Calculating people's attraction based on their facial expressions, body movement, voice tone and words spoken

Mitrea Ioan David

November 18, 2025

1 Data Collection

The quality and reliability of affective computing research depend critically on rigorous and multimodal data collection pipelines capable of capturing complex human emotional expressions. Traditional laboratory-based acquisition frameworks prioritize controlled environments and high-fidelity measurements; however, they often suffer from limited ecological validity and reduced generalizability [1]. Recent research has therefore shifted toward hybrid approaches that integrate audiovisual, physiological, and behavioral modalities collected in more naturalistic settings, thereby enabling richer affective modeling [2]. Following this methodological direction, our study implements a multimodal extraction pipeline designed to capture fine-grained facial, acoustic, and bodily behavioral cues from video recordings. The complete pipeline is summarized below.

1.1 Person-Specific Video Segmentation

To ensure that downstream feature extraction is performed only during time intervals in which the target participant is visible, we implemented person-specific temporal segmentation based on facial recognition. For each video frame, we computed face embeddings using `face_recognition` and compared them to a reference embedding of the target participant. A cosine distance threshold determined whether the detected face corresponded to the participant of interest. This yielded a set of visibility intervals

$$S = \{(t_i^{\text{start}}, t_i^{\text{end}}) \mid i = 1, \ldots, N\},$$

representing all temporal segments in which the participant was visually present. These intervals functioned as the structural backbone for subsequent audio and behavioral feature extraction.

1.2 Audio Extraction Aligned With Speaker Visibility

Because conventional speaker diarization models can be unreliable in multi-speaker naturalistic environments, we leveraged the person-visibility segments S to construct a participant-specific audio track. Using `moviepy`, audio was extracted only from the intervals in which the participant was detected onscreen. Silent padding ensured the temporal alignment of the resulting WAV file with the original video duration. This visibility-constrained audio track facilitates the analysis of vocal affect—such as prosody, energy, and temporal speaking patterns—while minimizing cross-speaker contamination.

1.3 Facial Action Unit Extraction

Facial behavior was captured by extracting facial Action Units (AUs) based on a two-stage mapping process. First, emotion probabilities were obtained using the DeepFace facial expression recognizer. Second, these categorical emotion estimates were mapped to continuous AU vectors using a rule-based mapping derived from established facial expression literature. Each frame within the visibility intervals produced a 17-dimensional AU vector:

$$\mathbf{a}_t = [\mathrm{AU1}_t, \ldots, \mathrm{AU17}_t].$$

Frames containing no detectable face (e.g., motion blur, occlusion) were assigned the zero vector to preserve temporal coherence. This procedure enabled robust modeling of dynamic facial affect across the entire temporal sequence.

1.4 Body, Hand, and Upper-Body Movement Features

To capture nonverbal expressive behaviors beyond the face, we employed MediaPipe's FaceMesh, Hands, and Pose modules to extract fine-grained kinematic and gestural information. For each frame, we tracked 3D keypoints for the hands, face contour, and upper body. Time-aligned features included:

- **Hand movement dynamics**: frame-to-frame velocity, acceleration, and movement smoothness.
- **Gesture frequency**: counts of meaningful movements exceeding a motion threshold.
- **Face-touch gestures**: detection of hand-to-face contact events using spatial proximity between hand landmarks and facial meshes.
- **Upper-body motion**: displacement and temporal derivatives of shoulder, torso, and arm keypoints.

By integrating these multimodal cues—facial expression, vocal behavior, and gesture/movement dynamics—we obtain a rich representation of affective behavior capable of supporting downstream analyses such as affect recognition, behavioral profiling, and interpersonal dynamics modeling. This multimodal pipeline aligns with contemporary trends in affective computing that emphasize ecological validity, multimodal integration, and fine-grained temporal annotation [1, 2].

2 Methodology

Our approach models short temporal sequences of multimodal behavioral signals to predict binary romantic-interest labels. The methodology consists of per-second feature aggregation, sequence construction, preprocessing, and sequential LSTM-based modeling. This section provides a precise description of each step, including hyperparameters and architectural details.

2.1 Sequence Construction and Feature Representation

Per-participant video recordings are preprocessed to produce **per-second feature vectors $\mathbf{x}_t \in \mathbb{R}^{28}$** comprising:

- 17 **Facial Action Units (AUs)**: frame-level categorical emotions are mapped via a rule-based scheme to continuous AU intensities, then averaged within each second. Missing frames (no face detected) are zero-filled before averaging.
- 4 **Hand and upper-body movement features**: left/right hand velocity, cumulative gesture frequency, and cumulative face-touch events computed from MediaPipe landmarks.
- 3 **Acoustic features**: energy (dB), pitch (Hz), and speaking rate extracted from participant-specific audio segments constrained by visual presence.
- 4 **Sentiment features**: compound, positive, neutral, and negative sentiment scores aggregated from manual-adjusted ScribeFlow transcripts for utterances overlapping each second. Zero-filled when participant is silent.

These per-second vectors are concatenated to form sequences of length $T = 15$ seconds:

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T] \in \mathbb{R}^{T \times 28}.$$

Sequences are generated with a sliding window (stride 1) to capture temporal dependencies, producing overlapping samples for model training. Missing modalities are handled deterministically via zero-filling, allowing the model to learn from absent or occluded signals without introducing NaNs or imputed values.

2.2 Feature Normalization

All features are standardized using a **StandardScaler (z-score normalization)** fit on the training set. For efficiency, the training sequences are reshaped from (N_{train}, T, D) to $(N_{\text{train}} \cdot T, D)$ prior to fitting. Validation and test sequences are transformed using the same scaler. This ensures consistent normalization without leaking information from held-out data.

2.3 Sequential LSTM Architecture

The model is a stacked LSTM followed by fully connected layers:

- **Input:** sequences of shape $(T = 15, D = 28)$.
- **LSTM layers:**
  1. LSTM(64), `return_sequences=True`, dropout=0.2
  2. BatchNormalization
  3. LSTM(32), `return_sequences=True`, dropout=0.2
  4. BatchNormalization
  5. LSTM(16), `return_sequences=False`, dropout=0.2
  6. BatchNormalization
- **Dense layers:** Dense(32) with ReLU + Dropout(0.3), Dense(16) with ReLU + Dropout(0.2)
- **Output:** Dense(1) with sigmoid activation, producing probability $\hat{y} \in [0, 1]$

Formally, the model learns a mapping:

$$f_\theta : \mathbb{R}^{T \times D} \to [0, 1], \qquad \hat{y} = f_\theta(\mathbf{X}),$$

where θ denotes all learnable parameters.

2.4 Training Objective and Optimization

The model is trained to minimize **binary cross-entropy**:

$$\mathcal{L}(\theta) = -\frac{1}{N} \sum_{i=1}^{N} w_{y_i} \left[y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \right],$$

where w_{y_i} are optional class weights computed from the training data to address class imbalance. The optimizer is **Adam**, with default parameters (learning rate = 0.001).

2.5 Training Protocol

Training uses:

- **Batch size:** 16
- **Epochs:** up to 100 with EarlyStopping (monitor=`val_loss`, patience=20, restore_best_weights=True)
- **Learning rate reduction:** ReduceLROnPlateau (factor=0.5, patience=10, min_lr=1e-6)
- **Metrics:** accuracy, precision, recall

Person-specific sequences are split such that validation data contains the last 20% of each participant's sequences, preventing data leakage and ensuring evaluation reflects generalization to unseen individuals.

2.6 Summary

This methodology integrates multimodal behavioral signals into a sequential modeling framework. Temporal aggregation, sequence construction, normalization, and LSTM-based learning collectively enable robust prediction of romantic interest while preserving interpretability and reproducibility. The approach adheres to best practices in affective computing and sequence-based behavioral modeling [2, 1].

3 References

References

[1] R. M. Schmidt. Recurrent neural networks (rnns): A gentle introduction and overview. *arXiv preprint arXiv:1912.05911*, 2019.

[2] L. A. M. Lebois, C. D. Wilson-Mendenhall, W. K. Simmons, L. F. Barrett, and L. W. Barsalou. Learning situated emotions. *Neuropsychologia*, 145:106637, 2018.