	18386	1	0.4702	0.4999
	64	50082	1	0.5752	0.4312
	128	202562	1	0.4172	0.4634
	256	1207938	1	0.463	0.5149
	512	8772866	1	0.5642	0.5298
1e-04	16	8298	0.9787	0.4979	0.5344
	32	18386	0.9984	0.5049	0.5093
	64	50082	1	0.4597	0.3722
	128	202562	1	0.5459	0.5415
	256	1207938	1	0.5942	0.4282
	512	8772866	1	0.5245	0.5572
1e-05	16	8298	0.7848	0.4953	0.3752
	32	18386	0.9231	0.6177	0.5387
	64	**50082**	**0.9689**	**0.418**	**0.5675**
	128	202562	0.9889	0.4821	0.5097
	256	1207938	0.9968	0.4452	0.5092
	512	8772866	1	0.5194	0.5356

We also investigate the impact of preprocessing on speech content representations before feature extraction, as mentioned in Section 4.2, which includes: tokenization, conversion to lowercase, removal of whitespace, exclusion of numerical characters, elimination of punctuation, filtering out stop words (e.g., "the," "a," "on," "is," "all"), removal of sparse terms and specific words, stemming (e.g., "books" to "book," "looked" to "look"), and lemmatization, which employs lexical knowledge bases to determine the correct base forms of words. To evaluate the effect of these preprocessing techniques, we employ the optimal hyperparameter configuration for the Feed-forward model with Word2Vec representations. Our analysis indicated that preprocessing did not lead to substantial improvements in performance. The accuracy differences observed were approximately 1%, which can be attributed to the minor inherent variability in the learning process. This could be the result of the pre-trained feature extractors having already processed the input transcripts, thereby diminishing the impact of additional preprocessing.

Table 5.5: Performance of the CNN model on the RLT dataset using Word2Vec representations.

Learning Rate	**Embedding Size**	**CNN Kernel Size**	**Number Parameters**	**CCR**		
				Train	**Validation**	**Test**
	1024	**3**	**10182554**	**0.9571**	**0.5563**	**0.592**
1e-03	1024	5	10111474	1	0.5347	0.5406
	1024	8	9971254	1	0.571	0.5497
	1024	3	10182554	1	0.5485	0.5612
1e-04	1024	5	10111474	1	0.5399	0.5757
	1024	8	9971254	1	0.5641	0.5895
	1024	3	10182554	0.9984	0.5881	0.5216
1e-05	1024	5	10111474	1	0.5478	0.5399
	1024	8	9971254	1	0.5047	0.5327

To evaluate the effectiveness of a CNN model for deceit detection, we utilize the TextCNN architecture [55], as detailed in Section 3 and Section 4. Table 5.5 demonstrates that the TextCNN architecture exhibits the highest parameter count, exceeding that of the Transformer architecture with the largest embedding size, as TextCNN has a fixed embedding size of 1024. While the dimensionality and number of Word2Vec speech content representations are comparable to those of the Handcrafted voice representations. Furthermore, the CCR values for the TextCNN model do not fall below 0.5 across any hyperparameter

configurations, highlighting its superior performance among non-transformer architectures.

As illustrated in Table 5.6, the results remain inconsistent between the validation and test sets. The model's inability to overfit the training set may be attributed to the relatively simple nature of the Word2Vec speech content representations. Nonetheless, the LSTM model appears capable of extracting valuable information from these representations, particularly when utilizing a larger embedding size, achieving a maximum CCR value of 0.61 for the test dataset.

Table 5.6: Performance of the LSTM model on the RLT dataset Word2Vec using representations.

Learning Rate	**Embedding Size**	**Number Parameters**	**CCR**		
			Train	**Validation**	**Test**
1e-03	16	7122	0.608	0.4946	0.4524
	32	18338	0.6979	0.4124	0.484
	64	53058	0.8058	0.5052	0.5094
	128	171650	0.9084	0.4789	0.5048
	256	605442	0.9723	0.4542	0.5088
	512	2259458	0.9942	0.509	0.447
1e-04	16	7122	0.5939	0.4923	0.5286
	32	18338	0.6306	0.49	0.4016
	64	53058	0.6883	0.569	0.4841
	128	171650	0.7636	0.4497	0.5452
	256	605442	0.8721	0.5216	0.5423
	512	2259458	0.9399	0.4894	0.5862
1e-05	16	7122	0.5174	0.4716	0.4536
	32	18338	0.5166	0.4661	0.3899
	64	53058	0.5644	0.5345	0.4739
	128	171650	0.5487	0.4789	0.5229
	256	**605442**	**0.6346**	**0.5609**	**0.6186**
	512	2259458	0.7578	0.535	0.4774

The Transformer architecture yields the best results for Word2Vec speech content representations, as detailed in Table 5.7. It achieves a maximum CCR value of 0.65 for the test set, utilizing a relatively compact model with only 914434 parameters and an embedding size of 256, which is close to the dimensionality of the representations (300). Furthermore, this architecture demonstrates superior generalization, with validation results often outperforming test results in most cases. This suggests that integrating a shallow pre-trained model for extracting representations with a Transformer model for optimization may be the most effective approach. This combination leverages the strengths of both models, achieving superior results with a relatively compact architecture.

Table 5.7: Performance of the Transformer model on the RLT dataset using Word2Vec representations.

Learning Rate	**Embedding Size**	**Number Attention Heads**	**Number Parameters**	**CCR**		
				Train	**Validation**	**Test**
1e-03	32	8	28290	0.992	0.5490	0.5566
	64	8	81154	1	0.5542	0.5402
	64	16	81154	1	0.5337	0.5961
	128	16	260610	1	0.5495	0.5085
	128	32	260610	1	0.5776	0.6072
	256	**32**	**914434**	**1**	**0.603**4	**0.6562**
	256	16	914434	1	0.6500	0.6244
1e-04	32	8	28290	0.6767	0.4946	0.4427
	64	8	81154	0.7942	0.568	0.6175
	64	16	81154	0.7975	0.5306	0.5639
	128	16	260610	0.8895	0.5128	0.4659
	128	32	260610	0.8725	0.5766	0.5866
	256	32	914434	0.971	0.6408	0.5253
	256	16	914434	0.9609	0.6298	0.6094
1e-05	32	8	28290	0.4978	0.5215	0.5063
	64	8	81154	0.5302	0.5745	0.5168
	64	16	81154	0.5186	0.5275	0.4493
	128	16	260610	0.5786	0.5493	0.3601
	128	32	260610	0.5700	0.4461	0.5888
	256	32	914434	0.6482	0.5150	0.5079
	256	16	914434	0.6413	0.4929	0.5492

5.2.2 MPNet representations

The fine-tuning of hyperparameters for the Feed-Forward model using MPNet representations is detailed in Table 5.8. When utilizing the Transformer-based pre-trained MPNet model for speech content representation alongside the Feed-Forward model for optimization, the performance is comparable to that of using the Transformer model with simpler Word2Vec representations. However, the MPNet and Feed-Forward architecture has a significantly smaller number of parameters, see Table 5.4 even with an embedding size of 128, due to its architectural simplicity. These results highlight the superior quality of MPNet speech content representations compared to Word2Vec, and also demonstrate the good generalization capabilities of this architecture.

Table 5.8: Performance of the Feed-Forward model on the RLT dataset using MPNet representations.

Learning Rate	**Embedding Size**	**Number Parameters**	**CCR**		
			Train	**Validation**	**Test**
1e-03	32	28114	0.9405	0.546	0.4789
	64	69538	1	0.5806	0.5845
	128	241474	1	0.6596	0.4887
	256	1285762	1	0.6175	0.5362
1e-04	32	28114	0.9974	0.4307	0.5372
	64	69538	1	0.6004	0.6364
	128	241474	1	0.5225	0.5855
	256	1285762	1	0.6279	0.5612
1e-05	32	28114	0.8352	0.5776	0.5662
	64	69538	0.8856	0.4918	0.5565
	128	**241474**	**0.9488**	**0.6476**	**0.6500**
	256	1285762	0.9789	0.6061	0.5954

The CNN model appears to encounter greater difficulty overfitting the training set when using MPNet speech content representations, as illustrated in Table 5.9. The increased representation size of 768 for MPNet, combined with its fixed embedding size of 1024, results in a substantial rise in the number of parameters. Despite this, the parameter configuration that achieves the highest CCR value does not overfit on the training set and demonstrates good generalization. Additionally, the kernel size seems to be appropriately calibrated to match the

dimensions of the MPNet speech content representations.

Table 5.9: Performance of the CNN model on the RLT dataset using MPNet representations.

Learning Rate	Embedding Size	CNN Kernel Size	Number Parameters	CCR		
				Train	Validation	Test
	1024	3	18116534	0.7169	0.558	0.5162
1e-03	1024	5	17817334	0.6504	0.5057	0.5125
	1024	8	17223682	0.5968	0.4812	0.5205
	1024	3	18116534	0.9984	0.5126	0.5416
1e-04	1024	5	17817334	0.9969	0.6228	0.5344
	1024	8	17223682	0.9849	0.5847	0.5552
	1024	3	18116534	0.9347	0.5325	0.4864
1e-05	1024	5	17817334	0.8766	0.5127	0.5414
	1024	**8**	**17223682**	**0.7995**	**0.5863**	**0.5613**

Since MPNet speech content representations have the lowest maximum number of representations per sample, the LSTM model should effectively capture temporal dependencies in the speech data. As shown in Table 5.10, the best results achieved with the LSTM model are comparable to those from our previous transformer-based approaches, detailed in Table 5.8 and Table 5.7.

Table 5.10: Performance of the LSTM model on the RLT dataset using MPNet representations.

Learning Rate	Embedding Size	Number Parameters	CCR		
			Train	Validation	Test
	32	33314	0.6934	0.5851	0.4884
	64	83010	0.769	0.4035	0.6134
1e-03	128	231554	0.8735	0.6041	0.5131
	256	725250	0.9613	0.5366	0.5078
	512	2499074	0.9897	0.6338	0.573
	32	33314	0.638	0.5676	0.4925
	64	83010	0.6994	0.5388	0.525
1e-04	128	231554	0.7651	0.5218	0.5644
	256	**725250**	**0.8301**	**0.6218**	**0.6224**
	512	2499074	0.9166	0.6068	0.562
	32	33314	0.5001	0.5206	0.4477
	64	83010	0.5512	0.4614	0.4893
1e-05	128	231554	0.5972	0.4982	0.5175
	256	725250	0.6695	0.5104	0.6057
	512	2499074	0.7259	0.6519	0.5592

The MPNet speech content representations achieve a notable performance while optimized by the Transformer model, exhibiting good generalization across both validation and test sets and easily overfitting on the training set, as demonstrated in Table 5.11. The optimal Transformer configuration also benefits from a smaller setup, with an embedding size of 32 and 8 attention heads, resulting in a relatively compact architecture. This efficiency results from the limited number of MPNet speech content representations, which directly influences the size of the Transformer model.

Table 5.11: Performance of the Transformer model on the RLT dataset using MPNet representations.

Learning Rate	**Embedding Size**	**Number Attention Heads**	**Number Parameters**	**CCR**		
				Train	**Validation**	**Test**
	32	**8**	**38018**	**0.999**	**0.6404**	**0.6164**
	64	8	100610	1	0.6181	0.5763
	64	16	100610	1	0.5756	0.5975
1e-03	128	16	299522	1	0.6205	0.5879
	128	32	299522	1	0.5668	0.5661
	256	32	992258	1	0.6010	0.5570
	256	16	992258	1	0.6396	0.5507
	32	8	38018	0.7488	0.5766	0.5461
	64	8	100610	0.8375	0.5611	0.5499
	64	16	100610	0.8763	0.5603	0.5823
1e-04	128	16	299522	0.9325	0.7172	0.6066
	128	32	299522	0.9358	0.5263	0.5530
	256	32	992258	0.9831	0.4984	0.6016
	256	16	992258	0.982	0.5026	0.5682
	32	8	38018	0.5267	0.5385	0.4924
	64	8	100610	0.5532	0.4794	0.4974
	64	16	100610	0.5841	0.4137	0.5634
1e-05	128	16	299522	0.5960	0.4996	0.4697
	128	32	299522	0.5719	0.6323	0.4919
	256	32	992258	0.6665	0.5333	0.4228
	256	16	992258	0.6714	0.4420	0.5408

5.2.3 RoBERTa representations

For the RoBERTa representations, we use the same set of modeling architectures applied to Word2Vec and MPNet. The dimension of the speech content representations obtained with the pre-trained RoBERTa model is proportional to their maximum number per sample.

The Feed-Forward architecture also delivers notable results with RoBERTa speech content representations, as shown in Table 5.12. This confirms that combining a transformer-based representation type with a shallow model for refinement can produce effective outcomes. Despite the large size of the RoBERTa speech content representations, the architecture remains relatively compact, achieving optimal results with a 64 embedding size and demonstrating robust generalization through its high validation CCR value.

Table 5.12: Performance of the Feed-Forward model on the RLT dataset using RoBERTa representations.

Learning Rate	**Embedding Size**	**Number Parameters**	**CCR**		
			Train	**Validation**	**Test**
1e-03	16	24682	0.5426	0.5877	0.495
	32	51154	0.848	0.428	0.5708
	64	115618	1	0.5008	0.5315
	128	333634	1	0.5002	0.4502
	256	1470082	1	0.4198	0.4981
	512	9297154	1	0.4185	0.4318
1e-04	16	24682	0.784	0.5115	0.567
	32	51154	0.9793	0.5329	0.4065
	64	115618	1	0.4223	0.4439
	128	333634	1	0.4979	0.588
	256	1470082	1	0.4999	0.5378
	512	9297154	1	0.5189	0.5325
1e-05	16	24682	0.8082	0.4952	0.6189
	32	51154	0.9706	0.4664	0.5265
	64	**115618**	**0.987**	**0.6718**	**0.6236**
	128	333634	0.9876	0.55	0.5699
	256	1470082	0.9952	0.5611	0.5752
	512	9297154	1	0.5383	0.5554

The results for the CNN model using RoBERTa speech content representations, as presented in Table 5.13, are very similar to those achieved with MPNet representations using the same CNN model, shown in Table 5.9. This indicates that the two pre-trained models, MPNet and RoBERTa, are quite similar to our task, as both are enhanced versions of the Bidirectional Encoder Representations from Transformers (BERT) architecture [41]. However, the maximum number of representations per sample differs significantly, ranging from 18 for MPNet to 738 for RoBERTa, resulting in the RoBERTa-CNN architecture having nearly nine times more parameters. This larger representation dimension appears to be redundant for the speech content deceit classification task, as it does not lead to improved results.

Table 5.13: Performance of the CNN on the RLT dataset using RoBERTa representations.

Learning Rate	**Embedding Size**	**CNN Kernel Size**	**Number Parameters**	**CCR**		
				Train	**Validation**	**Test**
1e-03	1024	3	124008374	0.6361	0.4913	0.5319
	1024	5	123432694	0.5633	0.5146	0.5542
	1024	8	122286082	0.5674	0.5302	0.526
1e-04	1024	3	124008374	1	0.5392	0.5144
	1024	5	123432694	0.9856	0.5207	0.5244
	1024	8	122286082	1	0.5296	0.5286
1e-05	1024	3	124008374	1	0.5524	0.4836
	1024	5	123432694	1	0.5412	0.5298
	1024	**8**	**122286082**	**1**	**0.523**	**0.5364**

The LSTM model's results using RoBERTa speech content representations, as shown in Table 5.14, reveal lower CCR values for the test set compared to the LSTM architecture using MPNet representations. This highlights the superiority of the pre-trained MPNet model in producing more compact and effective speech content representations. Furthermore, the model's inability to overfit the training set may be due to the redundancies present in the RoBERTa speech content representations.

Table 5.14: Performance of the LSTM model on the RLT dataset using RoBERTa representations.

Learning Rate	Embedding Size	Number Parameters	CCR		
			Train	Validation	Test
1e-03	32	33314	0.6274	0.4925	0.5891
	64	83010	0.6626	0.5796	0.5363
	128	231554	0.6939	0.5212	0.4497
	256	725250	0.7691	0.5471	0.5188
	512	2499074	0.8426	0.5517	0.5172
1e-04	32	33314	0.5728	0.5239	0.5873
	64	83010	0.5905	0.5343	0.4936
	128	231554	0.6296	0.5556	0.5590
	256	725250	0.6844	0.5564	0.5524
	512	2499074	0.7476	0.4958	0.4964
1e-05	32	33314	0.4937	0.5272	0.5179
	64	83010	0.5280	0.4735	0.4217
	128	231554	0.5742	0.5135	0.5524
	256	725250	0.5780	0.5348	0.5512
	512	2499074	0.6192	0.5246	0.5529

The Transformer model appears to be able to capture significant deceit information from RoBERTa speech content representations, as shown in Table 5.15, achieving better performance compared to MPNet representations. This might suggest that the Transformer architecture is more effective at handling complex features, with the results indicating robust generalization and effective learning.

Table 5.15: Performance of the Transformer model on the RLT dataset using RoBERTa representations.

Learning Rate	Embedding Size	Number Attention Heads	Number Parameters	CCR		
				Train	Validation	Test
	32	8	61058	0.999	0.5817	0.4429
	64	8	146690	1.000	0.5258	0.4850
	64	**16**	**146690**	**1.000**	**0.6369**	**0.6416**
1e-03	128	16	391682	1.000	0.5468	0.5237
	128	32	391682	1.000	0.5526	0.5801
	256	32	1176578	1.000	0.6545	0.5590
	256	16	1176578	1.000	0.5455	0.5666
	32	8	61058	0.6935	0.4628	0.5626
	64	8	146690	0.8131	0.5144	0.5133
	64	16	146690	0.8178	0.5550	0.5512
1e-04	128	16	391682	0.8935	0.4957	0.4843
	128	32	391682	0.8646	0.4840	0.6232
	256	32	1176578	0.9552	0.5469	0.5513
	256	16	1176578	0.9770	0.5927	0.5494
	32	8	61058	0.5096	0.4917	0.4867
	64	8	146690	0.5384	0.5269	0.5034
	64	16	146690	0.5092	0.5352	0.4654
1e-05	128	16	391682	0.5657	0.4615	0.4970
	128	32	391682	0.5516	0.4916	0.5302
	256	32	1176578	0.6237	0.4950	0.4955
	256	16	1176578	0.6291	0.4206	0.4263

5.3 Comparison to other studies

Our best results, presented in Table 5.16, indicate that the speech content modality provides more information for capturing deceit patterns compared to the voice modality.

Combining a shallow model with a transformer-based model proves effective, and similarly, the combination of two transformer-based models can achieve strong performance when the representations contain sufficient information, as demonstrated by the Transformer model using RoBERTa representations. Furthermore, despite their large number of parameters, CNN models tend to yield the lowest performance results. Additionally, the number of representations, detailed in Table 5.17, directly affects training time, with the LSTM architecture generally requiring the longest training duration across most experiments.

Table 5.16: Best-performing models on RLT dataset, based on Correct Classification Rate (CCR) and training time expressed in minutes per fold.

Modality	Feature Type	Architecture	Embedding Size	Number Parameters	Training Time	CCR
Voice	Handcrafted	LSTM	128	168322	7	**0.5972**
		Transformer	64	73090	7	0.5807
	Wav2Vec	LSTM	32	33314	30	0.4165
		Transformer	256	2029826	15	**0.5385**
Speech content	Word2Vec	Feed-Forward	64	50082	4	0.5675
		CNN	1024	10182554	6	0.5920
		LSTM	256	605442	8	0.6186
		Transformer	256	914434	6	**0.6562**
	MPNet	Feed-Forward	128	241474	4	**0.6501**
		CNN	1024	17223682	7	0.5613
		LSTM	256	725250	5	0.6224
		Transformer	32	38018	4	0.6164
	RoBERTa	Feed-Forward	64	115618	4	0.6236
		CNN	1024	122286082	8	0.5364
		LSTM	32	33314	11	0.5891
		Transformer	64	146690	7	**0.6416**

Table 5.17: Comparison of input representations dimensions for voice and speech content.

Modality	Method	Number representations	Representation dimension
Voice	Handcrafted	82	274
	Wav2vec	4071	768
Speech content	Word2vec	182	300
	MPNet	18	768
	RoBERTa	738	768

When analyzing the influence of class and gender on our best-performing architecture for the voice modality, specifically the Handcrafted representations and LSTM model, as shown in Table 5.18, we observe that female subjects appear to be more frequently associated with deceit than male subjects. This may be attributed to the higher pitch of female voices, which is more commonly found in deceitful samples.

However, similar patterns are observed in the gender results for transcribed speech, presented in Table 5.19. This consistency suggests that the imbalanced distribution of male and female subjects in the dataset may have a more significant

Table 5.18: Fold, gender and class results for best-performing voice architecture, Handcrafted representations and LSTM model, on RLT dataset, based on Correct Classification Rate (CCR)

Folds	Train Acc				Validation Acc				Test Acc			
	Truth		Deceit		Truth		Deceit		Truth		Deceit	
	Male	Female	Male	Female	Male	Female	Male	Female	Male	Female	Male	Female
1	0.88	0.8235	0.8889	0.9091	0.5	0.6667	1	0.4286	0	0.6667	0	0.5714
2	0.9412	0.8	1	0.7727	0.5	0.6	0	0.7222	0	0.5	0	0.2857
3	1	0.9	0.8	0.92	0	1	0	0.25	0.75	0.6	0	0.5556
4	0.9231	0.875	0.875	0.9762	0.5	0	0	1	0.6667	1	0	0.75
5	0.8462	0.8636	1	0.9268	0.6667	0.6667	0	0.2	0.5	1	0.5	0
6	0.9259	1	1	0.9268	0	0	0.5	1	1	0.3333	0	0.2
7	0.8889	0.875	0.875	0.9524	0	0	0	0.25	1	1	1	1
8	1	1	1	0.9767	0	1	0.3333	0	0.3333	1	0	0.5
9	0.9667	0.8571	1	0.9574	0	0.5	1	0	0	1	0.6667	0
10	0.9615	0.9524	1	0.875	0.25	0	0	0.5714	1	0.5	0.5	0
Average	1	1	1	1	0.2417	0.4433	0.2833	0.4422	0.5250	0.7600	0.2667	0.3863
std	0	0	0	0	0.2734	0.4128	0.4161	0.3714	0.4232	0.2666	0.3702	0.3460

impact on these results.

We present the performance of various methods from the literature that utilize the RLT dataset. It is important to note that the experimental setups for the RLT dataset vary, therefore, the results with these methods are not directly comparable. Additionally, given the limitations of the RLT dataset, characterized by its sparse data availability, these results may fluctuate with larger datasets and necessitate further validation.

For voice modality, our best model archives lower accuracy compared to other similar models, as shown in Table 5.20. It is even outperformed by the human annotators, whose performance is detailed in the work of Sen et al. [55]. On the other hand, for the speech content modality, our LSTM architecture using Handcrafted speech representations outperforms all other models, except for the one that we implemented as our CNN architecture, as shown in Table 5.21. The significant difference in accuracy for this model may be attributed to the different fold configuration used by Krishnamurthy et al. [10] in their work.

Table 5.19: Fold, gender and class results for best-performing speech content architecture, pre-trained Word2Vec and Transformer, on RLT dataset, based on Correct Classification Rate (CCR)

Folds	Train Acc				Validation Acc				Test Acc			
	Truth		Deceit		Truth		Deceit		Truth		Deceit	
	Male	Female	Male	Female	Male	Female	Male	Female	Male	Female	Male	Female
1	1	1	1	1	1	0.3333	0	0.5714	0.5	1	0	0.4286
2	1	1	1	1	0.75	0.4	0	0.3333	0	0.5	0	0.7143
3	1	1	1	1	0.6667	0	0	0.75	0.75	0.8	0	0.3333
4	1	1	1	1	0.5	1	0	1	0.6667	1	0	0.5
5	1	1	1	1	0.3333	0.3333	0	0.6	0.5	1	0.5	0
6	1	1	1	1	1	1	0.5	1	1	0.3333	0	0.8
7	1	1	1	1	0.6667	1	0	0.75	1	1	0.5	1
8	1	1	1	1	0	0.3333	0.6667	0	0.6667	1	0	0.75
9	1	1	1	1	1	0.5	0.5	0	0	0.6667	0.3333	0
10	1	1	1	1	0.75	1	0	0.7143	1	1	1	1
Average	1	1	1	1	0.6667	0.5900	0.1667	0.5719	0.6083	0.8300	0.2333	0.5526
std	0	0	0	0	0.3216	0.3745	0.2722	0.3593	0.3728	0.2487	0.3443	0.3657

Table 5.20: Comparison to other methods for voice modality on RLT dataset, based on Correct Classification Rate (CCR)

Method	Year	Architecture	Feature Type	Accuracy
Krishnamurthy et al. [10]	2018	Multi-Layer Perceptron (MLP)	openSMILE	52.38%
Sen at al. [55]	2020	Random Forest (RF)	Pitch	71.19%
Chebbi et al. [56]	2023	K-Nearest Neighbor (KNN)	Handcrafted	60%
Human annotators [55]	2020	-	-	69.22%
Ours	2024	LSTM	Handcrafted	59.72%

Table 5.21: Comparison to other methods for speech content modality on RLT dataset, based on Correct Classification Rate (CCR)

Method	Year	Architecture	Feature Type	Accuracy
Krishnamurthy et al. [10]	2018	TextCNN (static)	Word2Vec	82.31%
Sen at al. [55]	2020	Random Forest (RF)	Unigrams	64.41%
Chebbi et al. [56]	2023	K-Nearest Neighbor (KNN)	Lexical representations	58%
Human annotators [55]	2020	-	-	64.10%
Ours	2024	Transformer	Word2Vec	65.62%

Chapter 6

Conclusion

This research presents a comprehensive analysis of vocal and speech content data for deceit detection. By exploring a variety of input representation types and modeling architectures, we aimed to identify the most effective methods for detecting deceit. Our experiments were conducted on the Real-Life Trial (RLT) dataset, which embodies high-stakes real-life scenarios, thereby enhancing the practical relevance of our findings.

We employed handcrafted input representations—including Mel-Frequency Cepstral Coefficients (MFCCs), chroma features, mel spectrograms, and tonnetz—to effectively capture the emotional nuances of the voice signals. Additionally, renowned pre-trained models such as Wav2Vec, Word2Vec, RoBERTa, and Sentence-BERT (MPNet) were leveraged for advanced representation learning. These models facilitated the extraction of high-quality information from the data, contributing to more accurate deceit detection.

Our findings indicate that transcribed speech content provides more substantial information for deceit detection than vocal characteristics alone. This suggests that the semantic content of speech is a more critical factor than the vocal attributes when identifying deceptive behavior. Moreover, transformer architectures demonstrated exceptional effectiveness in representation learning and

modeling, capturing complex patterns and dependencies within the data. This effectiveness was evident in both the feature extraction and optimization stages of our analysis.

In addition to Transformer architectures, we employed simpler models such as Feed-Forward Networks (NNs) and more advanced models like Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNNs). The Feed-Forward Networks provided a baseline for comparison, while the LSTM networks were particularly useful in modeling sequential data due to their ability to retain information over time. The CNNs were applied to extract key features from transcribed speech input by utilizing filters of varying sizes to capture different n-grams in the speech content. These approaches allowed us to compare the effectiveness of different levels of architectural complexity.

While transformer-based models showed high performance individually, combining two transformer-based techniques resulted in a decrease in efficiency in most of the cases when compared to other approaches. Despite this, our best results were comparable to, and in some cases surpassed, existing leading models for speech content and voice deceit detection. This highlights the potential of transformer architectures when properly configured and underscores the importance of model selection and architectural design.

A significant consideration in our study is the limited size and poor gender distribution of the RLT dataset, which may influence the overall performance and generalization potential of our approaches. Thus, we created a balanced 10-fold cross-validation setup to alleviate this problem. To further explore this, we provided gender and class results for our best-performing models, which consisted of the Transformer model with Word2Vec representations for speech content modality and the LSTM model with handcrafted representations for voice modality. From these results, it became evident that data scarcity negatively impacts the performance of deceit detection, even though we implemented a balanced cross-validation fold configuration. Regarding gender differences, deceit was mostly attributed to female subjects despite their majority in the dataset as unique subjects. For the voice analysis, this could be due to their higher pitch,

often associated with deceit. Nevertheless, this issue is also evident in speech content analysis, further reinforcing the hypothesis that data sparsity is the primary contributing factor.

Although our best-performing model achieved performance levels comparable to human evaluators, deceit detection remains a challenging task. Addressing this challenge will require overcoming data limitations and pursuing further research on generalization methods to enhance model robustness across diverse contexts.

In summary, our systematic analysis emphasizes the superior informativeness of speech content over vocal characteristics in deceit detection and showcases the advantages of Transformer, LSTM, CNN, and Feed-Forward architectures together with handcrafting representations for identifying complex data patterns. Furthermore, through our results and comparison with state-of-the-art models, we showcase the crucial role of the dataset and its configuration in enhancing the reliability and applicability of automated deceit detection systems. Future research should focus on expanding the number and quality of deceit detection datasets to mitigate the data scarcity issue and on exploring more representation types and advanced modeling techniques. Such efforts will be crucial in enhancing the reliability and applicability of automated deceit detection systems, ultimately contributing to more effective identification of deceptive behaviors in real-world scenarios.

Bibliography

[1] C. W. Martin, *The philosophy of deception*. Oxford University Press, 2009.

[2] M. H. Boynton, D. B. Portnoy, and B. T. Johnson, "Exploring the ethics and psychological impact of deception in psychological research," *IRB*, vol. 35, no. 2, p. 7, 2013.

[3] W. Von Hippel and R. Trivers, "The evolution and psychology of self-deception," *Behavioral and brain sciences*, vol. 34, no. 1, pp. 1–16, 2011.

[4] B. L. Verigin, E. H. Meijer, G. Bogaard, and A. Vrij, "Lie prevalence, lie characteristics and strategies of self-reported good liars," *PloS one*, vol. 14, no. 12, p. e0225566, 2019.

[5] B. M. DePaulo, D. A. Kashy, S. E. Kirkendol, M. M. Wyer, and J. A. Epstein, "Lying in everyday life.," *Journal of personality and social psychology*, vol. 70, no. 5, p. 979, 1996.

[6] E. Harmon-Jones and J. Mills, "An introduction to cognitive dissonance theory and an overview of current perspectives on the theory.," 2019.

[7] J. E. Stets and P. J. Burke, "Identity theory and social identity theory," *Social psychology quarterly*, pp. 224–237, 2000.

[8] G. Iñiguez, T. Govezensky, R. Dunbar, K. Kaski, and R. A. Barrio, "Effects of deception in social networks," *Proceedings of the Royal Society B: Biological Sciences*, vol. 281, no. 1790, p. 20141195, 2014.

[9] P. Ekman and W. V. Friesen, "Nonverbal leakage and clues to deception," *Psychiatry*, vol. 32, no. 1, pp. 88–106, 1969.

[10] G. Krishnamurthy, N. Majumder, S. Poria, and E. Cambria, "A deep learning approach for multimodal deception detection," in *International Conference on Computational Linguistics and Intelligent Text Processing*, pp. 87–96, Springer, 2018.

[11] D. Grubin and L. Madsen, "Lie detection and the polygraph: A historical review," *The Journal of Forensic Psychiatry & Psychology*, vol. 16, no. 2, pp. 357–369, 2005.

[12] R. Adelson, "Detecting deception," *APA Monitor on Psychology*, vol. 35, no. 7, pp. 70–73, 2004.

[13] P. Ekman and W. V. Friesen, "Facial action coding system," *Environmental Psychology & Nonverbal Behavior*, 1978.

[14] L. Su and M. D. Levine, "High-stakes deception detection based on facial expressions," in *2014 22nd International Conference on Pattern Recognition*, pp. 2519–2524, IEEE, 2014.

[15] D. Avola, L. Cinque, G. L. Foresti, and D. Pannone, "Automatic deception detection in rgb videos using facial action units," in *Proceedings of the 13th International Conference on Distributed Smart Cameras*, pp. 1–6, 2019.

[16] H. U. D. Ahmed, U. I. Bajwa, F. Zhang, and M. W. Anwar, "Deception detection in videos using the facial action coding system," *arXiv preprint arXiv:2105.13659*, 2021.

[17] M. Zuckerman, "Verbal and nonverbal communication of deception," *Advances in experimental social psychology/Academic Press*, 1981.

[18] M. L. Newman, J. W. Pennebaker, D. S. Berry, and J. M. Richards, "Lying words: Predicting deception from linguistic styles," *Personality and social psychology bulletin*, vol. 29, no. 5, pp. 665–675, 2003.

[19] L. Zhou, J. K. Burgoon, J. F. Nunamaker, and D. Twitchell, "Automating linguistics-based cues for detecting deception in text-based asynchronous computer-mediated communications," *Group decision and negotiation*, vol. 13, pp. 81–106, 2004.

[20] J. T. Hancock, L. E. Curry, S. Goorha, and M. Woodworth, "On lying and being lied to: A linguistic analysis of deception in computer-mediated communication," *Discourse Processes*, vol. 45, no. 1, pp. 1–23, 2007.

[21] J. W. Pennebaker, "Linguistic inquiry and word count: Liwc 2001," 2001.

[22] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock, "Finding deceptive opinion spam by any stretch of the imagination," *arXiv preprint arXiv:1107.4557*, 2011.

[23] X. Wang, X. Zhang, C. Jiang, and H. Liu, "Identification of fake reviews using semantic and behavioral features," in *4th International Conference on Information Management (ICIM)*, pp. 92–97, IEEE, 2018.

[24] D. P. Jayathunga, R. I. S. Ranasinghe, and R. Murugiah, "A comparative study of supervised machine learning techniques for deceptive review identification using linguistic inquiry and word count," in *Computational Intelligence in Information Systems: Proceedings of the Computational Intelligence in Information Systems Conference (CIIS 2020)*, pp. 97–105, Springer, 2021.

[25] K. Stevens, "Autocorrelation analysis of speech sounds," *The Journal of the Acoustical Society of America*, vol. 22, no. 6, pp. 769–771, 1950.

[26] C. P. Smith, "A phoneme detector," *The Journal of the Acoustical Society of America*, vol. 23, no. 4, pp. 446–451, 1951.

[27] C. G. Howard, "Speech analysis-synthesis scheme using continuous parameters," *The Journal of the Acoustical Society of America*, vol. 28, no. 6, pp. 1091–1098, 1956.

[28] A. Rihaczek, "Signal energy distribution in time and frequency," *IEEE Transactions on information Theory*, vol. 14, no. 3, pp. 369–374, 1968.

[29] F. Horvath, "Detecting deception: the promise and the reality of voice stress analysis," *Journal of Forensic Sciences*, vol. 27, no. 2, pp. 340–351, 1982.

[30] X. Zhou and R. Zafarani, "A survey of fake news: Fundamental theories, detection methods, and opportunities," *ACM Computing Surveys (CSUR)*, vol. 53, no. 5, pp. 1–40, 2020.

[31] T. Randhavane, U. Bhattacharya, K. Kapsaskis, K. Gray, A. Bera, and D. Manocha, "The liar's walk: Detecting deception with gait and gesture," *arXiv preprint arXiv:1912.06874*, 2019.

[32] D. Avola, L. Cinque, M. De Marsico, A. Fagioli, and G. L. Foresti, "Lietome: Preliminary study on hand gestures for deception detection via fisher-lstm," *Pattern Recognition Letters*, vol. 138, pp. 455–461, 2020.

[33] A. Gallardo-Antolín and J. M. Montero, "Detecting deception from gaze and speech using a multimodal attention lstm-based framework," *Applied Sciences*, vol. 11, no. 14, p. 6393, 2021.

[34] J. Joy, A. Kannan, S. Ram, and S. Rama, "Speech emotion recognition using neural network and mlp classifier," *Ijesc*, vol. 2020, pp. 25170–25172, 2020.

[35] R. Arya, D. Pandey, A. Kalia, B. J. Zachariah, I. Sandhu, and D. Abrol, "Speech based emotion recognition using machine learning," in *IEEE Mysore Sub Section International Conference (MysuruCon)*, pp. 613–617, IEEE, 2021.

[36] T. Mikolov, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[37] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.

[38] Q. Li, Q. Hu, Y. Lu, Y. Yang, and J. Cheng, "Multi-level word features based on cnn for fake news detection in cultural communication," *Personal and Ubiquitous Computing*, vol. 24, pp. 259–272, 2020.

[39] S. Venkatesh, R. Ramachandra, and P. Bours, "Video based deception detection using deep recurrent convolutional neural network," in *Computer Vision and Image Processing: 4th International Conference, CVIP 2019, Jaipur, India, September 27–29, Revised Selected Papers, Part II 4*, pp. 163–169, Springer, 2020.

[40] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[41] J. Devlin, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[42] R. K. Kaliyar, A. Goswami, and P. Narang, "Fakebert: Fake news detection in social media with a bert-based deep learning approach," *Multimedia tools and applications*, vol. 80, no. 8, pp. 11765–11788, 2021.

[43] T. Fornaciari, F. Bianchi, M. Poesio, D. Hovy, *et al.*, "Bertective: Language models and contextual information for deception detection," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Association for Computational Linguistics, 2021.

[44] Y. Liu, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[45] P. Gupta, S. Gandhi, and B. R. Chakravarthi, "Leveraging transfer learning techniques-bert, roberta, albert and distilbert for fake review detection," in *Proceedings of the 13th Annual Meeting of the Forum for Information Retrieval Evaluation*, pp. 75–82, 2021.

[46] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 11 2019.

[47] A. Turnip, M. F. Amri, H. Fakrurroja, A. I. Simbolon, M. A. Suhendra, and D. E. Kusumandari, "Deception detection of eeg-p300 component classified by svm method," in *Proceedings of the 6th international conference on software and computer applications*, pp. 299–303, 2017.

[48] V. Pérez-Rosas and R. Mihalcea, "Experiments in open domain deception detection," in *Proceedings of the conference on empirical methods in natural language processing*, pp. 1120–1125, 2015.

[49] J. B. Hirschberg, S. Benus, J. M. Brenier, F. Enos, S. Friedman, S. Gilman, C. Girand, M. Graciarena, A. Kathol, L. Michaelis, *et al.*, "Distinguishing deceptive from non-deceptive speech," 2005.

[50] H. Nasri, W. Ouarda, and A. M. Alimi, "Relidss: Novel lie detection system from speech signal," in *IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*, pp. 1–8, IEEE, 2016.

[51] V. Pérez-Rosas, M. Abouelenien, R. Mihalcea, and M. Burzo, "Deception detection using real-life trial data," in *Proceedings of the ACM on international conference on multimodal interaction*, pp. 59–66, 2015.

[52] V. Gupta, M. Agarwal, M. Arora, T. Chakraborty, R. Singh, and M. Vatsa, "Bag-of-lies: A multimodal dataset for deception detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 0–0, 2019.

[53] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[54] F. Eyben, F. Weninger, F. Gross, and B. Schuller, "Recent developments in opensmile, the munich open-source multimedia feature extractor," in *Proceedings of the 21st ACM international conference on Multimedia*, pp. 835–838, 2013.

[55] M. U. Şen, V. Perez-Rosas, B. Yanikoglu, M. Abouelenien, M. Burzo, and R. Mihalcea, "Multimodal deception detection using real-life trial data," *IEEE Transactions on Affective Computing*, vol. 13, no. 1, pp. 306–319, 2020.

[56] S. Chebbi and S. B. Jebara, "Deception detection using multimodal fusion approaches," *Multimedia Tools and Applications*, vol. 82, no. 9, pp. 13073–13102, 2023.

[57] G. Sharma, K. Umapathy, and S. Krishnan, "Trends in audio signal feature extraction methods," *Applied Acoustics*, vol. 158, p. 107020, 2020.

[58] S. B. Davis and P. Mermelstein, *Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences*, vol. 28. IEEE Transactions on Acoustics, Speech, and Signal Processing, 1980.

[59] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *The journal of the acoustical society of america*, vol. 8, no. 3, pp. 185–190, 1937.

[60] P. Grosche, M. Müller, and J. Serra, "Audio content-based music retrieval," in *Dagstuhl Follow-Ups*, vol. 3, Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2012.

[61] E. J. Humphrey, T. Cho, and J. P. Bello, "Learning a robust tonnetz-space transform for automatic chord recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 453–456, 2012.

[62] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12449–12460, 2020.

[63] S. Hochreiter, "Long short-term memory," *Neural Computation MIT-Press*, 1997.