Robust Emotion Recognition via Attentive Denoising Autoencoders and Iterative Self-Training

İlke Başak Baydar
Department of Computer and Informatics
Istanbul Technical University
Istanbul, Turkiye
baydaril22@itu.edu.tr
Student ID: 150140709

***Abstract*—Emotion recognition from physiological signals is challenging due to high-dimensional feature spaces, limited labeled data, and the presence of measurement noise. In this project, 1793-dimensional feature vectors are classified into four valence-arousal categories: LVLA, HVLA, LVHA, and HVHA. To address these challenges, a compact Denoising Autoencoder (DAE) architecture with residual connections and a channel-wise attention mechanism was explored as part of a semi-supervised learning framework.**

Rather than relying on large ensemble models, the proposed approach focuses on learning stable latent representations through reconstruction-based regularization. Unlabeled data were incorporated using an iterative Teacher–Student self-training strategy with confidence-based pseudo-labeling and Mixup augmentation. Model performance was evaluated under subject-independent conditions using 5-fold Group Cross-Validation. Under this setting, the final configuration reached a macro F1-score of 0.4980 locally and 0.47644 on the Kaggle test set. These results suggest that combining structural regularization with cautious pseudo-label expansion can be effective for high-dimensional tabular emotion recognition tasks.

***Index Terms*—Emotion Recognition, Denoising Autoencoder, ResNet, Pseudo-Labeling, Semi-Supervised Learning, Feature Attention**

I. Introduction

This project's goal is to categorize 1793 dimensional feature vectors according to the valence-arousal representation into four emotional quadrants: LVLA, HVLA, LVHA, and HVHA. The main success metric is to maintain high macro F1-scores while achieving generalization across unseen subjects (person IDs).

By switching from conventional black-box classifiers to a structural regularization approach based on Autoencoders (AE) and Denoising Autoencoders (DAE), which were introduced during the course assignments and recitations, the model achieved more stable subject-independent performance. Via a channel-wise Feature Attention mechanism with a DAE-ResNet architecture, a robust classification pipeline was established. This model, further enhanced by an iterative Teacher-Student pseudo-labeling strategy, focuses on learning stable latent representations rather than overfitting to subject-dependent noise. This report presents the design, implementation, and evaluation of the proposed semi-supervised framework which achieved a local macro F1-score of 0.4980 and 0.47644 F1-score in Kaggle competition.

II. Datasets and Preprocessing

A. Data Description and EDA

The provided dataset consists of 22,496 training samples and 10,656 unlabeled test samples, with each sample represented by a 1793-dimensional feature vector. To interpret the structure of this high-dimensional manifold, t-SNE (t-Distributed Stochastic Neighbor Embedding) was utilized. t-SNE is a non-linear dimensionality reduction technique particularly suited for embedding high-dimensional data into a low-dimensional space for visualization by preserving local structures.

As illustrated in Fig. 1, the t-SNE projection reveals that the emotion classes are not linearly separable and exhibit a complex manifold structure with significant overlap. Furthermore, the analysis of class frequencies, summarized in Table I, highlights a severe class imbalance. Specifically, Class 3 (HVHA) represents the smallest fraction of the dataset, which poses a significant risk for biased classification towards majority classes.

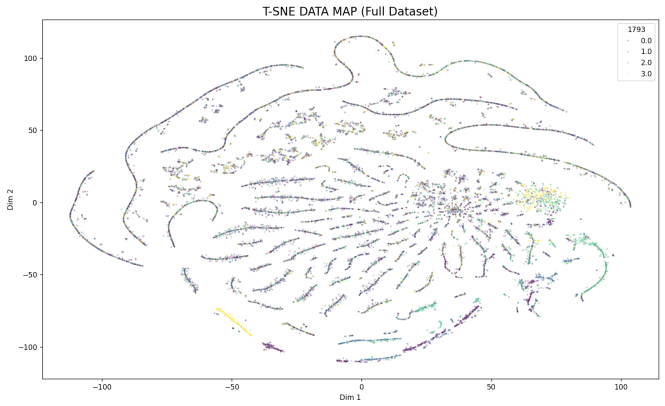

Fig. 1. t-SNE Visualization of the Train Data

B. Data Preprocessing

The preprocessing stage was designed to transform raw, high-dimensional physiological features into a format suitable

for deep neural representations. Based on the implementation in `main.py`, the pipeline follows these rigorous steps:

1) Feature Scaling and Outlier Management: During the Exploratory Data Analysis (EDA), extreme outlier values reaching magnitudes of 10^{32} were detected among the 1793 features. Given that such values are physically impossible to measure in human context, they were identified as sensor noise or recording artifacts. Traditional scaling methods, such as `StandardScaler`, proved ineffective as they were heavily biased by these extreme magnitudes.

- **Implementation Detail:** A `QuantileTransformer` with a Gaussian output distribution was utilized. This method maps the features to a normal distribution, effectively neutralizing the impact of 10^{32} scale outliers.
- **Global Fitting Strategy:** The scaler was fitted on a concatenation of both training and unlabeled test features using `np.vstack((X, X_test_kaggle_raw))`. Since the test set contains only features without class labels, this global fit is statistically valid and ensures that the model is exposed to the entire distributional range of the features, preventing distribution shift during inference.

2) Data Augmentation (Mixup): To enhance the model's robustness and prevent overfitting to the limited labeled samples, Mixup augmentation was integrated into the training loop.

- **Implementation Detail:** Virtual training examples are created by taking convex combinations of feature vectors and labels using a mixing coefficient $\lambda \sim Beta(0.4, 0.4)$.
- **Reasoning:** This encourages the model to learn smoother decision boundaries, making the classifier less sensitive to subject-specific variations.

III. Methods

The used deep learning framework named **Attentive Compact DAE-ResNet** utilizes a hybrid architecture that integrates channel-wise attention mechanisms with a Denoising Autoencoder (DAE) enhanced by Residual connections. This design was selected to better handle noise and instability observed during preliminary experiments on high-dimensional physiological features, while residual connections helped stabilize training. The architecture and the learning pipeline can be seen in Figure 2 (Schematic illustration of the learning pipeline was prepared via AI tools).

The implementation detail is given below:

- **Feature Attention Module:** To handle the 1793-dimensional feature space, a dynamic weighting mechanism is employed. A bottleneck structure (FC $\rightarrow$ ReLU $\rightarrow$ FC $\rightarrow$ Sigmoid) was used to reweight input features, as preliminary trials showed that uniform feature scaling led to unstable training behavior.
- **Denoising Autoencoder (DAE) with ResNet:** The encoder compresses input features into a 128-dimensional bottleneck, while the decoder serves as an auxiliary task for input reconstruction. Residual connections are integrated within the encoder to facilitate stable gradient flow.

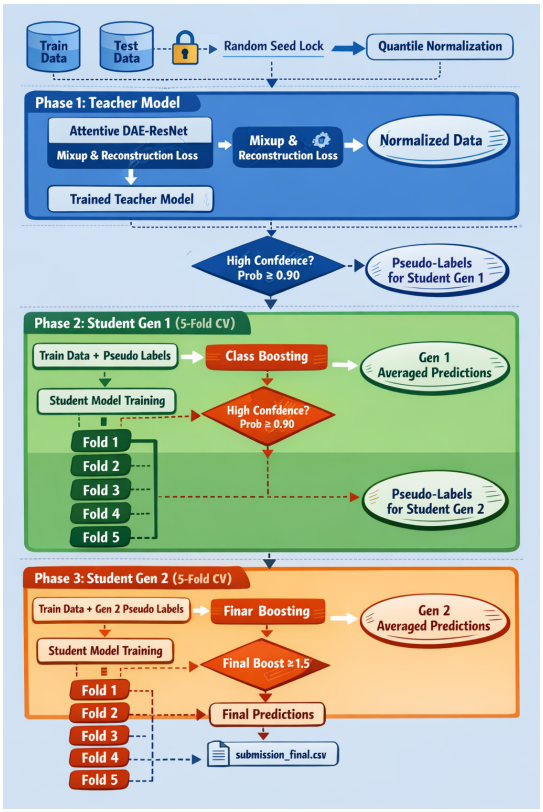

Fig. 2. Learning pipeline with Denoising Autoencoder architecture

A. Network Architecture

The model consists of three integrated modules:

1) **Feature Attention Module:** A channel-wise attention mechanism (FC $\rightarrow$ ReLU $\rightarrow$ FC $\rightarrow$ Sigmoid) scales the input features, allowing the network to suppress irrelevant noise and focus on informative signals.
2) **Denoising Autoencoder (DAE):** The network features a bottleneck structure (Encoder-Decoder). A multi-task loss function combining Cross-Entropy (L_{CE}) for classification and Mean Squared Error (L_{MSE}) for reconstruction is utilized:

$$L_{total} = L_{CE}(y, \hat{y}) + 0.1 \times L_{MSE}(x, \hat{x}) \quad (1)$$

The reconstruction task acts as a regularizer, ensuring the latent representation preserves the essential structure of the input data.
3) **ResNet Connections:** Residual connections are employed to facilitate gradient flow and allow deeper feature extraction.

B. Iterative Self-Training (Teacher-Student)

To exploit the unlabeled test data, a semi-supervised strategy was adopted:

- **Phase 1 (Teacher):** A base model is trained on the full labeled training set without subject-based splitting in order to maximize representation learning capacity.

- **Phase 2 (Pseudo-Labeling):** Labels for the test set are predicted by the Teacher. Samples with a confidence score ≥ 0.90 are selected as "Pseudo-Labels" and added to the training set.
- **Phase 3 (Student):** A new Student model is trained on the augmented dataset (Labeled + Pseudo). This process was iterated twice (Gen 1 and Gen 2).

From an implementation perspective, the model was trained using the AdamW optimizer with weight decay to reduce overfitting in the high-dimensional parameter space. Batch normalization layers were placed after linear transformations to stabilize training dynamics, while dropout was applied selectively to prevent co-adaptation of features. All experiments were conducted with a fixed random seed to ensure reproducibility across folds.

C. Validation Strategy

Group K-Fold Cross-Validation ($k = 5$) was used, grouping by `person_id`. This choice was necessary because early experiments without subject-based grouping resulted in unrealistically high validation scores, indicating strong data leakage.

IV. Results and Conclusions

A. Experimental Results

The performance of the proposed framework was evaluated using a rigorous 5-fold Group Cross-Validation strategy. A gradual performance improvement was observed as additional high-confidence pseudo-labeled samples were incorporated into the training set across generations.

1) Generation 1 Performance: In the initial phase of semi-supervised learning, **3,488** high-confidence samples were integrated into the training set based on the Teacher model's probabilistic outputs ($T > 0.90$). Under this configuration, the Student 1 model achieved an average macro F1-score of **0.4654**. The fold-specific performance metrics are detailed in Table I.

2) Generation 2 Performance: By applying a class-specific probability boosting factor (1.5x for Class 3) to the Student 1 predictions, the pseudo-label pool was expanded to **5,551** samples. This refined dataset, coupled with the Student 2 architecture, improved the average Group CV F1-score to **0.4980**. The comparative progression is summarized in the following table:

TABLE I
Macro F1-Score Comparison: Baseline (Stage 0) vs. Iterative Student Generations (Grouped)

Fold	Stage 0 (Standard k-Fold)	Student Gen 1 (Group k-Fold)	Student Gen 2 (Group k-Fold)
Fold 1	0.9208	0.3724	0.3934
Fold 2	0.9182	0.6725	0.6969
Fold 3	0.9108	0.5064	0.5154
Fold 4	0.9216	0.3593	0.3549
Fold 5	0.9174	0.4163	0.5294
Average	**0.9177**	**0.4654**	**0.4980**

B. Discussion of Results

The experimental results provide a granular view of the model's evolution across different validation schemes. The **Stage 0 baseline** yielded a deceptive F1-score of **0.9177**. This performance was diagnosed as a direct consequence of **data leakage**; by failing to partition the data based on subject identifiers, the model effectively exploited intra-group correlations rather than learning universal feature mappings.

Transitioning to **Group 5-Fold Cross-Validation** in Phase 2 established a realistic performance floor. Although this "detoxification" initially lowered the apparent F1-scores compared to Stage 0, it ensured that the Attentive DAE prioritized subject-invariant characteristics over latent group identifiers.The largest fold-level improvement was observed in Fold 5, where the macro F1-score increased in Generation 2. This confirms that iterative refinement of decision boundaries through boosted pseudo-labels (increasing from **3,488** to **5,551** samples) successfully compensates for the complexity of subject-independent recognition.

C. Final Performance and Environmental Variations

The proposed model achieved a local Group Cross-Validation macro F1-score of **0.4980**. When evaluated on the Kaggle competition test set, the submitted predictions resulted in a leaderboard score of **0.47644**. However, executing the same codebase entirely within a Kaggle kernel environment and submitting the resulting predictions yielded a lower score of **0.44722**.

This discrepancy can be explained by differences in execution environments and evaluation conditions. Although the training and inference logic remained unchanged, variations in numerical libraries (NumPy 1.26.4 locally versus NumPy 2.0.2 on Kaggle), floating-point precision, and random state handling during GPU-accelerated training can lead to non-identical model parameters and prediction distributions. In addition, the competition test set is fully unseen and unlabeled, which amplifies the sensitivity of performance metrics to small numerical or stochastic variations. These observations highlight the importance of controlled evaluation settings when comparing local validation results with competition-based scores.

D. Extended Discussions

Before finalizing the proposed architecture, more than 90 experimental trials were conducted. These experiments included a range of ensemble-based approaches, such as Gradient Boosting models (LightGBM, XGBoost, CatBoost) and Support Vector Machines with RBF kernels, as well as different voting strategies. Under subject-independent evaluation, several of these complex ensembles exhibited overfitting and unstable validation behavior.

In contrast, a simplified architecture combining Denoising Autoencoders (DAE) with channel-wise Attention mechanisms showed more consistent performance. The reconstruction objective of the DAE acted as an effective form of structural regularization, encouraging the model to learn stable latent

representations rather than relying on subject-specific correlations. This observation aligns with the Occam's Razor principle, where purposeful architectural choices proved more effective than increasing model complexity.

It is also important to distinguish between Kaggle competition settings and academic evaluation protocols. While Kaggle leaderboards provide a practical benchmark on a fixed hidden test set, they are sensitive to implementation details, numerical precision, and stochastic training effects. In this project, Group Cross-Validation was therefore prioritized as the primary evaluation strategy, as it prevents subject-level data leakage and offers a more reliable estimate of subject-independent generalization. Kaggle scores are reported as complementary results rather than definitive performance indicators, ensuring that model development was guided by methodological considerations rather than leaderboard optimization.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning," presentation slides, UC Davis ECS289G, 2016. [Online]. Available: https://web.cs.ucdavis.edu/ yjlee/teaching/ecs289g-fall2016/xiaoyun.pdf. Accessed Jan. 2026.

[2] P. Muruganantham, S. Wibowo, S. Grandhi, and N. Islam, "A Deep Learning Approach for Dealing with Tabular Data in Crop Classification," in *Proc. 15th Int. Conf. Information and Communication Technology Convergence (ICTC)*, Jeju Island, Republic of Korea, 2024, pp. 2054–2059, doi: 10.1109/ICTC62082.2024.10826760.

[3] Scikit-learn, "sklearn.manifold.TSNE - t-Distributed Stochastic Neighbor Embedding," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html. Accessed: Jan. 2026.

[4] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in ICLR, 2018.

[5] M. Ali et al., "A Globally Generalized Emotion Recognition System Involving Different Physiological Signals," Sensors, vol. 18, no. 6, p. 1905, 2018.

[6] C. D. Lima et al., "tqdm: A Fast, Extensible Progress Bar for Python," https://tqdm.github.io/, accessed Jan. 2026.

[7] Scikit-learn Developers, "QuantileTransformer," https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.QuantileTransformer.html. Accessed: Jan. 2026.