Stress Level Prediction from Wearable Sensor Data

Course: MSDS692 – Data Science Practicum 1

Author: Balarama Raju Saripalli
Email: bsaripalli@regis.edu
GitHub: https://github.com/varma1234/stress-project

Table of Contents

Project Overview

This project aims to predict stress levels from physiological and motion data collected passively via wearable devices. Stress detection is an increasingly vital field in health informatics, as it enables insights into personal well-being, workplace safety, and clinical monitoring applications.

By leveraging multimodal bio signals — including heart rate variability (HRV), electrodermal activity (EDA), respiration, temperature, and accelerometer data — the objective is to classify stress versus non-stress states and extract key physiological patterns associated with stress responses.

The goal is to develop an end-to-end predictive and interpretive pipeline with real-world applicability in wellness tracking and affective computing.

Problem Definition and Motivation

The core challenge is to construct a robust classifier for stress detection from complex, noisy, multimodal wearable sensor data. This involves carefully defining the stress detection problem, including labelling from available datasets and managing data heterogeneity.

The problem is framed as:

- Supervised classification task (stress vs. non-stress) using time-series physiological data
- Incorporating time-series prediction techniques and producing meaningful visualizations and dashboards
- Leveraging publicly available datasets while integrating domain-specific feature engineering

Stress monitoring makes significant contributions to mental health research, workplace safety programs, and healthcare diagnostics.

Data Description and Collection Effort

Data was sourced from publicly available repositories, integrating multiple datasets for comprehensive coverage:

- WESAD (Wearable Stress and Affect Detection): Contains multimodal recordings (ECG, EDA, accelerometer, temperature) labelled for stress and affective states.
- PhysioNet Wearable Exam Stress Dataset: Physiological recordings with unlabelled or semi-labelled data, including EDA, HR, and accelerometer signals.
- HRV Sleep/Lifestyle Dataset (Springer Nature): Pre-extracted HRV features with sleep diary annotations.

Data Structure

- Multimodal time-series signals sampled at various frequencies (ECG/HR, EDA, respiration, accelerometer).
- Labels: Stress and non-stress conditions, proxy labels where necessary.

Collection Effort

- Data was publicly available but required significant manual processing and alignment of multiple modalities, including:
    - Timestamp synchronization
    - Sliding window segmentation (e.g., 4s windows)
    - Feature extraction of statistical and domain-specific metrics (mean, std, RMSSD, LF/HF ratio)
    - Handling missing data and noise

Analysis Approach and Methodology

Stage	Description
Data Preprocessing	Noise filtering, timestamp cleaning, normalization, and segmentation into windows
Feature Engineering	Extraction of HRV metrics, accelerometer activity features, respiration parameters
Baseline Modeling	Logistic Regression, Random Forest, XGBoost
Deep Learning	LSTM for sequence modeling, 1D CNN for temporal feature extraction, Autoencoder for anomaly detection
Multimodal Fusion	Integration of HRV, EDA, and accelerometer data streams
Evaluation	Metrics including Accuracy, F1-score, AUROC, Confusion Matrix
Visualization	PCA/t-SNE projections, ROC curves, SHAP feature importance plots
Dashboard	Interactive visualization via Tableau Public or Power BI

Key Tools and Libraries Used

- Python 3.10+ environment
- Core libraries: Pandas, NumPy, scikit-learn, LightGBM
- Deep learning: TensorFlow/Keras
- Visualization: Matplotlib, Seaborn, Plotly, Tableau Public, Power BI
- Environment: Google Colab, Kaggle Notebooks for free GPUs

Practicum Deliverables and Evaluation Rubric Alignment

Deliverable Aspect	Details & Highlights
Problem Source & Definition	Novel stress detection problem defined using multiple public dataset integrations
Problem Difficulty Level	High due to multisource noisy physiological signals and label proxying
Data Collection & Cleaning Effort	Manual synchronization, multiple signal modalities, and preprocessing
Feature Engineering	Intuition and ML-driven feature selection based on physiological relevance
Tools & Coding Effort	Fully coded pipeline in Python with deep learning and ML models
Analysis Methods & Optimization	Included hyperparameter tuning, imbalance handling (SMOTE, class weights)
Scientific Rigor	Appropriate metrics and interpretability via SHAP explanations
Problem Solving & Learning	Managed complex data challenges and runtime optimization independently
Project Time Management	Adhered to detailed 8-week plan with milestone completion
Online Presence	Well-structured, documented GitHub repo with code and summaries
Code Organization & Comments	Properly segmented, commented notebook scripts and pipeline steps
Project Summary & Visualizations	Clear project summary on landing page, rich interactive visualizations

Week-by-Week Timeline and Progress Summary

Week	Activities & Status
1	Proposal submission, project planning, task and dataset finalization
2	Data download, structure verification, environment setup
3	Preprocessing, cleaning, feature extraction
4	Baseline ML models: Logistic Regression, Random Forest, XGBoost implementation
5	Deep learning model construction: LSTM, 1D CNN, Autoencoder
6	Class imbalance treatment, hyperparameter tuning, multimodal fusion experiments

Week Activities & Status

7 Evaluation metrics, visualization, dashboard creation, GitHub cleanup

8 Final report writing, presentation preparation, submission

Roadblocks and Learnings

- Managing long runtimes and computational costs for deep learning models using Google Colab's free GPU, mitigated by early stopping and caching techniques.
- Synchronizing and fusing signals collected at different frequencies and modalities required intricate timestamp alignment and feature windowing.
- Practical experience gained in balancing model complexity, interpretability, and reproducibility.
- Learned to independently overcome data preprocessing challenges and maintain well-documented code for review.

Online Presence and Documentation

A comprehensive GitHub repository is maintained with:

- Organized notebooks with stepwise, commented code covering all analysis phases
- A landing-page summary that articulates problem motivation, approach, and outcomes clearly, accessible even to non-technical reviewers
- Visualizations embedded or linked on the main page for quick insight
- README and markdown files narrating the project flow, decisions, and lessons learned
- Issue tracking and project board for task management

Visualizations and Dashboard Summary

- Multidimensional data exploration with PCA and t-SNE cluster plots
- ROC and precision-recall curves comparing model performance
- Time-series stress probability trends showcasing temporal fluctuations
- SHAP (SHapley Additive exPlanations) used for feature importance, improving interpretability
- dashboard showing daily stress patterns, correlations with sleep, and activity impact

Project Structure

- ➔ stress-project/data_raw
  Wesad
  Physionet
  exam/hrv_sleep
- ➔ stress-project/ notebooks/ Colab Notebooks
  01_peek_data.ipynb
  02_preprocess_physionet.ipynb
  03_preprocess_wesad.ipynb
  04_preprocess_hrv.ipynb
  05_week4_Implement baseline ML models_Fail.ipynb
  06_week4_unsupervised_Baseline.ipynb
  07_Week 5 Plan – Deep Learning & Advanced Model.ipynb
  08_Finalize Models & Produce Artifacts, Hyperparameter Tuning & Final Model Selection.ipynb
  09_Visualization & Dashboard.ipynb
- ➔ stress-project/ models
- ➔ stress-project/ results
- ➔ stress-project/ processed
- ➔ stress-project/ README.md

Quick Start Guide

Prerequisites: Python 3.10+, Google Colab or local Jupyter environment.
Libraries: TensorFlow, scikit-learn, LightGBM, Pandas, NumPy, Matplotlib, Seaborn, Plotly, Streamlit, Tableau Public/Power BI for dashboard building.

Run stress prediction:

bash

python scripts/predict.py --input sample_data.csv --model results/models/lstm_model.keras

Sample output:

text

Predicted Stress Level: STRESSED

Confidence: 0.87

References

- Schmidt, P., Reiss, A., et al. (2018). *Introducing WESAD: A Multimodal Dataset for Wearable Stress and Affect Detection.*
- Shaffer, F., & Ginsberg, J. P. (2017). *An Overview of Heart Rate Variability Metrics and Norms.*
- Benedek, M., & Kaernbach, C. (2010). *A Continuous Measure of Phasic Electrodermal Activity.*
- Goldberger, A. L., et al. (2000). *PhysioBank, PhysioToolkit, and PhysioNet.*
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). *"Why Should I Trust You?" Explaining the Predictions of Any Classifier.*
- Lau, C. H. (2019, January 10). 5 Steps of a Data Science Project Lifecycle. Retrieved from https://towardsdatascience.com/5-steps-of-a-data-science-project-lifecycle-26c50372b492
- Nellutla, V. (n.d.). Applying Agile IT Methodology to Data Science Projects. Retrieved from https://www.datasciencecentral.com/profiles/blogs/applying-agile-it-methodology-to-data-science-projects
- Destin Gong. Top 6 Machine Learning Algorithms for Classification . Retrieved from https://towardsdatascience.com/top-machine-learning-algorithms-for-classification-2197870ff501/