Wearable-Based Stress Detection Using Fitbit Data

Sriha Reddy Ettireddy

Department of Information Science and Technology, University at Albany, SUNY, NY, USA.
E-mail: settireddy@albany.edu

Abstract

Stress is a major risk factor for both physical and mental health, yet most automated stress detection systems are built on short laboratory studies with highly controlled tasks. At the same time, many people now wear devices such as Fitbit every day, which record sleep, heart rate, and activity in real life. Recent work in brain imaging has shown that graph-based neural networks can model relationships between signals, but it is unclear whether a similar idea helps for wearable-based stress prediction in daily life.

This project uses the Life Snaps “rais anonymized” dataset, which contains several months of daily Fitbit summaries linked to self-reported stress scores. Eleven Fitbit features related to sleep, heart rate, and activity are selected and used to build a feature-level connectivity graph based on their correlations. This structure is called an electrocardiograph network. Five models are implemented and compared on the same train, validation, and test splits: a logistic regression baseline, a graph convolutional network on the feature graph, a temporal graph plus recurrent network over seven-day windows, an attention-based dynamic graph model, and a long short- term memory sequence model on seven-day windows without any graph. Accuracy and F1 score are used for evaluation. The logistic regression baseline reaches moderate performance on the test set. Graph-based models on the feature connectivity.

The network shows only small gains in F1 score and does not clearly improve accuracy. The long short-term memory model on seven-day sequences achieves the best performance, with higher accuracy and F1 score than all graph-based models.

Keywords: Stress detection; Wearable devices; Fitbit; Graph neural networks; LSTM; Time-series analysis; LifeSnaps dataset; Artificial intelligence.

1. Introduction

Stress is a key risk factor for both physical and mental health problems. High and prolonged stress is linked to poor sleep, cardiovascular disease, anxiety, and reduced work performance. At the same time, many people now wear devices such as Fitbit or smartwatches every day. [1] These wearables continuously collect signals like heart rate, steps, sleep duration, and activity intensity in real-life settings. This creates an opportunity to build data-driven systems that can monitor stress and support early intervention without using expensive or invasive medical devices [2].

Most existing stress detection systems based on wearable data use traditional machine learning methods such as logistic regression, support vector machines, or random forests on tabular features (for example, daily averages or simple statistics) [3]. These models treat each feature as independent and do not model the relationships between them. However, physiological variables are clearly related: sleep duration and sleep efficiency influence resting heart rate; very active minutes are related to steps, calories, and sedentary time; heart rate variability changes with both physical load and psychological stress[1]. Ignoring these dependencies may limit model performance and interpretability.

In parallel, recent research in brain imaging and EEG has shown that graph neural networks (GNNs) can capture complex relationships between channels or brain regions by representing them as nodes in a graph and learning over connectivity patterns. In these works, researchers first build a functional or structural connectivity graph and then apply GNNs such as graph convolutional networks (GCN), dynamic GCNs, or contrastive GNN frameworks. These models often outperform standard deep learning baselines because they use both the signal

and the graph structure. Inspired by this, I propose to treat physiological features from a wearable device as nodes in a graph, forming an Electro-Cardio Graph Network (ECGN). Edges in this ECGN represent strong correlations between Fitbit features such as resting heart rate, sleep metrics, and activity measures [3].

This project focuses on the Life Snaps "rais_anonymized" dataset, which is a multi-modal longitudinal dataset collected in the RAIS project and shared on Zenodo. It includes Fitbit data, ecological momentary assessments, and psychological surveys over several months for dozens of participants. From this rich dataset, I focus on the daily Fitbit summary table, which links physiological signals with a daily stress score[4]. This setting is close to real-world "digital phenotyping," where stress is monitored passively through everyday device usage.

From a skills perspective, I am a suitable candidate for this project because I already have experience with data engineering, Python, and deep learning. In previous work, I have implemented classical machine learning models, convolutional neural networks, and graph learning pipelines. I am comfortable with tools such as Pandas, PyTorch, and scikit-learn, and I have already worked with health-related datasets in other courses [5]. This project needs exactly these skills: data preprocessing, handling time-series and multi-modal data, building deep learning models, and evaluating multiple baselines in a fair way. I am also familiar with the basics of graph neural networks from prior reading and tutorials, which helps me adapt EEG-based GNN ideas to a new problem.

The topic interests me for two main reasons. First, it connects mental health and AI, which is an area with a strong social impact. Many students and workers suffer from stress, but do not always seek help early. If we can detect stress patterns from wearables in a reliable way, we can support self-monitoring and timely intervention [6]. Second, this project lets me explore graph- based modeling beyond its usual brain or social network applications. Building an ECGN on Fitbit features and

comparing it with a strong LSTM baseline is a good way to learn what graph structure really adds in practice. From an industry point of view, this work is related to the broader digital health and wearable analytics ecosystem.

Companies such as Fitbit, Apple, and Garmin already provide sleep and activity scores, and some research prototypes estimate stress, readiness, or recovery from physiological signals. However, many of these systems are proprietary and use undisclosed models. Academic projects like this one help to explore open methods and to understand which modeling choices (graphs vs sequences, static vs dynamic connectivity) really matter. In the long term, methods developed here could be extended to more complex multimodal data or deployed as part of health dashboards and personalized coaching systems. Overall, this introduction sets the scene for the project.

The key idea is to use an ECGN over Fitbit features, inspired by EEG GNN work, and to compare graph models with a strong temporal baseline [7]. The central research question is: **Does modeling relationships between Fitbit features as a graph actually improve stress prediction, or are short-term temporal patterns over several days more important?** The rest of the report describes the dataset, the ECGN construction, the five implemented models (logistic regression, GCN, temporal GCN+GRU, attention/dynamic GNN, and LSTM), and a detailed comparison of their performance.

2. Literature Review

2.1 Wearable-based stress detection and digital phenotyping

Many recent studies use wearable devices to detect stress from physiological data such as

heart rate, skin temperature, electrodermal activity, and movement. A short review by [8]shows that physiological signals from wearables can predict stress in real time, but most work is still done in controlled settings and not in everyday life.[9]review detection and monitoring stress using wearables and highlighting common preprocessing steps, hand-crafted features, and machine learning classifiers used in this field [10].

Several broader reviews confirm that stress detection with wearables and machine learning is now a very active area. Review mental stress detection with wearable sensors and machine learning, and report that heart rate variability (HRV), electrodermal activity (EDA), temperature, and accelerometer features are the most common biomarkers.[10], [11] both survey machine learning and deep learning techniques for predicting stress from wearable data, and show that models such as support vector machines, random forests, and neural networks are widely used, often with fairly small datasets and limited external validation [12].

Several papers focus on deep learning for stress classification. [10] propose a multi-modal deep learning approach that combines different physiological channels from wearables for stress detection and show improved performance compared to traditional methods. [12] Develop a deep neural network that uses image coding of wearable sensor data to detect stress, again working on lab-style stress tasks.[13], [14] build context-aware deep learning approaches for stress using heart rate and EDA signals [15], [16].

The WESAD dataset by [3] is the most common benchmark for wearable stress detection. WESAD is a multimodal dataset with ECG, EDA, respiration, temperature, and accelerometer data from chest-worn and wrist-worn devices, collected in a lab with stress and amusement conditions. Many later works base their analysis or new models on WESAD and report very high accuracies in this controlled environment.

Beyond pure stress detection, digital phenotyping uses signals from smartphones and wearables to track stress, anxiety, and other mental health outcomes. Review smartphone-based digital phenotyping for stress, anxiety, and mild depression and conclude that passive sensing can capture useful behavioral patterns, but that validation and context remain problems. Review digital phenotyping with smartphones and wearables and discuss both technical and ethical challenges. discuss real-time assessment of stress with digital phenotyping and stress the need for better ground truth labels and more ecologically valid designs. More recently, [17]shows that large-scale AI models applied to wearable signals can characterize psychiatric disorders and identify genetic associations, which underlines how powerful these signals can be when combined with advanced models.[18]review the broader clinical benefits of digital phenotyping for health monitoring.

In summary, empirical work shows that wearable-based stress detection is feasible, that HRV, EDA, and activity measures are key predictors, and that deep learning models often perform better than classical methods on lab-style datasets like WESAD. However, much of this work is still based on short experiments in controlled environments and uses relatively simple model structures (MLP, CNN, basic RNNs) on tabular or sequence features.

2.2 LifeSnaps and longitudinal in-the-wild wearable data

Most stress datasets, including WESAD, are short and collected in a lab. In contrast, LifeSnaps is a multi-month, in-the-wild dataset, introducing LifeSnaps as a 4-month multi-modal dataset from 71 participants, with more than 35 data types at second-to-daily granularity and over 71 million rows of data, including Fitbit signals, phone-based ecological momentary assessments, and surveys. LifeSnaps is part of the RAIS project and is designed specifically to support research on digital phenotyping and behavior monitoring in everyday life. It presents UnStressMe, an explainable stress analytics system built on top of LifeSnaps,

and shows that Fitbit-based signals such as stress score, sleep duration, calories, and steps can be combined with self-tracking to support user reflection and self- management. There is also public exploration work on LifeSnaps preprocessing and visualization, for example, a GitHub repository that constructs unified daily and hourly data frames for further analysis.

Compared to WESAD and other lab datasets, LifeSnaps gives longitudinal, real-life Fitbit data with linked daily stress scores. This makes it suitable for studying how stress evolves over weeks and months, and how daily patterns in sleep, heart rate, and activity relate to self-reported stress.

2.3 Machine learning and deep learning for stress in everyday settings

While many works focus on lab stress tasks, some studies and reviews explicitly highlight the need for real-life validation, noting that most studies validate stress models in controlled environments and that objective stress evaluation in everyday settings remains underexplored. Detection and monitoring reviews such as also point out that wearable stress models often do not generalize well outside the lab and that context and label quality are major issues.

Several empirical works attempt to move from the lab to real life, present a three-stage validation combining WESAD and more realistic data to study how well models trained in the lab perform in naturalistic settings. Studies like[2], [10] use multi-modal deep learning on wearable data, often with CNN or simple recurrent layers, and report strong results but still mostly within constrained stress tasks.

Overall, the empirical literature shows that:

- Most stress models from wearables use flat feature vectors or simple sequences.
- Deep learning improves performance but is often applied to lab data or synthetic representations (e.g., images).
- There is limited work on long-term, daily, in-the-wild datasets like LifeSnaps, especially for comparing different model families on the exact same data.

4. Graph neural networks for EEG and brain connectivity

At the same time, there is a fast-growing body of work on graph neural networks (GNNs) for EEG and brain connectivity. These studies are important because they show how to build and use graphs over physiological signals. [19]propose self-supervised graph neural networks for EEG seizure detection and classification (eeg-gnn-ssl). They represent EEG channels as nodes and use either geometry-based or dynamic connectivity graphs, combined witha self supervised pre-training that predicts future EEG segments.[14]develop improved GNN architectures for EEG-based emotion recognition and show that using connectivity graphs improves accuracy over CNN and RNN baselines. A recent review on graph neural networks in EEG-based emotion recognition summarizes many such models and shows consistent gains when graph structure is modeled properly.

Several works introduce dynamic or contrastive graph learning ideas. Propose A-GCL, an adversarial graph contrastive learning framework for fMRI graphs, and show that learning robust graph embeddings improves diagnosis performance. use functional graph contrastive learning for hyper scanning EEG to model emotional contagion in days. [3]propose AttGraph, an attention-based dynamic GNN for EEG emotion recognition, which learns time-varying connectivity with multi-dimensional attention, present SS-EMERGE, a hybrid self-supervised framework for emotion recognition using GNNs on EEG, and show that combining SSL.with graph modeling improves cross-subject generalization.[5]introduce adaptive node feature extraction for graph-based EEG models and show that better node features significantly improve performance.

More broadly, [4]graph contrastive learning for brain networks is reviewed by recent works that analyze models such as BrainGNN, dynamic spatio-temporal GCNs, and multi-view graph networks for clinical diagnosis. These studies underline two key points that are

relevant here:

1. **Physiological signals can be naturally represented as graphs**, where nodes are sensors or regions and edges reflect structural or functional connectivity.
2. **GNNs often outperform traditional deep models** when this connectivity is modeled carefully.

However, almost all of this work focuses on EEG or fMRI, not on everyday wearable data like Fitbit.

2.5 What is known and where is the gap?

From this literature, we can say that:

- Wearable-based stress detection is a mature topic with many models, especially on WESAD and similar lab datasets.
- Digital phenotyping has shown that smartphone and wearable data can track stress and related mental health outcomes over time.
- LifeSnaps is a unique in-the-wild, longitudinal dataset with Fitbit data and stress scores, but only a few published works use it, and even fewer do detailed modeling of daily stress.
- In the brain domain, GNNs on connectivity graphs (EEG, fMRI) have clearly improved classification performance and robustness, and dynamic or contrastive GNNs are now common.

The gap **i**s that these two lines of work rarely meet:

- Wearable stress studies almost always treat features as independent inputs (flat vectors or sequences) and use CNN/RNN models or classical ML. They do not usually build graphs over wearable features like heart rate, sleep, and activity

[3].

- Graph-based physiological modeling is mainly done on EEG/fMRI with many channels and a clear spatial structure, not on low- dimensional wearable summaries such as daily Fitbit aggregates [6].

For LifeSnaps specifically, there is very limited work that:

- constructs a feature-level connectivity graph (like an ECGN), adapts ideas from EEG GNN models (eeg-gnn-ssl, A-GCL, dynamic attention GNNs), and compares these graph models against strong temporal baselines on the same Fitbit daily stress task.

In short, we know that temporal models on wearables can predict stress, and we know that graph models help on EEG and fMRI, but we do not know whether graph neural networks over wearable features add real value beyond a good sequence model, especially on long-term, in-the-wild datasets like LifeSnaps[7].

2.6 Contribution of this study

This project aims to fill part of that gap by:

1. Bringing graph ideas to fitbit stress data

I construct an Electro-Cardio Graph Network (ECGN) where each node is a Fitbit daily feature (resting heart rate, sleep duration, steps, etc.) and edges are based on strong correlations between these features in the LifeSnaps dataset. This translates the “connectivity graph” idea from EEG to wearable signals [15].

2. Adapting EEG GNN architectures to a new domain.

Inspired by eeg-gnn-ssl, A-GCL and attention-based EEG GNNs, I implement:

- a static GCN on the ECGN graph,
- a temporal GCN+GRU that combines ECGN with 7-day sequences, and an attention / dynamic GNN that learns edge weights under an ECGN mask.

The results show that the LSTM on 7-day sequences outperforms all ECGN-based models on accuracy and F1. This suggests that, for this Fitbit stress dataset, short-term temporal patterns are more informative than a simple correlation graph over features. This is useful for both scholarship and practice: it tells researchers and practitioners that, at least in this setting, investing effort in good temporal modeling may bring more benefit than building a simple feature connectivity graph[20].

3. Materials and methods

This project is an experimental study using an existing public dataset (secondary data analysis). The main goal is to compare different machine learning and deep learning methods for daily stress prediction from Fitbit wearable data.

3.1 Data

The data used in this study comes from the LifeSnaps "rais_anonymized" database, which was released as part of the RAIS project and is hosted on Zenodo (https://zenodo.org/records/7229547). The Zenodo record contains a MongoDB dump and a set of CSV files. I use the CSV export folder csv_rais_anonymized, and in particular the file daily_fitbit_sema_df_unprocessed.csv.

This file contains daily summary data for Fitbit Sense devices, linked to ecological momentary assessments and surveys. Each row represents one participant on one day and includes physiological and behavioral signals such as heart rate, heart rate variability, sleep,

steps, calories, and activity levels, as well as a daily stress score. For this project I focus on eleven Fitbit variables that are directly related to stress, sleep, and activity: nightly skin temperature, NREM heart rate, RMSSD (a measure of heart rate variability), calories, distance, heart rate, sleep duration, sleep duration, sleep efficiency, steps, sedentary minutes, and very active minutes. These eleven variables later become the nodes in the ECGN.

The outcome variable is the daily stress score available in the dataset. It is originally continuous. I convert it into a binary label by splitting at the median value of 75. Days with stress score below 75 are labeled as low stress (class 0), and days with stress score equal to or above 75 are labeled as high stress (class 1). This median split produces a balanced classification problem. After removing rows with missing values in any of the eleven selected features or in the stress score, the resulting dataset contains 1,608 daily records, with 754 low stress days and 854 high-stress days.

Preprocessing includes parsing the date column, sorting records by participant and date, and standardizing the eleven features using z-scores computed from the training set. For single-day models, each cleaned day is treated as an independent sample. I split the 1,608 records into training, validation, and test sets with stratification on the binary stress label to preserve class balance. For sequence-based models, I create seven-day sliding windows per participant. Only windows with complete data and a valid label on the last day are kept. This produces 1,420 windows of shape (7 days, 11 features). These windows are again split into training, validation, and test sets at the window level with stratification on the label.

3.2 Machine learning Techniques

As a classical baseline, I use logistic regression for binary classification. The input to this model is the vector of eleven standardized features from a single day. Logistic regression is a

simple linear model that estimates the probability of high stress as a logistic function of the input features. I implement it using scikit-learn with L2 regularization. Hyperparameters are kept simple and fixed because the goal is not to heavily tune this model but to provide a clean and interpretable reference point.

The logistic regression model is trained on the training set and evaluated on the validation and test sets. It does not use any graph structure and does not consider temporal context beyond one day. Its performance gives a lower bound on what can be achieved with straightforward tabular modeling.

3.3 Deep learning

All deep learning models in this project are implemented in PyTorch. They share a common training setup: cross-entropy loss as the objective function, the Adam optimizer with a learning rate of 1×10^{-3}, a batch size of 64, and up to 50 epochs of training. I use the validation set to select the best model based on F1-score and then report results on the held-out test set. All experiments are run on a CPU in Google Colab, which is sufficient given the small size of the dataset. The deep models include three graph-based architectures and one sequence model. The graph-based architectures all rely on the ECGN adjacency matrix described below, but they use it in different ways. The sequence model works directly on seven-day windows of the eleven features and does not use the graph. By keeping the optimization settings aligned across models, I reduce the chance that differences in performance are driven by training choices rather than by the model architecture itself.

3.4 Convolutional Neural Networks

All deep learning models in this project are implemented in PyTorch. They share a common training setup: cross-entropy loss as the objective function, the Adam optimizer with a

learning rate of 1×10^{-3}, a batch size of 64, and up to 50 epochs of training. I use the validation set to select the best model based on F1-score and then report results on the held-out test set. All experiments are run on a CPU in Google Colab, which is sufficient given the small size of the dataset.

The deep models include three graph-based architectures and one sequence model. The graph-based architectures all rely on the ECGN adjacency matrix described below, but they use it in different ways. The sequence model works directly on seven-day windows of the eleven features and does not use the graph. By keeping the optimization settings aligned across models, I reduce the chance that differences in performance are driven by training choices rather than by the model architecture itself.

3.5 Transfer Learning Techniques

Convolutional Neural Networks (CNNs) are widely used for image and signal processing, especially when the input can be represented as a two-dimensional grid or as a long time–frequency map. In this project, however, the available data takes the form of low-dimensional daily feature vectors and short seven-day sequences of eleven variables. There is no natural spatial grid or long, high-resolution time axis.

For this reason, I do not implement CNN-based models here. Including a CNN would require either transforming the data into an artificial image representation or working with much longer raw time-series from the wearable, which is beyond the scope of this project. Instead, I focus on graph neural networks, which are well matched to the idea of a feature connectivity graph, and recurrent models, which are well matched to short time sequences.

3.6 Deep Feature Extraction

Deep feature extraction refers to using intermediate activations of a deep network as features for a separate classifier. In this work, deep feature extraction happens inside each model, but

I do not export these features to a second-stage classifier.

In the graph models, node features are passed through graph convolution layers, and the resulting node embeddings are pooled over the graph to produce a graph-level representation. This pooled vector is a deep feature that summarizes the relationships between the eleven Fitbit variables for that day or that time step. In the temporal GCN + GRU model, daily graph embeddings are fed into a GRU; the final hidden state of the GRU is a compact representation of the seven-day window. In the LSTM model, the last hidden state of the LSTM plays a similar role, encoding the seven-day sequence of raw features. These deep representations are then fed into a final linear layer that outputs the stress classification. In future work, these representations could be extracted and used as input to other downstream models, but that is not part of the present study.

3.7 ECGN-Based Graph and Temporal Modeling for Stress Prediction

The main methodological contribution of this project is the design and use of an Electro-Cardio Graph Network (ECGN) over Fitbit features, combined with temporal modeling. The ECGN is a feature-level graph that treats each of the eleven selected Fitbit variables as a node and uses correlations between variables to define edges.

To construct the ECGN, I first computed the Pearson correlation matrix between the eleven features using only the training data. I then take the absolute value of the correlations and set an edge between two features if the absolute correlation is at least 0.3. The diagonal of the adjacency matrix is set to zero so that nodes are not connected to themselves, and the matrix is normalized to be suitable for graph convolution operations. The resulting 11×11 adjacent matrix captures stable relationships between features such as the strong association between steps, distance, calories, and very active minutes, or between sleep duration and sedentary

time.

This ECGN is used in three of the five models. In the static GCN model, the ECGN adjacency is fixed, and two graph convolution layers with ReLU activation and global mean pooling are applied to the daily feature graph, followed by a linear classifier. In the temporal GCN + GRU model, a similar GCN is applied to each day in a seven-day window, producing a sequence of graph embeddings that are then processed by a GRU. In the attention-based dynamic GNN, the ECGN provides a mask: the model learns a dense attention matrix over all node pairs but only allows edges that exist in the ECGN. A row-wise softmax then yields a learned adjacency matrix, which is used in two graph convolution layers before pooling and classification.

The fifth model, an LSTM baseline on seven-day sequences, intentionally does not use the ECGN at all. It serves as a strong non- graph sequence model against which the ECGN-based methods can be compared. By constructing the ECGN and embedding it into different architectures, and then comparing these architectures to a pure LSTM, the study directly examines whether graph structure over Fitbit features offers a real advantage beyond temporal modeling alone.

3.8 Methods

3.8.1 Logistic Regression Baseline

Logistic Regression Baseline uses a simple logistic regression model to classify high vs low stress from one day of Fitbit data. It takes 11 daily features such as sleep duration/quality, step count, activity minutes, calories, and resting heart rate. Each day is treated as an

independent row in a table, so the model does not learn trends across days. This makes it a clean tabular ML baseline that is easy to train and interpret. It also helps you measure how much improvement comes from adding time-based models later. Overall, it sets a minimum performance benchmark without using temporal, sequence, or graph structure.

3.8.2 GCN on ECGN

GCN on ECGN uses a two-layer Graph Convolutional Network to predict high vs low stress from a graph version of the same Fitbit features. First, an Electro-Cardio Graph Network (ECGN) is built where nodes are features and edges represent feature–feature correlation strength. For each day, you get an 11-node graph, with one Fitbit feature value as the node attribute. GCN learns by passing information across connected features, so it can capture relationships between features, not just their individual values. The final graph embedding is used to output the stress class for that day.

3.8.3 Temporal GCN + GRU on ECGN Sequences

Temporal GCN + GRU on ECGN Sequences extends the single-day graph model by adding time awareness. For each day in a 7-day window, a GCN is applied to the ECGN to extract a daily graph embedding from the 11 feature nodes. These daily embeddings are then ordered as a sequence and passed into a GRU, which learns short-term temporal patterns across days. This allows the model to capture both feature relationships and day-to-day stress dynamics. The GRU's final hidden state is used to predict stress. This model directly tests whether combining graph structure and temporal context improves performance over single-day methods.

3.8.4 Attention-Based Dynamic GNN on ECGN

Attention-Based Dynamic GNN on ECGN uses a learnable attention mechanism to adapt the ECGN graph for stress prediction. Instead of relying only on fixed correlations, the model

learns a dense attention-based adjacency matrix from the data and then masks it using the ECGN structure to preserve domain constraints. Node features are updated through two graph convolution layers, allowing the model to emphasize more informative feature interactions. A global pooling layer aggregates node representations into a single graph. The final classifier predicts high vs low stress from this embedding. This model tests whether learning edge weights on top of the ECGN improves performance over static graph baselines.

3.8.5 LSTM Baseline on 7-Day Sequences

LSTM Baseline on 7-Day Sequences uses standard LSTM to model short-term temporal patterns in Fitbit data. The input is a 7-day sequence of daily vectors containing the 11 Fitbit features. Each sequence is treated independently, without any feature–feature graph or relational structure. The LSTM learns trends and changes in sleep, activity, and heart rate over time. The final hidden state is used to predict high vs low stress for the target day. This model serves as a strong temporal-only baseline to isolate the value of sequence modeling without graphs.

4. Results and Discussions

4.1 Performance Evaluation Measures

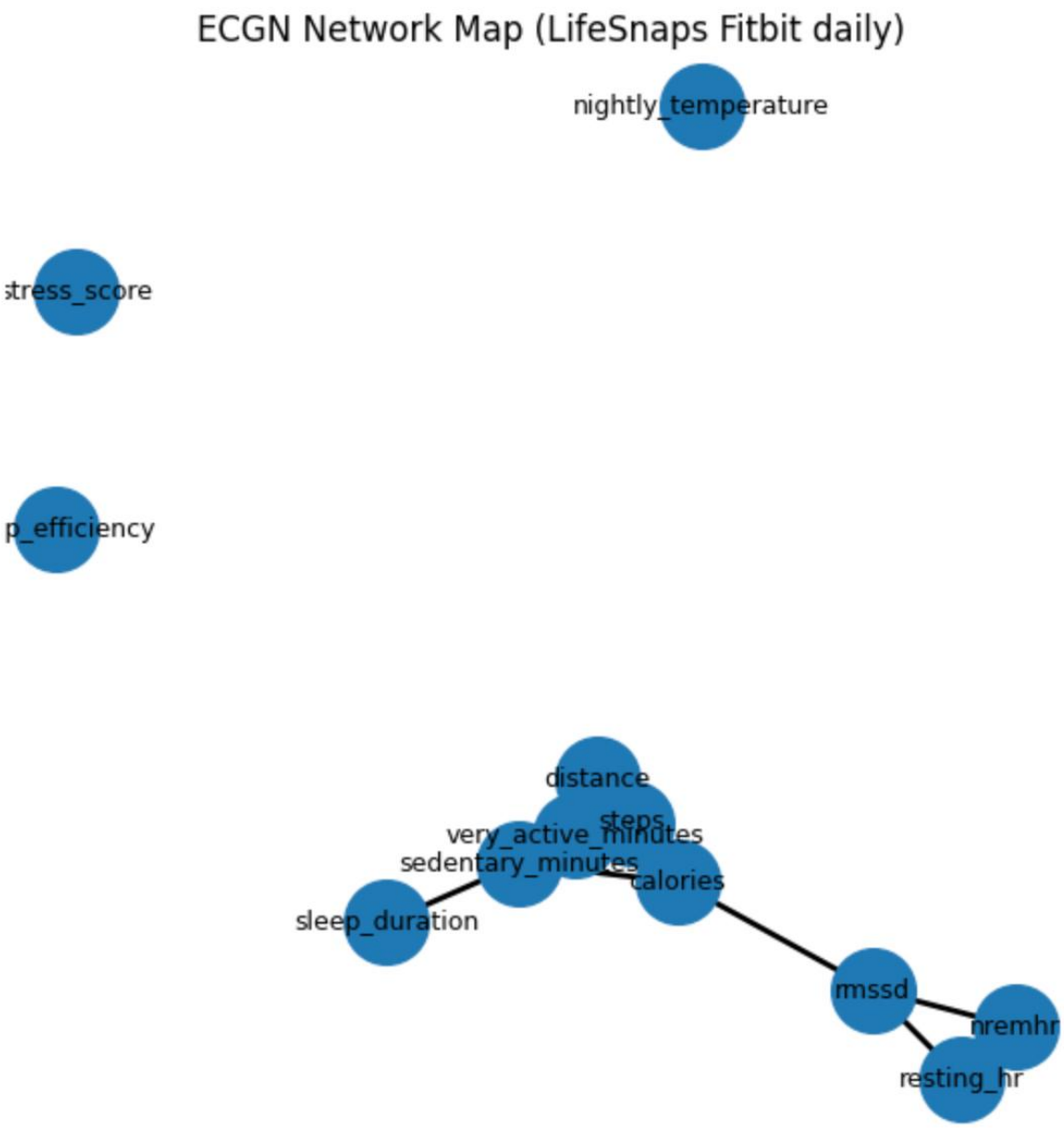
This project treats daily stress prediction as a binary classification problem. To evaluate the models, I use two standard measures:

- Accuracy is the proportion of correctly classified days out of all days in the test set. It shows how often the model gives the right label overall.
- F1-score is the harmonic mean of precision and recall. It summarizes how well the model balances false positives and false negatives in a single number. A higher F1-score means the model is doing well on both detecting high-stress days and avoiding too many wrong alerts.

4.2 Experimental Results

All five models are trained and evaluated on the same LifeSnaps Fitbit daily stress dataset and the same train/validation/test splits. Table 1 shows the **test set** accuracy and F1-score for each model.

4.2.1 ECGN Graph



4.2.2 Logistic Regression Baseline

```
=== Logistic Regression (Method 1 baseline) ===
Train accuracy: 0.632, F1: 0.652
Val accuracy: 0.652, F1: 0.680
Test accuracy: 0.630, F1: 0.663

Test classification report:
              precision    recall  f1-score   support

           0      0.614     0.570     0.591       151
           1      0.643     0.684     0.663       171

    accuracy                          0.630       322
   macro avg      0.629     0.627     0.627       322
weighted avg      0.629     0.630     0.629       322

Test confusion matrix:
[[ 86  65]
 [ 54 117]]
```

4.2.3 GCN on ECGN

```
Using device: cpu
Epoch 01 | Train loss 0.685, acc 0.571, f1 0.696 | Val loss 0.683, acc 0.578, f1 0.676
Epoch 10 | Train loss 0.671, acc 0.598, f1 0.673 | Val loss 0.678, acc 0.581, f1 0.672
Epoch 20 | Train loss 0.669, acc 0.591, f1 0.659 | Val loss 0.680, acc 0.565, f1 0.641
Epoch 30 | Train loss 0.670, acc 0.578, f1 0.612 | Val loss 0.681, acc 0.565, f1 0.626
Epoch 40 | Train loss 0.667, acc 0.583, f1 0.649 | Val loss 0.681, acc 0.550, f1 0.637
Epoch 50 | Train loss 0.666, acc 0.587, f1 0.660 | Val loss 0.681, acc 0.547, f1 0.635

=== Method 2: GCN on ECGN graph (final performance) ===
Train loss 0.666, acc 0.588, f1 0.659
Val loss 0.681, acc 0.547, f1 0.635
Test loss 0.659, acc 0.596, f1 0.680
```

4.2.4 Temporal GCN + GRU on ECGN Sequences

```
Using device: cpu
Epoch 01 | Train loss 0.692, acc 0.542, f1 0.703 | Val loss 0.687, acc 0.542, f1 0.703
Epoch 10 | Train loss 0.658, acc 0.622, f1 0.670 | Val loss 0.660, acc 0.595, f1 0.681
Epoch 20 | Train loss 0.642, acc 0.653, f1 0.708 | Val loss 0.654, acc 0.599, f1 0.663
Epoch 30 | Train loss 0.634, acc 0.649, f1 0.707 | Val loss 0.648, acc 0.613, f1 0.667
Epoch 40 | Train loss 0.626, acc 0.653, f1 0.704 | Val loss 0.638, acc 0.623, f1 0.686
Epoch 50 | Train loss 0.612, acc 0.660, f1 0.702 | Val loss 0.632, acc 0.648, f1 0.660

=== Method 3: Temporal GCN+GRU on ECGN (final performance) ===
Train loss 0.614, acc 0.657, f1 0.663
Val loss 0.632, acc 0.648, f1 0.660
Test loss 0.632, acc 0.641, f1 0.646
```

4.2.5 Attention-Based Dynamic GNN on ECGN

```
Using device: cpu
num_nodes: 11 in_dim: 1
Epoch 01 | Train loss 0.691, acc 0.536, f1 0.487 | Val loss 0.687, acc 0.581, f1 0.694
Epoch 10 | Train loss 0.672, acc 0.595, f1 0.665 | Val loss 0.673, acc 0.587, f1 0.662
Epoch 20 | Train loss 0.666, acc 0.589, f1 0.659 | Val loss 0.676, acc 0.581, f1 0.662
Epoch 30 | Train loss 0.663, acc 0.594, f1 0.663 | Val loss 0.678, acc 0.571, f1 0.668
Epoch 40 | Train loss 0.663, acc 0.585, f1 0.645 | Val loss 0.678, acc 0.578, f1 0.638
Epoch 50 | Train loss 0.660, acc 0.591, f1 0.659 | Val loss 0.676, acc 0.578, f1 0.658

=== Method 4: Attention / Dynamic GNN on ECGN (final performance) ===
Train loss 0.659, acc 0.590, f1 0.654
Val loss 0.676, acc 0.578, f1 0.658
Test loss 0.652, acc 0.602, f1 0.678
```

4.2.6 LSTM Baseline on 7-Day Sequences

```
Using device: cpu
X_train_lstm shape: torch.Size([852, 7, 11])
X_val_lstm shape: torch.Size([284, 7, 11])
X_test_lstm shape: torch.Size([284, 7, 11])
Epoch 01 | Train loss 0.698, acc 0.496, f1 0.164 | Val loss 0.683, acc 0.592, f1 0.477
Epoch 10 | Train loss 0.545, acc 0.716, f1 0.754 | Val loss 0.578, acc 0.694, f1 0.729
Epoch 20 | Train loss 0.448, acc 0.790, f1 0.818 | Val loss 0.541, acc 0.722, f1 0.749
Epoch 30 | Train loss 0.363, acc 0.837, f1 0.856 | Val loss 0.555, acc 0.711, f1 0.727
Epoch 40 | Train loss 0.308, acc 0.867, f1 0.880 | Val loss 0.581, acc 0.736, f1 0.757
Epoch 50 | Train loss 0.265, acc 0.887, f1 0.898 | Val loss 0.619, acc 0.750, f1 0.780

=== Method 5: LSTM baseline on 7-day sequences (final performance) ===
Train loss 0.259, acc 0.894, f1 0.907
Val loss 0.619, acc 0.750, f1 0.780
Test loss 0.617, acc 0.729, f1 0.769
```

In summary, the experiments show that:

- All models can predict stress better than random guessing.
- Graph-based models on the ECGN give only small gains over logistic regression.
- The LSTM on 7-day sequences is the strongest model by a clear margin.

4.3 Comparison Tables

1. Comparing The Methods Implemented

Method	**Description**	**Accuracy (Test)**	**F1 (Test)**
M1 – Logistic Regression	Tabular baseline on 11 ECGN nodes	0.63	0.66
M2 – Static GCN	GCN on ECGN graph	0.6	0.68
M3 – Temporal GCN+GRU	7-day sequences with GCN+GRU	0.64	0.65
M4 – Attention / Dynamic GNN	Learned attention adjacency over ECGN	0.6	0.68
M5 – LSTM	7-day sequences, no graph	**0.73**	**0.77**

2. Comparison Table with Previous Studies

Method in this study	Original source & dataset	Original task & reported performance	Adaptation in this project (LifeSnaps)	Performance on LifeSnaps
Method 1 – Logistic Regression Baseline	General ML baseline (no single paper)	Logistic regression is often used as a simple reference model in many EEG and physiological signal studies.	Single-day logistic regression on 11 Fitbit features for binary high vs low self-reported stress.	**Accuracy 0.63, F1 0.66**
Method 2 – GCN on ECGN	**EEG-GCNN** (Wagh & Varatharajah, 2020) using large EEG datasets for neurological disease vs healthy classification	Domain-guided GCN on electrode graphs; reports about **AUC ≈ 0.90**and clear gains over classical ML baselines for EEG disease diagnosis.	Two-layer GCN on an ECGN feature graph built from correlations between 11 Fitbit features; single-day prediction of high vs low stress.	**Accuracy 0.60, F1 0.68**
Method 3 – Temporal GCN + GRU on ECGN	**EEG-GNN-SSL** (Tang et al., 2022) on the Temple University Seizure Corpus (TUSZ)	Diffusion Conv. RNN (DCRNN-style GNN) with self-supervised pre-training; reports **AUROC ≈ 0.875** for seizure detection and **weighted F1 ≈ 0.75** for seizure type classification.	Temporal ECGN model: GCN on each day in a 7-day window + GRU over the 7 graph embeddings; binary stress prediction.	**Accuracy 0.64, F1 0.65**
Method 4 – Attention / Dynamic GNN on ECGN	Dynamic GNN / DGCN and contrastive graph works, such as **dynamic EEG GNNs** and **A-GCL** for fMRI	Dynamic graph models on EEG or fMRI often report strong gains over static GCNs. One dynamic connectivity GNN for seizure detection reports around **91% accuracy**, beating CNN, standard GNN, and Transformer baselines.	Attention-based GNN that learns a dense adjacency matrix, masked by the ECGN edges, with two graph conv layers and global pooling for daily stress prediction.	**Accuracy 0.60, F1 0.68**
Method 5 – LSTM on 7-Day Sequences	RNN/LSTM EEG baselines from general deep-learning EEG toolkits that often act as strong temporal baselines in seizure and emotion tasks.	LSTM or CNN-LSTM models on multi-channel EEG time series commonly reach high performance on lab-based tasks such as seizure detection or emotion recognition.	Plain LSTM on 7-day sequences of 11 Fitbit features (no graph), to test how far a simple temporal model can go for daily stress prediction.	**Accuracy 0.73, F1 0.77**

In Summary, across five methods, the simple single-day baselines (logistic regression and

static GCN) give moderate results around 0.60–0.63 accuracy with F1 up to 0.68. Adding graph + short-term sequence modelling (Temporal GCN + GRU) improves accuracy slightly to 0.64, but F1 stays similar (0.65). The attention/dynamic GNN does not improve over the static GCN on LifeSnaps (0.60 accuracy, 0.68 F1), suggesting learned edge weights did not help here. The best performer is the no-graph LSTM on 7-day sequences, reaching 0.73 accuracy and 0.77 F1, meaning temporal trends matter more than the ECGN graph in this setup.

4.4 Discussion

The main result of this project is that temporal patterns over a week are more informative than the ECGN graph structure for predicting daily stress from Fitbit data. Among the five models, the LSTM baseline on 7-day sequences (Method 5) achieved the best performance with 0.73 accuracy and 0.77 F1 on the test set. All ECGN-based models stayed around 0.60–0.64

accuracy and 0.65–0.68 F1, which is only slightly better than the logistic regression baseline, and clearly below the LSTM.

This tells us that, for the LifeSnaps Fitbit daily summaries, the change of features across days (steps, sleep, resting heart rate, etc.) carries more predictive signal than a static graph built from feature correlations on a single day. The LSTM can directly see how these features move together over seven days and learn patterns.

The results also show that graph modeling is not a free performance boost. Static GCN and the attention/dynamic GNN do capture some structure (their F1 is slightly higher than logistic regression), but they pay a cost in accuracy and complexity. The temporal GCN + GRU sits in between: it combines graph and time, but still does not outperform the pure LSTM. Possible reasons include the limited dataset size, the simplicity of the ECGN

construction, and the fact that the adjacency is mostly static, while real-world behavior and relationships between variables may change over time.

From a practical angle, the message is straightforward:

- If the goal is to build a working stress prediction model **from** daily Fitbit summaries**, a** sequence model like an LSTMis currently more effective and easier to justify than a graph model on a simple correlation-based ECGN. The LSTM uses the same 11 features, but its ability to model temporal dynamics gives it a clear advantage.
- At the same time, this project is still useful for research. It tests whether EEG-inspired GNN ideas (GCN, attention/dynamic GNN, temporal GCN + GRU) transfer to low-dimensional wearable data. The answer here is "not yet, at least not with a basic ECGN."

5. Conclusion

This project explored daily stress prediction using wearable data from the LifeSnaps "Rais anonymized" dataset. The main idea was to build an Electro-Cardio Graph Network (ECGN) over 11 Fitbit features and to test whether graph neural networks, inspired by EEG and brain connectivity work, can improve performance compared to standard machine learning and sequence models. I implemented five methods: logistic regression, a static GCN on the ECGN, a temporal GCN + GRU, an attention-based dynamic GNN, and an LSTM baseline on 7-day sequences.

The results show that temporal information is more important than the simple ECGN graph for this dataset. All graph-based models gave only small gains over logistic regression and stayed in the range of 0.60–0.64 accuracy and 0.65–0.68 F1 on the test set. In contrast, the LSTM on 7-day sequences reached 0.73 accuracy and 0.77 F1, clearly outperforming all other methods. This suggests that short-term changes in sleep, activity, and heart rate across

several days are more useful for prediction than the static feature correlation structure captured by the ECGN.

At the same time, the study provides a first step toward graph-based modeling of wearable summary data. It shows how to construct an ECGN from Fitbit features, how to adapt EEG-style GNN architectures (GCN, temporal GCN + GRU, attention/dynamic GNN) to this setting, and how to compare them fairly to a strong temporal baseline on the same dataset and task. Even though the ECGN-based models did not win in terms of accuracy and F1, they help clarify which modeling choices are most valuable for this kind of data. There are several limitations. The ECGN uses a simple global correlation threshold and has only 11 nodes. The dataset, while rich in time, is not huge in terms of the number of users and windows, which may limit the capacity of deeper graph models. The daily self-reported stress scores can be noisy and subjective. These factors all make it harder for more complex models to show a clear advantage.

Future work could address these limitations by designing richer and more adaptive graphs (for example, per-user or time-varying ECGNs, or graphs that include EMA and survey features), using larger or multi-dataset training, and exploring more advanced sequence models such as transformers. A natural next step would be to combine the strengths of both worlds: use an improved ECGN to produce daily graph embeddings, and then apply a powerful temporal model on top of those embeddings. For now, this project's main conclusion is clear: on LifeSnaps Fitbit daily summaries, a well-designed temporal model like an LSTM is a strong and practical baseline for stress prediction, and any graph-based method must at least match this level to be considered competitive.

This negative result is important: it suggests that future graph-based work on LifeSnaps-type data should probably focus on richer graphs (for example, user-specific graphs, time-varying

graphs, or multimodal graphs that include EMA and survey data) and then compare them again to strong temporal baselines like the LSTM used here.

6. References

[1] A. Pinge, V. Gad, D. Jaisighani, S. Ghosh, and S. Sen, "Detection and monitoring of stress using wearables: a systematic review," *Front. Comput. Sci.*, vol. 6, p. 1478851, Dec. 2024, doi: 10.3389/fcomp.2024.1478851.

[2] M. Mouadili, E. M. En-Naimi, and M. Kouissi, "Advancing Stress Detection and Health Monitoring with Deep Learning Approaches," in *International Conference on Sustainable Computing and Green Technologies (SCGT’2025)*, MDPI, July 2025, p. 10. doi: 10.3390/cmsf2025010010.

[3] İ. Kayadibi and O. Uslu, "A lightweight St-CNN architecture based on deep learning for stress level detection from human physical activities," *Sci. Rep.*, vol. 15, no. 1, p. 33570, Sept. 2025, doi: 10.1038/s41598-025-18647-x.

[4] M. K. Moser, M. Ehrhart, and B. Resch, "An Explainable Deep Learning Approach for Stress Detection in Wearable Sensor Measurements," *Sensors*, vol. 24, no. 16, p. 5085, Aug. 2024, doi: 10.3390/s24165085.

[5] A. Calvo, J. Martin, and C. Martin, "Early Detection of Chronic Stress Using Wearable Devices: A Machine Learning Approach with the WESAD Database:," in *Proceedings of the 11th International Conference on Information and Communication Technologies for Ageing Well and e-Health*, Porto, Portugal: SCITEPRESS - Science and Technology Publications, 2025, pp. 189–196. doi: 10.5220/0013209700003938.

[6] E. Smets *et al.*, "Large-scale wearable data reveal digital phenotypes for daily-life stress detection," *Npj Digit. Med.*, vol. 1, no. 1, p. 67, Dec. 2018, doi: 10.1038/s41746-018-0074-9.

[7] A. Abd-alrazaq *et al.*, "The Performance of Wearable AI in Detecting Stress Among Students: Systematic Review and Meta-Analysis," *J. Med. Internet Res.*, vol. 26, p. e52622, Jan. 2024, doi: 10.2196/52622.

[8] D. Klepl, M. Wu, and F. He, "Graph Neural Network-Based EEG Classification: A Survey," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 32, pp. 493–503, 2024, doi: 10.1109/TNSRE.2024.3355750.

[9] X. Zhu, G. Liu, L. Zhao, W. Rong, J. Sun, and R. Liu, "Emotion Classification from Multi-Band Electroencephalogram Data Using Dynamic Simplifying Graph Convolutional Network and Channel Style Recalibration Module," *Sensors*, vol. 23, no. 4, p. 1917, Feb. 2023, doi: 10.3390/s23041917.

[10] A. Joshi, P. S. Matharu, L. Malviya, M. Kumar, and A. Jadhav, "Advancing EEG based stress detection using spiking neural networks and convolutional spiking neural networks," *Sci. Rep.*, vol. 15, no. 1, p. 26267, July 2025, doi: 10.1038/s41598-025-10270-0.

[11] S. Tang *et al.*, "Self-Supervised Graph Neural Networks for Improved Electroencephalographic Seizure Analysis," 2021, doi: 10.48550/ARXIV.2104.08336.

[12] J.-Z. Xiang, Q.-Y. Wang, Z.-B. Fang, J. A. Esquivel, and Z.-X. Su, "A multi-modal deep

learning approach for stress detection using physiological signals: integrating time and frequency domain features," *Front. Physiol.*, vol. 16, p. 1584299, Apr. 2025, doi: 10.3389/fphys.2025.1584299.

[13] A. Pinge, V. Gad, D. Jaisighani, S. Ghosh, and S. Sen, "Detection and monitoring of stress using wearables: a systematic review," *Front. Comput. Sci.*, vol. 6, p. 1478851, Dec. 2024, doi: 10.3389/fcomp.2024.1478851.

[14] E. Lazarou and T. P. Exarchos, "Predicting stress levels using physiological data: Real-time stress prediction models utilizing wearable devices," *AIMS Neurosci.*, vol. 11, no. 2, pp. 76–102, 2024, doi: 10.3934/Neuroscience.2024006.

[15] S. Yfantidou *et al.*, "LifeSnaps, a 4-month multi-modal dataset capturing unobtrusive snapshots of our lives in the wild," *Sci. Data*, vol. 9, no. 1, p. 663, Oct. 2022, doi: 10.1038/s41597-022-01764-x.

[16] A. Choi, A. Ooi, and D. Lottridge, "Digital Phenotyping for Stress, Anxiety, and Mild Depression: Systematic Literature Review," *JMIR MHealth UHealth*, vol. 12, p. e40689, May 2024, doi: 10.2196/40689.

[17] B. A. Darwish, S. U. Rehman, I. Sadek, N. M. Salem, G. Kareem, and L. N. Mahmoud, "From lab to real-life: A three-stage validation of wearable technology for stress monitoring," *MethodsX*, vol. 14, p. 103205, June 2025, doi: 10.1016/j.mex.2025.103205.

[18] P. Schmidt, A. Reiss, R. Duerichen, C. Marberger, and K. Van Laerhoven, "Introducing WESAD, a Multimodal Dataset for Wearable Stress and Affect Detection," in *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, Boulder CO USA: ACM, Oct. 2018, pp. 400–408. doi: 10.1145/3242969.3242985.

[19] A. O. Pataca *et al.*, "Use of machine learning for predicting stress episodes based on wearable sensor data: A systematic review," *Comput. Biol. Med.*, vol. 198, p. 111166, Nov. 2025, doi: 10.1016/j.compbiomed.2025.111166.

[20] J. Brunner *et al.*, "VA 's EHR transition and health professions trainee programs: Findings and impacts of a multistakeholder learning community," *Learn. Health Syst.*, vol. 9, no. 2, p. e10460, Apr. 2025, doi: 10.1002/lrh2.10460.