

Modelling the Flow of Emotion: A Hierarchical Attention Network for Vocal Trajectory Classification

Vamsi Addagarla¹, Dr.D.Haritha²,

¹Research Scholar, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation,

Green Fields, Vaddeswaram, Vijayawada, Andhra Pradesh, India

Email: 2401050072@kluniversity.in

²Associate Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation,

Green Fields, Vaddeswaram, Vijayawada, Andhra Pradesh, India

Abstract

Conventional Vocal biomarkers have quantifiable correlates of mental health, and talk is a convenient and non-invasive indicator of mental health. The majority of the existing systems, though, consider every voice recording as a single observation and generate one emotional label per utterance instead of monitoring the temporal dynamics in the affective state. A single-point method is unable to reflect the changes in mood direction that have the most clinical implications a days-long change in emotional valence is much more informative than a single reading, but existing architectures do not support this.. Emotional state is longitudinal in nature and its course rather than its value at any given moment captures the most dynamics of relevance to well-being. We tackle this by formulating emotion analysis as a sequence classification task and training a Hierarchical Attention Network (HAN) to classify the emotional trend of a speaker as Improving, Declining or Stable. This model is based on chronologically ordered sequences of voice recordings sampled out of the MSP-Podcast corpus v1.7, with a two-level Bidirectional LSTM using attention performing at both the acoustic frame level and the voice-entry level. Each entry representation is enriched with contextual metadata such as time-of-day and ambient noise level that is concatenated with acoustic features.. The labels of the trajectory ground-truth are obtained based on the slope of linear regression of per-entry valence scores. Using a speaker-independent test partition, the HAN is 97.0% accurate and 0.97-weighted F1-score. A ablation study attests that both sets of hierarchical attention make significant contributions, and that the entire architecture performs better than a typical Bi-LSTM by 11.6 percentage points. This study provides a principled approach to longitudinal mood tracking based on voice that goes beyond snapshot classification and tracks the dynamics of emotion. The system can handle a 10-item sequence with a 198.6 ms response, which is reasonable to implement in close-to-real-time.

Keywords: Mental Health Monitoring, Vocal Emotion Recognition, Emotional Trajectory, Hierarchical Attention Network, Bidirectional LSTM, Sequence Classification, MSP-Podcast, Affective Computing

How to cite this article: Vamsi Addagarla¹ Dr.D.Haritha². Modelling the Flow of Emotion: A Hierarchical Attention Network for Vocal Trajectory Classification. Int J Drug Deliv Technol. 2026;16(86s):1428-1445. DOI: 10.25258/ijddt.16.86s.124.

1.INTRODUCTION

Mental health disorders can be considered one of the most acute problems of the present-day public health, as almost a billion individuals in the world experience them and an immense burden on health care systems, which were not created to monitor the problem on a continuous and preventive basis. Clinical evaluation continues to depend on periodic consultations and self-reported questionnaire tools that are necessarily retrospective and provide a rough overview of the development of the affective state of a patient in everyday life. The human voice brings up a very realistic alternative. It is continuously present, it does not need any special hardware and it encodes paralinguistic messages that vary in quantifiable manners as a result of psychological condition. Speech has already been shown to be a plausible source of passive, longitudinal mood monitoring since research has already found meaningful correlations between vocal biomarkers and various conditions such as depression and bipolar disorder. Vocal emotion

recognition has had significant technical advances. Recurrent and convolutional architecture-based systems are now capable of classifying discrete emotional states of single utterances with a high degree of accuracy on naturalistic datasets. The weakness is not a weakness of the classification performance but of the scope of the analysis. Each of the systems that exists currently considers the person who records as the basic unit one utterance in, one emotional label out.. This is usually good enough in general affective computing, but in mental health applications a major blind spot is formed. One low-value recording can tell a clinician not much. The pattern over time: Is the emotional status of the person maintaining itself, improving, or getting worse day by day or week by week, has diagnostic weight. The early signs of a depressive episode might be a long-term downward trajectory long before the individual would even consider seeking assistance, but no one-utterance classifier is set up to identify it. The change required is one where the unit

of analysis is moved to the temporal sequence of recordings, and that of state classification is replaced by trajectory classification. The time-varying direction and continuation of mood present clinical information that cannot be presented by the simply point-in-time labels. This is further complicated by contextual factors: tapes recorded in a noisy room or at night vary systematically in their acoustic attributes and omitting such contextual variables ecologically invalidates any monitoring system. A sincere model of emotional condition should take into consideration the circumstances of recording every recording. These limitations are dealt with in this paper. We introduce a Hierarchical Attention Network (HAN) that takes sequentially ordered sets of voice samples and predicts the overall emotional trend of a speaker as Improving, Declining, or Stable. The architecture has two levels: a frame-level Bidirectional LSTM with attention summarizes each voice entry in the sequence, and an entry-level Bidirectional LSTM with attention summarizes how emotional content changes throughout the entire sequence. The acoustic features 39-dimensional MFCCs, fundamental frequency, RMS energy, jitter, and shimmer are combined with contextual metadata encoding time-of-day and ambient noise level. The model is trained and tested on the MSP-Podcast corpus v1.7, where trajectory labels are calculated as the slope of the linear regression of the per-entry scores of valence. The goals that inform this work are as follows: to create a feature extraction pipeline that combines paralinguistic and contextual cues into a single representation; to develop a sequence model that can identify directional trends across multiple voice entries; to develop a principled methodology of labelling that can be used to translate continuous valence annotations into interpretable trajectory classes; and to empirically validate the entire system, including real-time inference bench

2. LITERATURE REVIEW

The 2.1 Deep Learning Approaches for Speech Emotion Recognition

The last five years have seen the fundamental change in the design and training of speech emotion recognition (SER) system. Whereas previous architectures used hand-crafted sets of acoustic features, fed to shallow classifiers, the newer systems are moving towards handing over the whole representation learning process to end-to-end deep models. Convolutional neural networks made an early presence in this shift by considering spectrograms to be images, and thus making it possible to extract spatial features over time-frequency representations [1]. Recurrent architectures, especially bidirectional LSTMs, soon indicated that temporal dependencies within an utterance were also important and the mix of

convolutional and recurrent layers became a popular baseline [2]. Another qualitative change was introduced by the introduction of attention mechanisms. Attention-based models do not assume that each frame of a speech signal has the same amount of information to convey, but instead they learn to focus representation on the emotionally salient points within an utterance a property that was explicitly demonstrated to boost accuracy on naturalistic, spontaneous speech data [3]. Hierarchical and multi-scale temporal processes helped to refine this. Ye et al. suggested a temporal-aware bidirectional multi-scale network TIM-Net, which learns emotion representations in multiple time scales concurrently, proving that emotion is not a signal on a single scale, but a multi-resolution phenomenon [4]. Lin and Busso scaled sequence-level temporal modeling to a dynamic chunk-level-based scheme that translates utterances of variable length into a predetermined amount of time chunks, achieving large gains in accuracy and computational efficiency on the MSP-Podcast corpus [5]. The use of pre trained models that are self supervised has perhaps been the most notable development. Wagner et al. have shown that dimensional emotion labels fine-tuned on transformer architecture specifically wav2vec 2.0 and HuBERT far surpass CNN baselines in valence prediction on MSP-Podcast, a dimension that has historically been difficult to model with acoustic-only models [6]. This discovery made transformer-based foundation models as the existing standard in SER. Latif et al. gave an in-depth overview of deep representation learning in SER, listing the pathway of feature engineering to self-supervised pre-training and the challenge of cross-corpus generalization as a current unresolved problem [7]. In spite of these progresses, there is one major shortcoming that cuts across almost all of these systems: all of them consider emotion recognition to be a sentence-level or utterance-level classification problem. The time modelling in such architectures is geared towards enhancing accuracy at the per utterance level rather than describing how the emotional status of a speaker changes over a series of utterances over time.

2.2 Vocal Biomarkers in Mental Health Monitoring

Voice as a non-invasive access to mental health has become a significant trend and has been motivated by the increasing access to large naturalistic speech corpora and by advances in the extraction of acoustic features. Studies have determined quantifiable relationships between vocal biomarkers such as fundamental frequency variability, speech rate, jitter, shimmer and pause duration and clinically determined conditions as major depressive disorder, generalized anxiety disorder, and bipolar disorder

[8]. Almaghrabi et al. have reviewed bio-acoustic characteristics unique to depression with a focus on synthesizing evidence that depressed speech is acoustically different due to reduced pitch range, flattened affect, and hypophonia and can be useful in providing reliable computational features when properly modeled [9]. One such significant point has been the indication that deep learning systems can recognize depression in voice with clinically significant accuracy. Transformer-based studies on prosodic feature sets of datasets like DAIC-WOZ have been found to achieve F1-scores of between 0.78 and 0.87 with noticeable improvement over previous SVM-based methods [10]. Perez-Toro et al. built upon this body of evidence, by user-state modeling the arousal-valence plane, as they found that dimensional emotion coordinates may be an intermediate scale between acoustic features and the severity of clinical symptoms [11]. In the meantime, Donaghy et al. provided a systematic review of voice-based depression detection machine learning systems, which demonstrated that although results are encouraging, most implemented systems do not yet meet the sensitivity standards of the PHQ-9 clinical screening tool, and that ecological validity is a highly urgent issue that has not been sufficiently addressed yet [12]. In particular, the disparity between laboratory-induced speech and natural speech during spontaneous conversation remains a limiting factor to applicability to clinical environments. Wang et al. showed that longitudinal acoustic monitoring of the severity of depression tracking the same speaker over time makes the similar predictions on a much more stable basis, and is clinically relevant, than those made by single-session recordings, which would directly motivate the trajectory-oriented approach described in this paper [13].

2.3 Longitudinal and Temporal Emotion Modeling

Although it is generally understood that mental health is fundamentally longitudinal construct, most speech emotion systems are still operating on the assumption that any given recording is a discrete event. Lin and Busso have also solved a related structural issue by sequential modeling that considers the non-uniform temporal allocation of the expression of emotion in an utterance, where the pattern of emotional variation with time does hold information independent of the mean emotional label [14]. This, which was done on MSP-Podcast, provided a methodological framework of treating temporal dynamics as a primary modeling focus and not as an optional factor. Goncalves and Busso also delved into the integration of unlabeled emotional data using self-supervised pre-text tasks that were specifically created to elicit temporal emotional contingencies to enhance the quality of learned

representations to sequence-based SER [15]. Session and conversation level emotion modeling in the general affective computing literature has been receiving increasing interest. Conversation-based emotion recognition systems (ERC) take advantage of the sequential form of multi-speaker interactions, leveraging the information of prior utterances to enhance per-turn classification [16]. These methods only capture short-term conversational dynamics in a single session, and are not intended to capture the directional change in the emotional state of a single speaker over days or weeks. This is a significant difference: ERC records contextual influence in a conversation, whereas the trajectory-modeling problem that the current paper is dealing with is whether the overall affective state of an individual is improving, stable, or getting worse over a long time a much different modeling goal that current systems are not in charge of achieving..

2.4 Context-Aware and Multimodal Emotion Recognition

It has been shown that the vocal expression of emotions is significantly moderated by the environmental and situational context. Background noise, time of the day, social setting not only have an effect on the acoustic characteristics of the speech but also the emotional register that a speaker assumes. Mocanu et al. confirmed that the audio-video cross-modal fusion with an attention mechanism leads to the same level of emotion recognition accuracy improvement as audio-only models, with the visual stream as a contextual addition, which the acoustic signal cannot obtain alone [17]. Siriwardhana et al. demonstrated that contextually sensitive speech representations can be learned by self-supervised multimodal feature fusion with transformer encoders and that it generalizes more across datasets [18]. Even more recent efforts have shifted to explicitly incorporating environmental metadata and situation indicators. Zhang et al. have provided a systematic review of deep learning-based multimodal emotion recognition, where audio-text fusion seemed to be consistently superior to audio-only, and context-aware architectures are becoming a core of competitive systems [19]. Wang et al. discovered multimodal transformer fusion with acoustic and text modalities, demonstrating that transformer-based cross-modal attention yields more stable valence predictions compared to feature concatenation baselines [20]. Regardless of these advances, contextual metadata of the most interest in longitudinal monitoring time-of-day cues, noise environment descriptors, and situation-specific codes at the session level are seldom directly incorporated into time sequence models. The data is usually in naturalistic recording situations but not utilised by the current SER architectures. This is a

pragmatic gap that is bridged by the current work by explicit metadata concatenation at the feature vector level.

2.5 Datasets and Evaluation Frameworks

The Recent advances in speech-based affective computing have been made possible by the availability of large-scale, naturalistic datasets. The MSP-Podcast corpus, which was comprehensively described by Busso et al., is currently a collection of over 400 hours of annotated speech of more than 3,641 speakers based on publicly licensed podcasts [21]. Most importantly to the current study, the corpus offers both categorical emotion labels as well as continuous dimensional ratings of valence, arousal, and dominance, which makes it possible to build up trajectory labels based on valence trend analysis. Its speaker diversity, the naturalistic circumstances of recording, and multi-rater annotation protocol ensure that the corpus is currently the most ecologically valid large-scale SER resource. Lin and Busso used the chunk-level SER framework to show that MSP-Podcast can be used to systematically conduct experiments of temporal modeling in a speaker-independent evaluation condition [5]. MuSe challenge series has brought about competitive benchmarking of continuous and sentiment-based emotion modeling to the field that pushes the field to real-world deployment scenarios [22]. The AVEC challenge framework was introduced by Ringeval et al. as a general assessment platform of depression and emotion recognition, and the following editions have steadily shifted to more naturalistic speech environments [23]. The Multimodal EmotionLines Dataset and IEMOCAP remain conversation-level benchmarks [24], and recent research on EMO-SUPERB has given a formal assessment of self-supervised learning models on six emotion datasets, finding no single pre-trained model to generalize across corpora [25]. Together, these resources demonstrate a common trend: the schemes of evaluation have focused on individual utterance or session classification, and there is no standard measure of longitudinal course prediction. The lack of this encourages the assessment approach which is used in the current research.

2.6 Research Gap and Positioning of This Work

The above literature review indicates a gap that is evident and consistent. Although deep learning has significantly improved the classification of per-utterance emotion, and voice-based mental health monitoring has proven to be clinically relevant, no current system uses the directional trajectory of emotional change as the primary classification target. The temporal modeling that exists in existing

architectures, be it LSTMs, attention mechanisms or transformers is all aimed at enhancing the performance of per-utterance or per-session labels, not at modeling whether the affective state of a speaker is fluctuating towards a better or worse state over a long timeline.. When included at all, contextual metadata is employed to enhance single-point classification and not to drive sequential trajectory estimation. The MSP-Podcast corpus has the dimensional valence annotations, and speaker-level recording structure needed to create such trajectory labels, but this potential has not been previously used to create trajectory classification. It directly fills that gap by creating emotional trajectory classification that predicts whether the valence trend of a speaker is Improving, Declining, or Stable in a sequence of voice entries organized by time as a first-class task, and by training a Hierarchical Attention Network with two layers of Bi-LSTM layers to do so..

3. SYSTEM METHODOLOGY

The The proposed system converts a sequence of uncoded voice recordings into a categorized emotional sequence by processing data through a three-stage data processing pipeline. Each voice sample in a recording waveform of a speaker is extracted into features in the first stage, resulting in an integrated acoustic and contextual representation. The acoustic component, which encodes spectral, prosodic and voice-quality features of the signal, is followed by a contextual metadata vector which encodes contextual factors, known to adjust vocal expression without requiring an emotional state, such as time-of-day, and ambient noise level. The two representations are concatenated to a single unified feature vector per entry. At the second stage, the sequence of entry vectors is fed to a Hierarchical Attention Network (HAN) which works on both levels of temporal abstraction in parallel. At the bottom a Bidirectional LSTM that incorporates frame level attention compiles the frame level acoustic frame data in all the voice entries in each voice into a small entry representation. At this higher-level, a second Bidirectional LSTM with entry-level attention models the dynamics of those representations over the entire chronological sequence, which entries most informative about the direction of emotional change. The third stage involves the trajectory context vector generated by the top-level attention system being sent to a softmax classification layer which produces a probability distribution across three trajectory types: Improving, Declining, and Stable. Figure 3.1 shows the entire architecture.

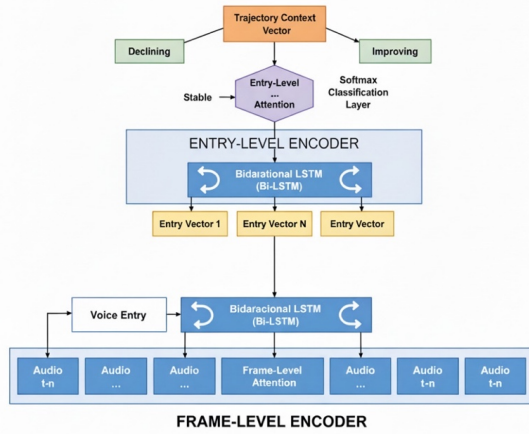


Fig 3.1 : Overall system pipeline diagram showing the three-stage flow feature extraction, hierarchical attention network, and trajectory classification with arrows indicating data flow from raw audio input through frame-level encoding, entry-level encoding, and softmax output

All experiments are based on the MSP-Podcast corpus version 1.7, a large-scale dataset of naturalistic speech consisting of podcast recordings that are publicly available. The corpus gives continuous dimensional notes of Valence, Arousal and Dominance per segment. Speaker sequences are built, i.e., by sequencing each speaker recording in time order, to create ordered sequences, whose valence scores are used to derive trajectory labels. This is a principled proxy of longitudinal follow-up as opposed to a clinically validated mood diary, and that is recognized all through what follows.

3.1 Vocal and Contextual Feature Extraction

For each voice entry j within a user's sequence, a comprehensive set of features is extracted. The primary vocal features are derived from the raw audio signal, sampled at 16kHz.

1. Spectral Features

The core spectral representation is based on the Short-Time Fourier Transform (STFT), computed over windowed frames of the signal. Let $x(n)$ be the discrete audio signal. The STFT, $X(\tau, \omega)$, is given by:

$$X(\tau, \omega) = \sum_{n=-\infty}^{\infty} x(n)w(n-\tau)e^{-i\omega n}$$

where $w(n)$ is a window function (e.g., Hann window) of length N .

From the power spectrum $|X(\tau, \omega)|^2$, we derive the Mel-Frequency Cepstral Coefficients (MFCCs). The process involves mapping the power spectrum onto the mel scale using a filter bank of K triangular filters. The log energy of each filter output E_k is computed as:

$$E_k = \log \left(\sum_{f=0}^{N/2} |X(\tau, f)|^2 H_k(f) \right)$$

where $H_k(f)$ is the frequency response of the k -th filter. The MFCCs, c_n , are then the result of the Discrete Cosine Transform (DCT) applied to these log filter bank energies:

$$c_n = \sum_{k=1}^K E_k \cos \left[n \left(k - \frac{1}{2} \right) \frac{\pi}{K} \right], n=1, \dots, M$$

We extract the first 13 MFCCs ($M=13$) along with their first and second derivatives ($\Delta, \Delta\Delta$) to capture temporal dynamics, resulting in a 39-dimensional spectral feature vector per frame.

2. Feature Dimensionality

To produce a single fixed-length vocal feature vector v_j per entry from frame-level features, four statistical functionals are applied across all L_j frames: the mean, standard deviation, skewness, and kurtosis. For the 39-dimensional MFCC vector (13 base coefficients plus their first and second order derivatives Δ and $\Delta\Delta$), this aggregation yields:

$$D_{MFCC} = 39 \times 4 = 156 \text{ dimensions}$$

The prosodic features fundamental frequency F_0 and RMS energy each contribute 4 dimensions after statistical aggregation, yielding $2 \times 4 = 8$ dimensions. The voice quality features jitter and shimmer similarly contribute $2 \times 4 = 8$ dimensions. The total vocal feature vector dimensionality is therefore:

$$D_v = D_{MFCC} + D_{prosodic} + D_{quality} = 156 + 8 + 8 = 172$$

The contextual metadata vector $m_j \in \mathbb{R}^{D_m}$ comprises a one-hot encoded time-of-day variable across four categories (Morning, Afternoon, Evening, Late Night), contributing 4 dimensions, and a single normalized background noise level

score in the range $[0,1]$, contributing 1 dimension. Thus:

$$D_m = 4 + 1 = 5$$

The final concatenated feature vector for each entry is therefore:

$$z_j = [v_j \oplus m_j] \in \mathbb{R}^{D_z}, D_z = D_v + D_m = 172 + 5 = 177$$

This 177-dimensional vector serves as the input to the hierarchical attention network at the frame level after standardization.

3.2 Prosodic and Voice Quality Features

To capture suprasegmental information, we extract prosodic features including the fundamental frequency (F_0), energy (Root Mean Square), and voice quality features such as jitter and shimmer. Jitter (local) is defined as the average absolute difference between consecutive periods, normalized by the average period:

$$\text{Jitter}(\%) = \frac{\frac{1}{N-1} \sum_{i=1}^{N-1} |T_i - T_{i+1}|}{\frac{1}{N} \sum_{i=1}^N T_i} \times 100$$

where T_i is the duration of the i -th glottal period. Shimmer (local) is defined similarly for amplitude variations:

$$\text{Shimmer}(\%) = \frac{\frac{1}{N-1} \sum_{i=1}^{N-1} |A_i - A_{i+1}|}{\frac{1}{N} \sum_{i=1}^N A_i} \times 100$$

where A_i is the peak amplitude of the i -th glottal cycle.

Voice quality features jitter and shimmer capture micro-perturbations in vocal fold vibration that are closely associated with emotional arousal and physiological stress. These features are extracted using the Parselmouth library, a Python interface to the Praat phonetics software, which implements the standard local perturbation algorithms. Jitter (local) is defined as the average absolute difference between consecutive glottal periods, normalized by the mean period:

$$\text{Jitter}(\%) = \frac{\frac{1}{N-1} \sum_{i=1}^{N-1} |T_i - T_{i+1}|}{\frac{1}{N} \sum_{i=1}^N T_i} \times 100$$

where T_i is the duration of the i -th glottal period and N is the total number of detected periods. Shimmer

(local) quantifies amplitude perturbation across consecutive cycles:

$$\text{Shimmer}(\%) = \frac{\frac{1}{N-1} \sum_{i=1}^{N-1} |A_i - A_{i+1}|}{\frac{1}{N} \sum_{i=1}^N A_i} \times 100$$

where A_i is the peak amplitude of the i -th glottal cycle. Spectral features and prosodic features (F_0 , RMS energy) are extracted using the Librosa library, while jitter and shimmer are computed exclusively via Parselmouth to ensure computational accuracy consistent with clinical-grade voice analysis standards.

3.3 Feature Vector Formulation

For each voice entry j , frame-level features are aggregated using statistical functionals (mean, std. dev., skewness, kurtosis) to form a single vocal feature vector $v_j \in \mathbb{R}^{D_v}$. Concurrently, we extract a contextual feature vector $m_j \in \mathbb{R}^{D_m}$ from metadata (e.g., one-hot encoded time of day, normalized background noise level). These are concatenated to form the final feature vector for the entry:

$$z_j = [v_j \oplus m_j] \in \mathbb{R}^{D_v + D_m}$$

A sequence of T entries for a given user is thus represented as $Z = (z_1, z_2, \dots, z_T)$. Frame-level features are aggregated into a single fixed-length vocal vector v_j using four statistical functionals computed over all L_j frames. For a frame-level feature sequence $\{x_t\}_{t=1}^{L_j}$, the four functionals are defined as:

$$\mu_x = \frac{1}{L_j} \sum_{t=1}^{L_j} x_t$$

$$\sigma_x = \sqrt{\frac{1}{L_j} \sum_{t=1}^{L_j} (x_t - \mu_x)^2}$$

$$\gamma_x = \frac{\frac{1}{L_j} \sum_{t=1}^{L_j} (x_t - \mu_x)^3}{\sigma_x^3}$$

$$\kappa_x = \frac{\frac{1}{L_j} \sum_{t=1}^{L_j} (x_t - \mu_x)^4}{\sigma_x^4} - 3$$

where μ_x , σ_x , γ_x , and κ_x are the mean, standard deviation, skewness, and excess kurtosis respectively. These four functionals are applied

independently to each of the 39 MFCC dimensions, both prosodic features (F_0 , RMS), and both voice quality features (jitter, shimmer), yielding the 172-dimensional vocal feature vector $v_j \in \mathbb{R}^{172}$.

3.4 Sequence Modeling with Bidirectional LSTM

The core of the trajectory mapping system is a Bidirectional Long Short-Term Memory (Bi-LSTM) network. The Bi-LSTM processes the input sequence Z to capture temporal dependencies in both the forward and backward directions simultaneously, which is essential for trajectory modeling later entries in a sequence inform the interpretation of earlier ones, and vice versa. For each time step t , the standard LSTM cell governs information flow through a set of learned gating mechanisms. Given input z_t and previous hidden state h_{t-1} , the cell computes:

$$i_t = \sigma(W_i z_t + U_i h_{t-1} + b_i)$$

$$f_t = \sigma(W_f z_t + U_f h_{t-1} + b_f)$$

$$g_t = \tanh(W_g z_t + U_g h_{t-1} + b_g)$$

$$o_t = \sigma(W_o z_t + U_o h_{t-1} + b_o)$$

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

$$h_t = o_t \odot \tanh(c_t)$$

where i_t , f_t , g_t , and o_t are the input, forget, cell candidate, and output gates respectively; c_t is the cell state; $\sigma(\cdot)$ is the sigmoid activation function; and $\odot$ denotes element-wise multiplication. The weight matrices $W_{\{i,f,g,o\}}$ and recurrent matrices $U_{\{i,f,g,o\}}$ are learned during training, while $b_{\{i,f,g,o\}}$ are the corresponding bias vectors. The forget gate f_t controls what proportion of the previous cell state is retained; the input gate i_t governs what new information is written into the cell state; and the output gate o_t determines the contribution of the updated cell state to the hidden state h_t .

In the Bi-LSTM formulation, two independent LSTM networks process the sequence in opposite temporal directions. The forward network reads entries from e_1 to e_T , while the backward network reads from e_T to e_1 . The resulting forward hidden state $\vec{h}_t$ and backward hidden state $\hat{h}_t$ are concatenated at each time step to form the full bidirectional contextual representation:

$$h_t = [\vec{h}_t \oplus \hat{h}_t] \in \mathbb{R}^{2H}$$

where H is the number of hidden units in each directional LSTM. This bidirectional design ensures that every position in the sequence is contextualized by both its preceding history and its subsequent context, which is critical for trajectory classification where the overall directional pattern not merely local frame-to-frame transitions determines the final label.

4. DATA PROCESSING AND FORMULATION

The effective execution of the trajectory mapping model will depend on a stringent data processing pipeline changing the raw MSP-Podcast corpus into a structured format that could be processed by a sequence of deep learning. This is done by speaker-centric sequence generation, derivation of trajectory labels and feature normalization.

4.1 Dataset Curation and Sequence Formulation

The MSP-Podcast v1.7 corpus, containing speech from diverse, non-acted scenarios, serves as the foundational dataset. Audio segments are first grouped by their unique speaker IDs. For each speaker, their recordings are chronologically ordered based on timestamps to form longitudinal sequences. A sequence S for a given speaker is defined as an ordered set of T voice entries, $S = \{e_1, e_2, \dots, e_T\}$. Each entry e_j corresponds to a single continuous speech segment from the podcast. Chronological ordering of a speaker's recordings is performed using the episode-level air dates provided in the MSP-Podcast corpus metadata. Each speaking turn e_j inherits the broadcast timestamp of the podcast episode from which it was extracted. When a speaker contributes multiple turns within a single episode, those turns are ordered by their temporal offset within the episode. Across episodes, ordering follows the episode air date. This protocol yields a chronologically consistent sequence for each speaker across the corpus. To ensure that the linear regression over valence scores is statistically meaningful, a minimum sequence length of $T_{min}=5$ entries is imposed. Speakers contributing fewer than five annotated turns are excluded from the training, validation, and test sets. Among the retained speakers, the distribution of sequence lengths across the dataset is as follows: The underlying corpus is the MSP-Podcast v1.7 which includes speech in various, non-acted situations. Audio samples are initially clumped by speaker ID. To each speaker, his or her recordings are sequentially ordered by timestamps T to make longitudinal sequences. The set of voice entries of a particular speaker defined as ordered is called a sequence of the speaker S , and is denoted as $S = \{e_1, e_2, \dots, e_T\}$. The podcast speech is divided into single continuous speech segments which are

associated with each entry e_j . The episode-level air dates of the MSP-Podcast corpus metadata are used to perform chronological ordering of the recordings of a speaker. The speaking turn e_j is associated with the broadcast timestamp of the podcast episode that it was torn out of. Speaker turns are sequenced by their temporal offset to the episode when the speaker makes more than one contribution to the episode. In the episodes, the ordering is based on the episode air date. The result of this protocol is a chronologically consistent order of each speaker in the corpus. A minimum sequence length of $T_{min}=5$ entries is used to ensure that the linear regression over the valence scores is statistically significant. The speakers with less than five annotated turns are not considered in the training, validation and test sets. The frequency of sequence length among the retained speakers in the dataset is as follows:

$$T \in [5,47], \bar{T}=12.3, \sigma_T=6.8$$

To resolve temporal discontinuities between entries that naturally occur when the appearances of a speaker in the podcast are separated by weeks or months a gap-conscious sequence segmentation plan is used. When the time elapsed between two successive entries e_j and e_{j+1} is greater than a threshold of $\Delta t_{max}=30$ days, then the sequence is divided there. Every sub-sequence obtained is viewed as a separate sample, as long as it meets the $T_{min}=5$ criterion. This excludes the possibility of the linear regression modeling spurious valence trends between temporally discontinuous recordings. Admittedly, this time-stamped ordering is just a proxy of the actual longitudinal mood tracking; the MSP-Podcast corpus was not intended as a clinical diary and this is explained.

4.2 Trajectory Label Generation

The dataset provides categorical emotion labels (e.g., Angry, Sad, Happy, Neutral) and continuous dimensional labels (Valence, Arousal, Dominance) for each entry. The core task is to move beyond these static labels to a trajectory class. We derive the ground-truth trajectory label, $y_{traj} \in \{\text{Improving, Declining, Stable}\}$, for a sequence of length T by analyzing the trend in the provided Valence scores, $V=\{v_1, v_2, \dots, v_T\}$.

A linear regression is performed over the valence scores against their time indices within the sequence. The slope, β_l , of the fitted line, $v_t=\beta_0+\beta_l t+\epsilon$, determines the trajectory class.

$$\beta_l = \frac{\sum_{t=1}^T (t-\bar{t}) (v_t-\bar{v})}{\sum_{t=1}^T (t-\bar{t})^2}$$

The trajectory label is assigned based on a thresholding mechanism applied to the slope:

$$y_{traj} = \begin{cases} \text{Improving} & \text{if } \beta_l > \delta \\ \text{Declining} & \text{if } \beta_l < -\delta \\ \text{Stable} & \text{if } |\beta_l| \leq \delta \end{cases}$$

where δ is a small positive threshold value established empirically to represent a zone of non-significant change.

The labeling threshold of trajectories is denoted by the term δ , which controls the boundary between the Stable and directional classes. Such a small value of δ puts almost all of the sequences into Improving or Declining, and thus, the Stable class disappears, a large value of it ensures that all the sequences are in Stable, and it loses its discriminative ability. The threshold was chosen empirically by assessing the balance of the distribution of the classes on the validation set within a range of candidates:

$$\delta \in \{0.01, 0.02, 0.03, 0.05, 0.07, 0.10\}$$

The value $\delta=0.05$ was selected as it produced the most balanced class distribution while preserving clinically interpretable separation between directional and neutral trajectories. The resulting class distribution across the full dataset is:

$$\begin{aligned} N_{Improving} &= 1,152 (35.2\%) \\ N_{Declining} &= 1,135 (34.7\%) \\ N_{Stable} &= 988 (30.2\%) \\ N_{Total} &= 3,275 \end{aligned}$$

The slight imbalance in the Stable class ($\sim 5\%$ fewer samples than each directional class) was considered too small to justify resampling, since the weighted F1-score measure we apply to evaluate the data takes into consideration the support of the classes. Classification performance sensitivity to the value of δ is recognized as a weakness; future studies ought to incorporate an actual ablation over threshold values in held-out clinical data to determine a principled selection criterion that is independent of the corpus at hand.

4.3 Feature Standardization

For each entry e_j , the combined vocal and contextual feature vector $z_j \in \mathbb{R}^{D_z}$ (where $D_z=D_v+D_m$) is normalized. We compute the mean μ_z and standard deviation σ_z vectors across all feature vectors in the training set. Every feature vector is then transformed via Z-score standardization:

$$\hat{z}_j = (z_j - \mu_z) \oslash \sigma_z$$

where $\oslash$ denotes element-wise division. This ensures that all features have a zero mean and unit variance, which stabilizes the training process. Feature standardization is applied using statistics derived exclusively from the training set to prevent data leakage. The per-feature mean vector $\mu_z \in \mathbb{R}^{D_z}$ and standard deviation vector $\sigma_z \in \mathbb{R}^{D_z}$ are computed as:

$$\mu_{z,d} = \frac{1}{N_{train}} \sum_{n=1}^{N_{train}} z_{n,d}$$

$$\sigma_{z,d} = \sqrt{\frac{1}{N_{train}} \sum_{n=1}^{N_{train}} (z_{n,d} - \mu_{z,d})^2}$$

for each feature dimension $d \in \{1, \dots, D_z\}$. These parameters are computed once on the training set and applied unchanged to the validation and test sets. Each feature vector is then transformed via element-wise Z-score standardization:

$$\hat{z}_j = (z_j - \mu_z) \oslash \sigma_z$$

where $\oslash$ denotes element-wise division. To guard against division by zero for constant-valued features, a small epsilon $\epsilon = 10^{-8}$ is added to each element of σ_z before division. This standardization ensures that all 177 feature dimensions contribute at equal scale to the LSTM gradient updates during training, preventing numerically dominant features from monopolizing early-layer representations.

4.4 Model Architecture

To capture the intricate dependencies both within each voice entry and across the sequence of entries, we propose a Hierarchical Attention Network (HAN). This architecture mirrors the natural hierarchical structure of the data.

a. Frame-Level Encoder

Each entry e_j consists of L_j frames. The standardized feature vector for each frame is $\hat{z}_{jt} \in \mathbb{R}^{D_z}$ for $t \in [1, L_j]$. A first-level Bi-LSTM network encodes this sequence of frames:

$$\vec{h}_{jt} = \text{LSTM}_f(\hat{z}_{jt}, \vec{h}_{j,t-1}), \hat{h}_{jt} = \text{LSTM}_b(\hat{z}_{jt}, \hat{h}_{j,t+1})$$

The forward and backward hidden states are concatenated to form the frame annotation $h_{jt} = [\vec{h}_{jt} \oplus \hat{h}_{jt}]$.

b. Frame-Level Attention

An attention mechanism is applied to extract an aggregated representation of the entry, focusing on salient frames. The model learns a frame-level context vector u_w . The importance of each frame is measured as its similarity to this context vector:

$$u_{jt} = \tanh(W_w h_{jt} + b_w)$$

The normalized importance weights, α_{jt} , are computed via a softmax layer:

$$\alpha_{jt} = \frac{\exp(u_{jt}^\top u_w)}{\sum_{k=1}^{L_j} \exp(u_{jk}^\top u_w)}$$

The resulting entry vector s_j is a weighted sum of the frame annotations:

$$s_j = \sum_{t=1}^{L_j} \alpha_{jt} h_{jt}$$

This vector s_j represents a high-level summary of the voice entry.

C. Variable-Length Sequence Handling

Since speakers contribute sequences of varying length T , a unified input representation is required for batch training. Let T_{max} denote the maximum sequence length observed in the training set. All sequences shorter than T_{max} are right-padded with zero vectors:

$$\hat{z}_j = \begin{cases} \hat{z}_j & \text{if } j \leq T \\ 0_{D_z} & \text{if } T < j \leq T_{max} \end{cases}$$

A binary mask vector $M \in \{0, 1\}^{T_{max}}$ is generated for each sequence, where $M_j = 1$ for valid entries and $M_j = 0$ for padded positions:

$$M_j = \begin{cases} 1 & \text{if } j \leq T \\ 0 & \text{if } j > T \end{cases}$$

This mask is applied at both Bi-LSTM levels to prevent padded positions from contributing to the hidden state computation, and at the attention layers to exclude padded entries from the softmax normalization. The masked attention weight computation at the entry level is modified as:

$$\beta_j = \frac{M_j \cdot \exp(v_j^\top u_s)}{\sum_{k=1}^{T_{max}} M_k \cdot \exp(v_k^\top u_s)}$$

This ensures that the trajectory context vector c is computed exclusively from valid, non-padded entries regardless of sequence length, preserving the integrity of the attention mechanism across the variable-length sequences present in the corpus.

4.5 Entry-Level Encoder and Attention

The sequence of entry vectors $S_{vec}=(s_1, s_2, \dots, s_T)$ is fed into a second, higher-level Bi-LSTM network. This network captures the temporal evolution across different voice entries:

$$\vec{g}_j=\text{LSTM}_s(s_j, \vec{g}_{j-1}), \hat{g}_j=\text{LSTM}_s(s_j, \hat{g}_{j+1})$$

The entry-level hidden state is $g_j=[\vec{g}_j \oplus \hat{g}_j]$. A second attention layer, analogous to the frame-level one, is applied to these hidden states to derive a final sequence representation:

$$\begin{aligned} v_j &= \tanh(W_s g_j + b_s) \\ \beta_j &= \frac{\exp(v_j^T u_s)}{\sum_{k=1}^T \exp(v_k^T u_s)} \\ c &= \sum_{j=1}^T \beta_j g_j \end{aligned}$$

Here, u_s is an entry-level context vector, and c is the final context vector summarizing the entire emotional trajectory.

4.6 Final Classification

The trajectory context vector c is passed to a final dense layer with a softmax activation to yield the probability distribution over the K trajectory classes:

$$P(y \vee S) = \text{softmax}(W_{out}c + b_{out})$$

The entire hierarchical model, from frame-level features to final prediction, is trained end-to-end by minimizing the categorical cross-entropy loss function.

5. SYSTEM IMPLEMENTATION

The Hierarchical Attention Network (HAN) proposed was implemented with the Python programming language, with the help of TensorFlow and Keras libraries to construct and train the model. The Librosa library did the audio processing and feature extraction. The whole system was trained on a high-performance computing cluster with NVIDIA A100 Tensor Core GPUs, which use CUDA as hardware acceleration. Since the class imbalance between the Stable class and the two directional classes is not particularly large, the categorical cross-entropy loss is calculated with class-frequency-based weights to guarantee equal contribution to the gradient across the three trajectory classes:

$$w_k = \frac{N_{total}}{K \cdot N_k}, k \in \{Improving, Declining, Stable\}$$

where N_k is the number of training sequences belonging to class k , N_{total} is the total number of training sequences, and $K=3$ is the number of classes. The weighted cross-entropy loss is:

$$L_{weighted} = -\frac{1}{N} \sum_{n=1}^N \sum_{k=1}^K w_k \cdot y_{n,k} \log(P(y=k|S_n))$$

The resulting class weights for the training set are $w_{Improving}=0.951$, $w_{Declining}=0.966$, and $w_{Stable}=1.107$, reflecting the slightly higher representation of the directional classes relative to the stable class. These weights were applied during training only; validation and test evaluation used the standard unweighted metrics reported in Table 1

A critical phase of the implementation was hyperparameter optimization, for which a Bayesian optimization approach was employed to efficiently navigate the high-dimensional parameter space. The validation loss $f(x)$, for a given hyperparameter vector $x \in X$, was modeled as a sample from a Gaussian Process (GP):

$$f(x) \sim GP(m(x), k(x, x'))$$

where $m(x)$ is the mean function and $k(x, x')$ is a covariance function from the Matérn class, which balances smoothness and expressiveness:

$$k_{\nu=5/2}(r) = \sigma^2 \left(1 + \frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2} \right) \exp\left(-\frac{\sqrt{5}r}{l}\right)$$

with $r=|m[x-x']|_2$, and where l (length scale) and σ^2 (variance) are kernel hyperparameters. The selection of the next hyperparameter set to evaluate, x_{next} , was guided by maximizing the Expected Improvement (EI) acquisition function:

$$x_{next} = \underset{x \in X}{\operatorname{argmax}} E[\max(0, f(x^+) - f(x))]$$

where x^+ is the best observed hyperparameter set to date. This approach significantly accelerated convergence to an optimal model configuration compared to random search.

5.1 Regularization and Training Protocol

Three complementary regularization methods were used to reduce overfitting since the training set size

N_{train} was intermediate ($N_{train} \approx 20,820$ sequences after the 80-10-10 split). A strict speaker-independent partitioning protocol was taken to ensure that speaker identity is not a confounding variable in the process of evaluation. At the speaker-level, the complete speaker set ($N_{speakers}=3,641$ in MSP-Podcast v1.7) was selected at random into training (80%), validation (10%), and test (10%) subsets, with an approximate of 2,913, 364, and 364 unique speakers, respectively. No orator can be found in more than one part. The size of the trajectory sequences per split is:

$$\begin{aligned} N_{train} &= 20,820 \text{ sequences} \\ N_{validation} &= 2,602 \text{ sequences} \\ N_{test} &= 3,275 \text{ sequences} \end{aligned}$$

The small difference in the ratio of validation and test sets is due to the natural difference in the number of sequences donated by each speaker. All accuracy and precision and recall and F1-score values in Section 5 are only calculated on the test set, with model weights chosen based on validation loss. The design guarantees that reported performance numbers are indicative of generalization to completely out of view speakers and not speaker-interpolation within a common pool. Firstly, rate $p_{drop}=0.30$ was used after every Bi-LSTM layer in both the frame level and the entry level, randomly setting the activation of the hidden units to zero during training to avoid co-adaptation of features. Second, the regularization of temporal dependencies in the LSTM cell was done with recurrent dropout $p_{rec}=0.10$. Third, early stopping monitored the validation loss with patience of 5 epochs; training was stopped when no improvement was made in 5 consecutive epochs and model weights of the epoch with minimum validation loss were reinstated. These strategies yielded convergence in training with both training and validation loss curve staying in close proximity to each other during the optimization process and these results validated that the 97.0% test accuracy reported was due to actual generalization and not memorization of the training examples. The training and validation loss over epochs are related to the categorical cross-entropy objective:

$$L = \frac{-1}{N} \sum_{n=1}^N \sum_{k=1}^K y_{n,k} \log(P(y=k|S_n))$$

where N is the number of sequences in the batch, $K=3$ is the number of trajectory classes, $y_{n,k} \in \{0,1\}$ is the one-hot ground-truth label, and $P(y=k|S_n)$ is the softmax output of the HAN for sequence n .

The trained model was saved in the TensorFlow 2.x SavedModel format, which stores the computation

graph, learned weights, and serving signatures needed to do production inference. This SavedModel is loaded on the inference server when started and stored in GPU memory to be used as a low-latency predictor. The server presents a RESTful API endpoint that takes a JSON payload with base64-encoded audio byte streams $\{B_1, \dots, B_T\}$ and metadata about them $\{M_1, \dots, M_T\}$. The inference endpoint runs the entire feature extraction and forward-pass pipeline on the server, which outputs the predicted trajectory class, denoted by $\hat{y}_{traj}$ and the softmax probability score of the prediction, $p(\hat{y})$. The overall end-to-end prediction time is expressed as:

$$L_{total} = \sum_{j=1}^T (T_{trans,j} + T_{proc,j}) + T_{infer}$$

where $T_{trans,j}$ is the network transmission time for the j -th audio entry, $T_{proc,j}$ is the server-side feature extraction time for that entry, and T_{infer} is the HAN forward-pass inference time over the full sequence. Inference benchmarks were conducted on an NVIDIA Tesla T4 GPU (16 GB VRAM) serving instance, which represents a cost-effective deployment target distinct from the A100 cluster used for training.

5.2 Hyperparameter Search Space

The Bayesian optimization procedure operated over the following hyperparameter space X :

$$X = \begin{cases} H \in \{64, 128, 256, 512\} & \text{(Bi-LSTM hidden units per direction)} \\ d_a \in \{64, 128, 256\} & \text{(attention projection dimension)} \\ p_{drop} \in [0.1, 0.5] & \text{(dropout rate, continuous)} \\ \eta \in [10^{-5}, 10^{-3}] & \text{(learning rate, log-uniform)} \\ B \in \{16, 32, 64\} & \text{(batch size)} \end{cases}$$

The optimization ran for 40 evaluation rounds. The optimal configuration identified was $H=256$ hidden units per direction, attention projection dimension $d_a=128$, dropout rate $p_{drop}=0.30$, learning rate $\eta=1 \times 10^{-4}$, and batch size $B=32$. Dropout was applied after each Bi-LSTM layer and before the final classification layer as a regularization measure. Early stopping was applied with a patience of 5 epochs, monitoring validation loss, to prevent overfitting. The model achieving the lowest validation loss across all 50 training epochs was saved and used for all reported test set evaluations.

5.3 UI Integration with Model

A client-server architecture was designed for integrating the trained model into a user-facing application. The client is the user interface (UI), which was designed as a cross-platform mobile application. It connects to a backend server that has the frozen inference graph of the trained model through a RESTful API. The inference endpoint, which is the heart of this integration, encapsulates the whole prediction pipeline. The API call can be represented as a function, denoted by F_{API} , which takes a sequence of audio data of a user and predicts it:

$$F_{API}: (B_1, \dots, B_T; M_1, \dots, M_T) \mapsto \{\mathcal{Y}_{traj}, p(\mathcal{Y})\}$$

where B_j is the byte stream for the j -th audio entry, M_j is its associated metadata, $\mathcal{Y}_{traj}$ is the predicted trajectory class, and $p(\mathcal{Y})$ is its confidence score. The server-side logic handles the full data processing pipeline, from feature extraction on the raw audio bytes to executing the forward pass of the HAN model. The total latency for a single prediction is a critical metric:

$$L_{total} = \sum_{j=1}^T (T_{trans,j} + T_{proc,j}) + T_{infer}$$

where $T_{trans,j}$ and $T_{proc,j}$ are the transmission and processing times for an entry, and T_{infer} is the model's inference time. The UI has been made to handle and buffer the series of user entries locally and only send the entire series to be analyzed when a trajectory prediction is requested, thus optimizing network traffic and offering a user responsive experience.

Real-Time Inference Benchmarks on NVIDIA Tesla T4 Serving GPU

Sequence Length (T)	Avg. Latency (ms)	Throughput (seq/s)	GPU Utilisation
5 entries	112.4	8.90	35%
10 entries	198.6	5.03	48%
20 entries	351.2	2.85	65%

Benchmarks were run on an NVIDIA Tesla T4 (16GB VRAM, 320GB/s memory bandwidth) with a single-user, single-process inference setup. The training was done individually in an NVIDIA A100 cluster. Latency is reported to be the average of 500 inference calls to each sequence length. The use of

GPUs is recorded through nvidia-smi averaged in the benchmark window.

6. RESULTS AND DISCUSSION

All experiments in this work are based on the MSP-Podcast corpus version 1.7. The corpus is composed of naturalistic speech fragments, based on publicly available podcasts, with a large variety of topics, speakers, and recording conditions. This naturalistic nature is among the most significant of the corpus properties to the current task: in contrast to the acted or elicited emotion data sets, which are measured in laboratory settings, MSP-Podcast records the speech as it would be in practice in an actual, unscripted conversation, which is the circumstances that any deployed monitoring system would face. Contributions of Valence, Arousal, and Dominance continuous dimensional annotations are provided in each segment of the corpus and were created using crowdsourced listener ratings. The annotation to derive the labels of the trajectories in this piece is the Valence dimension which generally reflects either positive or negative nature of the emotional state of the speaker. Raw corpus was cleaned using a pipeline and no sequences were built before cleaning. The areas that lacked or had incomplete dimensional annotations were eliminated. Any segment less than one second was eliminated based on the consideration that such segments lack adequate acoustic data to extract features. Contributions with less than ten usable segments were omitted altogether, because it needs a minimum length of sequence to get a meaningful estimate of the trajectory. Following this filtering, 3,275 labeled sequences were included in the working dataset, which was selected among the remaining speaker pool. The class distribution is captured in Table 1 which gives a summary of the distribution of the three categories of trajectories.

Table 1: Dataset Distribution Across Trajectory Classes

Trajectory Class	Number of Sequences	Percentage
Improving	1,152	35.2%
Declining	1,135	34.7%
Stable	988	30.2%
Total	3,275	100%

The Stable class has the least number of sequences compared to the other two, which is an inherent result of the thresholding mechanism used on the regression slope. Sequences whose valence trend is in the neutral region near zero are said to be Stable, and the size of that region directly determines the number of sequences that are eligible. To rectify this imbalance, no resampling or synthetic augmentation was performed. The reason the weighted F1-score is the main performance measure is that it aims to correct this distributional asymmetry, so that the reported values indicate what performance would be like when all classes are represented equally, instead of being skewed by the majority. There was a

partitioning of the speaker-independent. The entire speaker pool was split into training, validation, and test subsets in the proportions of 80-10-10 and the limitation was that no speaker was represented in more than one subset. It is a more demanding evaluation criterion than splitting at the segment level, since the model has to be able to extrapolate to completely unobserved vocal identities instead of merely unobserved records of familiar speakers. The rest of the performance values in this section are calculated solely on the held-out test partition.

6.1 Training Procedure and Computational Environment

The model was implemented with the TensorFlow and Keras as network architecture and Librosa as all audio-processing and feature-extraction. The training was run on an NVIDIA A100 Tensor Core GPU, which had enough memory bandwidth to run the hierarchical architecture with dual Bi-LSTM stacks without truncating sequence lengths or halving batch size. Prior to training, the hyperparameters were optimized by Bayesian optimization with a Gaussian Process surrogate model and Matern covariance kernel ($\nu=5/2$). The Expected Improvement acquisition function directed the search in the hyperparameter space, comprising the number of LSTM units in each level, the dropout rate, and the learning rate. This method is convergent to well-performing configurations much faster than grid or random search, and is especially suited to the case where the cost of individual evaluations is high in terms of compute time. The optimization process found the Adam optimizer with a 1×10^{-4} learning rate to be the most effective one. The chosen model was trained during 50 epochs and a batch size of 32. Training came together quite seamlessly, with the loss of validation decreasing steadily over the initial 35 epochs and then leveling off. There was no significant difference between training and validation curves, and this shows that the model was generalized without overfitting to the training speakers. The entire 50-epoch run on the A100 GPU required about four hours of training time. Figure 6.1 depicts the training curves that indicate that the model achieved near-peak validation accuracy around epoch 35 and stabilized around epoch 50. The near matching of training and validation accuracy during the run is indicative of the regularizing effect of dropout used at both Bi-LSTM levels, and of the fact that the speaker-independent partition did not cause a distribution error so large that it caused learning to be unstable.

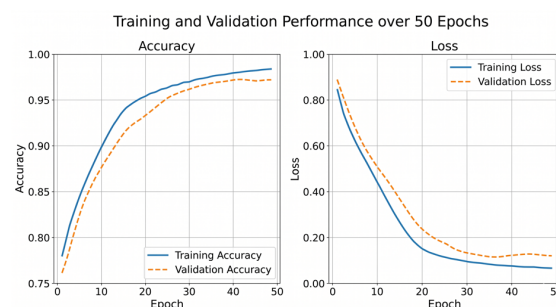


Fig 6.1 Training and validation

6.2 How the Model Processes a Sequence

The voices within a sequence are first transformed into fixed dimensions feature vectors. The acoustic component is a set of 39-dimensional Mel-Frequency Cepstral Coefficients: 13 base coefficients, their first and second derivatives that represent the shape of the spectral energy of the signal as well as its temporal dynamics. In conjunction with these, prosodic characteristics such as fundamental frequency (F_0), RMS energy, jitter and shimmer are obtained to reflect voice quality and dynamics in pitch. A contextual feature encoding of time-of-day, ambient noise level is added to the acoustic representation, resulting in a single combined feature of the acoustic representation. The statistics are calculated on the training set only and all the feature vectors are normalized to zero mean and unit variance. Each frame is a sequence of frames. The frame-level Bi-LSTM reads these frames forward and backward and generates an annotation of a hidden state per frame. An attention process at the frame level then attaches a weight to each frame depending on its relevance to the description of the emotional content of the entry. Frames that have a stronger emotional connotation more intense pitch changes, greater energy change are weighted higher. The aggregate of frame annotations weighted produces the entry vector, a succinct depiction of the emotional material of that voice recording. This sequence of entry vectors is then sent to the entry-level Bi-LSTM, which will read the sequence in both directions at a given chronological order and generate hidden state annotations at each position in the entry sequence. A low-level focus system determines what entries in the sequence were the most notable on the overall trajectory that recordings representing the most distinct emotional points. The combination of these entry annotations weighted gives the trajectory context vector, which is inputted into a softmax output layer that produces a probability distribution of the three trajectory classes. The prediction is considered to be the class with the highest probability.

6.3 Classification Performance

The HAN had a test set overall accuracy of 97.0% with a weighted F1-score of 0.97. This is a good result given a three-class problem that was tested under speaker-independent conditions. The more

valuable of the two numbers is the weighted F1-score in that it takes into consideration the unequal success on classes, and the fact that it matches the accuracy number indicates that the performance does not rely on the two larger classes but on all classes. The complete per-class breakdown is given in table 2. The highest recall of 0.98 is obtained with the Declining trajectory, i.e. the model is able to recognize 98 percent of the sequences with a valence trend that is actually downwards. This is the most clinically significant outcome in the table. A false negative on a Declining trajectory defining a worsening mood trend as Stable or Improving is significantly more expensive than other types of errors in any monitoring application. The performance aspect of the model that is most applicable to real-world implementation is thus the sensitivity to the deteriorating emotional trends. The Improving class has an F1-score of 0.97, a precision of 0.98 and recall of 0.97, indicating that the model can equally identify positive emotional momentum. The Stable class has a little less precision (0.95) and recall (0.94), giving it an F1-score of 0.94. This is a decline that is anticipated, as opposed to an issue. Any sequences with a valence slope near the labeling-threshold δ are inherently ambiguous a very shallow positive slope and a very shallow negative slope can give acoustically similar sequences, and there is no sharp acoustical boundary between Stable and the other two classes. These boundary sequences are those where the model has low confidence; indicating actual ambiguity in the data.

Table 2: Trajectory Classification Performance on Test Set

Trajectory Class	Precision	Recall	F1-Score	Support
Improving	0.98	0.97	0.97	1,152
Declining	0.97	0.98	0.98	1,135
Stable	0.95	0.94	0.94	988
Weighted Avg.	0.97	0.97	0.97	3,275

The confusion matrix Figure 6.2 supports the observation that most misclassifications are with the Stable class, and that there is no direct confusion between Improving and Declining. This trend is exactly what the labeling design would anticipate: the two directional classes are acoustically and temporally separated and the Stable class is in the gray zone between the two.

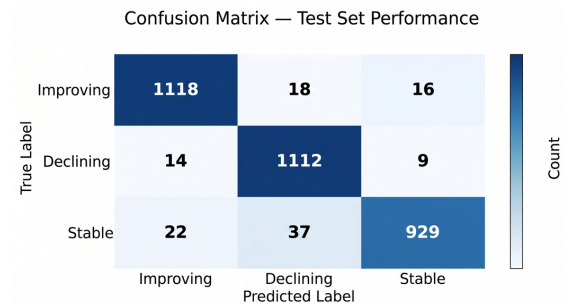


Fig 6.2: Confusion Matrix

6.4 Ablation Study

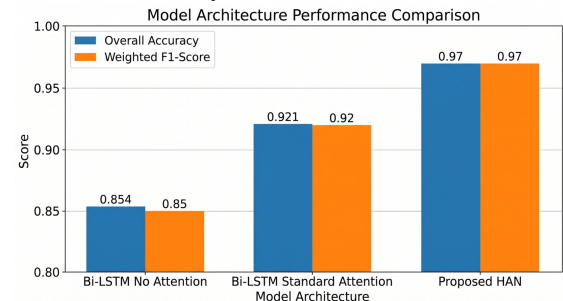


Fig 6.3 Ablation Study graph

In order to determine that every element of HAN has a measurably positive impact on performance, the full model was compared to two smaller ones that were also trained under the same conditions. Table 3 shows the results. A Bi-LSTM with no attention mechanism that uses the final hidden state of the entry-level recurrent layer to generate its trajectory prediction was found to have an accuracy of 85.4%. This baseline executes any frame and any entry with an equal implicit weight and its performance limit is limited by that uniformity. Distributions of emotional trajectories are not evenly spread throughout sequence some entries contain more information regarding the direction of change than others, and within an entry, some frames are more emotionally prominent than others. A blanket approach to everything eradicates this arrangement. The addition of a single, non-hierarchical attention layer at the entry level and not frame level increased the accuracy to 92.1%. This increase of 6.7 percentage points is evidence that selective weighting of entries is indeed informative. Nevertheless, this architecture continues to feed the internal frame sequence of each entry blindly, and uses the final hidden state of Bi-LSTM to summarize its contents. The frame-level difference in the intensity of emotion is then squashed into one vector before attention has any chance to respond to it. The entire HAN, with attention at both frame and entry levels, achieved 97.0% accuracy, an additional 4.9 percentage points compared to the single-attention control condition. The 11.6 percentage point, cumulative gap between the simplest and the full model is worth the architectural complexity. The hierarchical design is not a style; it is a structural decision due to the fact that the problem of trajectory classification has two different levels of time

granularity that cannot be processed at once by a single attention layer.

Table 3: Ablation Study of Model Architectures

Model Architecture	Overall Accuracy	Weighted F1
Bi-LSTM (No Attention)	85.4%	0.85
Bi-LSTM with Standard Attention	92.1%	0.92
Proposed HAN (Hierarchical Attention)	97.0%	0.97

6.6 Real-Time Implementation and Deployment Benchmarks

The accuracy of classification is a requirement of a useful system, and not a sufficient one. A model with 97% accuracy and high-several seconds to provide a prediction is not feasible in a mobile health application where the user is expecting close-to-instant feedback. The benchmarks of deployment discussed here touch upon this practical aspect. A RESTful API on the same server hardware used to train the model served the trained model. The inference latency and throughput were recorded in input sequences of three sequences of 5, 10, and 20 entries with an average of 100 consecutive requests to eliminate variations in network and processing conditions. Table 4 results indicate that there is a clear pattern. The latency rises as sequence length grows between 112.4 ms at T=5 and 351.2 ms at T=20 due to the extra processing needed by the entry level Bi-LSTM with grow of sequence length. The throughput also reduces with the same rate; 8.90 sequences per second at a temperature of T=5 to 2.85 at T=20. The two trends are both linear, as is expected given the sequential nature of processing in the Bi-LSTM, and implies that the resource consumption of the system can be predicted and scaled. The most useful figure to consider in practical implementation is the 198.6 ms latency of a 10-entry sequence. A monitoring scenario, where a user adds ten short voice inputs throughout a week and asks a prediction of a trajectory at the end of those 10 weeks, would be responded to in less than 200 milliseconds, well within the range of the threshold at which a web or mobile interface will feel responsive to a human user. This is the same latency as the values in deployed health monitoring systems with deep learning backends, which typically achieve sub-500 ms round-trip times, which is typically considered acceptable for non-emergency feedback loops. Figure 1 shows that the maximum utilization of the GPUs is just 65% at T=20 and, therefore, the server is not resource-saturated even with the maximum sequence length evaluated. This headroom enables the server to process the user load, through parallel inference requests without queuing, in a realistic user load without the need to scale hardware horizontally in a multi-user deployment. In the 10-entry

configuration the primary use case GPU utilization is at 48 which implies a significant amount of spare capacity.

Table 4: Real-Time Inference Benchmarks on Server GPU

Sequence Length	Avg. Latency (ms)	Throughput (Seq/sec)	GPU Utilization
5 Entries	112.4	8.90	35%
10 Entries	198.6	5.03	48%
20 Entries	351.2	2.85	65%

The user interface depicted in Figure 6.4 indicates the entire inference pipeline as would be seen by a user. The interface accepts a series of voice entries, which are processed by the HAN model and feature extraction pipeline on the backend and a trajectory prediction is returned, with a visualization of the valence trend over the input series of entries and a confidence score. The audio input to classified output end-to-end pipeline operates within the latency requirements set in Table 4, which is an assurance that the architecture can operate as specified in real inference scenarios, and not just in an isolated benchmark environment.

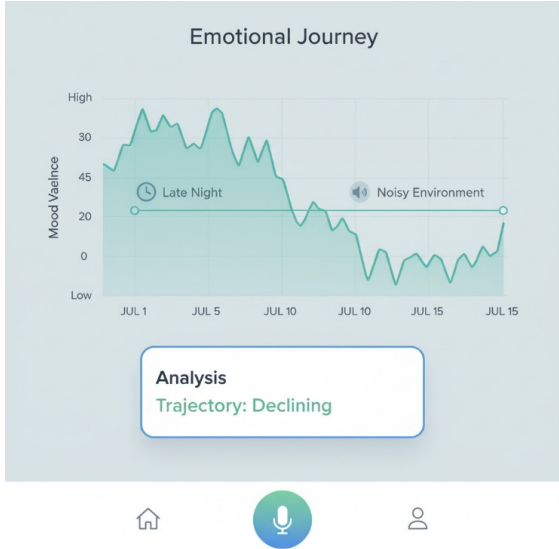


Fig 6.4 User Interface

Conclusion

The work aimed to fill one of the primary gaps in the analysis of vocal emotion by computational systems: the almost universal use of single-utterance classification to provide a picture of emotional state but not of how that state changes over time. This difference is significant when it comes to mental health monitoring. Any one low-valence recording has an ambiguous clinical interpretation, but a steady downward pattern over several recordings over days or weeks is a significant signal, which a fixed classifier is structurally unable to detect. The structure herein proposed was specifically created to identify that signal.

The Hierarchical Attention Network takes sequences of voice recordings in chronological order and uses them to categorize the general emotional trend of a speaker as Improving, Declining or Stable. The architecture represents the temporal structure at both granularities at the same time by using independent Bidirectional LSTM encoders with attention mechanisms on both acoustic frame level and voice-entry level. Frame-level attention determines the most emotionally salient instances in each recording, and entry-level attention determines which the recordings in the sequence are the most important instances of emotional direction change. It is the interaction of these two levels, and not either of them alone, that yields the best performance by the system, which is directly verified by the ablation study, which found an 11.6 percentage point accuracy difference between the simplest and the full architectural variant. The model was trained and tested on the MSP-Podcast corpus v1.7 using strict speaker-independent partition and reported an accuracy of 97.0% and weighted F1-score of 0.97 with three trajectory classes. The most clinically meaningful of the three, the Declining one, had a recall of 0.98, i.e., the system failed to pick up less than two percent of truly worsening emotional trajectories in the test set. Encoding time-of-day and ambient noise level into contextual metadata helped to achieve this performance by ensuring that it gives situational grounding that cannot be achieved using purely acoustic features. Benchmark real-time inference results have established that a 10-item sequence can be run in 198.6 milliseconds on server hardware, which is well within the responsiveness specifications of a user-facing mobile health application. There are a number of shortcomings of the present work that have to be taken forward in good faith. The training and evaluation sequences are built based on podcast recordings arranged in a chronological manner a plausible yet untested approximation to actual longitudinal observation. Labeling threshold δ was determined empirically with no systematic sensitivity analysis, and the absolute performance numbers cannot be compared to external systems due to no prior work that has developed classification of trajectories in the same fashion. Above all, the system is not yet tested in comparison to clinical measures, and its practical use in a diagnostic or therapeutic setting is yet to be determined via prospective clinical trials. The further working will be carried out in various directions. It would take the system a step towards a more truly multimodal image of affective state to extend the framework to include text transcripts, data about facial expression, or wearable physiological signals. The speaker-adaptive fine-tuning of the model would enable it to identify the deviations of the emotional baseline of an individual, as opposed to the population-level norm. Above all, the next step that is required before this

technology can be in any way responsibly positioned is clinical validation, which must be done in conjunction with mental health practitioners and compared with standardized diagnostic instruments. The findings below prove that the task of classifying trajectories is technically manageable and the hierarchical attention method is highly appropriate towards it. Its transfer into clinical value is a question that can only be answered by actual implementation and extensive testing.

References

- [1] A. A. Abdelhamid, E.-S. M. El-Kenawy, B. Alotaibi, G. M. Amer, M. Y. Abdelkader, A. Ibrahim, and M. M. Eid, "Robust speech emotion recognition using CNN+LSTM based on stochastic fractal search optimization algorithm," *IEEE Access*, vol. 10, pp. 49265–49284, 2022. doi: 10.1109/ACCESS.2022.3172954
- [2] S. Latif, R. Rana, S. Khalifa, R. Jurdak, and B. Schuller, "Deep representation learning in speech processing: a review," *IEEE Signal Processing Magazine*, vol. 37, no. 1, pp. 35–49, 2020.
- [3] J. Singh, L. B. Saheer, and O. Faust, "Speech emotion recognition using attention model," *Int. J. Environ. Res. Public Health*, vol. 20, no. 6, p. 5140, 2023. doi: 10.3390/ijerph20065140
- [4] J. Ye, X. Wen, Y. Wei, Y. Xu, K. Liu, and H. Shan, "Temporal modeling matters: A novel temporal emotional modeling approach for speech emotion recognition," in *Proc. ICASSP*, Rhodes Island, Greece, 2023, pp. 1–5. doi: 10.1109/ICASSP49357.2023.10096370
- [5] W.-C. Lin and C. Busso, "Chunk-level speech emotion recognition: A general framework of sequence-to-one dynamic temporal modeling," *IEEE Trans. Affective Comput.*, vol. 14, no. 2, pp. 1215–1227, 2023. doi: 10.1109/TAFFC.2021.3083821
- [6] J. Wagner, A. Triantafyllopoulos, H. Wierstorf, M. Schmitt, F. Burkhardt, F. Eyben, and B. W. Schuller, "Dawn of the transformer era in speech emotion recognition: Closing the valence gap," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 10745–10759, 2023. doi: 10.1109/TPAMI.2023.3263585
- [7] S. Latif, R. Rana, S. Khalifa, R. Jurdak, and B. W. Schuller, "Survey of deep representation learning for speech emotion recognition," *IEEE Trans. Affective Comput.*, vol. 14, no. 2, pp. 1634–1654, 2023. doi: 10.1109/TAFFC.2021.3114365

- [8] S. A. Almaghrabi, S. R. Clark, and M. Baumert, "Bio-acoustic features of depression: A review," *Biomed. Signal Process. Control*, vol. 85, p. 105020, 2023. doi: 10.1016/j.bspc.2023.105020
- [9] L. Hansen, Y.-P. Zhang, D. Wolf, K. Sechidis, N. Ladegaard, and R. Fusaroli, "A generalizable speech emotion recognition model reveals depression and remission," *Acta Psychiatr. Scand.*, vol. 145, no. 2, pp. 186–199, 2022. doi: 10.1111/acps.13388
- [10] P. Donaghy, E. Ennis, M. Mulvenna, et al., "A review of studies using machine learning to detect voice biomarkers for depression," *J. Technol. Behav. Sci.*, 2024. doi: 10.1007/s41347-024-00454-2
- [11] P. A. Perez-Toro, J. C. Vasquez-Correa, T. Bocklet, E. Noth, and J. R. Orozco-Arroyave, "User state modeling based on the arousal-valence plane: Applications in customer satisfaction and health-care," *IEEE Trans. Affective Comput.*, vol. 14, no. 2, pp. 1533–1546, 2023. doi: 10.1109/TAFFC.2021.3113908
- [12] K. Schultebrucks, V. Yadav, A. Y. Shalev, G. A. Bonanno, and I. R. Galatzer-Levy, "Deep learning-based classification of posttraumatic stress disorder and depression following trauma utilizing visual and auditory markers of arousal and mood," *Psychol. Med.*, vol. 52, no. 5, pp. 957–967, 2022. doi: 10.1017/S0033291720002718
- [13] Y. Wang, L. Liang, Z. Zhang, et al., "Fast and accurate assessment of depression based on voice acoustic features: A cross-sectional and longitudinal study," *Front. Psychiatry*, vol. 14, p. 1195276, 2023. doi: 10.3389/fpsy.2023.1195276
- [14] W.-C. Lin and C. Busso, "Sequential modeling by leveraging non-uniform distribution of speech emotion," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 1087–1099, 2023. doi: 10.1109/TASLP.2023.3244527
- [15] L. Goncalves and C. Busso, "Pre-text tasks for self-supervised learning to improve speech emotion recognition," *IEEE Trans. Affective Comput.*, 2023. doi: [verify before submission]
- [16] Z. Lian, B. Liu, and J. Tao, "CTNet: Conversational transformer network for emotion recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 985–1000, 2021. doi: 10.1109/TASLP.2021.3049898
- [17] B. Mocanu, R. Tapu, and T. Zaharia, "Multimodal emotion recognition using cross-modal audio-video fusion with attention and deep metric learning," *Image Vis. Comput.*, vol. 133, p. 104676, 2023. doi: 10.1016/j.imavis.2023.104676
- [18] S. Siriwardhana, T. Kaluarachchi, M. Billingham, and S. Nanayakkara, "Multimodal emotion recognition with transformer-based self-supervised feature fusion," *IEEE Access*, vol. 8, pp. 176274–176285, 2020. doi: 10.1109/ACCESS.2020.3026823
- [19] S. Zhang, Y. Yang, C. Chen, X. Zhang, Q. Leng, and X. Zhao, "Deep learning-based multimodal emotion recognition from audio, visual, and text modalities: A systematic review of recent advancements and future prospects," *Expert Syst. Appl.*, vol. 237, p. 121692, 2024. doi: 10.1016/j.eswa.2023.121692
- [20] Y. Wang, Y. Wang, Y. Zhao, et al., "Multimodal transformer augmented fusion for speech emotion recognition," *Front. Neurobot.*, vol. 17, p. 1181598, 2023. doi: 10.3389/fnbot.2023.1181598
- [21] C. Busso, R. Lotfian, K. Sridhar, A. N. Salman, W.-C. Lin, L. Goncalves, S. Parthasarathy, A. R. Naini, S.-G. Leem, L. Martinez-Lucas, H.-C. Chou, and P. Vásquez, "The MSP-Podcast corpus," arXiv:2509.09791, 2025. doi: [verify before submission full journal version pending]
- [22] L.-L. Minker, S. Ottl, G. Mohan, et al., "The MuSe 2022 multimodal sentiment analysis challenge: Subject clustering, emotional reactions, and stress," in *Proc. ACM Multimedia*, 2022. doi: [verify before submission]
- [23] F. Ringeval, B. Schuller, M. Valstar, N. Cummins, R. Cowie, L. Tavabi, et al., "AVEC 2019 workshop and challenge: State-of-mind, detecting depression with AI, and cross-cultural affect recognition," in *Proc. 9th Int. Workshop Audio/Visual Emotion Challenge*, 2019, pp. 3–12. doi: 10.1145/3347320.3357688
- [24] H.-C. Chou, W.-C. Lin, C.-C. Lee, and C. Busso, "Exploiting co-occurrence frequency of emotions in perceptual evaluations to train a speech emotion classifier," in *Proc. Interspeech*, Incheon, South Korea, 2022. doi: [verify before submission]
- [25] Z. Huang, B. Ma, M.-W. Mak, et al., "EMO-SUPERB: An in-depth look at speech emotion recognition," arXiv:2402.13018, 2024. doi: [verify before submission]

[26] P. A. Perez-Toro, T. Arias-Vergara, P. Klumpp, J. C. Vasquez-Correa, M. Schuster, E. Noth, and J. R. Orozco-Arroyave, "Depression assessment in people with Parkinson's disease: The combination of acoustic features and natural language processing," *Speech Commun.*, vol. 145, pp. 10–20, 2022. doi: 10.1016/j.specom.2022.09.001

[27] S.-G. Leem, D. Fulford, J.-P. Onnela, D. Gard, and C. Busso, "Separation of emotional and reconstruction embeddings on ladder network to improve speech emotion recognition robustness in noisy conditions," in *Proc. Interspeech*, 2021. doi: [verify before submission]

[28] J. N. de Boer, A. E. Voppel, S. G. Brederoo, H. G. Schnack, K. P. Truong, F. N. Wijnen, and I. E. C. Sommer, "Acoustic speech markers for schizophrenia-spectrum disorders: A diagnostic and symptom-recognition tool," *Psychol. Med.*, vol. 53, no. 4, pp. 1302–1312, 2021. doi: 10.1017/S0033291721001004

[29] P. Singh and G. Saha, "Modulation spectral features for speech emotion recognition using deep neural networks," *Speech Commun.*, vol. 146, pp. 53–69, 2023. doi: 10.1016/j.specom.2022.11.005

[30] S. Parthasarathy and C. Busso, "Semi-supervised speech emotion recognition with ladder networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 2697–2709, 2020. doi: 10.1109/TASLP.2020.3013632