Chapter 3

Materials and Methodology

This chapter gives a detailed explanation of the dataset and techniques used in this work. The first section describes the methodology workflow, followed by the database description and pre-processing in the second section. The decomposition of EDA into phasic and tonic using two methods is explained in the third section. The fourth section describes the time-frequency representation and time-encoded generation techniques employed in the work. The fifth section includes the feature extraction framework. The following section explains the statistical technique employed in this work. The final section discusses the classification procedures and their performance validation to differentiate the categorical emotional states.

3.1 Process Pipeline

The proposed process pipeline, illustrated in Figure 3.1, aims to analyze emotional states through a systematic approach consisting of four distinct stages: (1) Data pre-processing, (2) Decomposition Optimization, (3) Segment Optimization, and (4) Windowing Optimization. In the initial phase, EDA signals are obtained from the publicly available CASE dataset (Sharma et al. 2019). These signals undergo pre-processing, involving categorization based on emotional states and downsampling from their original 1000 Hz recording

rate to 20 Hz. This adjustment simplifies computational processes and ensures efficient signal processing while retaining essential information (Hernando-Gallego, Luengo, and Artés-Rodríguez 2017). Following this, a Butterworth low-pass filter with a 2 Hz cutoff frequency is applied to eliminate high-frequency noise present in the EDA signals (Zhang et al. 2020), (Y. Wu, Daoudi, and Amad 2023).

Moving to the next stage, the pre-processed EDA signals undergo decomposition using two algorithms, cvxEDA and Bayesian EDA, yielding tonic and phasic signals. Temporal features are then extracted from these signals to optimize the EDA components and the decomposition technique. Following this, the significance of these features is evaluated using the Kruskal-Wallis test (p<0.05), ensuring the selection of features that exhibit significant differences across emotional states, strengthening their relevance for classification tasks. Finally, the significant features are input into SVM, RF, and XGB ML models to classify four emotional states: amusing, boring, relaxing, and scary. The performance of these classifiers is evaluated using six performance metrics: accuracy, sensitivity, specificity, precision, F1-score, and AUC. The classification process incorporates 10-fold cross-validation and Grid search-CV for robust evaluation and hyperparameter tuning.

In the subsequent stage, the optimized components obtained from the decomposition process are utilized to optimize the segments of EDA signals for emotional classification. In this stage, the optimized component is segmented into two parts, namely the first-half and second-half segments, as shown in Figure 3.2. This stage aims to optimize the segments for emotion analysis by considering three parts: First-half, Second-half, and Whole signal. The segmented signals are transformed into time-frequency representations using STFT and MFC to generate spectrograms. Further, the GLCM and GLRLM features are extracted from these spectrograms. These features are employed in emotion classification using SVM, RF, and XGB classifiers, which are evaluated using the six performance measures.

In the final stage, the optimized components from the decomposition process and opti-

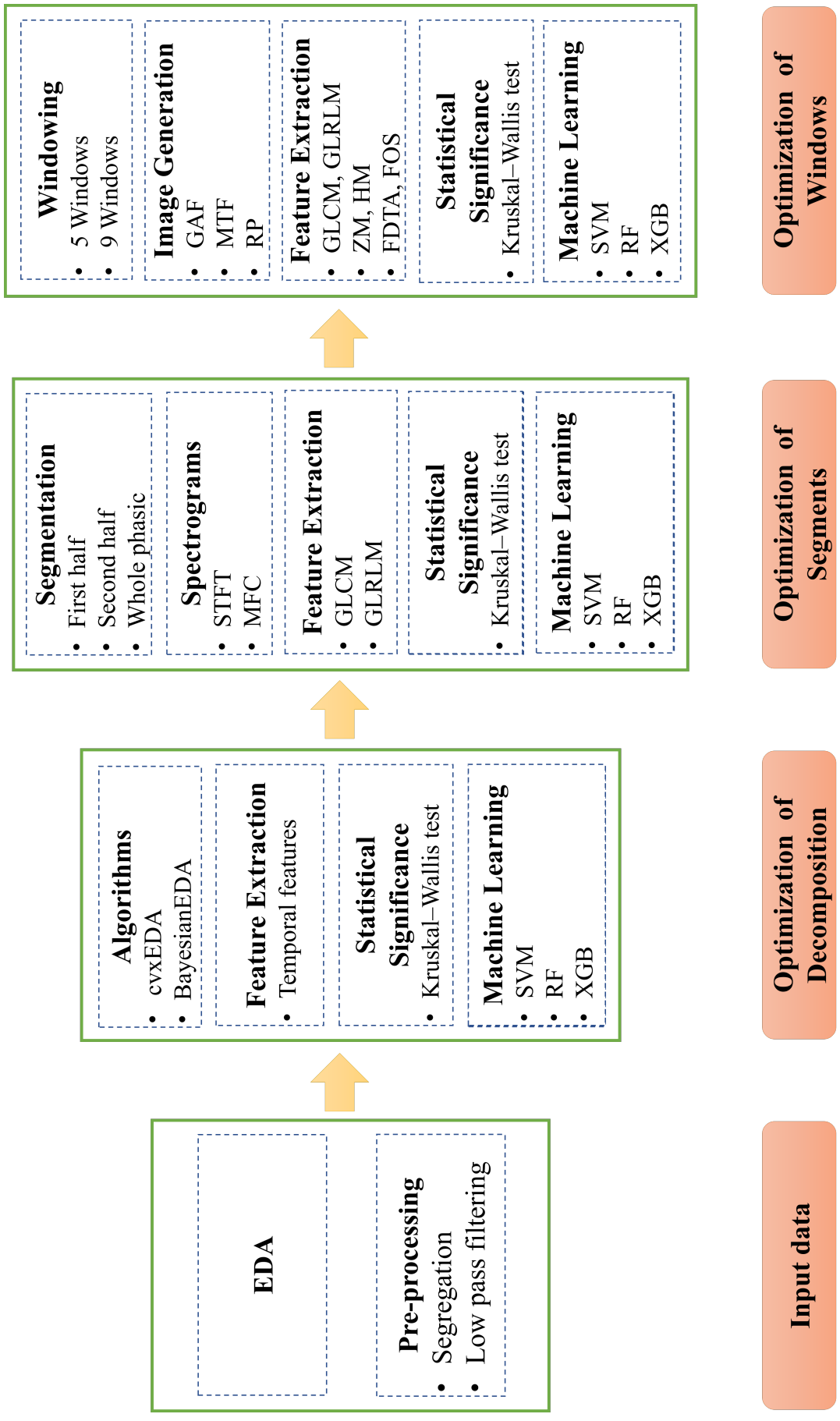

Figure 3.1: Proposed process pipeline for this study.

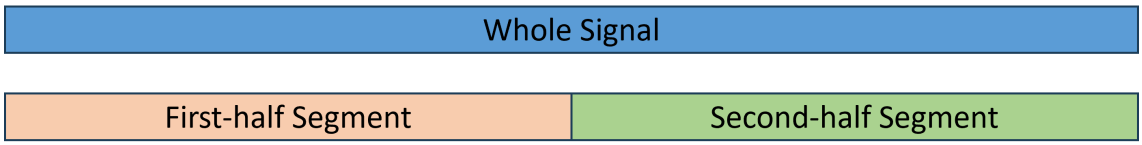

Figure 3.2: Representative segmented signals.

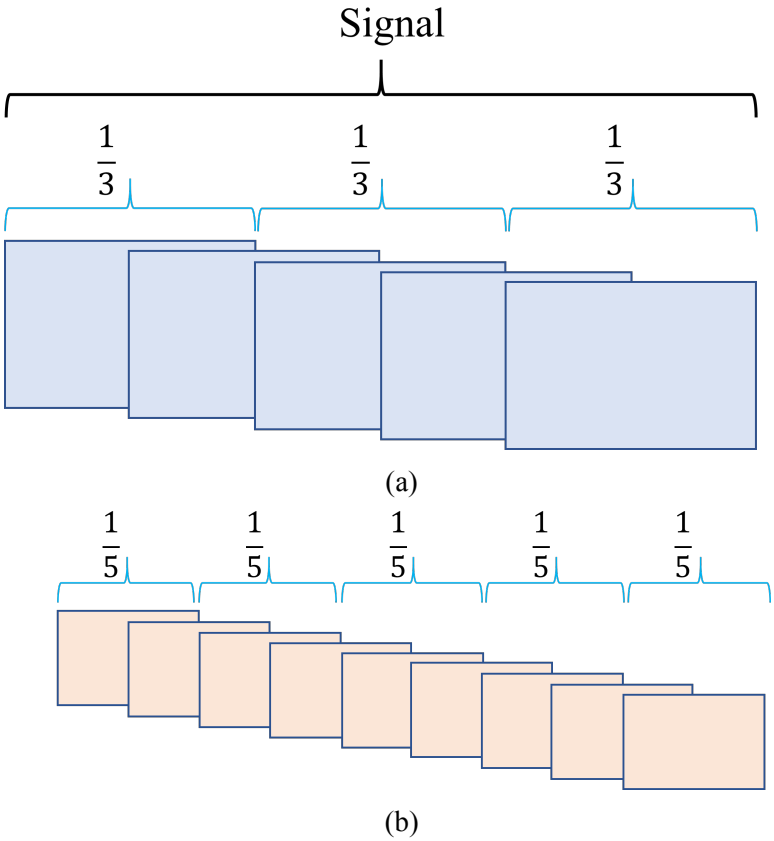

Figure 3.3: Representative signal with (a) 5 windows and (b) 9 windows

mized segment undergo further analysis using windowing approaches. The segments are further divided into signal windows with a 50% overlap (K. Yang et al. 2021). Window sizes are 33.33% (1/3) and 20% (1/5) of the signal, resulting in obtaining 5 and 9 windows, respectively, as shown in Figure 3.3. The windowed signals are converted into image representations using GAF, MTF, and RP methods. Features such as GLCM, GLRLM, FDTA, ZM, HM, and FOS are extracted from these two-dimensional images. These features are then utilized in SVM, RF, and XGB models to classify the four emotional states and optimize the windowing approach.

3.2 Dataset

This study is evaluated using a widely used publicly available CASE dataset. The participant information, stimuli, experiment protocol, and signal description are given below.

3.2.1 Participant Information

The dataset involved 30 healthy participants, comprising 15 males and 15 females, with an average age of 28.6 years for males and 25.7 years for females. These participants were recruited from diverse cultural backgrounds, reflecting a varied sample population. Recruitment was conducted through an organization-wide call for volunteers at the Institute of Robotics and Mechatronics. Participants received detailed information about the experiment, including instructions, consent forms, and guidelines. All participants had a working proficiency in English, the language used for communication during the experiment.

3.2.2 Ethical Framework and Institutional Approval

Ethical considerations were paramount in the study, with the experiment designed in compliance with the World Medical Association's Declaration of Helsinki. Data collection was approved by the institutional ethical review board for data privacy protection and the work council of the German Aerospace Center, ensuring the participants' rights and well-being were safeguarded throughout the study.

3.2.3 Stimuli Information

The experiment included a carefully curated set of eight video stimuli to induce four emotional experiences such as amusement, boredom, relaxation, and scary. These stimuli were selected to represent a diverse range of emotional content and intensity levels to evoke distinct emotional responses from the participants. To prevent potential carry-over effects and ensure each stimulus's independent evaluation, the order of the videos was randomized for each participant. This approach aimed to minimize any bias or influence

from previous stimuli on the perception and annotation of subsequent videos. Additionally, two-minute blue screen intervals were strategically inserted between the videos to allow participants to rest, reset their emotional state, and avoid emotional contamination between stimuli.

3.2.4 Experimental Protocol

The experimental setup occurred within a controlled laboratory setting. On the experiment's day, participants received oral and written protocol explanations. Following this, participants provided informed consent and received a brief introduction to the 2D circumplex model. Physiological sensors were then attached, and participants were seated facing a 42-inch flat-panel TV to watch various videos designed to elicit emotions, as depicted in Figure 3.4. The central figure displays the video-playback window with the embedded annotation interface. The right-most figure zooms in on the annotation interface, showing the self-assessment manikin on the valence and arousal axes. The setup involves a participant watching videos and annotating with a joystick-based emotion reporting interface to continuously report their emotional experiences on the self-assessment manikin for valence and arousal levels. Feedback on the annotation system was collected using the system usability scale questionnaire at the end of the experiment. The experiment lasted approximately 40 minutes, during which the supervisor monitored data acquisition from a distance. Sensors were removed, and participants were offered refreshments, with the opportunity to share any additional insights encouraged.

3.2.5 EDA Signal Description

The dataset comprises EDA recordings from 30 participants collected to assess emotional states triggered by viewing eight emotional videos to elicit four distinct emotions: amusement, boredom, relaxation, and scary. Data acquisition was performed during the experiment using LabVIEW software, sampling physiological signals at 1000 Hz and annotations at 20 Hz. Alongside EDA signals, other physiological signals were also gathered during the experiments. The EDA signals, varying in length and corresponding to the four

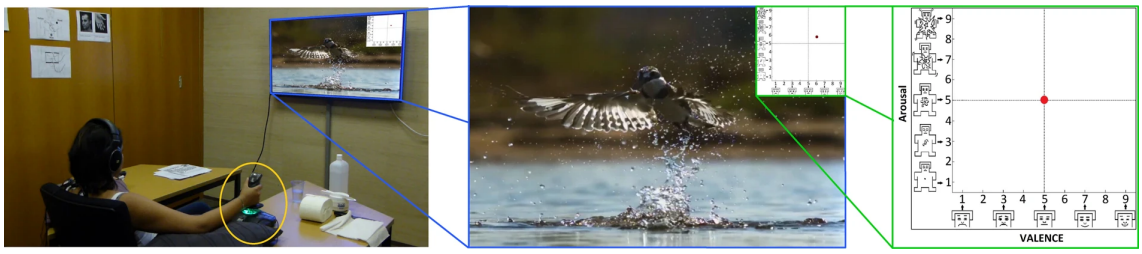

Figure 3.4: Experimental setup of the CASE dataset.

emotions, are utilized for further analysis, resulting in a total of 240 EDA signals in the database. The length of each EDA signal is determined by the duration of the video stimuli. The summary of demographic information for the CASE dataset is shown in Table 3.1.

Table 3.1: Description of CASE dataset

Parameter	**Range**
Number of subjects	30 (15 males, 15 females)
Age	22-37 years
Emotional videos	8
Video duration	118-197 seconds
Recorded signals	**EDA**, ECG, BVP, EMG, RSP and SKT
Sampling rate	1000 Hz (Physiological signals) 20 Hz (Annotations)
Annotation	Continuous self-assessment
Rating scales	Valence (V): [0.5-9.5] Arousal (A): [0.5-9.5]

3.3 Pre-processing

The EDA signals from the CASE dataset were initially categorized into four emotions: amusement, boredom, relaxation, and scary, based on provided emotion labels. To optimize computational efficiency, these signals underwent down-sampling to 20 Hz, preserv-

ing crucial information while reducing processing demands (Hernando-Gallego, Luengo, and Artés-Rodríguez 2017). Additionally, a low-pass Butter-worth filter with a 2 Hz cut-off frequency was applied to eliminate artifacts, ensuring the accuracy and reliability of the signals for analysis (Zhang et al. 2020; Y. Wu, Daoudi, and Amad 2023). This filtering process refines the signals, improving their quality for detailed examination of emotional responses in subsequent analysis stages.

3.4 Decomposition Methods

This study utilizes two decomposition methods: the widely adopted cvxEDA and the recently proposed BayesianEDA, based on the poral valve theory, to estimate the tonic and phasic components. Concise descriptions of these methods are presented below.

3.4.1 Convex Optimization to EDA

Greco, Valenza, Lanata, et al. 2015 developed cvxEDA, a quadratic programming-based method for decomposing tonic and phasic activity, incorporating two distinct dictionaries for representation. The cvxEDA introduced a novel approach to model SCRs as a linear time-invariant system output driven by a sparse non-negative driver signal. The model proposed that the observed skin conductance (y) comprises the phasic activity (r), a slow tonic component (t), and an additive independent and identically distributed zero-average Gaussian noise term (ε). The equation is represented as:

$$y = r + t + \epsilon \tag{3.1}$$

Physiologically plausible characteristics, such as temporal scale and smoothness of the tonic input signal, are achieved using a cubic spline with equally spaced knots every 10 seconds, along with an offset and a linear trend term:

$$t = Bl + Cd \tag{3.2}$$

Here, B is a tall matrix containing cubic B-spline basis functions as columns, l is the vector of spline coefficients, C is an $N \times 2$ matrix (where N is the length of the skin conductance

time series) with $C_{i,1} = 1$ and $C_{i,2} = \frac{i}{N}$, and d is a 2×1 vector representing the offset and slope coefficients for the linear trend.

The phasic component arises from the convolution between the Sudo Motor Nerve Activity (SMNA) p and an impulse response $h(t)$ shaped as a bi-exponential Bateman function:

$$h(t) = (e^{-\frac{t}{\tau_1}} - e^{-\frac{t}{\tau_2}})u(t) \tag{3.3}$$

Here, 1 and 2 are the slow and fast time constants of the phasic curve shape, and $u(t)$ is the unitary step function. By taking the Laplace transform of $h(t)$ and then approximating it in discrete time with sampling time , an auto-regressive moving average model is obtained, represented in matrix form as:

$$H = M^{-1}A \tag{3.4}$$

where M and A are tridiagonal matrices with the moving average and auto-regression coefficients along the diagonals. Using an auxiliary variable q such that:

$$q = A^{-1}p, \quad r = Mq \tag{3.5}$$

the final observation model is expressed as:

$$y = Mq + Bl + Cd + \epsilon \tag{3.6}$$

The objective is to identify the Maximum A Posteriori (MAP) neural driver SMNA (p) and tonic component (t) parametrized by $[q, l, d]$, given the measured skin conductance signal (y). Then, cvxEDA reformulates the MAP problem as a constrained convex minimization Quadratic Programming (QP) problem. Subject to the constraint $Aq \geq 0$, the objective function seeks to minimize the expression:

$$\frac{1}{2}\|Mq + Bl + Cd - y\|_2^2 + \alpha\|Aq\|_2^2 + \frac{\gamma}{2}\|l\|_2^2 \tag{3.7}$$

After suitable matrix manipulations, this optimization problem is rewritten in the standard QP form and efficiently solved using available sparse-QP solvers. Upon obtaining the

optimal $[q, l, d]$, the tonic component t is derived from the spline representation, while the SMNA driving the phasic component can be readily determined as $p = Aq$. The objective function to be minimized reflects a quadratic measure of misfit between the observed data and the model predictions. Additionally, prior knowledge regarding the sparsity and nonnegativity of SMNA (p) and the smoothness of the tonic component are incorporated through regularizing terms and constraints. The cvxEDA algorithm is implemented in MATLAB, and the software is accessible online.
(http://www.mathworks.com/matlabcentral/fileexchange/53326-cvxeda).

3.4.2 Bayesian EDA

This proposed physiological model for EDA is based on the poral valve model by Edelberg. This model explains the secretion and diffusion of sweat, which affects skin conductance (SC). It aims to capture both the slow and fast components of EDA through a comprehensive state-space representation. Sweat secretion begins in response to ANS impulses, with initially empty sweat ducts. As sweat accumulates, hydraulic pressure increases within the ducts, leading to diffusion into the stratum corneum and deeper skin layers, resulting in a rise in SC. The model uses a three-compartment pharmacokinetic system, including direct secretion via pore opening (modeled by $S(t)$) and diffusion/reabsorption (modeled by $T(t)$). The observation equation for SC is

$$SC(t) = P(t) + T(t) + \nu(t) \tag{3.8}$$

where $SC_P(t) = P(t)$ and $SC_T(t) = T(t)$ represent the phasic and tonic components, respectively. To make the model suitable for real-time applications, it assumes a constant fraction of sweat secretion via pore opening and employs a linear approximation, represented in discrete state-space form as

$$x_k = Ax_{k-1} + Bu_k \tag{3.9}$$

and

$$y_k = Cx_k + \nu_k \tag{3.10}$$

Phasic and tonic components in this form are extracted as

$$SC_{P,k} = C_P x_k \tag{3.11}$$

and

$$SC_{T,k} = C_T x_k \tag{3.12}$$

The tonic component is represented by $T(t)$ in the continuous model and by $C_T x_k$ in the discrete model, while the phasic component is represented by $P(t)$ in the continuous model and by $C_P x_k$ in the discrete model. To ensure physiological validity and avoid over-fitting, parameters are constrained to be non-negative, with lower bounds set based on prior studies and specific ratio constraints to maintain relevance (R. Amin and Faghih 2022).

3.5 Time-Frequency Representation

This study uses two Time-Frequency Representation (TFR) methods, STFT and MFC, to produce spectrograms for optimizing segments. Spectrograms visually depict how a signal's frequency content changes over time. Below is a brief description of STFT and MFC.

3.5.1 Short Time Fourier Transform

The STFT is a powerful technique for analyzing time-varying signals and extracting their frequency information. It divides the signal into overlapping segments, applies the Fourier transform to each segment, and captures its frequency components. Mathematically, this is expressed as:

$$\text{STFT}(t, f) = \int x(\tau) w(t - \tau) e^{-j2\pi f \tau} d\tau \tag{3.13}$$

Here, $x(\tau)$ represents the phasic signal, $w(t - \tau)$ is a window function managing the segmentation process, and f is the frequency (Ganapathy and Swaminathan 2019). The resulting spectrogram is a two-dimensional plot where the x-axis represents time, the y-axis represents frequency, and the intensity indicates the magnitude or power of the frequency component at each time and frequency. This plot shows changes in the signal's frequency content over time. The chosen window function for this study is the Hann window.

3.5.2 Mel Frequency Cepstrum

MFC spectrogram is computed by transforming the signal from the linear frequency scale to the logarithmic Mel-scale using the STFT. The resulting spectrum is then passed through a filter bank, which measures signal energy distribution on the Mel-scale frequency vector. The corresponding eigenvalues can be approximated as signal energy distribution on the Mel-scale frequency. The relationship between the Mel-frequency (f_m) and the linear frequency (f) is given by:

$$f_m = 2595 \log_{10} \left(\frac{f}{700} + 1 \right) \tag{3.14}$$

Mathematically, for N triangular filters, the Mel spectrogram $M(f_m, t)$ at time t is given by:

$$M(f_m, t) = \sum_{k=1}^{N} S(f_k, t) \cdot H_m(f_k) \tag{3.15}$$

where $S(f_k, t)$ is the spectrogram magnitude at frequency f_k and time t, and $H_m(f_k)$ is the value of the m-th triangular filter at frequency f_k (Abdul and Al-Talabani 2022).

3.6 Time-series to Image Generation

This study uses time-encoded image methodologies, including GAF, MTF, and RP, to optimize the windowing approach. Below is a concise overview of each method:

3.6.1 Gramian Angular Field

GAFs present a methodology for transforming time series into a matrix format while maintaining the intrinsic spatial and temporal relationships within the data. Two primary types of GAF algorithms are commonly employed: the Gramian Angular Summation Field (GASF) and the Gramian Angular Difference Field (GADF). These techniques are designed to convert time series into images using a polar coordinates-based matrix. The Gramian matrix is formulated such that each element represents the cosine of the summed angles for GASF or the sine of the subtracted angles for GADF.

The initial step involves normalizing the given time series data $X = \{x_1, x_2, \ldots, x_n\}$ of n real-valued observations to ensure all values fall within the interval $[-1, +1]$. The normalization process is defined by the equation:

$$\hat{X} = \frac{X - \min(X)}{\max(X) - \min(X)} \times 2 - 1 \tag{3.16}$$

where $\hat{X}$ is the normalized time series, X is the original time series, $\min(x)$ and $\max(x)$ are the minimum and maximum values in the original time series, respectively.

The normalized time series $\hat{X} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n\}$ is then converted into polar coordinates (ϕ_i, r_i). Here, ϕ_i is the angle and r_i is the radius. The angle ϕ_i is calculated using the arc-cosine function:

$$\phi_i = \arccos(\hat{x}_i), \quad -1 \leq \hat{x}_i \leq +1, \quad \hat{x}_i \in \hat{X} \tag{3.17}$$

The radius r_i is determined by the timestamp value relative to the total number of observations (N) in the time series:

$$r_i = \frac{t_i}{N}, \quad t_i \in \mathbb{N} \tag{3.18}$$

where t_i is the timestamp for the i-th observation. The resulting polar coordinates (ϕ_i, r_i) for each observation in the time series are used to encode an image representation. These images preserve the temporal correlation between the observations in the original time series (Z. Wang and Oates 2015), (Jiang and H.-C. Chen 2023).

The GASF image is as follows:

$$\text{GASF} = \begin{bmatrix} \cos(\phi_1 + \phi_1) & \cos(\phi_1 + \phi_2) & \cdots & \cos(\phi_1 + \phi_n) \\ \cos(\phi_2 + \phi_1) & \cos(\phi_2 + \phi_2) & \cdots & \cos(\phi_2 + \phi_n) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(\phi_n + \phi_1) & \cos(\phi_n + \phi_2) & \cdots & \cos(\phi_n + \phi_n) \end{bmatrix} \tag{3.19}$$

The GADF image is as follows:

$$\text{GADF} = \begin{bmatrix} \sin(\phi_1 - \phi_1) & \sin(\phi_1 - \phi_2) & \cdots & \sin(\phi_1 - \phi_n) \\ \sin(\phi_2 - \phi_1) & \sin(\phi_2 - \phi_2) & \cdots & \sin(\phi_2 - \phi_n) \\ \vdots & \vdots & \ddots & \vdots \\ \sin(\phi_n - \phi_1) & \sin(\phi_n - \phi_2) & \cdots & \sin(\phi_n - \phi_n) \end{bmatrix} \tag{3.20}$$

where ϕ_i and ϕ_j denote the polar angles corresponding to the i^{th} and j^{th} elements of the time series.

3.6.2 Markov Transition Field

MTF is a method to encode one-dimensional time series signals into 2D images, capturing dynamic changes over time and frequency (Song et al. 2023), (Lu, Z. Chen, and Jia 2022). For a time series $X = \{x_1, x_2, \ldots, x_n\}$ of n real-valued observations, the sequence is divided into Q bins based on the magnitude of the values. Each x_i $(1 \leq i \leq n)$ in the sequence maps to a corresponding bin q_j $(1 \leq j \leq Q)$. The transitions between bins at each time step are calculated using a first-order Markov chain, resulting in a $Q \times Q$ Markov transition matrix W as follows:

$$W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1Q} \\ w_{21} & w_{22} & \cdots & w_{2Q} \\ \vdots & \vdots & \ddots & \vdots \\ w_{Q1} & w_{Q2} & \cdots & w_{QQ} \end{bmatrix} \tag{3.21}$$

where w_{ij} represents the probability of transitioning from bin i to bin j. Each row of W contains the transition probabilities from a specific bin to all other bins, ensuring that the sum of probabilities in each row equals 1. The transition probability from bin q_i to bin q_j at time t is given by:

$$w_{ij} = P(x_{t+1} \in q_i \,|\, x_t \in q_j) \tag{3.22}$$

The MTF M is constructed by arranging the probabilities from the W matrix according to the time series order. The MTF M is an $n \times n$ matrix, where each element represents the transition probability based on the time order of the original time series:

$$M = \begin{bmatrix} M_{11} & M_{12} & \cdots & M_{1n} \\ M_{21} & M_{22} & \cdots & M_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ M_{n1} & M_{n2} & \cdots & M_{nn} \end{bmatrix} \tag{3.23}$$

where each element M_{ij} represents the transition probability from the bin corresponding to x_i to the bin corresponding to x_j:

$$M = \begin{bmatrix} w_{q(x_1)q(x_1)} & w_{q(x_1)q(x_2)} & \cdots & w_{q(x_1)q(x_n)} \\ w_{q(x_2)q(x_1)} & w_{q(x_2)q(x_2)} & \cdots & w_{q(x_2)q(x_n)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{q(x_n)q(x_1)} & w_{q(x_n)q(x_2)} & \cdots & w_{q(x_n)q(x_n)} \end{bmatrix} \tag{3.24}$$

Here, $q(x_i)$ denotes the bin to which the value x_i belongs. Each element $w_{q(x_i)q(x_j)}$ corresponds to the probability of transitioning from the bin of x_i to the bin of x_j.

3.6.3 Recurrence Plot

It offers insights into the internal structure of time series data, providing information about similarity, periodicity, chaos, and non-stationarity within the time series (Jin and Shouyi Chen 2023). The initial step involves constructing a phase space from the time series data and selecting appropriate parameters such as the delay coefficient τ (specifies the temporal separation between points within the phase space), embedding function m (defines the dimensionality of the reconstructed space), and threshold ε (specifies the cutoff distance that defines the neighborhood of points in the phase space). The reconstruction of the phase space involves using a reconstructed vector

$$X = (x_i, x_{i+\tau}, \ldots, x_{i+(m-1)\tau}) \tag{3.25}$$

where $i = 1, 2, \ldots, N$ and $N = (n - (m-1)\tau)$. Computing the Euclidean distance between any two points in the phase space and given by

$$D_{ij} = ||X_i - X_j|| \tag{3.26}$$

where $i, j = 1, 2, \ldots, N$. The recurrence matrix R_{ij} is determined by comparing distances with the threshold ε is given by

$$R_{ij} = U(\varepsilon - D_{ij}) \tag{3.27}$$

where $U(\cdot)$ denotes the Heaviside function and can be defined as

$$U(x) = \begin{cases} 0 & \text{if } x < 0, \\ 1 & \text{if } x \geq 0. \end{cases} \tag{3.28}$$

Using the Heaviside function, R_{ij} produces two values: 0 and 1. When $R_{ij} = 1$, it signifies that the distance between two points in the phase space is less than the threshold ε, indicating a recursive relationship. When $R_{ij} = 0$, it implies that the distance between two points in the phase space exceeds the threshold ε, indicating no recursive relationship (Lu, Z. Chen, and Jia 2022).

3.7 Feature Extraction

3.7.1 Temporal Features

This study utilized the temporal features to optimize the EDA components and decomposition method. The features description and formulas are given below (Sánchez-Reolid, Martínez-Rodrigo, et al. 2020), (Greco, Valenza, Lázaro, et al. 2021).

1. **Mean (MN)** : It represents the average value of the signal.

$$\text{MN} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{3.29}$$

where N is the number of points in the signal and x_i is the value of the i-th point.

2. **Median (MDN)**: It is the middle value of a sorted list of signal values.

$$\text{MDN} = \text{Median of } \{x_1, x_2, \ldots, x_N\} \tag{3.30}$$

3. **Standard Deviation (STD)**: It measures the amount of variation or dispersion in a set of signal values.

$$\text{STD} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \text{MN})^2} \tag{3.31}$$

where N is the number of points in the signal and x_i is the value of the i-th point, and MN is the mean.

4. **Maximum Peak (MAP)**: It represents the maximum peak as the highest value in the

signal.

$$\text{MAP} = \max\{x_1, x_2, \ldots, x_N\} \tag{3.32}$$

5. **Minimum Peak (MIP)**: It represents the minimum peak as the lowest value in the signal.

$$\text{MIP} = \min\{x_1, x_2, \ldots, x_N\} \tag{3.33}$$

6. **Dynamic Range (DR)**: It is the difference between the maximum and minimum values in the signal.

$$\text{DR} = \text{MAP} - \text{MIP} \tag{3.34}$$

7. **Mean of First Derivative (MFD)**: It represents the average rate of change of the signal.

$$\text{MFD} = \frac{1}{N-1} \sum_{i=1}^{N-1} \frac{x_{i+1} - x_i}{t_{i+1} - t_i} \tag{3.35}$$

where t_i is the time corresponding to the i-th point in the signal.

8. **Mean of Second Derivative (MSD)**: It represents the average acceleration of the signal.

$$\text{MSD} = \frac{1}{N-2} \sum_{i=1}^{N-2} \frac{\frac{x_{i+2} - x_{i+1}}{t_{i+2} - t_{i+1}} - \frac{x_{i+1} - x_i}{t_{i+1} - t_i}}{t_{i+2} - t_i} \tag{3.36}$$

9. **Standard Deviation of First Derivative (SFD)**: It measures the variability of the rate of change of the signal.

$$\text{SFD} = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N-1} \left(\frac{x_{i+1} - x_i}{t_{i+1} - t_i} - \text{MFD} \right)^2} \tag{3.37}$$

10. **Standard Deviation of Second Derivative (SSD)**: It measures the variability of the

acceleration of the signal.

$$\text{SDSD} = \sqrt{\frac{1}{N-2}\sum_{i=1}^{N-2}\left(\frac{\frac{x_{i+2}-2x_{i+1}+x_i}{(t_{i+2}-t_{i+1})(t_{i+1}-t_i)} - \text{MSD}}{\Delta t_i}\right)^2} \tag{3.38}$$

where $\Delta t_i = t_{i+1} - t_i$ is the time difference between adjacent points in the signal.

3.7.2 Image based Features

In this study, the focus lies on leveraging GLCM and GLRLM features extracted from spectrograms to optimize signal segments. Moreover, GLCM, GLRLM, FDTA, ZM, HM, and FOS features are utilized and derived from time-encoded images to enhance the optimization of the windowing approach.

GLCM features are pivotal for texture analysis, examining pixel spatial relationships to uncover patterns and structures in the data. They offer valuable insights into the texture's complexity and composition, aiding in identifying subtle variations and distinguishing between different texture types. Moreover, GLCM features provide quantitative measures such as contrast, correlation, energy, and homogeneity, which further enhance our understanding of texture characteristics within an image. Similarly, GLRLM features capture the length of consecutive pixels with the same gray level, providing valuable information about texture and granularity within an image. They help quantify the distribution of run lengths, which can be indicative of different textures and structures present in the image (Öztürk and Akdemir 2018), (Haralick, Shanmugam, and Dinstein 1973), (Shabu and Jayakumar 2020), (Ronicko et al. 2020).

FDTA measures the complexity and roughness of an image by assessing its fractal properties. This feature is crucial for characterizing irregular and self-similar patterns, providing valuable information about the overall structure and composition of the image (Tsiaparas et al. 2010). ZM features offer a rotation-invariant feature set, capturing the shape and geometric information of objects within an image. This is particularly useful for consistent identification despite changes in orientation, allowing for robust object recognition

and classification (Shiyi Chen et al. 2023). Hu moments, consisting of seven invariant moments derived from image moments, are important for characterizing the shape and spatial distribution of pixel intensities in an image. They remain invariant to transformations such as translation, scale, and rotation, making them essential for robust object recognition and classification tasks (Anandhalli, Tanuja, and Baligar 2022). Lastly, FOS provides insights into the distribution of pixel intensities within an image, including metrics such as mean, variance, skewness, and kurtosis. These statistics offer valuable information about the overall intensity distribution, serving as foundational features for further image analysis and classification (Ahmed, Bari, and Gavrilova 2019). The comprehensive list of features encompassing GLCM, GLRLM, FDTA, ZM, HM, and FOS is outlined in Table 3.2.

3.8 Significance Test

The study employs the Kruskal-Wallis (KW) test to assess the statistical significance of the features within their respective optimization modules. This non-parametric test is selected to evaluate potential differences across three or more independent groups. It assesses whether the medians of the groups are statistically different without assuming the normality of the data, making it suitable for non-normally distributed or ordinal data. The test works by ranking all the data points across groups and calculating a test statistic based on the ranked data. This test statistic H for the KW test is calculated as:

$$H = \frac{12}{N(N+1)} \sum_{j=1}^{k} \frac{R_j^2}{n_j} - 3(N+1) \tag{3.39}$$

where: N is the total number of observations across all groups,

k is the number of groups,

R_j is the sum of the ranks of group j,

n_j is the number of observations in group j.

The computed H value is then compared against a critical value from the chi-squared distribution with $k-1$ degrees of freedom (in this case, 3 degrees of freedom). If H exceeds

this critical value, corresponding to $p < 0.05$, the null hypothesis of no difference between groups is rejected, indicating significant differences among the groups. Conversely, if H falls below the critical value, corresponding to $p \geq 0.05$, the null hypothesis is retained. This rigorous statistical approach ensures robust evaluation of feature significance across different optimization modules (Veeranki, Diaz, et al. 2024).

3.9 Machine learning

The significant features (p<0.05) identified through the KW test were input into three widely used machine learning classifiers, such as SVM, RF, and XGB, to recognize emotions. We utilized the GridSearchCV package to fine-tune the hyper-parameters of these classifiers, as detailed in Table 3.3. GridSearchCV performs an exhaustive search over a specified parameter grid, ensuring the selection of the best possible hyper-parameters for each classifier. This process involves evaluating different combinations of parameters, such as kernel types for SVM, the number of trees for RF, and learning rates for XGB, enhancing model performance and generalization. To ensure unbiased and robust classification, we implemented stratified 10-fold cross-validation. This technique divides the dataset into ten equal folds, maintaining the same proportion of each emotion category in every fold. The models were trained on nine folds and tested on the remaining fold, repeating this process ten times. This method guarantees that each sample is used for both training and validation, providing a comprehensive assessment of model performance while minimizing over-fitting.

3.9.1 Support Vector Machine

SVM is a powerful algorithm used for pattern recognition and classification tasks. It constructs a mathematical function that creates hyperplanes or boundaries to separate different classes of data points in a high-dimensional space. The discriminant function of SVM can be expressed as:

$$f(x) = \sum_{i=1}^{n} a_i^* K_s(x, x_i) + b \tag{3.40}$$

In this equation, x_i represents the training sample eigenvector, x is the recognizing sample eigenvector, a_i^* is the Lagrange operator, $K_s(x, x_i)$ is the kernel function, and b is the bias term (George E. Sakr et al., 2010; Jac Fredo et al., 2020)(Sakr, Elhajj, and Huijer 2010), (Ronicko et al. 2020). SVM uses linear or nonlinear kernel functions, such as the linear and radial basis kernel functions, to create hyper-planes that effectively classify data into different classes. Parameters such as cost, gamma, and kernel type were fine-tuned.

3.9.2 Random Forest

RF is a highly effective and interpretable ML technique for classification tasks. It aggregates the decisions of many decision trees to estimate the posterior probability density function. Each decision tree within the ensemble determines the leaf node for a given input feature vector through a series of simple threshold comparisons at decision nodes. During the training stage, a tree-specific local posterior distribution is computed, and the overall posterior distribution is obtained by combining all these local distributions. Features and their corresponding threshold values are selected at each node to construct each tree in the forest using the $GINI$ index measure. This measure evaluates the performance of feature-threshold pairs in separating training vectors of different classes. The $GINI$ index is calculated as:

$$GINI = \sum_{i=1}^{N} \sum_{j \neq i} \frac{|T_i|}{|T|} \frac{|T_j|}{|T|} \tag{3.41}$$

Here, T represents the given training set, C_i denotes the class to which a randomly selected sample belongs, and $\frac{f(C_i,T)}{|T|} \frac{f(C_j,T)}{|T|}$ is the probability that the selected case belongs to class C_i (Mostafa K. Abd El Meguid; and Martin D. Levine 2014, Jac Fredo et al. 2020) (Abd El Meguid and Levine 2014), (Ronicko et al. 2020).

3.9.3 Extreme Gradient Boosting

XGB, a prominent algorithm in the ensemble learning category, stands out for its exceptional performance in classification and regression tasks, boasting high predictive accuracy. This algorithm constructs a robust predictive model by aggregating the outputs of multiple weak learners, typically decision trees. XGB employs a regularized objective function, which comprises a loss function measuring the model's error and a regularization term penalizing complex models to minimize over-fitting. The objective function for XGB is defined as:

$$\text{Obj} = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{3.42}$$

Here, n denotes the number of training samples, L represents the loss function, y_i is the predicted output for the i-th sample, K signifies the number of weak learners (trees), and $\Omega(f_k)$ denotes the regularization term for each tree. The distinctive feature of XGB lies in its gradient-boosting framework, where each new tree is trained to correct the errors of the combined model. Moreover, XGB incorporates regularization terms and utilizes a Taylor expansion to optimize the objective function efficiently. This amalgamation of techniques renders XGB a favored choice across various ML tasks (Nikhil et al. 2024).

The performance of the ML models was evaluated using six metrics such as accuracy, sensitivity, specificity, precision, F1-score, and AUC derived from the confusion matrix of ML each model. The confusion matrix offers a comprehensive visualization of the model's performance across different classes. Accuracy represents the model's performance in accurately predicting all classes across the total number of instances. Sensitivity indicates the model's ability to identify positive cases correctly. Specificity reveals the model's potential to identify negative cases correctly. Precision indicates the model's capability to identify true positives and avoid false positives. The F1-score is the harmonic mean of precision and sensitivity, providing a single value that summarizes the model's performance in terms of both false positives (precision) and false negatives (sensitivity). The AUC reflects the model's ability to distinguish between different classes by measuring the area

under the receiver operating characteristic curve (Ronicko et al. 2020), (Rahman, M. Z. Hossain, and Gedeon 2019). Table 3.4 shows the classification performance evaluation metrics formulas.

Various toolboxes and platforms were utilized in this study to implement the proposed process pipeline and analyze the results. MATLAB 2023a handled preprocessing, decomposition, and segmentation tasks. Python (version 3.11.4) within Jupyter Notebook was used for spectrogram and time-encoded image generation, utilizing the Librosa and pyfeats packages, respectively. Spectrogram-based and image-based feature extraction, supported by the skimage.feature, glrlm, and pyfeates packages, as well as the execution of ML models, were also conducted using Python. All algorithms were executed on a Dell 12th Gen Intel(R) Core(TM) i7-12700K with a clock speed of 3.60 GHz and 64.0 GB RAM.

Table 3.2: Comprehensive list of the features obtained from GLCM, GLRLM, FDTA, ZM, HM, and FOS.

Texture/Shape analysis	**Features**
GLCM (38)	Contrast, Dissimilarity, Homogeneity, Energy, Correlation, Angular Second Moment (ASM), Entropy, Mean, Variance, and Sum of Squares (SS). The mean values following features: 1) Mean of Angular Second Moment (MASM), 2) Mean of Contrast (MCN), 3) Mean of Correlation (MCR), 4) Mean of Sum of Squares Variance (MSSV), 5) Mean of Inverse Difference Moment (MIDM), 6) Mean of Sum Average (MSA), 7) Mean of Sum Variance (MSV), 8) Mean of Sum Entropy (MSE), 9) Mean of Entropy (ME), 10) Mean of Difference Variance (MDV), 11) Mean of Difference Entropy (MDE), 12) Mean of Maximal Correlation Coefficient (MMCC), 13) Mean of Information Measures of Correlation-1 (MIMC1), and 14) Mean of Information Measures of Correlation-2 (MIMC2). The range of values of the following features: 1) Range of Angular Second Moment (RASM), 2) Range of Contrast (RCN), 3) Range of Correlation (RCR), 4) Range of Sum of Squares Variance (RSSV), 5) Range of Inverse Difference Moment (RIDM), 6) Range of Sum Average (RSA), 7) Range of Sum Variance (RSV), 8) Range of Sum Entropy (RSE), 9) Range of Entropy (RE), 10) Range of Difference Variance (RDV), 11) Range of Difference Entropy (RDE), 12) Range of Maximal Correlation Coefficient (RMCC), 13) Range of Information Measures of Correlation-1 (RIMC1), and 14) Range of Information Measures of Correlation-2 (RIMC2).
GLRLM (5)	Short Run Emphasis (SRE), Long Run Emphasis (LRE), Grey Level Uniformity (GLU), Run Length Uniformity (RLU), and Run Percentage (RPC).
FDTA (4)	Hurst Coefficients 1 to 4 (HC1 to HC4)
FOS (7)	Mean (FOSMN), Variance (FOSVRN), Skewness (FOSSKW), Kurtosis (FOSKRT), Energy (FOSERG), Entropy (FOSEN), and Coefficient of Variation (FOSCV)
HM (7)	Moments 1 to 7 (HM1 to HM7)
ZM (24)	Moments 1 to 24 (ZM1 to ZM24)

Table 3.3: Classifier hyper-parameters and values.

Classifier	**Hyperparameters**	**Values**
SVM	C	0.1, 1, 10
	Gamma	scale, auto, 0.1, 1
	Kernel	rbf, linear, poly
RF	n_estimators	50, 100, 200
	max_depth	None, 10, 20
	min_samples_split	2, 5, 10
XGB	Learning rate	0.01, 0.1, 0.2
	n_estimators	50, 100, 200
	max_depth	3, 5, 7
	Subsample	0.8, 1
	colsample_bytree	0.8, 1

Table 3.4: Performance metrics for classifiers and formulas.

Parameter	**Formula**
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Sensitivity	$\frac{TP}{TP+FN}$
Specificity	$\frac{TN}{TN+FP}$
Precision	$\frac{TP}{TP+FP}$
F1-score	$2 \times \frac{Sensitivity \times Precision}{Sensitivity+Precision}$

Notes: *TP-True Positive*
TN-True Negative
FP-False Positive
FN-False Negative