Ayushi Kate
COE379L
12 December 2025

Project 03 Report

Stress Detection from Photoplethysmography Data

1. Introduction and Problem Statement

Stress impairs concentration, memory, and decision making, which is not ideal for students that are shooting for high grades and taking many classes. Chronic stress makes it difficult for students to be engaged in their classes and perform well in their coursework. University students specifically experience high levels of stress, and without monitoring, stress can build until it becomes burnout. Understanding a person's stress helps identify early signs of mental-health concerns so students can get help before it becomes long-term exhaustion.

Stress and mental health monitoring is a highly emerging application area for machine learning interfacing with wearable devices. Wearables that are enhanced with ML models can be more effective for continuous monitoring of daily stress, detecting burnout symptoms early, or catching anxiety or PTSD symptoms. Stress is one of the physiological metrics that is the hardest for the everyday person to notice, because it is something that fluctuates and is not easy to see in the moment. Stress-focused wearables are especially critical for students in today's day and age, as a robust tool like this one can help promote mental well-being and take steps towards preventive mental health care.

This project aims to eventually integrate with a hardware system that utilizes photoplethysmography (PPG) sensor data, a sensor commonly used in wearables. A PPG uses light reflected in the human tissue to measure changes in blood volume to track physiological parameters like heart rate variability (HRV) and blood pressure. By leveraging machine learning, I focused on building a binary classification model that can distinguish between stressed and non-stressed physiological states.

2. Data Sources and Technologies Used

2.1 Data Preparation

The data preparation phase began with an examination of the initial dataset composition. The dataset used was sourced from Kaggle.com from a study published in 2022 by Anwar et al. (Anwar & Zakir, 2022). The study included 27 undergraduate student participants, 15 males and 12 females with a mean age of 21. Since they were all bachelor's students, this sample was best for my problem's focus on college students. The data is collected from a low-cost PPG sensor placed behind the earlobe in the same location for all of the students.

The data came prelabeled as stress and nonstress baseline episodes, which allowed me to better visualize the information. The method that they were evaluated between stressed and not stressed is by inducing stress using the Stroop test. The Stroop test is an assessment designed to induce and test a participant's state of mental stress. The test requires semantic and response conflicts which challenge a person's prefrontal cortex and other cognitive control systems (Lee, Byun, Lee, & Ha, 2025). This makes it an effective method for assessing a person's stress levels, because their inability to properly respond to the task is a strong indicator of inhibited performance of the prefrontal cortex, which typically works to suppress inappropriate responses and maintains focused behavior (Lee, Byun, Lee, & Ha, 2025). With the labels of stress and non-stressed state based on the results of the Stroop test, I was able to perform supervised learning with the dataset.

2.2 Training Procedure

All preprocessing steps were applied to the data, including scaling (StandardScaler), filtering (Savitky-Golay), and normalization.

The dataset was split into 80% training data and 20% testing data. I generated a random sample of stress and non-stressed data points and separated them into X_train, X_test, y_train, and y_test. To ensure reproducible results, I used the random seed 42.

I fit all of the classifiers using the training dataset. During training, the algorithms learned underlying patterns related to HRV metrics and PPG waveform characteristics which differentiate stressed and relaxed states.

3. Methods Employed

3.1 Data Exploration & Visualization

I handled the data using Pandas and NumPy to efficiently conduct exploratory data analysis (EDA), followed by visualization with Matplotlib and Seaborn to identify redundant features and evaluate correlations before preprocessing.

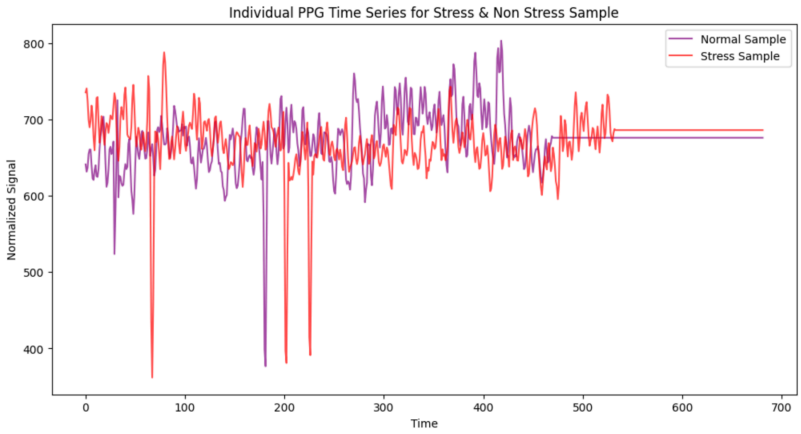

Figure 1: Individual PPG Time Series

Through EDA, I was able to understand the structure and distribution of PPG features. I initially focused on identifying missing features, because there were many columns that had null values if that subject did not have information for that time period. For the null columns, I used interpolation to fill the columns.

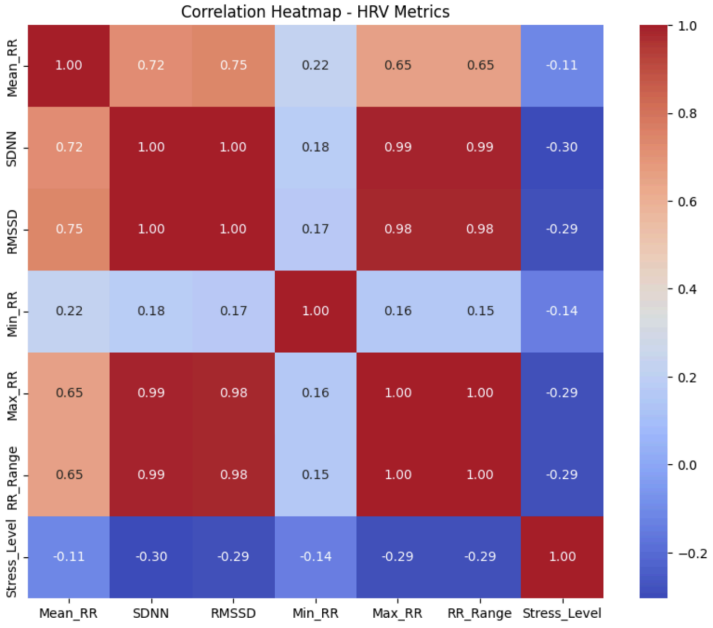

Figure 2: Correlation Heatmap of HRV Metrics

Basic features of the data, like reading in the shape and showing it on a time-series plot as seen in Figure 1, was useful before processing the data. Data was visualized using Matplotlib and Seaborn before conducting preprocessing. I generated correlation matrices, like in Figure 2, to evaluate what features are most valuable to the model.

A key observation was that episodes of stress had higher PPG signal irregularity and changes in amplitude. Visualizing these patterns supports the selection of features and helps inform the model's weighting of certain features to make it a better predictor for stress detection.

3.2 Feature Engineering

Feature engineering was extremely critical to transform the raw PPG signal data into meaningful descriptors that actually relate to physiological stress and Heart Rate Variability (HRV) data.
Time-domain features include minimum, maximum, median, mean, standard deviation, peak to peak intervals, variance, skew, and kurtosis. For frequency domain features, we applied a Fast-Fourier Transform (FFT) to be applied to the time-domain data values to extract values like power spectral density and entropy. Because stress often leads to reduced HRV and erratic patterns, capturing these features through feature engineering is especially important. Additionally, using StandardScaler to do feature scaling ensures that all features contributed proportionally during training.

4. Results

4.1 Model Algorithm & Fine Tuning

I utilized machine learning model algorithms to see what yields the best results. As per my research, Support Vector Machine (SVM) and Random Forest (RF) have been used in research papers before to accurately predict PPG data. Classical machine learning techniques were effective as long as proper feature engineering was done. I also chose to use XGBoost because it is a very robust model that we used in class a lot and I thought it would apply well to dynamic PPG signals.

Based on the initial performance comparisons, the best performing models, which were RF and XGBoost, were selected for hyperparameter tuning. Cross-validation of the models using testing data is necessary to fine-tune the hyperparameters of the model. By finding the right parameters, I was able to develop a more accurate model. For RF, the parameters I focused on were 'n_estimators', 'max_features', 'max_depth', 'min_samples_split'. For XGBoost, I tuned the parameters 'n_estimators', 'max_depth', 'learning_rate'. In this scenario, it is more important to me to minimize false negatives, because not identifying stress in a stressful situation is more detrimental to the user. Thus, I set the scoring method to roc_auc instead of accuracy. The final tuned models had a good balance between performance and generalization, so they weren't overfitting on the training data.

4.2. Model Evaluation

Model	Accuracy
SVM	0.5455
Random Forest	0.8182
XGBoost	0.9091

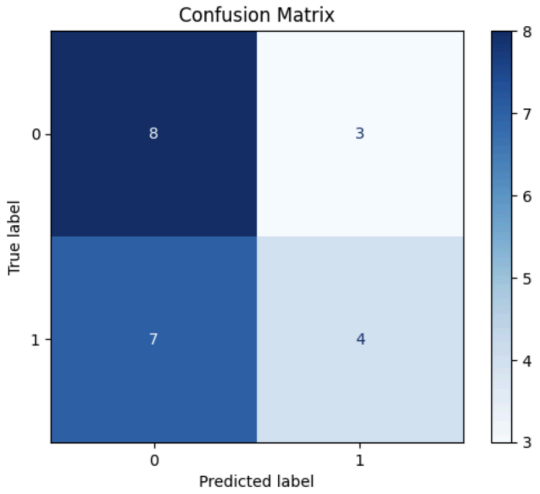

Figure 3: SVM Confusion Matrix

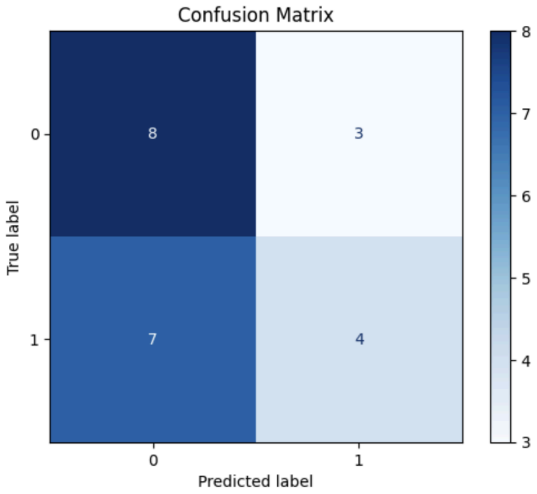

Figure 4: Random Forest Confusion Matrix

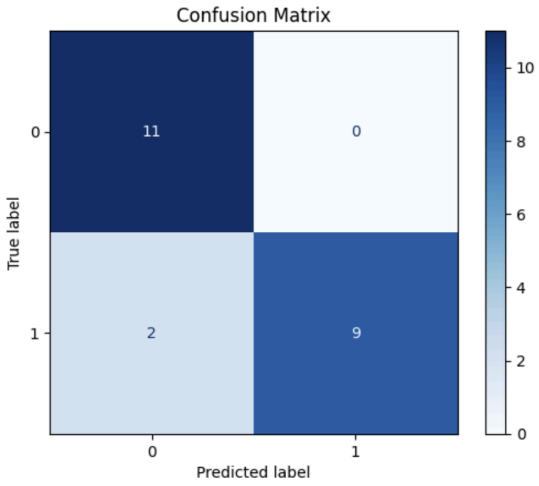

Figure 5: XGBoost Confusion Matrix

Hyperparameter Tuning

RF

```
Fitting 3 folds for each of 54 candidates, totalling 162 fits
Best RF parameters: {'max_depth': None, 'max_features': 'sqrt', 'min_samples_split': 5, 'n_estimators': 100}
Best RF ROC_AUC: 0.740702947845805
RF Test Accuracy: 0.7727272727272727
```

XGB

```
Best XGB parameters: {'learning_rate': 0.05, 'max_depth': 5, 'n_estimators': 100}
Best XGB ROC_AUC: 0.8115646258503402
XGB Test Accuracy: 0.8181818181818182
```

2.3. Class Prediction

Overall, the models performed well in the binary classification task. The final classifier was evaluated on the unseen test set using accuracy, precision, recall, and F1-score. Confusion matrices seen in Section 2.4 were generated to visualize correct vs. incorrect predictions. SVM really struggled with predictions, but Random Forest had some false negatives. With XGBoost, we were able to minimize the false negatives even further. Since RF and SVM did the best initially, I chose to do hyperparameter tuning on those two. I was able to achieve an XGB ROC_AUC of 0.8116, which was really high.

It was able to achieve a really high accuracy on the validation set, so I am able to trust its ability to generalize to other data. However, more validation is necessary to guarantee the model's performance in other contexts, perhaps with other datasets to confirm its results. Regardless, I am largely confident in the reliability of this model given its performance with this dataset. Future work can involve incorporating deep learning methods, expanding the dataset, and integrating the model into hardware for testing.

5. References

Anwar, T., & Zakir, S. (2022). Machine learning-based real-time diagnosis of mental stress using photoplethysmography. Journal of Biomimetics, Biomaterials and Biomedical Engineering, 55, 154–167. https://doi.org/10.4028/p-01r9mn.

Lee, J., Byun, K., Lee, M., & Ha, M. (2025). Disrupted practice effects and altered prefrontal activation in mild cognitive impairment: An fNIRS study using the Stroop task. Brain & Behavior, 15(11), 1–13. https://doi.org/10.1002/brb3.70942.