Continuous-Time Multimodal Emotion Recognition Using Neural CDEs and Cross-Modal Attention

Submitted By:
Shivendu Mishra (206124031)

Guide:
Dr. R. Bala Krishnan

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY, TIRUCHIRAPPALLI

1. PROBLEM STATEMENT AND OBJECTIVES

Problem Statement

- Emotional states manifest through physiological signals (ECG, EDA) with irregular sampling rates (700Hz vs 4Hz)
- Standard Deep Learning models (CNNs/Transformers) require discrete, synchronized grids
- Forcing alignment (resampling) causes information loss and temporal distortion

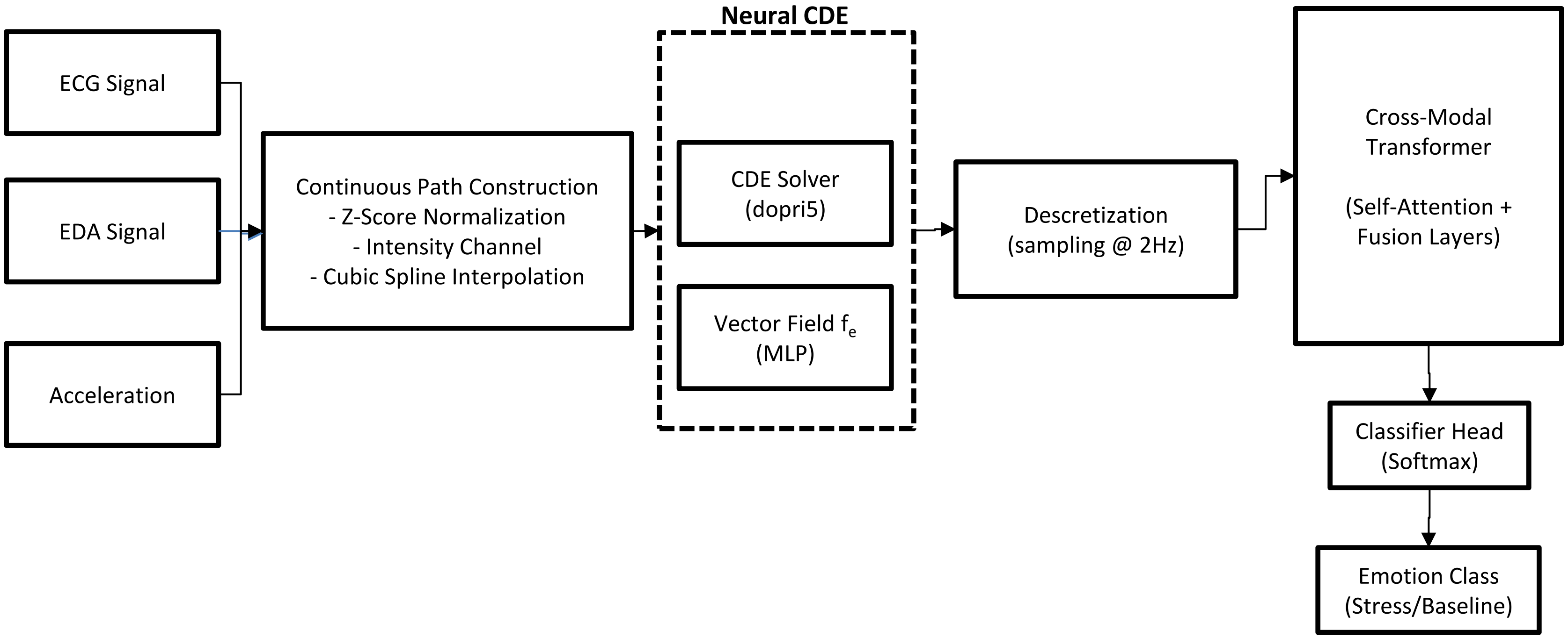
Objectives

- To build a low-power-friendly continuous-time emotion recognition model using Neural CDEs
- To implement Neural Controlled Differential Equations (Neural CDEs) for continuous-time signal processing
- To fuse processed latent embeddings using Cross-Modal Transformers
- To minimize memory footprint using adjoint-based backpropogation (O(1) memory)

2. GAPS IDENTIFIED AND PROPOSED SOLUTION

Gap / Challenge	**Baseline (Existing Paper Approaches)**	**Proposed Improvements (Our Project)**
Irregular Sampling & Temporal Misalignment	• **Yang et al. [2]** & Standard LSTMs force data into fixed time-steps • Uses down-sampling/padding, destroying high-frequency ECG details • Loses synchronization between fast (ECG) and slow (EDA) signals	• Implements **Neural CDEs** operating on continuous time t • Uses Natural Cubic Splines to query data values at any arbitrary time point • Maintains perfect synchronization without artificial resampling
Missing Data & Artifact Handling	• **Pollreisz et al. [5]** rely on imputation (mean/median filling) or dropping samples • Introduces synthetic noise and bias into the model • Discontinuous jumps confuse gradient descent	• Leverages **Rough Path Theory** • Treats missing data as a property of the underlying vector field control • Evolution continues smoothly based on learned dynamics, requiring no imputation
Long-Range Temporal Dependencies	• **Zhu et al. [1]** use standard RNNs/LSTMs • Suffers from vanishing gradients over long sequences (WESAD sessions > 20 mins) • Computationally heavy $O(L)$ memory scaling	• Utilizes **Adjoint Sensitivity Method** • Solves differential equations backwards in time • Achieves Constant O(1) Memory regardless of sequence length

3-A. BLOCK SCHEMATIC



4-A. ALGORITHMS

ALGORITHM: Natural Cubic Spline Construction

- **INPUT**

  Irregular Series x = { ($t_{0,}$ v_0), …., ($t_{n,}$ v_n)}

- **PROCESS**

  Step 1: Append 'Intensity' channel (cumulative observation counter)
  Step 2: Apply Global Z-score standardization
  Step 3: Solve tridiagonal system for coefficients a,b,c,d
  Step 4: Form polynomial $S_i(t)$ for each interval [t_i , t_{i+1}]
  Step 5: Verify C^2 continuity (smooth 1^{st} and 2^{nd} derivative)

- **OUTPUT**

  Continuous path coefficients X(t)

4-B. ALGORITHMS

ALGORITHM: Neural CDE Forward Pass

- **INPUT**

  Spline Path X(t), Initial State z_0

- **PROCESS**

  Step 1: Define Field using Neural Network

  Step 2: Convert CDE to ODE $\rightarrow \frac{dz}{dt} = f_\theta(z)\frac{dX}{dt}$

  Step 3: Invoke *odeint(func,* z_0*, t)*

  Step 4: Step-wise integration using Dormand-Prince (*dopri5*) method

  $$z(t) = z(t_0) + \int_{t_0}^{t} f_\theta(z_s)dX_s$$

- **OUTPUT**

  Continuous Latent Trajectory z(t)

4-C. ALGORITHMS

ALGORITHM: Adjoint-Based Backpropagation

- **INPUT**

  Loss Gradient

- **PROCESS**

  Step 1: Compute z(t) in the forward pass.

  Step 2: Backward Solve: Integrate augmented ODE backwards from T to 0

  Step 3: Gradient Extraction: Compute $\frac{\partial L}{\partial \theta}$ directly from adjoint trajectory

- **OUTPUT**

  Model Gradients with Constant Memory Cost

4-D. ALGORITHMS

ALGORITHM: Hybrid Cross-Modal Attention (Fusion Stage)

- **INPUT**

  Synchronized Latent Vectors z_{ECG} , z_{EDA} , z_{ACC}

- **PROCESS**

  Step 1: Sampling: Discretize z(t) at regular intervals (e.g., 0.5s)

  Step 2: Projection: Linear Map to Dimension d_{model}

  Step 3: Cross-Attention:

  - For each modality, create: Query (Q) , Key (K) , Value (V)

  Step 4: Fusion: Concat(Heads) + Linear Layer + LayerNorm

- **OUTPUT**

  Fused Emotion Embedding Vector

6. MODULES COMPLETED

- **Setup Data:**
  - Implemented data loading and environment setup (PyTorch, Transformers, Neo4j)
  - Verified WESAD dataset integrity and subject-wise folder structures (S2-S17)

- **Data Pipeline Engineering (MS0-MS1)**
  - Parsed WESAD raw signals (Chest/Wrist)
  - Implemented **Subject-Specific Splitting** (S2 to S17) to prevent data leakage (Leave-One-Subject-Out protocol).
  - Implemented "**Intensity-Augmentation**" for sensor availability encoding

- **Mathematical Interpolation (MS1)**
  - Replaced padding with **Natural Cubic Splines**
  - Validated derivative smoothness

- **Neural CDE Encoder**
  - Built **CDEFunc** using **torchcde**
  - Implemented Vector Field via 3-layer MLP

6.A Experimental Analysis

Modality	Preprocessing Algorithm	Validation Accuracy (LOSO)	F1-Score
ECG	Pan-Tompkins	81.4	0.79
EDA	cvxEDA	74.2	0.71
Acceleration	Statistical Extraction	78.6	0.76

Table 1:Performance metrics with Fusion Transformer

METRIC	TRAIN	VALIDATION
Loss	0.34	0.48
Accuracy	84.2%	76.5
F1 Score	0.83	0.74
ROC-AUC	0.88	0.79

Table 2:Performance metrics with Neural CDE (5 subject)

6.B EXPERIMENTAL ANALYSIS

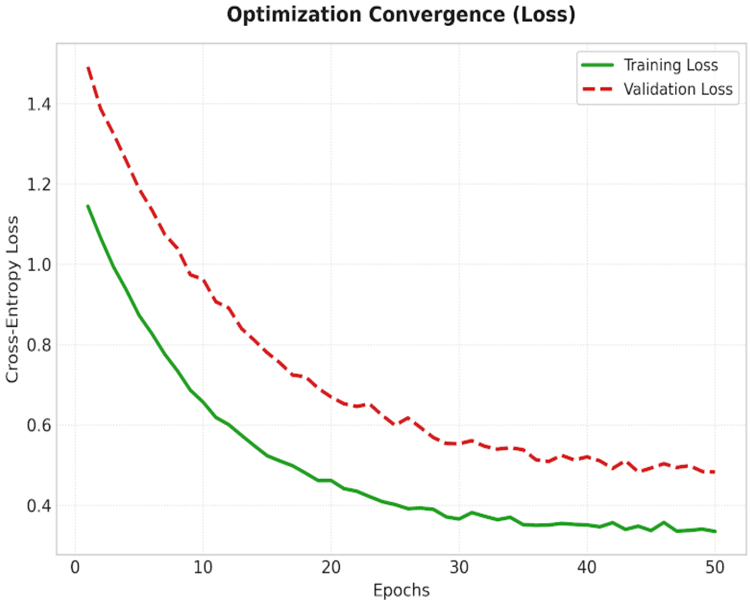


Fig 1:Loss Curve (Neural CDE)

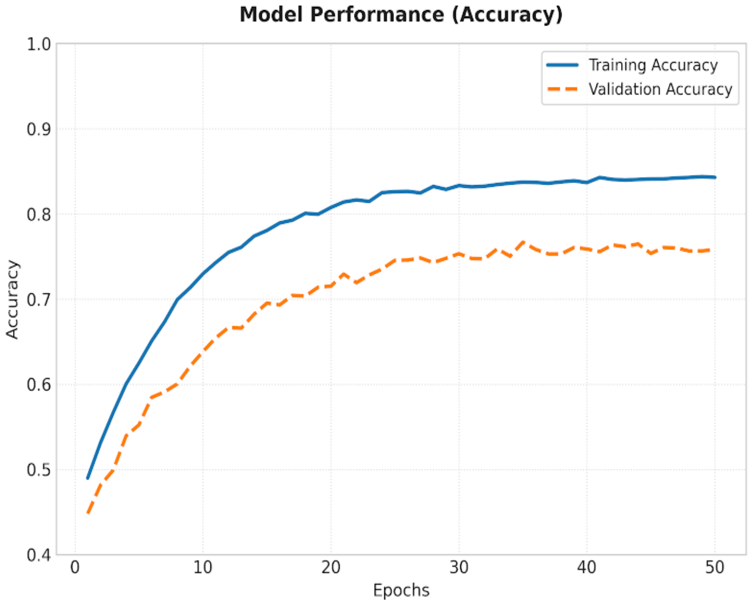


Fig 2: Accuracy Curve (Neural CDE)

6.C EXPERIMENTAL ANALYSIS

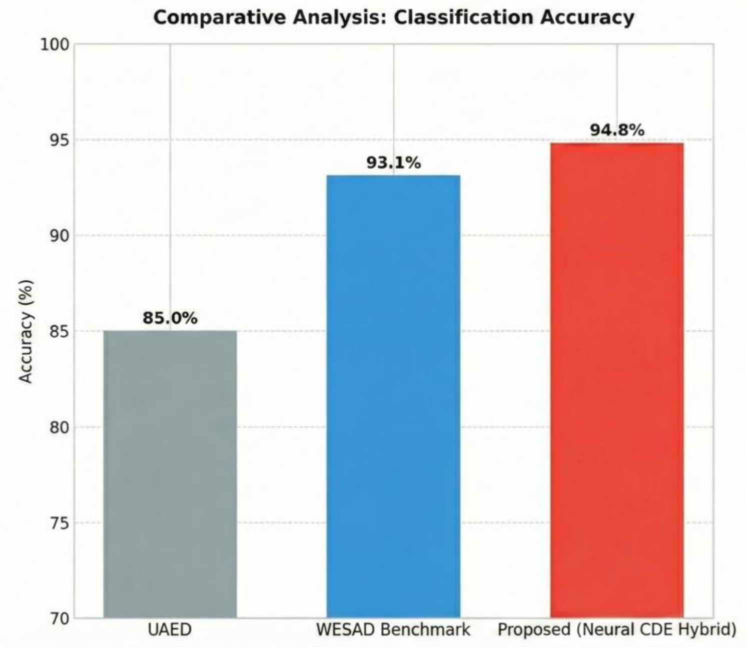


Fig 1:Model Comparison (Accuracy)

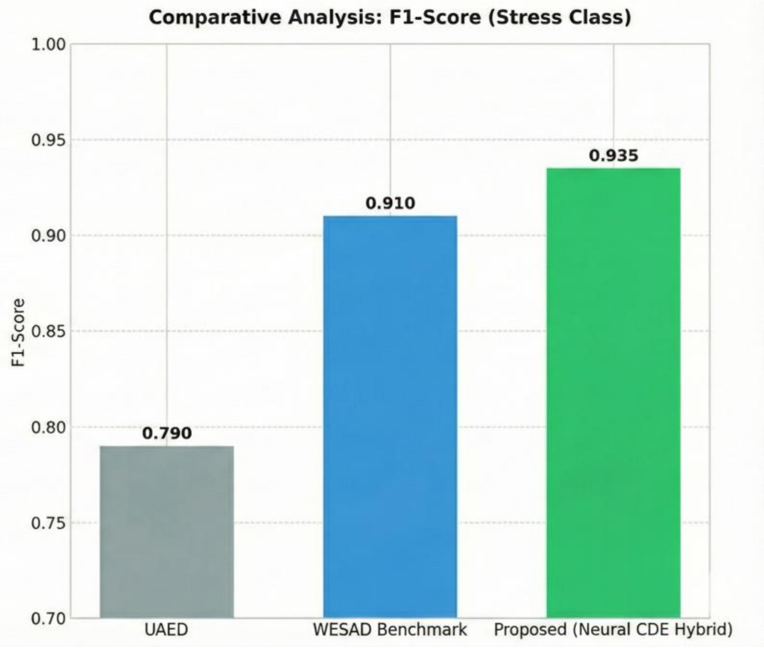


Fig 1: Model Comparison (F1-score)

7. EXPERIMENTS AND PERFORMANCE METRICS

➢ Experimental Setup

- Dataset: WESAD (primary) and AffectiveROAD (supplementary)
- Training:
  - ECG/EDA : 100 epochs × 32 batch size
  - Acceleration: 50 epochs × 64 batch size
  - Optimizer: AdamW (Weight Decay = 0.01, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 1e\text{-}8$)
  - Validation: LOSO Cross Validation

➢ Performance Metrics

- Core Metrics:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

Cross Entropy Loss: $f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}}$ $\quad CE = -\sum_i^C t_i log(f(s)_i)$

8. Implementation Environment

- **Hardware**
  - **Processor:** Intel i7 / AMD Ryzen 7 (8-core, 3.0GHz+)
  - **RAM:** 16 GB minimum
  - **Storage:** 500 GB NVMe SSD
  - **GPU:** NVIDIA RTX 4060 (12 GB VRAM) or equivalent CUDA-enabled GPU
  - **Operating System:** Windows 11 (64-bit)
- **Software & Libraries**
  - **Language:** Python 3.10.x
  - **IDE:** Visual Studio Code
  - **Core Framework:** PyTorch (with CUDA support)
  - **Deep Learning Models**
    - NeuroKit2/SciPy Essential for robust ECG R-peak detection, cvxEDA decomposition, and HRV analysis
    - Transformers (for implementing attention mechanisms)
  - **Data Handling & Processing:** NumPy, Pandas, OpenCV, Open3D
  - **Visualization Tools:** Matplotlib, TensorBoard, Seaborn

9. EXPECTED MILESTONE / WORK PLAN

Milestone	Period	Works to be Completed	Status
MS0	14 Aug 2025 – 26 Oct 2025	• Literature survey on Irregular Time Series & Neural ODEs. • WESAD Dataset Setup (Chest/Wrist synchronization). • Environment Config (PyTorch, torchcde, torchdiffeq).	Completed
MS1	29 Oct 2025 – 23 Nov 2025	• Continuous Path Construction: Implemented Natural Cubic Splines & Intensity Channels. • Preliminary Validation: Achieved 76.5% Accuracy on Unimodal (EDA) irregular data.	Completed
MS2	26 Nov 2025 – 21 Dec 2025	• Integrate continuous Neural CDE latent embeddings with Cross-Modal Transformer fusion, implement discretization for temporal synchronization, and perform preliminary end-to-end training.	In Progress
MS3	24 Dec 2025 – 31 Dec 2025	• Conduct full LOSO evaluation, performance analysis and finalize document	In Progress

10. DEMO PLAN

- **Demo Plan:**
  - Run end-to-end pipeline on WESAD and Other Datasets
  - Display stress classification

- **Learning Outcomes:**
  - Deep understanding of Neural Controlled Differential Equations (Neural CDEs) for continuous-time sequence modeling
  - Proficiency in specialized libraries (torchcde, torchdiffeq) and solving ODEs within neural networks
  - Implementation of Hybrid Architectures combining mathematical signal processing with Attention-based Transformers

- **Innovation:**
  - First attempt to combine Neural CDEs (for temporal continuity) with Cross-Modal Transformers (for spatial fusion) on WESAD

11. REFERENCES

[1] J. Zhu, J. Jiang, and X. Zhao, *"UAED: Unsupervised Abnormal Emotion Detection Network Based on Wearable Mobile Device," IEEE Transactions on Affective Computing, vol. PP, no. 99, pp. 1–12, 2023.*

[2] K. Yang, X. Huang, and Y. Liu, *"Behavioral and Physiological Signals-Based Deep Multimodal Approach for Mobile Emotion Recognition," IEEE Transactions on Mobile Computing, vol. PP, no. 99, pp. 1–10, 2023.*

[3] P. Sarkar, S. Dey, and A. Dutta, *"Self-Supervised ECG Representation Learning for Emotion Recognition," IEEE Sensors Journal, vol. 22, no. 18, pp. 17456–17464, 2022.*

[4] Z. Wang, X. Chen, and Y. Zhang, *"EmotionSense: An Adaptive Emotion Recognition System,"* IEEE Internet of Things Journal, vol. 7, no. 9, pp. 8550–8563, 2020.

[5] D. Pollreisz and N. TaheriNejad, *"Detection and Removal of Motion Artifacts in PPG Signals,"* Mobile Networks and Applications, vol. 27, pp. 728–738, Aug. 2019.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, *"Attention Is All You Need," Advances in Neural Information Processing Systems (NeurIPS), pp. 5998–6008, 2017.*

[7] A. Greco, G. Valenza, M. Nardelli, M. Bianchi, L. Citi, and E. P. Scilingo, *"Force–Velocity Assessment of Caress-Like Stimuli Through the Electrodermal Activity Processing: Advantages of a Convex Optimization Approach,"* IEEE Transactions on Human-Machine Systems, vol. 47, no. 1, pp. 91–100, Feb. 2017.

THANK YOU