A Reproducible Stress Prediction Pipeline with Mobile Sensor Data

PANYU ZHANG, KAIST, Republic of Korea
GYUWON JUNG, KAIST, Republic of Korea
JUMABEK ALIKHANOV, Inha University, Republic of Korea
UZAIR AHMED, KAIST, Republic of Korea
UICHIN LEE*, KAIST, Republic of Korea

Recent efforts to predict stress in the wild using mobile technology have increased; however, the field lacks a common pipeline for assessing the impact of factors such as label encoding and feature selection on prediction performance. This gap hinders replication, especially because of a lack of common guidelines for reporting results or privacy concerns that limit access to open codes and datasets. Our study introduces a common pipeline based on a comprehensive literature review and offers comprehensive evaluations of key pipeline factors, promoting independent reproducibility. Our systematic evaluation aimed to validate the findings of previous studies. We identified overfitting and distribution shifts across users as the major reasons for performance limitations. We used K-EmoPhone, a public dataset, for experimentation and a new public dataset—DeepStress—to validate the findings. Furthermore, our results suggest that researchers should carefully consider temporal order in cross-validation settings. Additionally, self-report labels for target users are key to enhancing performance in user-independent scenarios.

CCS Concepts: • **Human-centered computing** → **Ubiquitous and mobile computing**; • **Applied computing** → *Health informatics*.

Additional Key Words and Phrases: Stress Prediction, Mobile Health, Reproducibility

ACM Reference Format:
Panyu Zhang, Gyuwon Jung, Jumabek Alikhanov, Uzair Ahmed, and Uichin Lee. 2024. A Reproducible Stress Prediction Pipeline with Mobile Sensor Data. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8, 3, Article 143 (September 2024), 35 pages. https://doi.org/10.1145/3678578

1 Introduction

Chronic stress is a growing concern, particularly among college students [71]. The prevalence and impact of stress have prompted researchers to explore various strategies for alleviating it. One such approach is the development of mobile applications specifically designed to cope with everyday stressors [28]. Furthermore, the mobile sensor data collected from wearable sensors and mobile phones offer novel opportunities for extracting invaluable insights into everyday stressors and enabling mobile context-aware interventions. The efficacy of such context-aware interventions depends on accurately detecting when an individual is stressed.

Thus, prior studies have extensively explored how mobile sensors and interaction data can be used to predict self-reported stress levels in real-world settings [35, 63]. In our work, we mainly focused on the in-the-wild,

*The corresponding author.

Authors' Contact Information: Panyu Zhang, panyu@kaist.ac.kr, KAIST, Daejeon, Republic of Korea; Gyuwon Jung, gwjung@kaist.ac.kr, KAIST, Daejeon, Republic of Korea; Jumabek Alikhanov, juma@inha.edu, Inha University, Incheon, Republic of Korea; Uzair Ahmed, uziahmd@kaist.ac.kr, KAIST, Daejeon, Republic of Korea; Uichin Lee, uclee@kaist.edu, KAIST, Daejeon, Republic of Korea.

ACM 2474-9567/2024/9-ART143
https://doi.org/10.1145/3678578

self-report stress prediction using mobile sensor data. In addition, we aim to use machine learning without considering end-to-end deep learning because end-to-end deep learning models may be unsuitable for handling event-based data (details regarding this are included in Section 6).

After a decade of development, very few literature reviews have summarized the common stress prediction pipeline in this field [18, 64, 75]. For example, Fukazawa et al. [18] described the pipeline as comprising data collection, feature design, machine learning and statistical analysis, and an evaluation setting. Certain "factors" played a role in each step. One "factor" in the evaluation setting step is cross-validation, which can be conducted through the k-fold or leave-one-subject-out (LOSO) approaches. Fukazawa et al. observed potential data leakage using the k-fold cross-validation. However, insights from previous studies regarding stress prediction pipelines have not been validated with implementations.

Recently, researchers have increasingly focused on reproducibility and generalizability issues in the broader scope of mobile sensing [3, 7, 45, 47, 53, 79]. Most recent papers aim to achieve generalization across datasets collected by different users and devices or in various countries. This is established with cross-dataset validation by training on dataset *A* and testing on dataset *B*. Mishra et al. [53], referred to this type of cross-dataset validation as "cross-dataset reproducibility". Cross-dataset generalizability and reproducibility have advanced; however, research on within-dataset reproducibility using slightly different codes or analyses remains limited-Albertoni et al. [5] defined this as "independent reproducibility." In practice, each factor in the aforementioned pipeline may impact the performance on the final performance. Reproducing results using the same dataset without comprehensively understanding the impact of each factor can be challenging.

Despite a decade of efforts in this field, the performance of in-the-wild, self-reported stress prediction in user-independent settings remains limited. Yu et al. [84] achieved a 63% macro F1 score in a group 5-fold cross-validation, whereas Toshnazarov et al. [70] achieved a 65.8% F1 score using only physiological data in a LOSO cross-validation setting with a pre-trained, in-the-lab best model. Further research is needed to shed light on this low performance and potentially push the current performance limits.

This study is the first attempt at deriving a common pipeline in stress prediction through reproducibility experiments. We aim to understand the impact of each pipeline 'factor' on the final performance, which will help establish a benchmark and facilitate reproducibility in this field with open code and datasets. In addition, we endeavor to enhance the performance limits by tuning the pipeline factors and debugging the reasons for the currently low performance.

Therefore, we set the following research questions:

- **RQ1** What is the common pipeline for stress prediction using mobile sensor data?
- **RQ2** What is the impact of each factor in a stress prediction pipeline on the final performance using a public dataset?
- **RQ3** How can the model performance in user-independent settings be improved?
- **RQ4** What is the primary reason for the low performance of real-world mobile stress prediction in user-independent settings?

In addition to presenting a common pipeline, this study also meticulously examined factors within the data analysis pipeline, adhering to the principles of independent reproducibility, using a public mobile sensor dataset [35] with open code. We observed that the temporal order is essential when considering cross-validation settings. In addition, removing samples of the neutral state may lead to overly optimistic results.

Furthermore, the sensor data require access to the target user's data for personalization; however, the last experience sampling method (ESM) stress label is valid for model improvement even in user-independent settings. This indicates that human behavior derived from sensor data may vary across users; however, the temporal dependency of stress states remains consistent. When sufficient labeled data from target users are available for partial personalization (i.e., using part of the labeled data of the target user for training), the sensor data become

more efficient for stress prediction. The Area Under the Receiver Operating Characteristic Curve (AUC-ROC) and user-independent cross-validation are also recommended for model evaluation.

We utilized the LOSO settings across two datasets, and integrating sensor data and previous self-reported stress labels was most effective for enhancing stress prediction. This approach is consistent with prior research [48, 52, 70] and highlights the importance of recent stress labels in improving the model performance. Additionally, this finding is consistent with the benefits of partial personalization, underscoring the importance of incorporating labeled data from target users to improve model accuracy. Our analysis indicated that overfitting and distribution shifts across users are major contributors to poor performance in user-independent settings, necessitating an enhanced focus on domain generalization and adaptation.

Moreover, the literature supports improving prediction accuracy through partial personalization and the including the previous stress label as a feature [45, 48, 52, 70], further validating our results. Based on these insights, we advocate a "user-in-the-loop" strategy, initially using the sensor data and the last stress label when data from the target user is limited and transitioning to partial personalization with solely sensor data as more user-specific labels are acquired. Thus, this work facilitates the development of more robust and accurate stress prediction systems by identifying these challenges and opportunities.

2 Related Work

2.1 Real-world Mobile Stress Detection

Stress detection has been extensively explored in laboratories with previous studies achieving success using an array of sensor data, including physiological metrics such as Electrocardiogram (ECG) readings [39, 57, 64]. Despite these advancements, transitioning from controlled environments to unpredictable real-world scenarios presents a critical challenge for stress detection technologies.

The quest for accurate ground-truth labeling in emotion research has long relied on self-reported data. Historically, these data strongly correlated with facial expressions [62]. This correlation lays the foundation for using

Table 1. Open DataSets for Real-world Mobile Stress Detection

Dataset	**Duration**	**#Users**	**Feature Types**	**Freq. of Labels**	**Year**
StudentLife [77]	10 weeks	48	Sensor data, ESM, pre- and post- survey	3-13 ESMs administered per day	2014
CrossCheck [76]	2-12 months	62	Sensor data	Administered every 2-3 days	2016
Tesserae [42]	56 days	649	Sensor data, ESM, pre- and post- survey	Administered daily	2019
TILES-2018 [55]	10 weeks	212	Sensor data, ESM, pre- and post- survey	Administered daily	2020
TILES-2019 [82]	3 weeks	57	Sensor data, ESM, pre- and post- survey	Twice a day	2022
GLOBEM [80]	10 weeks per year (2018-21)	497	Sensor data, Weekly ESM, pre- and post-survey	Weekly	2022
K-EmoPhone [35]	7 days	77	Sensor data, pre- and post- survey	10 ESMs administered per day	2023
DeepStress [33]	6 weeks	24	Sensor data, pre-survey	Avg. 4.9 ESMs sent per day	2024

self-reported labels as ground-truth benchmarks for emotion prediction. A pioneering study, AffectAura, allowed users to record their emotional states, engage in reflective practices, and predict these states [44].

The Experience Sampling Method (ESM) [40] is widely recognized in collecting self-report surveys. Its application spans various disciplines, offering a window into the participants' emotions, thoughts, and daily activities as they naturally occur. Upon receiving ESM notifications on their phones, individuals report these details daily. A surge in technological innovation has occurred in the last decade, yielding advanced tools that enhance the efficacy and reach of ESM, thereby enriching the data collected using this method [26].

In real-world mobile stress detection, combining mobile and wearable sensor data is a growing trend. As highlighted in previous studies [10, 12], this approach leverages the capabilities of wearable technologies for continuous, real-time stress monitoring. Moreover, some studies include past self-reported ESM data to enrich the model [29]. Integrating diverse sensor types such as image and speech data [27, 32, 36, 73, 74] has gained interest; however, practical challenges and privacy concerns in collecting such real-life data limit their feasibility. Owing to these constraints, our research focused on conventional sensor data types. Table 1 details the existing open datasets in this domain and provides a basis for future research to test our reproducible pipeline. In our study, the K-EmoPhone dataset [15] was selected for the reproducibility experiments because of its high labeling rate and raw sensor and phone interaction data availability.

2.2 Pipelines for Emotion Prediction Using Mobile Sensor Data

Prior studies have proposed common pipelines in the broader scope of emotion sensing using mobile or wearable data; however, studies specifically focusing on the pipelines for mobile stress prediction are lacking. Fukazawa et al. [18] proposed a common pipeline for mobile mental state detection, consisting of data collection, feature design, machine learning and statistical analysis, and evaluation settings. Vos et al. [75] reviewed the pipeline for wearable stress monitoring, comprising three main steps; preprocessing, feature engineering, and algorithm selection. Regarding mobile emotion sensing, Yang et al. [81] designed a pipeline that included signal perception, feature engineering covering handcrafted and deep feature extractions, and classification involving both traditional machine learning and deep models. These prior works conducted extensive literature reviews in related fields to develop the pipeline; however, a research gap exists because these studies have no real implementations. As illustrated in Section 1, none of the insights derived from the literature reviews for pipeline design has been validated using publicly available datasets.

2.3 Reproducibility, Generalizability, and Replicability

In machine learning, reproducibility, generalizability, and replicability have varied definitions. Albertoni et al. [5] summarized the existing terminologies based on whether the experiment was conducted by the same team, on the same data, using the same code and analysis, and so on. They defined *reproducibility* as using the same data and the same code and analysis by different teams. In contrast, *replicability* involves using different codes and analyses andor different data by different teams. *Repeatability* refers to replication by the same team, whereas *corroboration* aims to validate findings from previous studies. According to ACM definitions [58], reproducibility is ***the ability of a different team to arrive at the same scientific results using the same experimental setup***. According to Google Developers [25], generalization refers to a model's ability to adapt properly to new, previously unseen data. Raff et al. [61] defined ***independent reproducibility*** as using the same data and different code and analysis.

We outlined our terminologies based on Albertoni. et al.'s terminology figure and other related works (Figure 1). The key terminologies used in this paper are highlighted in bold. We specifically focused on "independent reproducibility," which involves employing different codes and analyses on the same dataset.

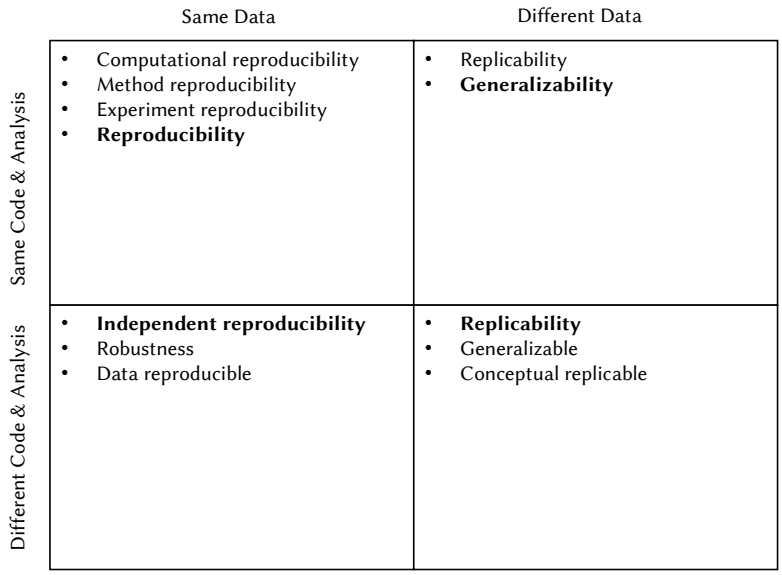


Fig. 1. Summarized Terminologies (developed based on the work of Albertoni et al. [5])

In the broad fields of emotion, mental health, and daily activity prediction using mobile sensors and/or wearable data, efforts have been made to explore generalizability and reproducibility. Xu et al. [79] explored the generalizability of the in-the-wild depression detection models across different datasets. They explored the generalizability across datasets collected from various institutions and years of data collection. Existing studies [7, 37, 45] have focused on the generalizability of mood inference, activity detection, and personality inference models across datasets collected from different countries. A study [3] also explored the cross-dataset generalizability but defined it as merging multiple datasets and training and testing using the merged dataset. Some studies [9, 53] have defined cross-dataset generalizability as transfer learning across datasets. However, using the same code and analysis on different datasets is referred to as generalizability in our context (Figure 1).

In this domain, no previous work has specifically focused on independent reproducibility, which serves as the central theme of this paper. Hereafter, any mention of 'reproducibility' denotes independent reproducibility.

2.4 Limited Performance in In-the-wild Mental State Prediction using Mobile Sensor Data and/or Wearable Sensor Data

In wearable stress detection, Yu et al. [84] achieved a maximum macro F1 score of 63% in subject-independent settings (group 5-fold cross-validation), even using semi-supervised learning. Meegahapola et al. [45] obtained a 53% Area Under the AUC-ROC for mobile mood inference for group 5-fold cross-validation. Recently, Toshnazarov et al. [70] attained an F1 score of 65.8% for in-the-wild stress prediction with the pre-trained, in-the-lab best model as part of a two-stage model with a LOSO cross-validation setting. Despite using pre-trained models, their performance remains limited in in-the-wild datasets.

Given the low field performance in the field, several studies have explored tuning the factors in the pipeline to improve the model performance limit. Sano et al. [63] explored different combinations of feature types and

machine learning models to enhance performance. Hung et al. [30] achieved their best-performing model by combining different time window sizes, feature selection methods, and machine learning models.

In-depth investigations into the causes of low performance have also been conducted. Pratap et al. [60] analyzed the accuracy of personalized mobile mood detection models at the individual-user levels. They uncover that personalization is effective for most users (80%), with an AUC-ROC of >50% and offers strong prediction results for 11.8% users, with an AUC-ROC of >80%. Gao et al. [19, 20] identified the uncertainty in self-reported labels as a factor contributing to poor mental well-being sensing performance in the wild. Xu et al. [79] highlighted overfitting as a major barrier to model generalization across unseen users and suggested early stopping as a mitigation strategy.

Despite previous attempts to enhance the model performance by tuning the pipeline factors, systematic approaches are lacking because the pipeline and its factors are not clearly defined. Furthermore, existing research merely identifies the phenomenon of low performance and lacks an in-depth investigation into the potential causes using publicly available datasets.

3 Common Pipeline

Regarding the common pipeline in the in-the-wild mobile stress prediction field, RQ1, we conducted a literature review and derived the steps and factors of a common pipeline. Based on three review papers published in 2017, 2020, and 2023 [18, 59, 81], we targeted 54 related papers. Given the in-the-wild mobile stress prediction scope of this paper, our selection criteria exclude papers that are unrelated to stress prediction, do not use mobile sensor data, do not target in-the-wild scenarios, and are not about building predictive models. Fifteen relevant papers met the selection criteria.

Figure 2 illustrates the common pipeline derived, which summarizes and combines the pipelines from the prior work. The pipeline comprises eight steps: preprocessing, feature extraction, feature preparation, feature selection, data splitting, oversampling, model training, and model evaluation. Each step comprises different factors with all possible alternatives. For example, in the preprocessing step (step 1), there are two options-**a.1** and **a.2**-for factor **a** of the preprocessing step (i.e., removing the invalid survey samples). We also provide supporting references from relevant literature for each factor for the given pipeline components in Tables 2, 3, 4.

Specifically, several factors were not derived from the 15 related papers but are common in the broader scope of emotion and human activity prediction using mobile and or wearable data. We also included important factors that were not explored in the literature. For instance, b.2 (i.e., removing users with extreme label distributions in Step 1 of preprocessing) has not been tested in previous works. However, we still included it in the common pipeline setting, which aims to remove users who may always be stressed or not stressed as a proxy for the low data quality of labels. Similarly, in the data splitting step (step 5), a.2 group k-fold cross-validation was not used in the 15 related papers on mobile stress prediction but was used in Yu et al.'s work on wearable stress detection [84] and Ferrari et al.'s work on human activity recognition [17]. Following this rationale, c.2, stratified partial personalization and c.3, random partial personalization were used by Meegahapola et al. and Tarzav et al. on mobile mood/food consumption level inference and wearable stress detection [45, 48, 69]. Lastly, b.1 and c.1 were added because of the time-series nature of the sensor data that were not tested in the previous work.

4 Methodology

4.1 Dataset

Given the scope of in-the-wild mobile self-report stress prediction, this study focused on predicting stress in real time using in-the-wild mobile self-report data. Consequently, datasets that meet our criteria should include self-reported stress labels and mobile sensor data, with a preference for those featuring in-situ stress labels. In a common data collection setting, in-situ labels are collected multiple times throughout the day to capture real-time

stress accurately. This contrasts with those gathered only at the day's start or end, or even less frequently. Frequent in-situ self-report collection is crucial because the aim is to predict real-time stress using mobile sensors. The StudentLife, K-EmoPhone, and DeepStress datasets currently contain in-situ labels (Table 1). However, owing to the large intervals between labels in the StudentLife dataset, which resulted in a lower data frequency, we used only the K-EmoPhone and DeepStress datasets. We primarily focused on the K-EmoPhone dataset, which features many users and a broader range of mobile and wearable sensor modalities than the DeepStress dataset. The DeepStress dataset was used to corroborate the findings from the K-EmoPhone. The results of DeepStress are discussed in Section 5.4.

4.1.1 K-EmoPhone Dataset. The K-EmoPhone dataset, introduced by Kang et al. [35], represents a rich amalgamation of sensor and self-reported data captured from 77 participants over a week-long period. This comprehensive dataset integrates heart rate metrics obtained from the Polar H10 ECG sensors with smartphone-sourced data, including GPS location, physical activity, application usage, voice call and SMS logs, Wi-Fi scanning history, device status, and battery consumption details. Complementing Polar H10 data and smartphone data, the Microsoft Band 2 wearable device provides additional physiological and behavioral data, such as heart rate via photoplethysmography (PPG), R-R intervals (RRI), galvanic skin response (GSR), skin temperature, three-dimensional acceleration, caloric expenditure, and step count. Table 15 lists the sensor data used in this study.

In addition to these sensor-based recordings, the dataset is enriched with personal information gathered through participant surveys. Emotional states were meticulously logged using the ESM method, where subjects rated their stress levels on a 7-point Likert scale ranging from -3 to +3. This method involved regular prompts

1. Preprocessing	2. Feature Extraction	3. Feature Preparation	4. Feature Selection
a. Remove invalid ESM samples **a.1** Remove expiratory **a.2** Remove neutral **b**. Remove invalid users **b.1** Remove users with too few ESM samples **b.2** Remove users with extreme label distribution **c**. Label encoding **c.1** Theoretical threshold **c.2** Statistical threshold for all users **c.3** Statistical threshold for each user	**a**. Feature type **a.1** Sensor data **a.2** Survey data **a.2.1** Participant information **a.2.2** EMA context data **a.2.3** Previous EMA labels **b**. Time window **b.1** Current (last value before label) **b.2** Immediate past (fixed time window before ESM) **b.3** Extended past (daily) **b.3.1** Epoch window **b.3.2** Whole time window	**a**. Feature normalization **a.1** For all users (the statistics measure such as mean and std is calculated from the training set) **a.2** For each user **b**. Impute missing values	**a**. Feature Selection a.1 Filter methods **a**.2 Wrapper methods **a**.3 Embedded methods
5. Data Splitting	**6. Oversampling/Undersampling**	**7. Model Training**	**8. Model Evaluation**
a. User-independent cross validation **a.1** Leave one subject out **a.2** Group-based k-fold cross-validation **b**. User-dependent cross validation **b.1** K-fold cross validation **b.2** Time series k-fold cross validation **c**. Partial personalization **c.1** Random **c.2** Stratified **c.3** Time series	**a**. Oversample the minority class or undersample the majority class a.1 Original Distribution a.2 Random oversampling a.3 Random undersampling a.4 SMOTE/SMOTE-NC	**a**. Personalized vs generalized **a.1** Fully personalized (only using a single user's data) **a.2** Similar-user model (only using similar user group's data) **a.3** Multi-task learning **a.4** Generalized model **b**. Model selection **b.1** Traditional machine learning models (**b.1.1** Gradient boosting, **b.1.2** RandomForest, **b.1.3** SVM, **b.1.4** logistic regression, **b.1.5** KNN, **b.1.6** decision tree, and **b.1.7** Naïve Bayes classifier) **b.2** Neural network models (e.g. MLP)	**a**. Metric selection **a.1** Accuracy **a.2** F1 score (positive) **a.3** macro F1 score **a.4** AUC-ROC **a.5** precision (PPV) **a.6** recall

Fig. 2. Stress Prediction Common Pipeline

Table 2. Related Work for Preprocessing & Feature Extraction

Steps	Factors	[54]	[8]	[24]	[16]	[30]	[21]	[50]	[23]	[14]	[65]	[67]	[3]	[35]	[31]	[70]
1.	a.1								✓			✓		✓		
	a.2														✓	
	b.1										✓	✓	✓	✓	✓	
	b.2															
	c.1		✓	✓	✓	✓	✓		✓	✓			✓	✓		
	c.2	✓									✓	✓			✓	
	c.3							✓								✓
2.	a.1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	a.2.1		✓									✓		✓		
	a.2.2										✓					✓
	a.2.3													✓		✓
	b.1													✓		
	b.2			✓		✓	✓		✓	✓				✓		✓
	b.3.1			✓							✓		✓			
	b.3.2	✓	✓		✓			✓	✓			✓	✓	✓	✓	

1. denotes preprocessing, and 2. denotes feature extraction. Tick means yes (i.e., presence of the alternative), and blank means no (i.e., absence of the alternative). Each row denotes one alternative of the factors in the pipeline. Please refer to the previous common pipeline figure for detailed information regarding the rows.

for emotion reporting, operational for 12 h daily-10 am to 10 pm-with an average prompt frequency of 45 min intervals, thereby fostering a nuanced timeline of emotional fluctuation. The stress label distribution is attached in Figure 10 in Appendix A.

4.1.2 DeepStress Dataset. For further analysis, we used the additional stress dataset named DeepStress which closely mirrors the K-EmoPhone but was gathered from 24 participants (i.e., fewer participants) over 6 weeks (i.e, a longer collection period) with fewer sensor modalities during the previous studies on the causal relationships in everyday life data [33, 34]. We selected this dataset because it encompasses both mobile sensor data and in-situ self-reported stress labels. Specifically, this dataset collected the participants' stress levels using a 5-point Likert Scale. Its composition is unidentical to that of the K-EmoPhone dataset; however, it offers crucial insights as it includes data on participants' locations, physical activities, and mobile app usage, which are vital for predicting stress levels. Detailed information on this DeepStress dataset including data types and ESM notifications is available at https://bit.ly/3whIEb2.

4.2 Baseline Pipeline

Establishing a baseline pipeline was crucial before designing the experiments to answer RQ 2, 3, and 4 (Figure 3). This baseline allows us to analyze the impact and sensitivity of each factor in the pipeline, which is essential for addressing RQ2 and reducing the complexity of potential factor combinations

Table 3. Related Work for Feature Preparation & Feature Selection & Data Splitting & Oversampling

Steps	Factors	[54]	[8]	[24]	[16]	[30]	[21]	[50]	[23]	[14]	[65]	[67]	[3]	[35]	[31]	[70]
3.	a.1		✓											✓	✓	
	a.2									✓						✓
	b												✓	✓	✓	✓
4.	a.1	✓	✓			✓						✓		✓		
	a.2	✓				✓	✓									
	a.3															
5.	a.1	✓		✓			✓			✓						
	a.2												✓	✓		✓
	b.1	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓			✓	
	b.2															
	c.1															
	c.2															
	c.3															
6.	a.1	✓	✓	✓	✓	✓	✓	✓			✓	✓			✓	✓
	a.2															
	a.3															
	a.4								✓	✓			✓	✓		

3. denotes feature preparation, 4. denotes feature selection, 5. denotes data splitting, and 6. denotes oversampling/undersampling.

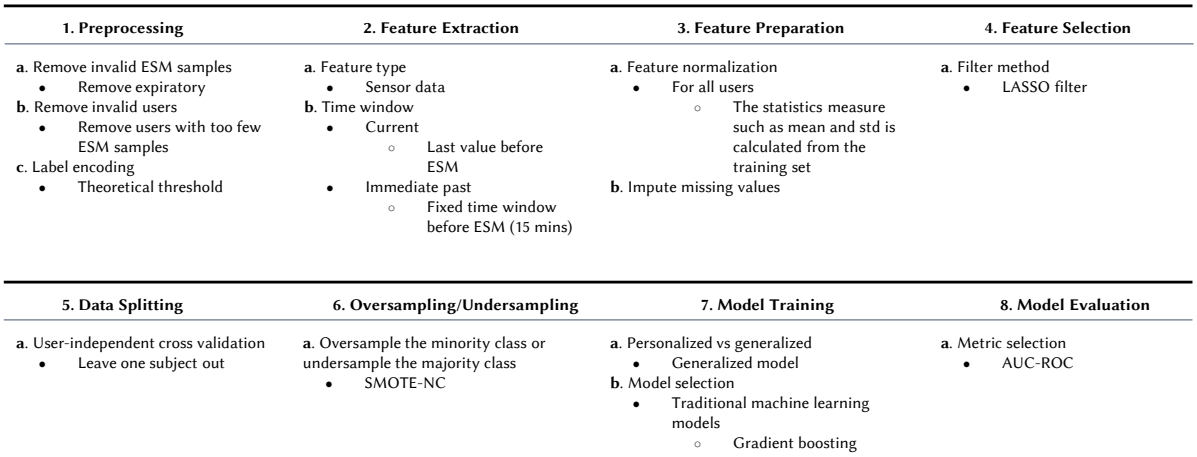


Fig. 3. Baseline pipeline

4.3 **Experiments for RQ2:** Impact of Pipeline Factors on Model Performance

4.3.1 Preprocessing. In the preprocessing step of our common pipeline, we focused on three main factors: removing invalid survey samples, removing invalid users, and label encoding (Figure 2). We consider two alternatives for removing invalid samples: expired samples-those answered 10 minutes past the scheduled time, and neutral state samples-representing the middle value on the Likert scale, specifically '0' from the 7-point Likert

Table 4. Related Work for Model Training & Model Evaluation

Steps	Factors	[54]	[8]	[24]	[16]	[30]	[21]	[50]	[23]	[14]	[65]	[67]	[3]	[35]	[31]	[70]
7.	a.1	✓			✓		✓			✓						
	a.2			✓			✓						✓			
	a.3										✓	✓				
	a.4	✓	✓		✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
	b.1.1		✓										✓	✓		✓
	b.1.2		✓	✓					✓					✓	✓	✓
	b.1.3		✓	✓	✓	✓			✓	✓					✓	✓
	b.1.4	✓							✓							✓
	b.1.5									✓					✓	
	b.1.6			✓		✓	✓			✓					✓	
	b.1.7					✓	✓								✓	
	b.2		✓					✓		✓	✓	✓			✓	✓
8.	a.1	✓	✓	✓	✓	✓	✓	✓				✓		✓	✓	✓
	a.2													✓		
	a.3							✓	✓	✓	✓			✓		✓
	a.4								✓			✓			✓	✓
	a.5						✓	✓					✓			✓
	a.6		✓				✓	✓								✓

7. denotes model training, and 8. denotes model evaluation.

scale in the K-EmoPhone dataset. The options for the removal of invalid users include either excluding users with insufficient labels (less than 35 in the K-EmoPhone dataset) or excluding those exhibiting extreme label distribution, where the count of the majority label is more than four times that of the minority label. Regarding label encoding, we explore three alternatives: (1) using a theoretical threshold which is the mid-value of the label range, (2) a statistical threshold averaged across all users, and (3) individual user mean thresholds. Note that statistical thresholds are applicable only if the data is split in a user-independent manner to avoid data leakage.

In contrast, our baseline pipeline, shown in Figure 3, simplifies the approach by only removing expired label samples, excluding users with too few ESM samples, and applying a theoretical threshold for label encoding. Additionally, the experiment design for preprocessing, detailed in the Goal-Question-Method (GQM) format in Table 5, outlines the structured approach we employed to evaluate these preprocessing steps.

4.3.2 Feature Extraction. Our feature extraction approach prepares hand-crafted features based on domain knowledge and literature review. Tables 13, 14, and 15 in Appendix A depict details such as reasoning for using this type of data, raw sensor data, preprocessing, information being aggregated into features, and extracted features.

Table 5. Experiment Design for Preprocessing, Feature Extraction & Preparation Steps

Goal	Questions	Methods
G1: Exploring how preprocessing affects stress detection model performance	Impact of excluding neutral states on performance	Evaluate performance without 'neutral state' (0) samples
	Effect of removing users with extreme labels on performance	Assess performance after excluding users with extreme label distributions (majority/minority >4)
	Influence of label binarization choice on the final performance	Explore effects of various label binarization thresholds
G2: Identify key features impacting real-world stress detection	Effect of adding participant information to sensor data on prediction	Compare sensor data only vs. sensor data + participant information
	Impact of including prior EMA labels on sensor-only predictions	Compare sensor data only vs. sensor data + previous stress label
	Model performance if using survey data only (participant information or the last EMA label)	Test participant information only and last stress label only
G3: Investigate how different time window sizes affect model performance	Performance fluctuations with time window adjustments	Experiment with immediate past time window sizes of 5, 10, 15, 30, and 45 mins before ESM survey label; max duration matches the average interval between labels
G4: Determine if current stress correlates with extended past events and if features from the extended past predict current stress levels	Impact of last night's sleep features on stress detection	Conduct experiments with features below excluding the first day's data for fairness: • Current and immediate past • Current, immediate past, and sleep data • Current, immediate past, and today's epoch • Current, immediate past, and yesterday's epoch • Current, immediate past, and today's full-time window (6 am to label timestamp) • Current, immediate past, and yesterday's full-time window (6 am to end of day)
	Effect of extended past daily features on stress detection	
	Optimal look-back period for extended past features	
	Effectiveness of epoch time window for extended past features	
G5: Evalute if user-specific normalization of sensor features enhance model generalization; aim to validate or refute this approach	Difference in performance between user-specific feature normalization and normalization for all users	Contrast user-specific standard normalization with standard normalization for all users, where mean and standard deviation are derived from the training set

In the study's common pipeline, we consider two primary factors: (a) feature type and (b) time window. Feature type involves both passive sensor data and survey data, which includes participant information, ecological momentary assessment (EMA) context data, and previous stress labels. Regarding time windows, we differentiate among the current time window, the immediate past time window, and the extended past/daily time window.

The rationale behind our time window settings draws heavily on prior research. We refer to Choy et al. [13] who experimented with both immediate and extended past time windows, as well as Gjoreski et al. [24] who utilized similar settings for short-term and relative epoch features. Kang et al. [35] introduced the use of the current time window, capturing the last value before the label timestamp. However, extended past/daily time windows have

Table 6. Experiment Design for Feature Selection, Data Splitting, Oversampling & Undersampling, and Model Training Steps

Goal	Questions	Methods
G6: Assessing filter methods for feature selection in high-dimensional sensor data	Difference in performance using only LASSO filter versus combined zero variance and pairwise correlation filters	Compare various combinations: • LASSO filter only • LASSO filter + zero variance filter • LASSO filter + pairwise correlation filter • LASSO filter + zero variance filter + pairwise correlation filter
G7: Examine k-fold cross-validation reliability with mixed user IDs and shuffled times	Success of k-fold cross-validation: impact of mixed users, shuffled order, or both	Compare group k-fold vs. standard k-fold cross-validation, time-series k-fold vs. standard k-fold cross-validation (requires temporal sorting)
G8: Assess partial personalization cross-validation's effects, its temporal relation, and the necessary data amount	Performance improvement with partial personalization	Compare LOSO partial personalization with 50% target user data for training and rest for testing vs. LOSO with 50% data for testing only and rest unused
	Relation between partial personalization's high performance and shuffled temporal order	Comparing time series partial personalization with stratified and random partial personalization
	Percentage of test user data needed for partial personalization	Partial personalization ratios: 10%, 30%, 50%, 70%, and 90% for time-series approach
G9: Test if oversampling or undersampling improves model performance and determine which method is the most effective	Effect of oversampling or undersampling on model performance	Compare original distribution with oversampling or undersampling methods
	Most effective method for improving performance: oversampling or undersampling	Evaluate model performance across different oversampling and undersampling methods
G10: Determine the optimal personalization level and model type for best performance, considering variations in existing studies	Superior performance comparison: similar-user model, multi-task learning, or generalized model	Compare the baseline model (generalized model) to multi-task learning and similar-user models
	Best performing model type	Performance evaluation with multiple models: XGBoost, RandomForest, SVM, Logistic Regression, KNN, decision tree, Naive Bayes classifier, and MLP

been criticized for diluting feature expressiveness, as pointed out by Choy et al. [13], who recommend epoch segmentation to better capture temporal variations. Similarly, Taylor et al. and Fukazawa et al. [18, 67] discuss the benefits of epoch time windows for extended past/daily feature extraction. Bogomolov et al. [8] explored using data from the past few days, discussing the impact of the size of the time windows, such as using one or two days for feature extraction.

For the extended past time window, as illustrated in Figure 4, we employ an epoch-based feature calculation approach. 'Yesterday epoch features' are computed within three-hour intervals starting at 6 a.m. and concluding at the end of the day, while 'today epoch features' are calculated using the same interval but only extend up to the time the label is recorded. This method ensures the extraction of distinct features for each epoch. Additionally, sleep data, as noted in the feature engineering table, is derived from screen events with the sleep event start time fixed between 9 p.m. the previous day and 7 a.m. on the current day, assuming a non-shift worker schedule.

In the baseline pipeline, shown in Figure 3, we utilize only sensor data among the feature types and consider the current and immediate past time window (past 15 minutes). The feature extraction process is organized

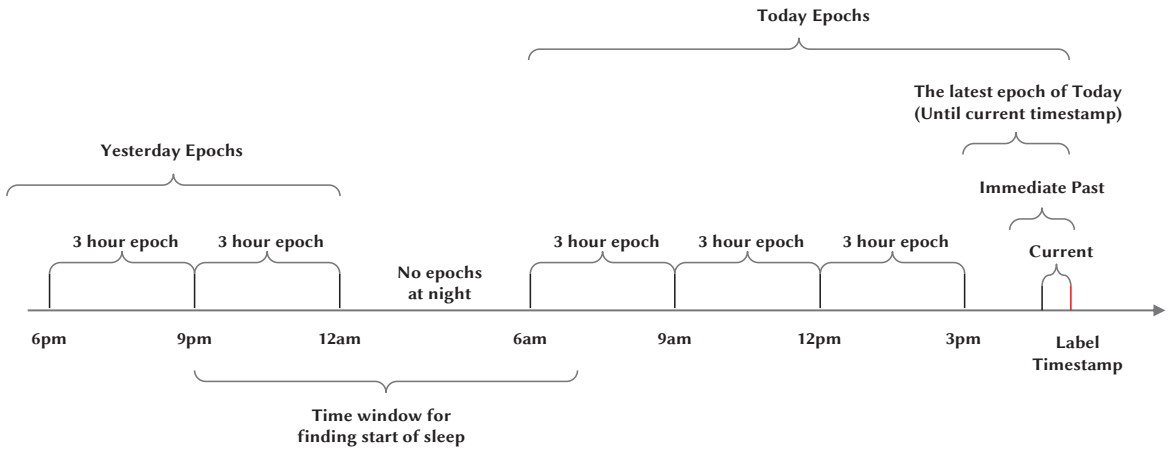


Fig. 4. Time window setting details

under the GQM format in G2, G3, and G4 of Table 5, structuring our approach to ensure thorough evaluation and analysis.

4.3.3 Feature Preparation. In the common pipeline for feature preparation, we focus on two primary factors: normalization and data imputation. There are two approaches to normalization: one involves normalizing features for each individual user, supported by some literature that suggests user-specific normalization can enhance the generalization capability of predictive models [53]. The other approach is to normalize features across all users, being careful to avoid data leakage by ensuring that the statistics used for normalization are derived solely from the training set. For data imputation, our standard method involves interpolating numeric features and forward-filling categorical features, and this approach is consistently applied throughout our experimental evaluations without variations in the imputation methods tested. In the baseline pipeline, as shown in Figure 3, normalization is applied across all users, and missing values are imputed using the aforementioned methods by default. The feature preparation phase of our study, described as step 3 in our experimental design, is methodically organized in G5 of Table 5, ensuring a structured and thorough evaluation process.

4.3.4 Feature Selection. In our common pipeline (Figure 2), feature selection is crucial, with the main factor being the type of feature selection method utilized. We consider three options: filter methods, wrapper methods, and embedded methods [4, 81]. Filter methods involve selecting features based on specific criteria, such as the LASSO to perform both variable selection and regularization, zero variance filtering, and elimination of high pairwise correlation [6]. Specifically, the LASSO filter removes features whose coefficients are reduced to zero; the zero variance filter removes features with nearly constant values; and the pairwise correlation filter removes features with high pairwise correlation, indicating redundancy. The second alternative, wrapper methods, involves selecting subsets of features and assessing their effectiveness by training models on them, though this method is less favored due to its high computational demands. The third alternative, embedded methods, incorporates feature selection directly into the model training process, as seen when using LASSO during model training, which automatically selects features.

Despite the available methods, as shown in Table 3, previous studies rarely apply embedded methods, and only a few have utilized wrapper methods. In this paper, we will focus exclusively on different filter methods for feature selection. In the baseline pipeline, as outlined in Figure 3, we default to using the LASSO filter for feature

selection. The approach to feature selection in our study is structured under the GQM format, organized in G6 of Table 6, ensuring a systematic evaluation of the methods tested.

4.3.5 Data Splitting. In the realm of mobile stress detection which is our common pipeline, data splitting techniques are categorized into three primary approaches. The first, subject-independent cross-validation, employs methods such as Leave-One-Subject-Out (LOSO) and group-based k-fold cross-validation. These methods segment the data based on unique user identifiers, ensuring that all data from a single user falls within the same fold, as illustrated in Figure 5. The second approach, subject-dependent cross-validation, includes techniques like time-series k-fold cross-validation, which considers the temporal order of data but ignores user identifiers, and the standard k-fold cross-validation, which treats the dataset as a collection of independent observations, disregarding both user ID and temporal sequence as shown in Figure 6. The third approach, partial personalization cross-validation, merges aspects of subject-independent validation (either LOSO or group-based k-fold) with the inclusion of a subset of the test users' data within the training set, as depicted in Figure 7. This method has been explored in previous studies [45, 69], highlighting its potential for improved model personalization.

In this paper, our focus will be on LOSO combined with a subset of the target user's data for partial personalization. This approach has three variations based on how the subset of the test user's data is selected. Previous research has primarily investigated the first two types: random selection and stratified selection, which consider both label distribution and user ID. The third type, which has not been explored in prior studies, involves selecting the subset based on temporal order. For instance, the first 50% of a test user's data might be included in the training set to enhance model personalization.

In our baseline pipeline, as shown in Figure 3, LOSO will be used as the default cross-validation method. The design and methodology of our data splitting process are meticulously outlined in the GQM format, presented in G7 and G8 of Table 6, to ensure a comprehensive evaluation of the various approaches.

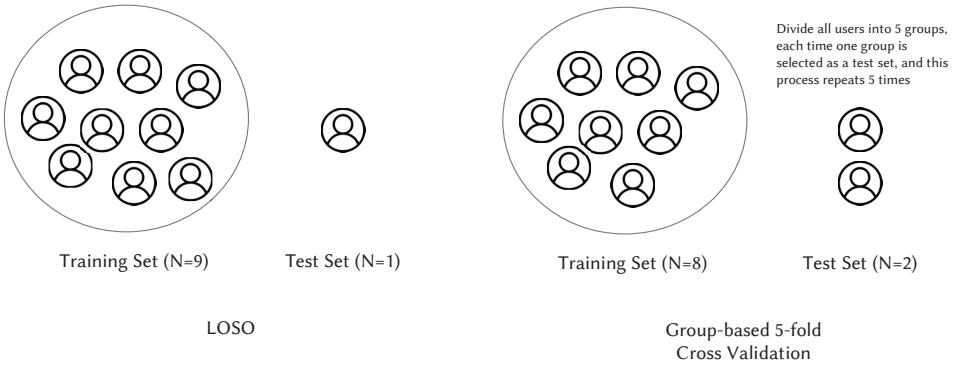
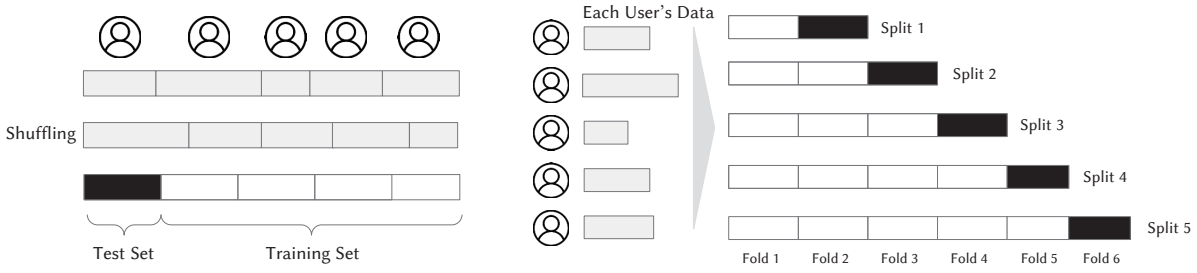


Fig. 5. Subject Independent Cross-Validation (assuming 10 users in total)

4.3.6 Oversampling and Undersampling. In the domain of mobile stress detection, the prevalence of non-stress states creates an inherent label imbalance challenge, skewing the dataset towards one (majority) class. Data imbalance is closely related to how labels are binarized (i.e., encoded) from Likert-scale stress self reports. In other words, label encoding is inherently linked to how labels are initially defined, which is influenced by factor c label encoding (more details in Preprocessing step). If labels are encoded using the mean of all users' Likert responses, it could lead to a situation where the labels for certain users are exclusively categorized as either positive or negative. This is due to baseline level of subjective stress received by each user. For instance, user A is



(a) Standard 5-fold Cross Validation
After random shuffling, divide all data samples into 5 groups. Each time 1 group is selected as a test set. This process repeats 5 times.

(b) Time Series 5-fold Cross Validation
Each user's data is first sorted by time and then split into 6 folds. For the first split, use all users' first fold as a training set and the second as a test set. Rep eat this process five times with increasing size of the training set.

Fig. 6. Subject Dependent Cross-Validation (assuming 5 users in total)

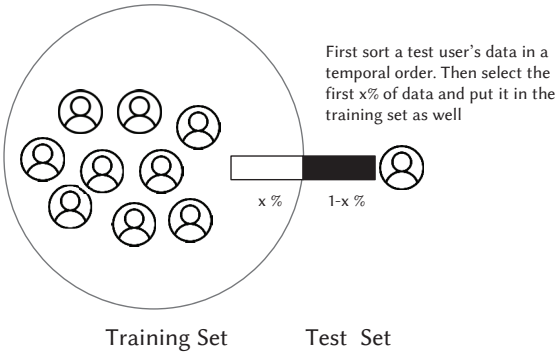


Fig. 7. Partial Personalization Cross-Validation (assuming 10 users in total)

usually not stressed, while user B is usually stressed. Furthermore, addressing how to categorize neutral states in a binary classification framework presents additional challenges and can contribute to label imbalance. For instance, if the categories of 'stressed' and 'not stressed' are initially balanced, reclassifying 'neutral' states as 'stressed' would disrupt this balance and exacerbate label imbalance. To counteract this imbalance, researchers often employ oversampling/undersampling strategies like the Synthetic Minority Over-sampling Technique for Nominal and Continuous data (SMOTE-NC) [11], random oversampling, and random undersampling.

As shown in the baseline pipeline Figure 3, SMOTE-NC will be used by default. This method is adept at augmenting minority class instances in mixed data types, containing both categorical and numerical features. Oversampling is done on the training set only since the test set should not be manipulated in the training stage to avoid data leakage. The experiment for step 6. Oversampling/Undersampling is organized in G9 of Table 6.

4.3.7 Model Training. In the domain of stress detection modeling, two pivotal considerations govern the training process: the degree of personalization and the selection of the modeling approach. Personalization ranges from fully personalized models, which utilize data from individual users for both training and testing, to models that are specific to groups of users clustered by demographic information or sensor data patterns. In our case, we will use demographic information for similar user clustering. Multi-task learning approaches expand upon group-specific

models by incorporating mechanisms for sharing information across user groups, thereby augmenting the models' versatility and performance. In contrast, generalized models are designed to operate across the entirety of the user base, encompassing a broad spectrum of user behaviors within a single predictive framework. In this paper, we will not test the fully personalized model because of the dataset size of the K-EmoPhone dataset and only consider a similar-user model, multi-task learning model, and generalized model. The detailed explanation for personalized models is available in the Supplementary Material. The selection of the appropriate model constitutes the second major consideration, with options spanning from conventional machine learning algorithms to sophisticated deep learning networks. However, we will only focus on traditional machine learning models. A more detailed discussion regarding the application of deep learning methods in this domain will be given in Section 6.

As shown in the baseline pipeline Figure 3, the generalized XGBoost model will be used by default as the baseline in the experiments. It is noteworthy that our default model, XGBoost, requires a separate validation set in addition to training and test sets. This is used for early stopping. One of the pitfalls in the experimental setting is to use of a test set as a validation which causes data leakage, subsequently leading to optimistic model performance metrics. Hence, we exclusively form a validation set as 20% of the training set. This setting ensures the test set is only used once in the end for model evaluation. The experiment for step 7. Model Training is organized in G10 of Table 6.

4.3.8 Model Evaluation. Evaluative metrics are critical to the assessment of model efficacy, with the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) metric being paramount, especially in the context of imbalanced datasets [6]. This metric's resilience to label imbalance makes it a preferred choice for validating models within this research. Given the prevalence of imbalanced stress labels, our study will concentrate on macro AUC-ROC as the primary measure of performance, ensuring a consistent and rigorous evaluation of the predictive capabilities of our stress detection models.

4.4 **Experiments for RQ3:** Improving Model Performance in User-independent Settings

To answer RQ3 on how to improve model performance in user-independent settings, we try to tune the factors in the common pipeline to push the model performance following the same way as Sano et al. and Hung et al. [30, 63]. The experiments started from the baseline pipeline (explained in experiments for RQ2). All factors except for LOSO cross-validation, generalized model, and macro AUC-ROC metric will be tuned to improve the model's performance (i.e., discriminatory ability).

4.5 **Experiments for RQ4:** Analysis of Low Performance in User-independent Settings

Similar to Pratap et al. [60], we dive into the individual user level of data and compare the high-performance users and low-performance users in the baseline pipeline model explained in experiments for RQ2. Confusion matrix, feature importance, and label distribution will be examined to find the difference between high-performance and low-performance users which could potentially cause the performance difference.

5 Results

5.1 **RQ2** Results for Reproducibility Experiments

Our study employs three baseline models as shown in Table 7. Baseline 1, depicted in Figure 3 (Section 4), serves as our primary baseline model. Baseline 2, which excludes the first day's stress labels, is used in extended past feature experiments (G4 in Table 5) to ensure comparability by avoiding null values on the initial day. Baseline 3 facilitates a fair comparison between the baseline model and a partial personalization approach that utilizes 50% of the target user's data. It employs LOSO cross-validation, where only 50% of the test user's data is used in the test set, leaving the remainder unused.

The results for Steps 1 to 3 in the pipeline are shown in Table 8, while results for the rest of the pipeline are in Table 9. Specifically, the stratified random partial personalization experiment (LOSO + stratified 50% test user) was not conducted due to the lack of labels per user in the K-EmoPhone dataset and challenges in stratifying samples based on label distribution, as labels for some users were unevenly distributed. Results from RQ2 experiments are summarized in Table 10, addressing all questions from the previous methodology experiment design.

Our findings suggest that removing neutral state samples may improve model performance, but this could be overly optimistic as the neutral state is crucial for affective computing [22]. Despite addressing label imbalance issues through label encoding for each user or removing extreme users, performance improvements remain modest. This finding aligns with [48, 52, 70], which noted the meaningful help provided by previous survey labels.

Regarding window size, no clear relationship was observed, yet it still influences model performance, echoing Yu et al.'s findings about sequence length [84]. Extended past features' improvement compared to baseline 2 was limited and epochs were not necessarily better than whole time windows, differing from Choy et al.'s observations in mobile interruptibility prediction [13]. However, their use of k-fold cross-validation may explain the discrepancy. Normalization for each user did not significantly enhance performance compared to normalization across all users, diverging from previous assumptions [53].

Partial personalization showed a slight performance improvement in the LOSO setting, albeit smaller than previous findings [45]. The original study's use of group-5-fold cross-validation and a larger dataset size might explain the modest improvements seen in our study. This is consistent with Trazav et al.'s work [69], where LOSO partial personalization showed minimal enhancement.

Furthermore, time-series partial personalization performed even worse than random partial personalization, emphasizing the importance of temporal order and suggesting that time-series methods are more applicable in real-world settings. Multi-task learning and similar-user models did not improve the model performance as reported in previous studies [65, 67], possibly due to the LOSO cross-validation and the small dataset size.

It is noteworthy that the previous results and comparisons are done based on LOSO cross-validation which may cause different conclusions and also on the limited dataset K-EmoPhone. The comparisons will be revisited with the DeepStress dataset offering more data per user in a subsequent subsection.

Through analysis of the results for reproducibility experiments, RQ2 impact of each factor on the model performance is answered.

5.2 **RQ3** Results for Pushing the Performance Limits

To enhance the performance of our generalized models, we experimented with various parameter combinations, focusing on the AUC-ROC due to its relevance in scenarios with imbalanced labels. These efforts, conducted in a LOSO setting, are detailed in Table 11. As shown in Table 7, Baseline 2 yields a higher AUC-ROC than Baseline 1. Consequently, our subsequent optimization experiments are based on Baseline 2, consistently excluding data from the first day as standard practice.

5.2.1 Sensor Data Only. The highest AUC-ROC achieved using only sensor data was 58%, combining features from the baseline model and yesterday's whole time window features. As in Table 11, the best model with sensor

Table 7. Results for Baseline

Experiment Type	Accuracy	F1-Score (pos. label)	Macro F1-Score	AUC-ROC	Precision	Recall
Baseline 1	0.600	0.294	0.494	0.518	0.521	0.517
Baseline 2 (removing 1st day's data)	0.620	0.302	0.502	0.556	0.536	0.528
Baseline 3 (using 50% data for testing)	0.604	0.281	0.485	0.511	0.506	0.511

Table 8. Results for Preprocessing, Feature Extraction & Preparation

Experiment Type	Details	Accuracy	F1 Sco. for Pos.	Macro F1 Score	AUC	Precision	Recall
G1: Label Processing	Removing Neutral States	0.551	0.408	0.497	0.543	0.552	0.544
	Removing Extreme Users	0.568	0.402	0.512	0.533	0.528	0.529
	Label binarization (mean for all users)	0.534	0.500	0.485	0.539	0.527	0.528
	Label binarization (mean for each user)	0.529	0.458	0.503	0.524	0.524	0.525
G2: Using Different Feature Types	Previous survey label data only	0.668	0.424	0.570	0.570	0.571	0.570
	Pre-experiment survey only	0.575	0.281	0.353	0.500	0.288	0.500
	Sensor + Pre-experiment survey only	0.599	0.303	0.492	0.533	0.512	0.508
	Sensor + Previous survey label data	0.626	0.343	0.525	0.568	0.545	0.536
G3: Using different time window sizes for immediate past time window	5 mins	0.606	0.313	0.501	0.518	0.523	0.521
	10 mins	0.611	0.334	0.516	0.542	0.539	0.535
	30 mins	0.603	0.321	0.506	0.546	0.529	0.527
	45 mins	0.597	0.305	0.499	0.535	0.529	0.520
	Today epochs only	0.604	0.340	0.511	0.542	0.538	0.532
	Noterday epochs only	0.547	0.289	0.453	0.500	0.475	0.499
G4: Using extended past features	Current + immediate past + sleep	0.607	0.288	0.492	0.561	0.516	0.521
	Current + immediate past + today epochs	0.621	0.303	0.504	0.559	0.551	0.525
	Current + immediate past + yesterday epochs	0.610	0.307	0.502	0.560	0.538	0.528
	Current + immediate past + today whole time window (aggregated over all epochs)	0.616	0.302	0.502	0.543	0.526	0.523
	Current + immediate past + Yesterday whole time window (aggregated over all epochs)	0.630	0.317	0.514	0.580	0.536	0.539
G5: Feature Normalization	Standard Normalization for each user	0.544	0.336	0.480	0.536	0.530	0.523

data alone only reached a 51.4% macro F1 score, potentially influenced by the universal binarization threshold (0.5) used for label probability. Future studies could enhance the macro F1 score by optimizing this threshold.

5.2.2 Sensor and ESM Data. Our combination of steps below produced the best-performing model:

- Exclusion of neutral state samples (0 values).
- Omission of users with an excessively skewed label distribution (majority label ratio > 0.8).
- Incorporation of immediate past sensor features and current ESM data (last stress label).
- Elimination of features with zero variance, high pairwise correlation, and application of LASSO for feature selection.
- Reduction of the evaluation set ratio from 20% to 10% of the training data.

This refined approach yielded our best AUC-ROC of 63.1%, surpassing the performance of models relying solely on ESM data (59.6% AUC-ROC) and sensor data only (58% AUC-ROC). This finding underscores the value of integrating sensor and ESM data for more accurate stress detection, demonstrating that the combination of these data sources is superior to using ESM data alone and sensor data only. However, there are some limitations as removing neutral states may not be realistic in real-world applications.

RQ3 regarding the best performance we can achieve is answered by tuning the factors in the pipeline. It is also noteworthy that the past ESM label is important for achieving the best performance.

Table 9. Results for Feature Selection, Data Splitting, Oversampling & Undersampling, and Model Training

Experiment Type	Details	Accuracy	F1 Sco. for Pos.	Macro F1 Score	AUC	Precision	Recall
G6: Feature selection	Remove 0 variance features + LASSO only	0.593	0.295	0.489	0.532	0.521	0.515
	Remove features with high pairwise correlation + LASSO only	0.597	0.307	0.496	0.525	0.519	0.512
	Remove 0 variance features + Remove features with high pairwise correlation + LASSO	0.602	0.318	0.503	0.539	0.524	0.518
G7 & G8: Data splitting	Group k fold	0.598	0.349	0.528	0.562	0.539	0.532
	Time-series k fold	0.615	0.389	0.553	**0.588**	0.558	0.553
	k fold	0.650	0.462	0.601	**0.650**	0.607	0.599
	LOSO + random 50% test user	0.632	0.319	0.510	**0.552**	0.549	0.528
	LOSO + stratified 50% test user	NA	NA	NA	NA	NA	NA
	LOSO + first 50% test user	0.599	0.255	0.466	0.534	0.504	0.509
	LOSO + first 10% test user	0.609	0.324	0.508	0.558	0.534	0.524
	LOSO + first 30% test user	0.628	0.317	0.512	0.570	0.532	0.533
	LOSO + first 70% test user	0.609	0.268	0.469	0.557	0.506	0.517
	LOSO + first 90% test user	0.619	0.257	0.510	0.576	0.529	0.539
G9: Oversampling & Undersampling	Original distribution	0.602	0.217	0.463	0.511	0.517	0.502
	Random oversampling	0.580	0.290	0.484	0.527	0.512	0.511
	Random undersampling	0.540	0.410	0.491	0.530	0.523	0.521
G10: Using different levels of personalization	Multi-task learning	0.562	0.335	0.467	**0.541**	0.498	0.514
	Similar-user model	0.520	0.339	0.457	0.514	0.502	0.500
G10: Model training using different machine learning models	RandomForest	0.631	0.242	0.487	0.535	0.539	0.524
	SVM	0.606	0.307	0.500	0.542	0.539	0.519
	Logistic Regression	0.569	0.327	0.483	0.508	0.510	0.510
	KNN	0.453	0.459	0.425	0.535	0.518	0.515
	Decision Tree	0.550	0.346	0.483	0.499	0.504	0.499
	Naïve Bayes classifier	0.504	0.415	0.466	0.524	0.521	0.524
	MLP	0.593	0.323	0.503	**0.548**	0.532	0.529

5.3 **RQ4** Results for Exploring Low-Performance Reasoning

As outlined in Section 4. Methodology, the study compares the top four highest-performing users with the four lowest to identify key differences, similarly done in Pratap et al. [60]. The primary distinction lies in label distribution: lower-performing users often exhibit a more skewed label distribution. Details are available in Figures 11, 12, and 13 in Appendix A. Attempts to balance this by binarizing labels using the user-specific mean led to only marginal performance improvements, as shown in Table 8 under the factor **mean for each user**. Performance enhancements were observed in some previously lower-performing users, while others experienced a decrease which is a sign of overfitting.

To further validate the existence of overfitting, we analyzed AUC-ROC results for all splits in the LOSO approach for both training and test sets (Figure 8). The figure reveals a marked performance discrepancy between training and test datasets, coupled with high variance across folds—classic indicators of overfitting.

Observations:

- **Performance Discrepancy**: A generalization gap suggests the model's inability to maintain its predictive accuracy on unseen data.

Table 10. Answers to Reproducibility Experiment Questions

G1: Preprocessing	After removing neutral state samples/extreme users/label binarization using the mean threshold of all users/each user, the labels are more balanced. The AUC-ROC improves using all these methods. Among all the techniques, removing neutral states gives us the best improvement but it is not realistic in real-world application.
G2: Feature Extraction: feature types	Pre-data-collection survey data slightly improves AUC-ROC. Previous ESM stress labels did help improve the performance. However, only using previous ESM stress labels achieves similar performance compared with using both previous labels and sensor data.
G3: Feature Extraction: size of time windows	There is no clear linear relationship between performance and time window size.
G4: Feature Extraction: extended past features	Sleep data does not improve AUC significantly. The improvement is ignorable. Removing 1st day's data improves performance. Extended past features do help improve the performance. It is better to look at yesterday. Epoch time window is not necessarily better than the whole time window.
G5: Feature Preparation	Normalizing features for each user improves AUC-ROC by around 2%.
G6: Feature Selection	Remove 0 variance features + Remove features with high pairwise correlation + LASSO works the best, slightly improve AUC-ROC
G7 & 8: Data Splitting	Standard k-fold cross-validation result is much higher than time-series k-fold (6% in AUC-ROC). Using standard k-fold cross-validation could be overly optimistic. Partial personalization does help improve performance. The first 30% of the test user's data will be good enough for partial personalization. Besides, if we ignore temporal order for partial personalization, we may achieve overly optimistic results as well.
G9: Oversampling & Undersampling	The oversampling and undersampling methods are slightly improving the performance. However, the difference between different oversampling and undersampling methods is very limited.
G10: Model Training	Multi-task learning >generalized model >similar-user model MLP works best in terms of AUC-ROC.

Table 11. Results for Best Performing Models

Experiment Type	Acc.	F1-Score (pos. label)	Macro F1-Score	AUC-ROC	Prec.	Recall
Using both ESM and sensor data	0.621	0.584	0.587	0.631	0.598	0.604
Only using sensor data	0.630	0.317	0.514	0.580	0.536	0.539

- **Inconsistent Optimization**: Efforts to recalibrate the model for lower-performing users did not yield a commensurate improvement in overall performance. This implies that the model, initially overfitted to high-performing users, merely shifted its overfitting bias to the newly optimized group.

Note that the issue of overfitting primarily occurs in user-independent cross-validation, which may arise from the distribution shift across different users, as reported in Xu et al.'s work[79]. As potential solutions, previous

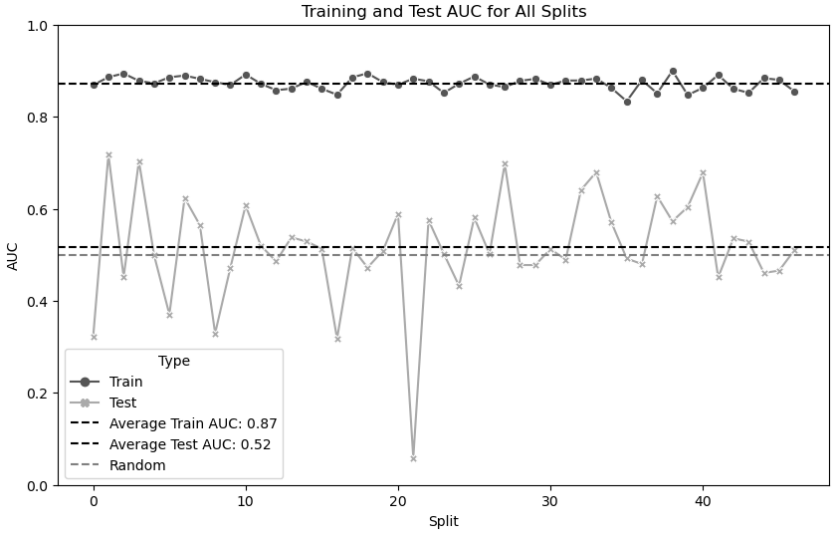


Fig. 8. Train & Test AUC-ROC for Baseline Pipeline

studies [79, 83] proposed tuning hyperparameters related to model complexity, regularization, and early stopping to mitigate overfitting issues. Detailed hyperparameters are available in Table 12. After parameter tuning with Hyperopt, we achieved 56.9% AUC-ROC. This marks some improvement compared to baseline performance; however, the overfitting issue persists, evidenced by the gap between the training set and test set performances.

Through a comparative analysis of high- and low-performance users, along with the visualization of training and testing performances, overfitting emerged as a potential cause of suboptimal performance in real-world mobile stress prediction. This issue appears to be particularly prevalent in user-independent scenarios, likely due to a distribution shift across users. Despite efforts to mitigate overfitting through hyperparameter tuning, the performance improvements remained modest.

5.4 Results of the DeepStress Dataset

As for RQ2 reproducibility experiments, we present the results of the DeepStress dataset in Tables 16, 17, 18 in Appendix A. We note that the combination of current, immediate past, and sleep features was not examined in the new dataset due to a lack of sleep data. In the K-EmoPhone dataset, sleep information was derived from the screen on/off events, which were not available in the new dataset.

Table 12. Hyperparameters Related for Addressing Overfitting in XGBoost

Parameters	**Value Range**	**Parameters**	**Value Range**
max_depth	3 to 10 (integers)	learning_rate	0.01 to 0.2 (log scale)
min_child_weight	1 to 6 (integers, step size of 1)	n_estimators	100, 250, 500 (discrete choices)
subsample	0.6 to 1.0 (continuous)	reg_lambda	0.5 to 5.0 (continuous)
colsample_bytree	0.5 to 1.0 (continuous)	reg_alpha	0.0 to 2.0 (continuous)
colsample_bylevel	0.5 to 1.0 (continuous)	num_parallel_tree	1, 10, 20 (discrete choices)
gamma	0.0 to 0.5 (continuous)	early_stopping_rounds	10, 30, 50 (discrete choices)

For label preprocessing, removing neutral state samples does not improve performance, which could be due to the label size difference (566 per user, as opposed to 70 per user in K-EmoPhone). When the label size is large, the influence of neutral state samples on the discriminative power of a classifier appears to be minimal. For feature types, the last ESM label is still working. The last ESM label and sensor data without hyperparameter tuning is still worse than the last ESM label only. For time window size setting, similarly, there is no obvious linear relationship between model performance and window size. For extended past features, removing 1st day's data does not significantly improve the model performance. This could be because the period of the new dataset is much longer than that of K-EmoPhone, and the effect of the first day is small. In contrast, we found that removing the first day in K-EmoPhone helped us slightly mitigate overfitting. Generally, this type of extended past feature still improves the model performance. For feature normalization & selection, similarly, there is some improvement, but it is limited. For data splitting, Group k works worse than LOSO, which is counterintuitive. This could be because the number of participants in the new dataset is small (n=24) compared with the K-EmoPhone dataset (n=77). In group k cross-validation, there will be far fewer users in the training set compared with LOSO.

Similarly, on the new dataset, the time-series k-fold is worse than the k-fold. For partial personalization, similarly, using time-series partial personalization is worse than random partial personalization. However, partial personalization is improving model performance even in LOSO settings. On K-EmoPhone, the improvement is limited to less than 2% in AUC-ROC but around 5% on the DeepStress dataset, probably due to many more labels per user (70 vs. 566). This may also imply that enough amount of labeled data from target users is important for the success of partial personalization. Note that partial personalization here is mainly based on LOSO. Results regarding group-k partial personalization will be elaborated in the discussion section, and the improvement is much bigger than LOSO, which is consistent with Lakmal's paper [45]. For oversampling and undersampling, there exists similar limited improvement even with different ways of oversampling and undersampling. For personalized models, the performance is even lower than generalized models. It could be due to the lack of Big-Five personality information in the new dataset, which may be necessary for the successful grouping of similar users. For model types, while the best model differs from the K-EmoPhone dataset, it is consistent with K-EmoPhone results that there is some model type impact though limited.

As for RQ3, we achieved 0.563 AUC-ROC by using SVM for sensor data only and 0.616 AUC-ROC by combining sensor data and the last ESM label as features. Consistently, the last ESM label is important for model performance improvement. As for RQ4, the overfitting phenomenon still exists on the new dataset even though there are fewer sensor data types and more labels per user. Similar to the K-EmoPhone dataset, we also conducted parameter tuning related to model complexity, regularization, and early stopping. After parameter tuning, the performance was 0.537 AUC-ROC, and there is a minor improvement compared with the baseline, which is consistent with the finding on the K-EmoPhone dataset.

6 Discussion

6.1 Independent Reproducibility for Mobile Stress Detection

As suggested by Fukazawa et al. [18], the temporal order is important for realistic cross-validation settings, as shown in Section 5. We observed that time-series k-fold and time-series random partial personalization underperform compared to common k-fold cross-validation and random partial personalization. This discrepancy might be due to the different amounts of data used for testing the time series and common k-fold cross-validations, potentially biasing the comparison. However, both time series and random partial personalization use the same amount of data for training and testing; therefore, random partial personalization could be overly optimistic, failing to consider the temporal order.

Removing neutral states may yield overly optimistic results that are unrealistic in real-world situations. The last stress label improves the model performance even in user-independent cross-validation settings, indicating

that the temporal dependency of stress states is universal across users. Conversely, sensor data are more useful in user-dependent cross-validation and partial personalization settings, indicating that human behavior in sensor data is heterogeneous across users and requires data from the target user for effective personalization. The last ESM label could be useful in scenarios lacking target user data. When substantial data from the target user are available, the sensor data become more critical for stress prediction.

Among the metrics used, AUC-ROC emerged as the most reliable measure compared to accuracy, macro F1 score, and others. The LOSO and group-k cross-validation methods reflect the real implementation of new users more. LOSO represents the most challenging scenario for stress prediction models, and a model that performs well in LOSO is likely to succeed in other cross-validation settings.

6.2 Improving Real-World Mobile Stress Prediction Performance

6.2.1 Improving Prediction Performance via User-in-the-loop Strategies. As suggested by Toshnazarov et al. and Meegahapola et al. [45, 48, 70], using the last ESM label as a feature and incorporating part of the labeled data of the target user into the training set improves the model performance, consistent with our findings. This underscores the importance of labels from the target user and the efficacy of a "user-in-the-loop" strategy, which comprises two stages: using the last ESM label and implementing partial personalization.

In the initial phase, when data for partial personalization or training a personalized model are scarce, collecting stress labels is crucial because the last stress label could serve as a vital predictor. Once a sufficient label corpus is amassed in the latter stage, we can strategically transition to partially personalized models. Partial personalization is useful for the DeepStress dataset, but it is limited for the K-EmoPhone dataset. On the DeepStress dataset, the LOSO time-series partial personalization improved AUC-ROC by 5%, and the group-k-fold time series partial personalization achieved a 67.6% AUC-ROC, marking a 17% improvement compared to the group-k-fold without partial personalization. However, the performance improvement was minor in LOSO and the group-k-fold cross-validation for the K-EmoPhone dataset. This disparity likely stems from the new dataset having six weeks of data per user, compared to only one week in the K-EmoPhone dataset, which may be insufficient for effective partial personalization. Therefore, sufficient labels should be collected for partial personalization.

Collecting sufficient labeled data from the target user is beneficial; nevertheless, gathering ESM labels in real life remains challenging. Strategies for collecting more labels without burdening users are related to receptivity prediction [6] and designing active learning algorithms [68]. Compared with receptivity prediction, designing active learning algorithms that optimize performance without collecting excessive labels requires further research.

Another essential research question is how to effectively utilize the data from the target user. Meegahapola et al. introduced a domain-adaptation method that requires only unlabeled data from the target user [46]. However, research on supervised multimodal domain adaptation that uses labeled data from target users for mobile stress prediction is lacking. Previous research, such as that of Cahoon et al. [9], typically focused on a single or few modalities, such as step count and heart rate, for stress prediction.

6.2.2 Dealing with Overfitting. Our investigation suggests that overfitting is a key factor diminishing the effectiveness of generalized models in mobile stress detection. This issue correlates with the challenges of working with heterogeneous populations [50] and limited labeling [84]. Following the approach of Xu et al. [79], employing hyperparameter tuning for model complexity, regularization, and early stopping may mitigate the overfitting issue; nonetheless, the extent of performance improvement is limited.

In addition, particularly for the XGBoost model, selecting an evaluation set for early stopping in the models can influence the final performance. Different ways of evaluating set splitting from the training set exist, such as random splitting, stratified splitting, and LOSO. In addition, the size of the evaluation set is crucial. Determining how to choose an evaluation set that accurately represents unseen users or the general population is a compelling research challenge for addressing overfitting in this field.

Theoretically, using fewer types of sensor data and more data per user would help reduce overfitting. However, comparisons between the K-EmoPhone and DeepStress datasets indicate that this is not necessarily effective in user-independent settings; however, noticeable improvements exist in user-dependent scenarios. More labeled data per user is beneficial only when there is access to target users' data.

Overall, overfitting in user-independent settings may stem from distribution shifts across users. There are two methods for handling distribution shifts; domain generalization [79] and domain adaptation [46].

6.2.3 Application of Personalized Models. While performance improvements using multitask learning and similar-user models are quite limited for the two datasets, personalized models have shown the potential for model improvement in prior studies [67]. The main challenge in applying personalized models lies in their computational demands. Training specific models for each user or group of users can be resource-intensive, given a large user base. Several strategies can be used to address these challenges.

Personalized models may be trained for groups of similar users rather than each user. Additionally, for deep learning, some layers can be shared across user groups using multitask learning approaches. Moreover, incremental learning can be used to incrementally update an existing model with new data, thereby minimizing the need for full retraining and making it suitable for devices with limited computational resources. Finally, on-device learning could be an option. This method involves training models on the device by leveraging federated learning, using a combination of on-device and cloud resources to reduce data transmission and computational overhead [78].

6.2.4 Application of Deep Learning. Our derived pipeline remains valid when manually crafted features are inputs for the deep learning models. However, if we use an end-to-end deep learning model, the entire pipeline would be different because designing and manually extracting features or conducting feature selection methods would not be necessary. Regarding end-to-end deep learning, recent mobile/wearable emotion prediction/detection papers using end-to-end deep learning models mainly focus on continuous high-frequency sensor data collected via wearable and mobile devices instead of event log data which is also common in such datasets [41, 49, 84].

The challenge is that event log data, such as call logs and other low-frequency event log data, may be unsuitable for automated deep learning-based feature extractors, in other words, representation learning, especially for in-situ stress prediction, because the sequence length will be relatively short in this scenario (Figure 9). Another essential issue in using end-to-end deep learning models is the lack of labeled data [84] and dataset size in this field. In real-world experiments, notifications that allow users to report their emotions can be bothersome, making it challenging to collect sufficient labels for deep learning.

Researchers have recently developed possible solutions for this first multimodal challenge. Xu et al. [79] extracted features at different frequencies (from an epoch of a day to multiple days) to address this issue instead of using a unified time window or sequence length setting to handle the event log data.

6.3 Limitations and Future Work

This paper focuses solely on in-the-wild mobile self-reported stress prediction and does not explore the combined effects of various factors, primarily because of the exponential increase in computational complexity and potential reproducibility challenges. In addition, this paper does not cover end-to-end deep learning because of concerns about multimodal and label numbers. Future work could explore end-to-end deep learning in multimodal mobile stress prediction to address these gaps.

In a stress prediction pipeline, several parameters have crucial roles, including the threshold for the LASSO feature selection (a fixed threshold or the mean value of all features), data sorting by temporal order per user or across all users, and an embedded feature selection method. The data sorting method can influence the model's interpretation of temporal sequences, thereby affecting its performance. Additionally, using XGBoost for feature

selection and model training improved performance. However, numerous other factors within the pipeline should be further explored.

Some biases could also exist regarding the baseline in the experiments, especially when comparing different cross-validation methods. For example, comparing time-series k-folds and common k-folds could be biased because they use different amounts of data for training and testing. We have added one more baseline, 3, using only 50% of the target user's data for testing, and the rest are unused to make a fair comparison with partial personalization with 50% of the target user's data. However, aside from the 50%, there were 10%, 30%, 70%, and 90% partial personalization, and we did not create baselines for each of them to avoid information overhead.

Future research should also consider designing active learning algorithms that optimize the model performance with minimal label collection from users. Addressing distribution shifts across users and mitigating overfitting in user-independent settings will require further exploration of domain-adaptation techniques.

7 Conclusion

In this paper, we derived a common pipeline for in-the-wild mobile stress prediction by experimenting with the impact of each factor on the model performance, which is defined as independent reproducibility. This is the first paper to introduce independent reproducibility in this field. This concept is essential for the field because, without sufficiently comprehending the impact of each factor, researchers would find it challenging to replicate existing works given the lack of open code and dataset because of privacy issues in this field [43]. Some insights were derived through reproducibility experiments. In particular, disregarding the temporal order for cross-validation settings, such as random k-fold cross-validation and random partial personalization settings, can be overly optimistic. We also observed that data from target users are crucial for improving model performance, regardless of whether the last stress label is used as a feature or partial personalization with part of the target user's data in the training set. Finally, overfitting was identified as a potential cause of low model performance in user-independent cross-validation settings, which could be related to distribution shifts across users. More research should be conducted on domain generalization and adaptation in this field to handle distribution shifts and overfitting [46, 79].

Acknowledgments

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Korean government (MSIT) (2022R1A2C2011536).

References

[1] Saeed Abdullah, Mark Matthews, Elizabeth L. Murnane, Tanzeem Choudhury, and Geri Gay. 2014. Towards Circadian Computing: "Early to Bed and Early to Rise" Makes Some of Us Unhealthy and Sleep Deprived. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '14)*. ACM, New York, NY, USA, Association for Computing Machinery, New York, NY, USA, 673–684. https://doi.org/10.1145/2632048.2632100

[2] L. Acharya, L. Jin, and W. Collins. 2018. College life is stressful today–Emerging stressors and depressive symptoms in college students. *Journal of American College Health* 66, 7 (2018), 655–664. https://doi.org/10.1080/07448481.2018.1451869

[3] D. A. Adler, F. Wang, D. C. Mohr, and T. Choudhury. 2022. Machine learning for passive mental health symptom prediction: Generalization across different longitudinal mobile sensing studies. *PLOS ONE* 17, 4 (2022), 1–20. https://doi.org/10.1371/journal.pone.0266516

[4] Charu C. Aggarwal. 2014. *Data Classification: Algorithms and Applications.* Chapman and Hall/CRC. https://doi.org/10.1201/b17320

[5] R. Albertoni, S. Colantonio, P. Skrzypczyński, and J. Stefanowski. 2023. Reproducibility of Machine Learning: Terminology, Recommendations and Open Issues. *arXiv preprint arXiv:2302.12691* (2023). https://arxiv.org/abs/2302.12691

[6] Jumabek Alikhanov, Panyu Zhang, YoungTae Noh, and Hakil Kim. 2023. Design of Contextual Filtered Features for Better Smartphone-User Receptivity Prediction. *IEEE Internet of Things Journal* (2023). https://doi.org/10.1109/JIOT.2023.3331715

[7] Karim Assi, Lakmal Meegahapola, William Droz, Peter Kun, Amalia De Götzen, Miriam Bidoglia, Sally Stares, George Gaskell, Altangerel Chagnaa, Amarsanaa Ganbold, Tsolmon Zundui, Carlo Caprini, Daniele Miorandi, José Luis Zarza, Alethia Hume, Luca Cernuzzi, Ivano Bison, Marcelo Dario Rodas Britez, Matteo Busso, Ronald Chenu-Abente, Fausto Giunchiglia, and Daniel Gatica-Perez. 2023.

Complex Daily Activities, Country-Level Diversity, and Smartphone Sensing: A Study in Denmark, Italy, Mongolia, Paraguay, and UK. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 506, 23 pages. https://doi.org/10.1145/3544548.3581190

[8] Andrey Bogomolov, Bruno Lepri, Michela Ferron, Fabio Pianesi, and Alex (Sandy) Pentland. 2014. Daily Stress Recognition from Mobile Phone Data, Weather Conditions and Individual Traits. In *Proceedings of the 22nd ACM International Conference on Multimedia* (Orlando, Florida, USA) *(MM '14)*. Association for Computing Machinery, New York, NY, USA, 477–486. https://doi.org/10.1145/2647868.2654933

[9] Jordan L. Cahoon and Luis A. Garcia. 2023. Continuous Stress Monitoring for Healthcare Workers: Evaluating Generalizability Across Real-World Datasets. In *Proceedings of the 14th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics (BCB '23)* (Houston, TX, USA). ACM, New York, NY, USA, Association for Computing Machinery, New York, NY, USA, 5. https://doi.org/10.1145/3584371.3612974

[10] Y. S. Can, N. Chalabianloo, D. Ekiz, and C. Ersoy. 2019. Continuous Stress Detection Using Wearable Sensors in Real Life: Algorithmic Programming Contest Case Study. *Sensors* 19, 8 (2019), 1849. https://doi.org/10.3390/s19081849

[11] N.V. Chawla, K.W. Bowyer, L.O. Hall, and W.P. Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357. https://doi.org/10.1613/jair.953

[12] J. Chen, J. Rogers, C. Chen, and D. Kotz. 2021. Stress Detection Using Context-Aware Sensor Fusion From Wearable Devices. *IEEE Internet of Things Journal* 8, 15 (2021), 12148–12161. https://doi.org/10.1109/JIOT.2023.3265768

[13] Minsoo Choy, Daehoon Kim, Jae-Gil Lee, Heeyoung Kim, and Hiroshi Motoda. 2016. Looking back on the current day: interruptibility prediction using daily behavioral features. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16)*. Association for Computing Machinery, New York, NY, USA, 1004–1015. https://doi.org/10.1145/2971648.2971649

[14] Matteo Ciman and Katarzyna Wac. 2018. Individuals' Stress Assessment Using Human-Smartphone Interaction Analysis. *IEEE Transactions on Affective Computing* 9, 1 (2018), 51–65. https://doi.org/10.1109/TAFFC.2016.2592504

[15] Sharon M Crook, Andrew P Davison, and Hans E Plesser. 2013. Learning from the past: approaches for reproducibility in computational neuroscience. In *20 Years of Computational Neuroscience*. Springer, 73–102. https://link.springer.com/chapter/10.1007/978-1-4614-1424-7_4

[16] R. Ferdous, V. Osmani, and O. Mayora. 2015. Smartphone app usage as a predictor of perceived stress levels at workplace. In *2015 9th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*. IEEE, 225–228. https://doi.org/10.4108/icst.pervasivehealth.2015.260192

[17] Anna Ferrari, Daniela Micucci, Marco Mobilio, and Paolo Napoletano. 2020. On the Personalization of Classification Models for Human Activity Recognition. *IEEE Access* 8 (2020), 32066–32079. https://doi.org/10.1109/ACCESS.2020.2973425

[18] Y. Fukazawa, N. Yamamoto, T. Hamatani, K. Ochiai, A. Uchiyama, and K. Ohta. 2020. Smartphone-based Mental State Estimation: A Survey from a Machine Learning Perspective. *Journal of Information Processing* 28, 3 (2020), 650–669. https://doi.org/10.2197/ipsjjip.28.650

[19] Nan Gao, Soundariya Ananthan, Chun Yu, Yuntao Wang, and Flora D. Salim. 2023. Critiquing Self-report Practices for Human Mental and Wellbeing Computing at Ubicomp. arXiv:2311.15496 [cs.HC]

[20] Nan Gao, Mohammad Saiedur Rahaman, Wei Shao, and Flora D Salim. 2021. Investigating the Reliability of Self-report Data in the Wild: The Quest for Ground Truth *(UbiComp/ISWC '21 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 237–242. https://doi.org/10.1145/3460418.3479338

[21] Enrique Garcia-Ceja, Venet Osmani, and Oscar Mayora-Ibarra. 2015. Automatic Stress Detection in Working Environments From Smartphones' Accelerometer Data: A First Step. *IEEE Journal of Biomedical and Health Informatics* 20 (2015), 1053–1060. https://api.semanticscholar.org/CorpusID:3738191

[22] Karen Gasper. 2023. A Case for Neutrality: Why Neutral Affect is Critical for Advancing Affective Science. *Affective Science* 4 (2023), 458–462. https://doi.org/10.1007/s42761-023-00214-0

[23] Surjya Ghosh, Niloy Ganguly, Bivas Mitra, and Pradipta De. 2017. Evaluating effectiveness of smartphone typing as an indicator of user emotion. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*. 146–151. https://doi.org/10.1109/ACII.2017.8273592

[24] Martin Gjoreski, Hristijan Gjoreski, Mitja Lutrek, and Matja Gams. 2015. Automatic Detection of Perceived Stress in Campus Students Using Smartphones. In *2015 International Conference on Intelligent Environments*. 132–135. https://doi.org/10.1109/IE.2015.27

[25] Google Developers. n.d.. Generalization - Machine Learning Crash Course. https://developers.google.com/machine-learning/crash-course/generalization/video-lecture. Accessed: 2024-03-25.

[26] Tom Gross and Tony Malzhacker. 2023. The Experience Sampling Method and its Tools: A Review for Developers, Study Administrators, and Participants. *Proceedings of the ACM on Human-Computer Interaction* 7, EICS (2023), 182:1–182:29.

[27] Megha V. Gupta, Shubhangi Vaikole, Ankit D. Oza, Amisha Patel, Diana Petronela Burduhos-Nergis, and Dumitru Doru Burduhos-Nergis. 2022. Audio-Visual Stress Classification Using Cascaded RNN-LSTM Networks. *Bioengineering* 9, 10 (2022), 510. https://doi.org/10.3390/bioengineering9100510

[28] Mathias Harrer, Sophia H. Adam, Harald Baumeister, Pim Cuijpers, Eirini Karyotaki, Randy P. Auerbach, Ronald C. Kessler, Ronny Bruffaerts, Matthias Berking, and David D. Ebert. 2019. Internet interventions for mental health in university students: A systematic review and meta-analysis. *International Journal of Methods in Psychiatric Research* 28, 2 (2019), 1–18. https://doi.org/10.1002/mpr.1759

[29] S. Hosseini, S. Katragadda, R. T. Bhupatiraju, Z. Ashkar, C. W. Borst, K. Cochran, and R. Gottumukkala. 2021. A multimodal sensor dataset for continuous stress detection of nurses in a hospital. *arXiv preprint arXiv:2108.07689* (2021). https://arxiv.org/abs/2108.07689

[30] Galen Chin-Lun Hung, Pei-Ching Yang, Chia-Chi Chang, Jung-Hsien Chiang, and Ying-Yeh Chen. 2016. Predicting Negative Emotions Based on Mobile Phone Usage Patterns: An Exploratory Study. 5, 3 (2016), e160.

[31] Salar Jafarlou, Jocelyn Lai, Iman Azimi, Zahra Mousavi, Sina Labbaf, Ramesh C Jain, Nikil Dutt, Jessica L Borelli, and Amir Rahmani. 2023. Objective Prediction of Next-Day's Affect Using Multimodal Physiological and Behavioral Data: Algorithm Development and Validation Study. *JMIR Formative Research* 7 (2023). https://doi.org/10.2196/39425

[32] Taejae Jeon, Han Byeol Bae, Yongju Lee, Sungjun Jang, and Sangyoun Lee. 2021. Deep-Learning-Based Stress Recognition with Spatial-Temporal Facial Information. *Sensors* 21, 22 (2021), 7498. https://doi.org/10.3390/s21227498

[33] Gyuwon Jung, Sangjun Park, and Uichin Lee. 2024. DeepStress: Supporting Stressful Context Sensemaking in Personal Informatics Systems Using a Quasi-experimental Approach. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA.

[34] Gyuwon Jung, Sangjun Park, Eun-Yeol Ma, Heeyoung Kim, and Uichin Lee. 2024. A Tutorial on Matching-based Causal Analysis of Human Behaviors Using Smartphone Sensor Data. *ACM Comput. Surv.* (feb 2024). https://doi.org/10.1145/3648356 Just Accepted.

[35] S. Kang, W. Choi, C. Y. Park, N. Cha, A. Kim, A. H. Khandoker, L. Hadjileontiadis, H. Kim, Y. Jeong, and U. Lee. 2023. K-EmoPhone: A Mobile and Wearable Dataset with In-Situ Emotion, Stress, and Attention Labels. *Scientific Data* 10 (2023). https://doi.org/10.1038/s41597-023-01234-6

[36] Maryam Khalid and Akane Sano. 2022. Exploiting Social Graph Networks for Emotion Prediction. *Scientific Reports* 13 (2022), 60691.

[37] Mohammed Khwaja, Sumer S. Vaid, Sara Zannone, Gabriella M. Harari, A. Aldo Faisal, and Aleksandar Matic. 2019. Modeling Personality vs. Modeling Personalidad: In-the-wild Mobile Data Analysis in Five Countries Suggests Cultural Impact on Personality Models. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 3, Article 88 (sep 2019), 24 pages. https://doi.org/10.1145/3351246

[38] T. Kim, H. Kim, H. Y. Lee, H. Goh, S. Abdigapporov, M. Jeong, H. Cho, K. Han, Y. Noh, S. J. Lee, and H. Hong. 2022. Prediction for Retrospection: Integrating Algorithmic Stress Prediction into Personal Informatics Systems for College Students' Mental Health. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 20 pages. https://doi.org/10.1145/3491102.3517701

[39] S. Koldijk, M. Sappelli, M. Neerincx, S. Verberne, and W. Kraaij. 2014. The SWELL Knowledge Work Dataset for Stress and User Modeling Research. In *Proceedings of the 16th ACM International Conference on Multimodal Interaction (ICMI '14)*. ACM, 291–298. https://doi.org/10.1145/2663204.2663257

[40] Reed Larson and Mihaly Csikszentmihalyi. 2014. *The Experience Sampling Method.* Springer Netherlands, Dordrecht, 21–34. https://doi.org/10.1007/978-94-017-9088-8_2

[41] Boning Li and Akane Sano. 2020. Extraction and Interpretation of Deep Autoencoder-based Temporal Features from Wearables for Forecasting Personalized Mood, Health, and Stress. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 2, Article 49 (jun 2020), 26 pages. https://doi.org/10.1145/3397318

[42] S. M. Matingly, J. M. Gregg, P. Audia, A. E. Bayraktaroglu, A. T. Campbell, N. V. Chawla, V. Das Swain, M. De Choudhury, S. D. D'Mello, A. K. Dey, G. Gao, K. Jagannath, K. Jiang, S. Lin, Q. Liu, G. Mark, G. J. Martinez, K. Masaba, S. Mirjafari, E. Moskal, R. Mulukutla, K. Nies, M. D. Reddy, P. Robles-Granda, K. Saha, A. Sirigiri, and A. Striegel. 2019. The Tesserae Project: Large-Scale, Longitudinal, In Situ, Multimodal Sensing of Information Workers. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts (CHI'19 Extended Abstracts)*. https://www3.nd.edu/~dwang5/documents/chi19extendedabstracts.pdf

[43] M.B.A. McDermott, S. Wang, N. Marinsek, R. Ranganath, M. Ghassemi, and L. Foschini. 2021. Reproducibility in machine learning for health research: Still a ways to go. *Science Translational Medicine* 13, 586 (2021), eabb1655. https://doi.org/10.1126/scitranslmed.abb1655

[44] D. McDuff, A. Karlson, A. Kapoor, A. Roseway, and M. Czerwinski. 2012. AffectAura: An Intelligent System for Emotional Memory. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Association for Computing Machinery, New York, NY, USA, 849–858. https://doi.org/10.1145/2207676.2208545

[45] L. Meegahapola et al. 2023. Generalization and Personalization of Mobile Sensing-Based Mood Inference Models: An Analysis of College Students in Eight Countries. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*. ACM. https://dl.acm.org/doi/abs/10.1145/3569483

[46] L. Meegahapola, H. Hassoune, and D. Gatica-Perez. 2024. M3BAT: Unsupervised Domain Adaptation for Multimodal Mobile Sensing with Multi-Branch Adversarial Training. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT/ Ubicomp)* (2024).

[47] L. Meegahapola, A.R. Mader, and D. Gatica-Perez. 2024. Learning about Social Context from Smartphone Data: Generalization Across Countries and Daily Life Moments. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. To appear.

[48] Lakmal Meegahapola, Salvador Ruiz-Correa, Viridiana del Carmen Robledo-Valero, Emilio Ernesto Hernandez-Huerfano, Leonardo Alvarez-Rivera, Ronald Chenu-Abente, and Daniel Gatica-Perez. 2021. One More Bite? Inferring Food Consumption Level of College

Students Using Smartphone Sensing and Self-Reports. 5, 1, Article 26 (2021), 28 pages. https://doi.org/10.1145/3448120

[49] Abhinav Mehrotra and Mirco Musolesi. 2018. Using Autoencoders to Automatically Extract Mobility Features for Predicting Depressive States. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 3, Article 127 (sep 2018), 20 pages. https://doi.org/10.1145/3264937

[50] G. Mikelsons, M. Smith, A. Mehrotra, and M. Musolesi. 2017. Towards Deep Learning Models for Psychological State Prediction using Smartphone Data: Challenges and Opportunities. In *Proceedings of the NIPS Workshop on Machine Learning for Healthcare 2017 (ML4H 2017)*. https://doi.org/10.48550/arXiv.1711.06350

[51] V. Mishra, T. Hao, S. Sun, K. N. Walter, M. J. Ball, C. H. Chen, and X. Zhu. 2018. Investigating the Role of Context in Perceived Stress Detection in the Wild. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers (UbiComp '18)*. Association for Computing Machinery, New York, NY, USA, Association for Computing Machinery, New York, NY, USA, 1754–1759. https://doi.org/10.1145/3267305.3274147

[52] Varun Mishra, Gunnar Pope, Sarah Lord, Stephanie Lewia, Byron Lowens, Kelly Caine, Sougata Sen, Ryan Halter, and David Kotz. 2020. Continuous Detection of Physiological Stress with Commodity Hardware. *ACM Trans. Comput. Healthcare* 1, 2, Article 8 (apr 2020), 30 pages. https://doi.org/10.1145/3361562

[53] Varun Mishra, Sougata Sen, Grace Chen, Tian Hao, Jeffrey Rogers, Ching-Hua Chen, and David Kotz. 2020. Evaluating the Reproducibility of Physiological Stress Detection Models. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 4, Article 147 (dec 2020), 29 pages. https://doi.org/10.1145/3432220

[54] Amir Muaremi, Bert Arnrich, and Gerhard Tröster. 2013. Towards Measuring Stress with Smartphones and Wearable Devices During Workday and Sleep. *Bionanoscience* 3, 2 (2013), 172–183. https://doi.org/10.1007/s12668-013-0089-2

[55] K. Mundnich, B. M. Booth, M. L'Hommedieu, T. Feng, B. Girault, J. L'Hommedieu, M. Wildman, S. Skaaden, A. Nadarajan, J. L. Villatte, T. H. Falk, K. Lerman, E. Ferrara, and S. Narayanan. 2020. TILES-2018, a longitudinal physiologic and behavioral data set of hospital workers. *Scientific Data* 7 (2020). https://doi.org/10.1038/s41597-020-00630-y

[56] M. L. Pariat, A. Rynjah, M. Joplin, and M. G. Kharjana. 2014. Stress Levels of College Students: Interrelationship between Stressors and Coping Strategies. *IOSR Journal Of Humanities And Social Science (IOSR-JHSS)* 19 (2014). Issue 8. www.iosrjournals.org

[57] C. Y. Park, N. Cha, S. Kang, A. Kim, A. H. Khandoker, L. Hadjileontiadis, A. Oh, Y. Jeong, and U. Lee. 2020. K-EmoCon: A multimodal sensor dataset for continuous emotion recognition in naturalistic conversations. *Scientific Data* 7, 1 (2020), 293. https://doi.org/10.1038/s41597-020-00630-y

[58] Limor Peer and Vicky Rampin. 2021. Reproducibility Principles: Taking the Pulse. https://reproducibility.acm.org/2021/04/08/reproducibility-principles-taking-the-pulse/ Accessed: 2024-03-25.

[59] Eugenia Politou, Efthimios Alepis, and Constantinos Patsakis. 2017. A survey on mobile affective computing. *Computer Science Review* 25 (2017), 79–100. https://doi.org/10.1016/j.cosrev.2017.07.002

[60] A. Pratap, D. C. Atkins, B. N. Renn, M. J. Tanana, S. D. Mooney, J. A. Anguera, and P. A. Areán. 2019. The accuracy of passive phone sensors in predicting daily mood. *Depression and Anxiety* 36, 1 (2019), 72–81. https://doi.org/10.1002/da.22822

[61] Edward Raff. 2019. A Step Toward Quantifying Independently Reproducible Machine Learning Research. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Curran Associates Inc., Red Hook, NY, USA, Article 492, 5485–5495. https://dl.acm.org/doi/10.5555/3454287.3454779

[62] E. L. Rosenberg and P. Ekman. 2005. *Coherence Between Expressive and Experiential Systems in Emotion*. Oxford University Press, 63–88. https://doi.org/10.1093/acprof:oso/9780195179644.003.0004

[63] Akane Sano and Rosalind W. Picard. 2013. Stress Recognition Using Wearable Sensors and Mobile Phones. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*. IEEE. https://doi.org/10.1109/ACII.2013.117

[64] Philip Schmidt, Attila Reiss, Robert Duerichen, Claus Marberger, and Kristof Van Laerhoven. 2018. Introducing WESAD, a Multimodal Dataset for Wearable Stress and Affect Detection. In *Proceedings of the 2018 on International Conference on Multimodal Interaction - ICMI '18*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3242969.3242985

[65] A. Shaw, N. Simsiri, I. Deznabi, M. Fiterau, and T. Rahman. 2019. Personalized Student Stress Prediction with Deep Multitask Network. In *Proceedings of the 1st Adaptive & Multitask Learning Workshop* (Long Beach, California).

[66] Thomas Stütz, Thomas Kowar, Michael Kager, Martin Tiefengrabner, Markus Stuppner, Jens Blechert, Frank H. Wilhelm, and Simon Ginzinger. 2015. Smartphone Based Stress Prediction. In *User Modeling, Adaptation and Personalization. UMAP 2015. Lecture Notes in Computer Science()*, Vol. 9146. Springer International Publishing, Cham, 240–251. https://doi.org/10.1007/978-3-319-20267-9_20

[67] S. Taylor, N. Jaques, E. Nosakhare, A. Sano, and R. Picard. 2020. Personalized Multitask Learning for Predicting Tomorrow's Mood, Stress, and Health. *IEEE Transactions on Affective Computing* 11, 2 (April-June 2020), 200–213. https://doi.org/10.1109/TAFFC.2017.2784832

[68] Ali Tazarv, Sina Labbaf, Amir Rahmani, Nikil Dutt, and Marco Levorato. 2023. Active Reinforcement Learning for Personalized Stress Monitoring in Everyday Settings. In *2023 IEEE/ACM Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*. Association for Computing Machinery, New York, NY, USA, 44–55. https://doi.org/10.1145/3580252.3586979

[69] A. Tazarv, S. Labbaf, S. M. Reich, N. Dutt, A. M. Rahmani, and M. Levorato. 2021. Personalized Stress Monitoring using Wearable Sensors in Everyday Settings. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 9630224.

[70] Kobiljon Toshnazarov, Uichin Lee, Byung Hyung Kim, Varun Mishra, Lismer Andres Caceres Najarro, and Youngtae Noh. 2024. SOSW: Stress Sensing With Off-the-Shelf Smartwatches in the Wild. *IEEE Internet of Things Journal* (2024), 1–1. https://doi.org/10.1109/JIOT.2024.3375299

[71] L.C. Towbes and L.H. Cohen. 1996. Chronic stress in the lives of college students: Scale development and prospective prediction of distress. *Journal of Youth and Adolescence* 25, 2 (1996), 199–217. https://doi.org/10.1007/BF01537344

[72] Fani Tsapeli and Mirco Musolesi. 2015. Investigating Causality in Human Behavior From Smartphone Sensor Data: A Quasi-Experimental Approach. *EPJ Data Science* 4, 1 (2015), 24. https://doi.org/10.1140/epjds/s13688-015-0061-1

[73] J. Vamsinath, B. Varshini, T. Sandeep, V. Meghana, and B. Latha. 2022. Stress detection using non-semantic speech representation. In *2022 32nd International Conference Radioelektronika (RADIOELEKTRONIKA)*. 1–6. https://doi.org/10.1109/RADIOELEKTRONIKA54537.2022.9764916

[74] J. Vamsinath, B. Varshini, T. Sandeep, V. Meghana, and B. Latha. 2023. A Survey on Stress Detection Through Speech Analysis Using Machine Learning. *International Journal of Scientific Research in Science and Technology* 9, 4 (2023), 326–333. https://doi.org/10.32628/IJSRST229436

[75] Gideon Vos, Kelly Trinh, Zoltan Sarnyai, and Mostafa Rahimi Azghadi. 2023. Generalizable Machine Learning for Stress Monitoring from Wearable Devices: A Systematic Literature Review. *International Journal of Medical Informatics* 173 (May 2023).

[76] R. Wang, M. S. H. Aung, S. Abdullah, R. Brian, A. T. Campbell, T. Choudhury, M. Hauser, J. Kane, M. Merrill, E. A. Scherer, V. W. S. Tseng, and D. Ben-Zeev. 2016. CrossCheck: Toward passive sensing and detection of mental health changes in people with schizophrenia. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. Association for Computing Machinery, New York, NY, USA, 886–897. https://ubicomplab.cs.washington.edu/pdfs/crosscheck.pdf

[77] R. Wang, F. Chen, Z. Chen, T. Li, G. Harari, S. Tignor, X. Zhou, D. Ben-Zeev, and A. T. Campbell. 2014. StudentLife: Assessing Mental Health, Academic Performance and Behavioral Trends of College Students using Smartphones. In *Proceedings of the ACM Conference on Ubiquitous Computing*. Association for Computing Machinery, New York, NY, USA, 3–14. https://studentlife.cs.dartmouth.edu/ubicomp2014.pdf

[78] Tong Xia, Jing Han, Abhirup Ghosh, and Cecilia Mascolo. 2023. Cross-Device Federated Learning for Mobile Health Diagnostics: A First Study on COVID-19 Detection. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 1–5. https://doi.org/10.1109/ICASSP49357.2023.10096427

[79] Xuhai Xu, Xin Liu, Han Zhang, Weichen Wang, Subigya Nepal, Yasaman Sefidgar, Woosuk Seo, Kevin S. Kuehn, Jeremy F. Huckins, Margaret E. Morris, Paula S. Nurius, Eve A. Riskin, Shwetak Patel, Tim Althoff, Andrew Campbell, Anind K. Dey, and Jennifer Mankoff. 2023. GLOBEM: Cross-Dataset Generalization of Longitudinal Human Behavior Modeling. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 4, Article 190 (jan 2023), 34 pages. https://doi.org/10.1145/3569485

[80] X. Xu, R. Wang, F. Chen, G. Harari, S. Tignor, X. Zhou, D. Ben-Zeev, and A. T. Campbell. 2022. GLOBEM Dataset: Multi-Year Datasets for Longitudinal Human Behavior Modeling Generalization. *arXiv preprint arXiv:2202.07051* (2022). https://arxiv.org/abs/2202.07051

[81] Kangning Yang, Benjamin Tag, Chaofan Wang, Yue Gu, Zhanna Sarsenbayeva, Tilman Dingler, Greg Wadley, and Jorge Goncalves. 2023. Survey on Emotion Sensing Using Mobile Devices. *IEEE Transactions on Affective Computing* 14, 4 (2023), 2678–2696. https://doi.org/10.1109/TAFFC.2022.3220484

[82] J. C. Yau, B. Girault, T. Feng, K. Mundnich, A. Nadarajan, B. M. Booth, E. Ferrara, K. Lerman, E. Hsieh, and S. Narayanan. 2022. TILES-2019: A longitudinal physiologic and behavioral data set of medical residents in an intensive care unit. *Scientific Data* 9 (2022). https://doi.org/10.1038/s41597-022-01234-6

[83] Xue Ying. 2019. An Overview of Overfitting and its Solutions. *Journal of Physics: Conference Series* 1168, 2 (feb 2019), 022022. https://doi.org/10.1088/1742-6596/1168/2/022022

[84] Han Yu and Akane Sano. 2023. Semi-Supervised Learning for Wearable-based Momentary Stress Detection in the Wild. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 2 (2023), Article 80. https://doi.org/10.1145/3596246

A Appendix

A.1 Data & Code

The K-Emophone dataset is accessible through Zenodo[1], and the new dataset can be found on the public GitHub page[2]. Code for personalization experiments (similar-user models and multi-task leaning) on both datasets

[1] https://zenodo.org/records/7606611
[2] https://bit.ly/3whIEb2

is hosted in an anonymized repository[3]. Additional experiment codes are divided between two anonymized repositories: one for the K-EmoPhone dataset[4] and another for the DeepStress dataset[5].

A.2 Figures & Tables

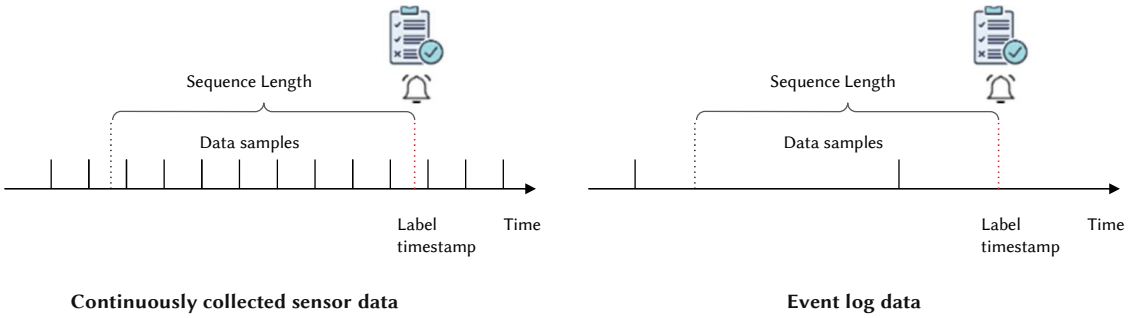


Fig. 9. Data Type Challenges for End-to-end Deep Learning

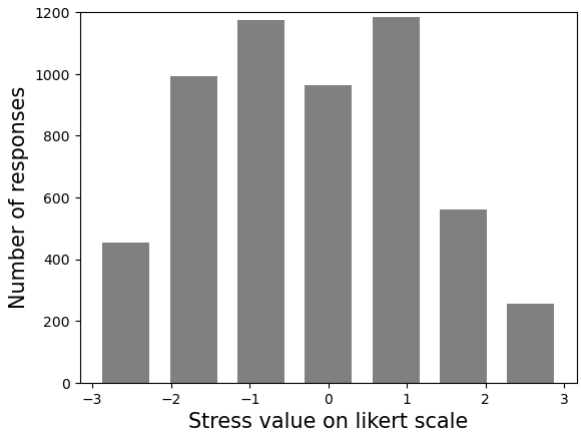


Fig. 10. K-EmoPhone Stress Label Distribution

[3] https://anonymous.4open.science/r/IndependentReproducibility_MTL-D1BC/
[4] https://anonymous.4open.science/r/IndependentReproducibility-E7BF/
[5] https://anonymous.4open.science/r/DeepStressReproducibility-D08F/

Table 13. Feature Extraction - Reasoning & Data Source

Type	Reasoning	Raw Data
Social interaction [2]	Change in social activities is one of the main stressors for college students	Call event, Message event
Physical activity [56]	Exercise is regarded as one of the positive coping strategies to relieve stress [56]	Accelerometer [54], Physical activity event, Steps, Distance, Calories
Context [51]	Contextual features have an effect on the users' perceived stress levels [51]	Location, Time Information, UltraViolet, Ambient light
Phone usage [16, 66] [38]	Phone usage including app usage patterns could be a good predictor of EMA stress level [16, 66]	App usage, Installed app, Screen event, OnOffEvent, Network connectivity, Battery event, Data traffic, WiFi events, Media events, System events
Physiological data [54] [53, 63]	Physiological data such as HRV and GSR are believed to be good predictors of stress levels [54, 63]	Skin temperature, RRInterval, Heart rate, EDA
Sleep [2]	Changes in sleep could be a good predictor of stress	Screen event

Table 14. Feature Extraction - Preprocessing

Type	Raw Data	Preprocessing [35]
Social interaction [2]	Call event	Filter negative and 0 duration
	Message event	Encode the categorical events to 1 as numeric values
Physical activity [56]	Accelerometer [56]	Calculate the magnitude of accelerometer data
	Steps	Calculate the step count between consecutive pedometer data recordings
	Distance	Calculate the distance between consecutive data recordings
	Calories	Calculate the calories consumed between consecutive data recordings
Context [51]	Location	Calculate haversine distance between consecutive GPS recordings; Cluster the GPS data for each user using poi clustering; Label clusters using semantic labels (home, work, google map API labels, and none) [72]
	UltraViolet	Calculate UV exposure between consecutive UV recordings
Phone usage [16, 66] [38]	App usage	Recategorize apps into predefined categories [66], calculate app usage duration for each app usage session
	Installed app	Calculate jaccard similarity index between consecutive installed app name
	Screen event	Calculate screen on duration for each screen on event session for each user
	WiFi events	Calculate cosine, euclidian, and manhattan distance between consecutive wifi rssi; calculate jaccard similarity index between consecutive wifi bssid
	Media events	Encode the categorical video, image, and all types events to 1 as numeric values
Physiological data [54] [53, 63]	Skin temperature	Remove outliers and conduct z-score normalization for each user
	RRInterval	Remove outliers and conduct z-score normalization for each user [53]
	Heart rate	Remove heart rate bigger than 220 bpm and smaller than 30 bpm; remove outlier using statistical method and normalization for each user [53]
	EDA	Calculate skin conductance from raw EDA data; apply median filter and min-max normalization for each user; decomposing EDA into phasic and tonic EDA [53]
Sleep [2]	Screen event	Filter out screen-on events caused by notifications (session shorter than 30 seconds); Calculate screen off duration for each screen off session for each user; Discard screen-off patterns that do not start between 9PM to 7AM (next day) [1]

Table 15. Feature Extraction - Extracted Features

Type	Raw Data	Information being aggregated into features	Features [35]
Social interaction [2]	Call event	Call duration and previous times contacted of the contact person	Numeric features, call frequency
	Message event	Message sent, received, and all events including both sent and received events.	Numeric features
Physical activity [56]	Accel. [56]	X, Y, Z values, and magnitude	Numeric features
	Physical activity transition events	ENTER_WALKING, ENTER_STILL, ENTER_IN_VEHICLE, ENTER_ON_BICYCLE, ENTER_RUNNING events	Categorical features
	Physical activity event	Confidence of unknown, OnFoot, Walking, InVehicle, OnBicycle, Running, and Titling	Numeric features
	Steps	Steps count	Numeric features
	Distance	Distance traveled, motions, speed and pace	Numeric features
	Calories	Calories consumed	Numeric features
Context [51]	Location	Distance traveled, location cluster, and location cluster semantic label	Numeric features for distance, categorical features for location cluster and cluster label, number of locations visited
	Time Information	Label timestamp	Day of week, weekend or not, hour name
	UltraViolet	UV exposure, intensity	Numeric features
	Ambient light	Ambient light brightness	Numeric features
Phone usage [16, 38, 66]	App usage	Different types of app usage events and their duration	Categorical features for app events and numeric features for usage duration
	Installed app	Jaccard similarity index between consecutive installed app name	Numeric features
	Screen event	Screen events and screen on duration	Categorical features for screen events while numeric features for screen_on duration
	OnOffEvent	Phone power on off events	Categorical features
	Network connectivity	Network connected events	Categorical features
	Battery event	Battery level, status, and temperature	Numeric features for battery level and temp. while categorical features for battery status
	Data traffic	Received and sent data in kbytes	Numeric features
	WiFi events	Three types of distances between consecutive rssi and jaccard similarity index between consecutive bssid	Numeric features
	Media events	Video, image, and all types of events	Numeric features
	System events	Ringer mode types, power save event types, and mobile charge event types	Categorical features
Physiological data [53, 54, 63]	Skin temperature	Skin temperature level	Numeric features
	RRInterval	Interval between the consecutive heart-beat	Mean, median, max, min, std, kurt, skw, slope, percentile_20/80, and rmssd
	Heart rate	Heart rate variability	Numeric features
	EDA	Skin conductance, phasic and tonic skin conductance	Mean, max, min, std, num_peaks, and AUC
Sleep [2]	Screen event	Screen off duration and corresponding start timestamp	Longest duration is sleep duration; its start time is sleep onset

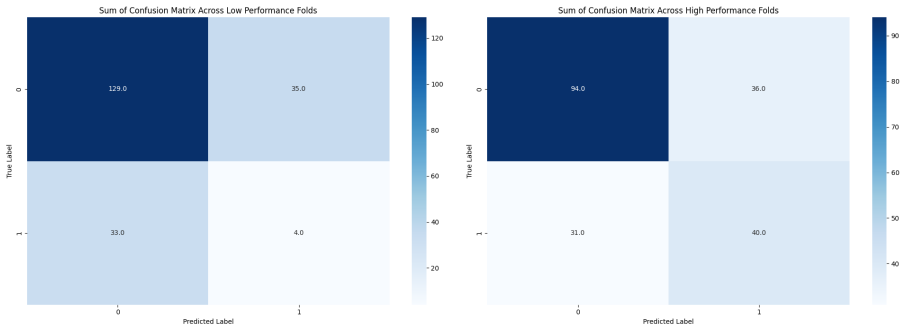


Fig. 11. Confusion Matrix for High and Low Performance Users

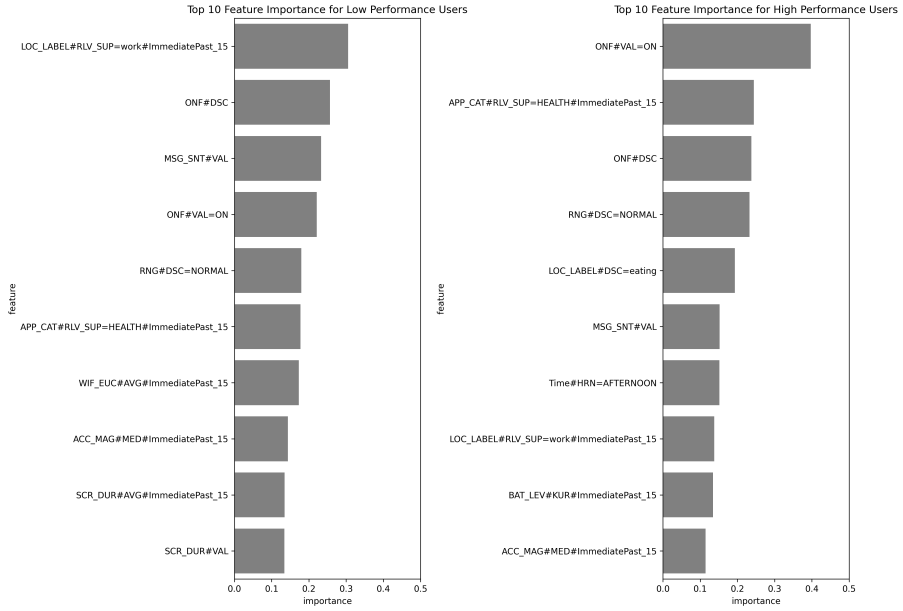


Fig. 12. Feature Importance for High and Low Performance Users

Table 16. Results for Baseline on New Dataset

Experiment Type	Accuracy	F1-Score (pos. label)	Macro F1-Score	AUC-ROC	Precision	Recall
Baseline 1	0.704	0.124	0.468	**0.522**	0.518	0.506
Baseline 2 (removing 1st day's data)	0.700	0.128	0.468	0.523	0.513	0.506
Baseline 3 (using 50% data for testing)	0.710	0.125	0.470	0.524	0.511	0.509

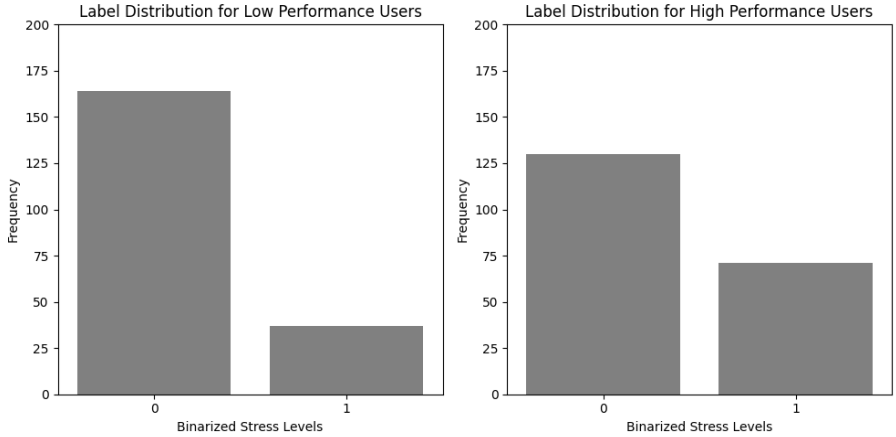


Fig. 13. Label Distribution for High and Low Performance Users

Table 17. Results for Preprocessing, Feature Extraction & Preparation on New Dataset

Experiment Type	Details	Accuracy	F1-Score (pos. label)	Macro F1-Score	AUC-ROC	Precision	Recall
G1: Label Processing	Removing Neutral States	0.649	**0.160**	0.461	0.511	0.518	0.507
	Removing Extreme Users	0.521	0.284	0.455	**0.554**	0.520	0.522
	Label binarization (mean 4 all users)	0.555	0.296	0.476	0.540	0.532	0.523
	Label binarization (mean 4 each user)	0.530	0.352	0.485	0.533	0.521	0.514
G2: Using Different Feature Types	Previous survey label data only	0.803	0.422	0.638	**0.640**	0.636	0.640
	Pre-experiment survey only	0.456	0.113	0.279	0.500	0.228	0.500
	Sensor + Pre-experiment survey only	0.668	0.146	0.460	0.554	0.515	0.530
	Sensor + Previous survey label data	0.752	0.254	0.546	**0.616**	0.590	0.568
G3: Using different time window sizes for immediate past time window	5 mins	0.707	0.139	0.476	0.525	0.515	0.515
	10 mins	0.699	0.135	0.472	0.527	0.517	0.518
	30 mins	0.699	0.131	0.470	0.530	0.515	0.515
	45 mins	0.690	0.125	0.463	0.527	0.506	0.503
G4: Using extended past features	Current + immediate past + sleep	NA	NA	NA	NA	NA	NA
	Current + immediate past + today epochs	0.710	0.132	0.474	0.535	0.513	0.513
	Current + immediate past + yesterday epochs	0.724	0.129	0.477	**0.542**	0.524	0.511
	Current + immediate past + today whole time window (aggregated over all epochs)	0.688	0.171	0.483	**0.544**	0.526	0.520
	Current + immediate past + yesterday whole time window (aggregated over all epochs)	0.710	0.141	0.479	0.527	0.525	0.509
G5: Feature Normalization	Standard normalization for each user	0.432	0.317	0.373	0.530	0.533	0.513

Table 18. Results for Feature Selection, Data Splitting, Oversampling & Undersampling, and Model Training on New Dataset

Experiment Type	Details	Accuracy	F1-Score (pos. label)	Macro F1-Score	AUC-ROC	Precision	Recall
G6: Feature selection	Remove 0 variance features + LASSO only	0.708 (0.126)	0.140 (0.086)	0.477 (0.052)	**0.539 (0.074)**	0.519 (0.037)	0.522 (0.044)
	Remove features with high pairwise correlation + LASSO only	0.695 (0.125)	0.134 (0.091)	0.469 (0.043)	0.535 (0.07)	0.512 (0.035)	0.506 (0.024)
	Remove 0 variance features + Remove features with high pairwise correlation + LASSO	0.697 (0.126)	0.135 (0.092)	0.470 (0.044)	0.536 (0.07)	0.514 (0.036)	0.508 (0.025)
G7&G8: Data splitting	Group k fold	0.704	0.140	0.480	0.503	0.489	0.494
	Time-series k fold	0.727	0.289	0.559	**0.636**	0.589	0.565
	k fold	0.803	0.471	0.675	**0.764**	0.713	0.657
	loso + random 50% test user	0.789	0.277	0.565	**0.636**	0.596	0.565
	loso + stratified 50% test user	0.782	0.256	0.553	**0.634**	0.573	0.556
	loso + first 50% test user	0.715	0.179	0.492	**0.573**	0.524	0.517
	loso + first 10% test user	0.696	0.158	0.480	0.552	0.527	0.528
	loso + first 30% test user	0.701	0.158	0.479	0.540	0.525	0.526
	loso + first 70% test user	0.737	0.155	0.511	0.539	0.551	0.523
	loso + first 90% test user	0.769	0.203	0.535	0.579	0.552	0.545
G9: Oversampling & Undersampling	Original distribution	0.741	0.062	0.451	0.529	0.481	0.501
	Random oversampling	0.680	0.168	0.479	0.533	0.533	0.522
	Random undersampling	0.533	0.223	0.433	0.507	0.505	0.493
G10: Using different levels of personalization	Multi-task learning	0.625	0.201	0.461	**0.483**	0.495	0.493
	Similar-user model	0.604	0.181	0.445	**0.505**	0.501	0.498
G10: Model training using different machine learning models	RandomForest	0.727	0.103	0.467	0.529	0.500	0.505
	SVM	0.689	0.154	0.475	**0.563**	0.527	0.528
	Logistic Regression	0.663	0.196	0.484	0.548	0.523	0.522
	KNN	0.458	0.297	0.416	0.523	0.513	0.519
	Decision Tree	0.597	0.231	0.474	0.503	0.508	0.503
	Naïve Bayes classifier	0.356	0.321	0.323	0.518	0.509	0.500
	MLP	0.650	0.194	0.479	0.546	0.517	0.519