Design of Contextual Filtered Features for Better Smartphone-User Receptivity Prediction

Jumabek Alikhanov, *Student Member, IEEE*, Panyu Zhang, *Student Member, IEEE*, Youngtae Noh, *Member, IEEE*, and Hakil Kim, *Member, IEEE*

***Abstract*—For the successful engagement of users with incoming interventions, the delivery of just-in-time (JIT) (or opportune moment–OM) interventions is of the utmost importance. This can be accomplished with a machine learning (ML) model that utilizes user context data. Smartwatch and smartphone interactions further open the space for further improvement by considering the user's state and environmental information. In this work, a novel feature engineering approach for predicting smartphone user receptivity to JIT interventions is proposed. Proposed approach utilizes rich information from user–smartphone interactions and smartwatch sensor data to extract contextually filtered features (CFFs). The superiority of the proposed feature engineering method is demonstrated by developing an ML model that predicts a participant's receptivity prior to administering an experience sampling method (ESM) questionnaire. The proposed approach is evaluated using the KEmoPhone data set, which contains 3334 ESM answers collected over the course of one week from 73 participants. To reduce the bias that may result in selecting specific classifier, multiple classifiers are tested and their average ROC-AUC metrics are compared. The results show that the proposed feature engineering method—CFFs improves the prediction performance, achieving an ROC-AUC of 56.5% as opposed to 54.7% obtained using conventional features (statistical moments of time series over an 80 min window). Such superior performance is consistent across multiple ML classification algorithms. Full research code will be released in Jupyter Notebooks for facilitating the research in the domain at https://github.com/Jumabek/receptivity upon the publication of this research work.**

***Index Terms*—Context-aware services, intelligent sensors, machine learning, ubiquitous computing.**

I. Introduction

MOBILE health (mHealth) and digital therapeutics (DTx) research have opened up unique potentials of smartphone-based digital health interventions in inducing favorable behavioral changes in response to diverse needs, such as smoking, alcohol disorders, physical inactivity, and depression [1], [2], [3]. The ultimate objective is to combine accurate sensing with appropriate DTx therapies to improve the quality of life for a specific population subgroup (e.g., increasing physical activity in sedentary people). Although, there is large body of research [4], [5], [6], [7] that is intended to examine and measure the (causal) effect of health interventions, this is not the topic of this study.

To properly engage consumers, it is essential to accurately predict the delivery timing of DTx interventions [8], [9]. This necessitates that an intervention design should incorporate a component for identifying opportune moments (OMs) for a specific intervention. However, the architecture of the OM detector component can differ according to the type of DTx. The opportune time (i.e., OM) for stress-relieving DTx could occur when the person is stressed. Similarly, for DTx that aims to increase physical activity, an OM might be described as the time when a user is able or feels compelled to carry out the proposed interventions (e.g., taking a walk to a hall). This can depict the moment after the user has completed a task or has been inactive for an extended period of time.

Significant research has been undertaken in the realm of ubiquitous computing for the modeling of a similar phenomenon: user interruptibility. Inference based on context sensing and machine learning (ML) has been proposed in a number of research [10], [11]. In general, interruptibility quantifies how opportune it is to interrupt a person [12]. Similarly, receptivity to health interventions, as defined by Nahum-Shani et al. [13], is a person's ability to receive, process, and utilize the support (intervention) provided. One approach to conceptualize receptivity involves encompassing the combination of interruptibility (willingness to receive an intervention), engagement (receive the intervention), and a person's subjective perception of the intervention provided (process and use the intervention) [14]. Receptivity can also be defined in multiple stages as proposed by Choi et al. [15]. Authors defined receptivity as stages which define if user has perceived the interruption, if he has perceived then next stage is if user is available for the intervention, finally if user is available then his receptivity can be gauged based on whether he decides to adhere to health intervention.

Fig. 1 depicts an overview of the receptivity prediction pipeline in this work, from data collection to receptivity prediction prior to intervention. Note that creating a forecast

Manuscript received 26 January 2023; revised 10 August 2023; accepted 3 November 2023. Date of publication 13 November 2023; date of current version 26 March 2024. This work was supported in part by the BK21 Four Program under Grant NRF-2021M3A9E4080780 funded by the National Research Foundation (NRF) of the Ministry of Education (MOE), South Korea; in part by the Ministry of Trade, Industry, and Energy (MOTIE) under Grant RS-2023-00236325; and in part by the Korea Institute of Energy Technology under Grant KRG2022-01-011. *(Corresponding authors: Youngtae Noh; Hakil Kim.)*

Jumabek Alikhanov and Hakil Kim are with the Department of Electrical and Computer Engineering, Inha University, Incheon 402-751, South Korea (e-mail: jumabek4044@gmail.com; hikim@inha.ac.kr).

Panyu Zhang is with the Department of Industrial and Systems Engineering, Korea Advanced Institute of Science and Technology, Daejeon 305-701, South Korea (e-mail: panyu@kaist.ac.kr).

Youngtae Noh is with the Department of Energy AI, KENTECH University, Naju 402-751, Republic of Korea (e-mail: ytnoh@kentech.ac.kr).

Digital Object Identifier 10.1109/JIOT.2023.3331715

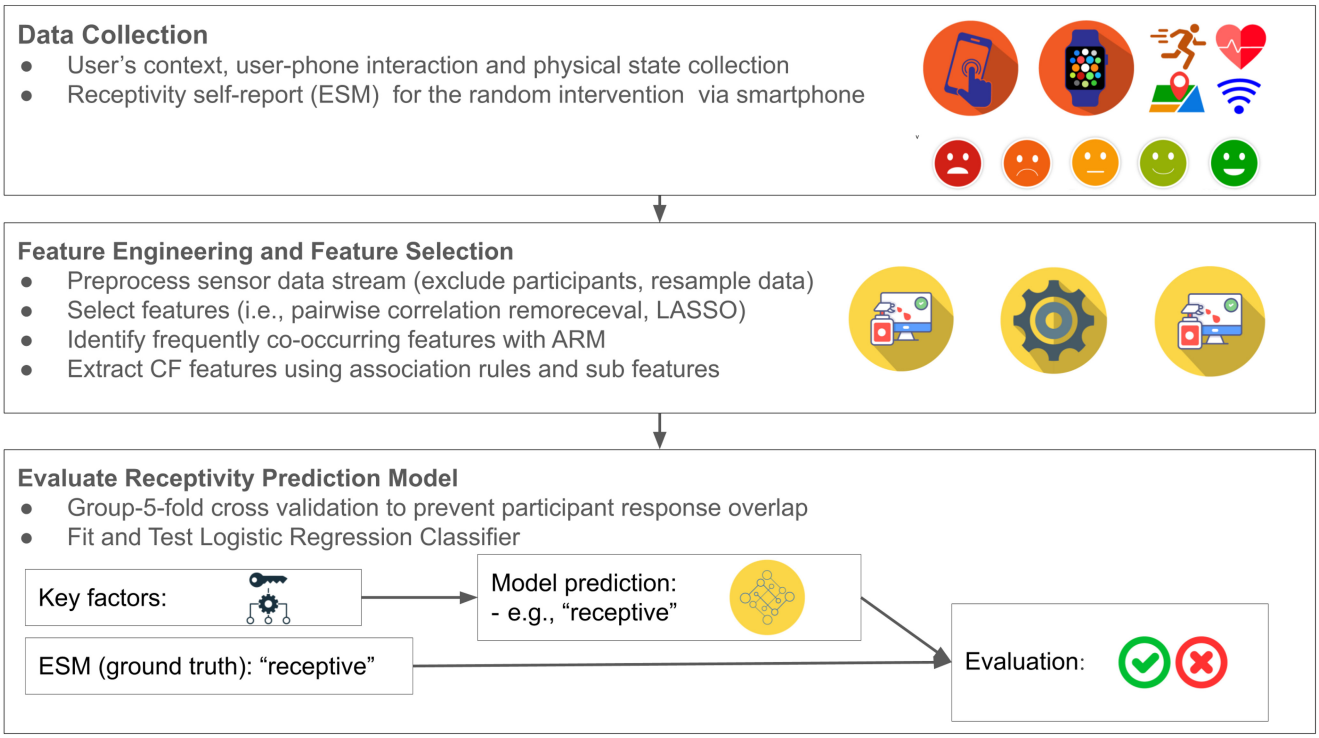

Fig. 1. Receptivity prediction pipeline.

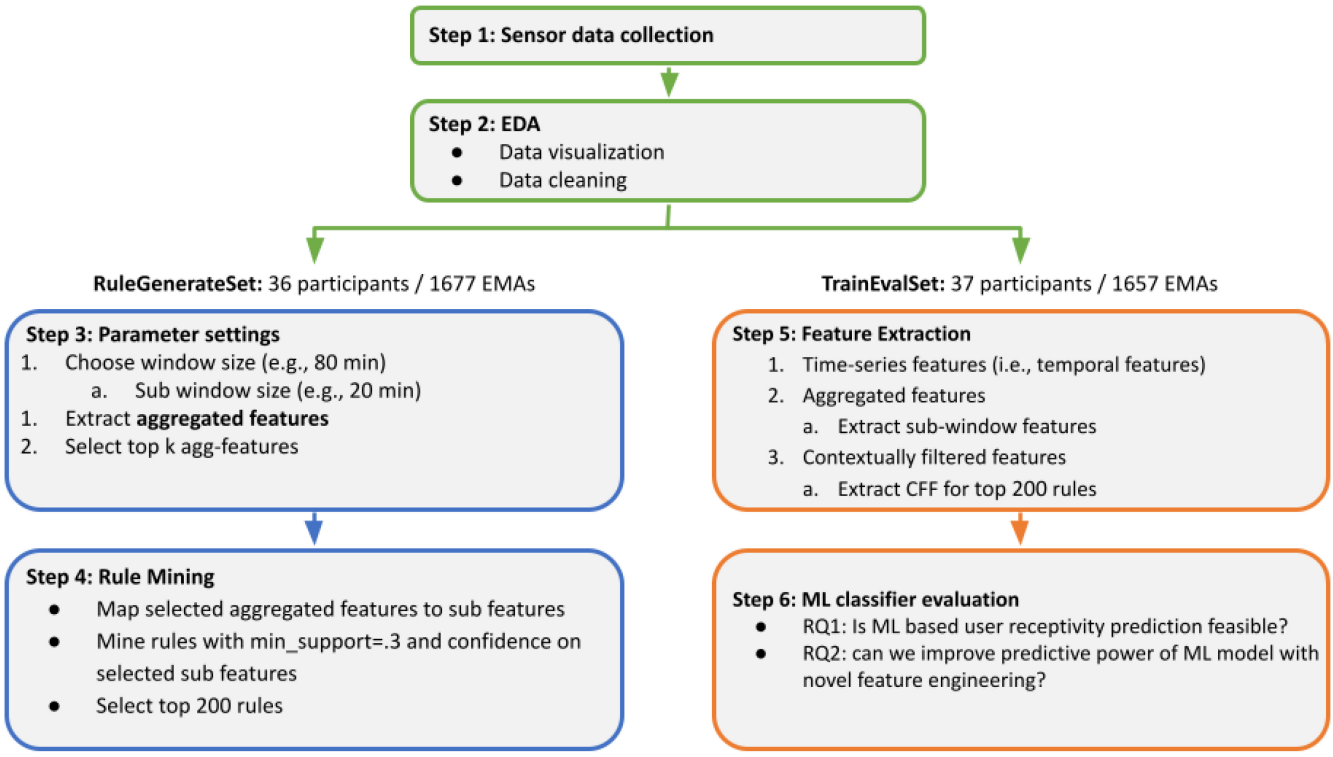

Fig. 2. Research overview.

based on previously collected data allowed us to evaluate how well a user's receptivity state might be anticipated using the proposed ML-based receptivity prediction pipeline, which resulted in improved engagement with the DTx intervention

An overview of this research can be seen in Fig. 2. First, collected sensor data from the K-EmoPhone data set is prepared. Then exploratory data analysis (EDA) is performed to understand and cleanse data. Then the data set is split into two sets: 1) for mining association rules and 2) for extracting contextually filtered features (CFFs). Research findings of this study contribute to the comprehension of the following research questions.

TABLE I
OVERVIEW OF RELEVANT STUDIES IN THE LITERATURE ON INTERRUPTIBILITY AND RECEPTIVITY

Authors	Year	n: # participants m: # responses	Ground-truth/binning	Classification results		Random Baseline	
				Metric	Performance	Strategy	Performance
Fogarty et al [16]	2004	n = 10 m = 975	Liker five scale/highly non-interruptible vs. rest	Accuracy	79.5	Most frequent	70.1
Poppinga et al [17]	2014	n = 79 m = 6,581	Whether a user will answer the questionaire/binary	Accuracy	77.85	Most frequent	77.08
Pevojic et al [10]	2014	n = 20 m = 2,334	Reaction presence/binary Reaction sentiment	Precision/ Recall	64 / 41 46 / 10	Stratified	39 / 38 27 / 26
Choy et al [18]	2016	n = 25 m = 4,103	Binary self-report on interruptibility/binary	Accuracy	92	*NA*	∼50
Pielot et al [11]	2017	n = 337 m = 78,930	Whether users are engaged with the recommended content/binary	F1 score	11.2	Most frequent	8.2
Sarker et al [19]	2014	n = 30 m = 360	Response delay/ two extremes slow and quick answers	Accuracy	74.7	Most frequent	50
Kunzler et al [14]	2019	n = 189 m = 13,942	Whether a user responds to the chatbot within 10 min/binary	F1 score	30	Stratified	20
Mishra et al [20]	2021	n = 83 m = NA	Whether a user responds to the chatbot within 10 min/binary	F1 score	36	Most frequent	25
The proposed app.	2023	n = 73 m = 3334	Likert-seven scale disturbance/ OM (-3, -2, -1, 0) vs. non-OM (+1, +2, +3)	F1 score	56	Most frequent/ Stratified	38/48

1) *RQ-1:* Can ML techniques be applied effectively to enhance users' receptivity to health interventions? To answer this, we use the KEmoPhone data set and evaluate whether ML plus user context data can produce more accurate predictions than a random baseline.
2) *RQ-2:* Is it possible to enhance ML-based receptivity prediction models by feature engineering alone? Here, the effectiveness of the proposed feature engineering method is evaluated.

Our contributions can be summarized as follows.

1) *Novel Feature Engineering Approach:* a novel feature engineering method called CFFs is proposed for predicting user receptivity to just-in-time (JIT) interventions. This approach leverages rich information from user–smartphone interactions and smartwatch sensor data to extract features that capture contextual information relevant to receptivity.
2) *Context Mining for Feature Extraction:* The process of context mining is incorporated into the proposed feature engineering methodology. By mining association rules from subwindow features, potentially valuable contexts for feature extraction is identified, allowing us to extract more meaningful features for predicting user receptivity.
3) *Evaluation and Comparison:* The performance of the proposed method is evaluated using the KEmoPhone data set, which contains a large number of experience sampling method (ESM) answers collected from participants over a one-week period. The performance of the proposed approach is compared against to conventional feature extraction methods and demonstrates its superiority in predicting user receptivity to JIT interventions.

II. RELATED WORK

In the realm of ubiquitous computing, the topic of interruptibility has been intensively investigated. Table I provides a summary of past research on interruptibility and receptivity to health intervention. The offered overview comprises the purpose of a certain research project, binning criteria for outcome variables, classification results together with their corresponding metrics, and a random baseline. Researchers utilized a variety of evaluation variables, including accuracy, precision/recall, and the F1 score. The table also includes the evaluation metric scores of the proposed approach for comparison. It is noteworthy to mention that, although the proposed approach obtains a higher F1 score (56%) compared to other studies (11.2%, 30%, and 36%), direct comparison is hard as the data, ground-truth, and its binning strategy differs for each study. In addition, the baseline procedures used to evaluate cases where there is no intelligent model are explained.

One of the earliest studies on user interruptibility detection was conducted by Fogarty et al. [16]. The authors used external sensors in an office environment to predict the interruptibility of an office worker. The goal was to develop an application that helped incoming visitors identify if an office worker was interruptible. Poppinga et al. [17] developed a decision tree-based model to predict if it is the OM for interrupting mobile users with notifications. Using contexts, such as the time of day, location, and posture of the device, they obtained an accuracy of 77.85%, whereas a random model that only predicted the majority label (non-OM) produced an accuracy of 77.08%.

Pejovic and Musolesi [10] studied the identification and utilization of OMs. They demonstrated that interruptions

delivered via the proposed intelligent InterruptMe library resulted in better user satisfaction and shorter response times. The authors attempted to gauge OMs from different perspectives, such as reaction presence (if the user did not overlook the notification) and sentiment (how obtrusive the interruption was from the user's perspective). The following were used as a data source for feature engineering: time, accelerometer, location, social setting, physical activity, and the subject's emotions (obtained through an ESM).

Choy et al. [18] explored the effect of the window size from which features are extracted on the performance of an ML model for predicting interruptibility (i.e., interrupting a smartphone user). They discovered that instead of simply looking at the immediate past (15 min) or current features, looking back at the entire day produced features with richer representations and achieved 91% accuracy on the KAIST data set when using the Naive Bayes classification algorithm.

Instead of merely predicting user interruptibility in response to notifications, Pielot et al. [11] attempted to model the engagement of a smartphone user with the suggested content (notification). In other words, a user is considered engaged if he or she not only opens the notification but engages with it (e.g., clicks the links and plays the suggested game). The data sources used for communication activities were as follows: context (e.g., being at home), demographics, phone status (e.g., phone unlocked), and use patterns (e.g., watching a game or surfing the Internet). The XGBoost classifiers built on engineered features achieved a recall (conversion) of 7.1%. Although this performance metric appears small, it is 66.6% better than the baseline recall (conversion rate) of 4.3%.

Compared with prior studies that predicted the interruptibility of users for smartphone notifications, OM prediction for health intervention is slightly different since the target health activity triggered by the health intervention is known and can be measured. Sarker et al. [19] predicted OMs for JIT health interventions for smoking cessation study. In addition to the available sensor features, such as location and stress, the authors used the ESM responses of a user as inputs to the ML model. The ML model demonstrated promising performance as opposed to random baseline results in terms of accuracy (74.7% versus 50%). The authors defined a moment to be available (opportune) for JIT interventions if the user answered with a small response delay, whereas a nonavailable moment would be implied by a larger response delay. In a similar study, to gauge receptivity, Kunzler et al. defined several metrics, such as the JIT response health chatbot, response delay, and engagement in conversation. The primary metric was the JIT response, which is a binary metric that defines whether a user responds to an incoming chatbot's—Ally's message within 10 min. The authors additionally investigated the association between different factors and the defined receptivity metrics. These factors included intrinsic characteristics (age, gender, and personality) and contextual factors (day/time, phone battery, phone interaction, and physical activity). With contextual factors as features, the ML model achieved an F1 score of 40% for Android users, whereas the random baseline score was 27%. A follow-up study conducted by Mishra et al. [20] deployed the chatbot Ally+ in a natural environment where interventions are sent to users only if the ML model identified the moment as OM. Results revealed that the ML model led to a 40% improvement in receptivity compared with the control (baseline) model. Their static support vector machine (SVM) model was trained before deployment, and it achieved an average F1 score of 36%, a score comparable to the 25% F1 score in the baseline scenario. A significant difference between the study and those reported in the literature lies in the fact that this research focused on improving the model's performance with novel feature engineering, namely, CFFs. In other words, this research is not proposing another ML classifier or algorithm. Instead, a novel way to engineer more richer feature representation which can improve the performance of any classification methods is proposed. In this regard, the model is also constrained to use only sensor and user–phone interaction features to investigate the effect of novel feature engineering. In other words, features, such as age, gender, personality scores, time of the day, weekend, or weekday, were not considered. With the inclusion of these features, further improvement in performance is expected as explored in [21]. However, this is beyond the scope of this study.

III. Research Design

A. Receptivity to Just-In-Time Health Interventions

In this study, we used the K-EmoPhone data set to explore aforementioned research questions. Note that this data set contains a questionnaire gauging a subject's emotion and stress level on a seven-point Likert scale. There are total of seven questions (Q1: valence; Q2: arousal; Q3: attention level; Q4: stress level; Q5: emotion duration; Q6: task disturbance level; and Q7: emotion change). Additional information on the seven-item questionnaire, such as the exact text and range of answers, is presented in Table II.

We can assume that the cognitive load arising from answering a seven-item ESM questionnaire is approximately equal to a JIT interventions (e.g., taking a short walk or taking a deep breath). In other words, receptivity to ESM intervention can be used as a proxy for receptivity to JIT interventions. Using ESM questionnaires as subjective ground truth on interruptibility was done in prior research [16]. Subsequently, as for subjective (e.g., self-reported) receptivity measurement, we used one of the ESM questionnaires, Q6: task disturbance level. The level of disturbance tells how opportune the moment of intervention was. In other words, the disturbance level shows the momentary receptivity when the user responds to the ESM questionnaire. This is because the ground-truth ESM answers are designed to capture a user's state with respect to the response time.

B. Data Collection

1) Collection Procedure: The data in the K-EmoPhone [22] data set were collected from 79 participants who used Android phones over the course of one week. The participants were educated through a session explaining details pertaining to the data collection and equipment. Furthermore, they provided institutional review board-signed consent forms to participate

TABLE II
K-EMOPHONE ESM QUESTIONNAIRE CONTENT

Item	Question	Range	Measurement
1	My emotion right before taking this survey was	very negative (-3)–very positive (+3)	Valence
2		very calm (-3)–very excited (+3)	Arousal
3	My attention level right before taking this survey could be rated as	very bored (-3)–very engaged (+3)	Attention level
4	My stress level right before taking this survey was	not stressed at all (-3)–very stressed (+3)	Stress level
5	My emotion regarding answering the above has not changed in the recent ___ min.	5, 10, 15, 20, 30, 60 min, or "I am not sure"	Emotion duration
6	Answering this survey disturbed my current activity	entirely disagree (-3)–entirely agree (+3)	Task disturbance level
7	Change in your emotions while answering this survey?	I felt more negative (-3)–I felt more positive (+3)	Emotion change

in the data collection. Each participant was provided with an MS Band 2 wearable device for the collection of biosignals, which are widely used in emotion- and stress-related research. Such biosignals include the electrocardiogram, galvanic skin response, plethysmography, and human skin temperature. Smartphones were used to collect usage data, such as location tracking, physical activities, and how people use smartphone apps. The participants were asked to answer the questionnaires every 45 min during their working hours from 10 A.M. to 10 P.M. The questionnaire included seven items, as shown in the following table.

2) Smartphone and Smartwatch Collected Sensor Data: In the data collection process, both Android smartphones and MS Band 2 smartwatches were used. Implementation of a special-purpose software on Android phones enabled the collection of sensor data reflecting various parameters like mobility, network traffic, social interactions, application usage, and device state. The MS Band 2 smartwatch data was collected in sync with ESM schedules, owing to battery limitations. The data collection software used three sampling methods—periodic, adaptive, and event-based, for the sensor data. The periodic method utilized a preset sampling rate, while the adaptive method adjusted the sampling rate depending on device OS policies. The event-based approach recorded sensor readings only when changes were detected. Through this, sensor data from smartphones and smartwatches are stored locally and uploaded to the database server every hour.

Smartphone data included the network connection history, network data usage, call and text message histories, application usage, installed applications, ringer mode changes, power-saving mode changes, charging state, media creation, battery status, physical activity, location data, WiFi data, and more. Each data point was collected with various methods, such as event-based or adaptive sampling. Smartwatch data included information like the wearer's acceleration, number of steps taken, distance walked, ambient brightness, exposure to ultraviolet radiation, heart rate, skin temperature, caloric burn rate, electrodermal activity, etc. The data was collected at various rates depending on specific measures. The data collection was thorough, capturing various aspects of user behavior and physical states, which provides a comprehensive overview of user receptivity.

TABLE III
CLEANING GROUND TRUTH (ESM ANSWERS)

Step Description	#participants	#responses
Original data	79	3715
Exclude ESMs outside campaign period	79	3592
Exclude ESMs without sensor data	77	3543
Exclude ESMs with currupted ts	77	3537
After excluding duplicate data	77	3426
After excluding participants with zero variance in their responses	74	3401
After excluding participants with most sensor sources missing	73	3334

IV. EXPLORATORY DATA ANALYSIS

We conduct EDA to understand, interpret, correct, and clean the data where necessary. Additionally, the correlation between response delay and self-reported, receptivity, (et al. disturbance) is investigated.

3) Cleaning: Unreliable and/or noisy data were filtered stepwise using the criteria listed in Table III. The table indicates that after the cleaning process, 73 participants were retained, who collectively provided 3334 ESM responses. Some ESMs had a corrupted (invalid) timestamp, and some had duplicates. Additionally, we found out that the data collected from four participants was unsuitable for use, as participants had zero variance in their responses all the time and some had no associated sensor data.

4) Visualization: Notably, during data collection, users sometimes did not respond to the ESM questionnaires. The number of missing answers varied across the participants. Fig. 8 visualizes the collected self-reports for three different campaigns. Each campaign corresponds to different participant groups where data is collected on different dates. Each row in the subfigure corresponds to one participant, and each dot corresponds to the answer on the ESM questionnaire.

5) Binning Outcome Variable: The distribution of self-reported user disturbances is illustrated in Fig. 3. As explained in Table II, the responses varied between -3 (least disturbed, i.e., not receptive at all) and $+3$ (most disturbed, i.e., most receptive). The figure indicates that the most common response

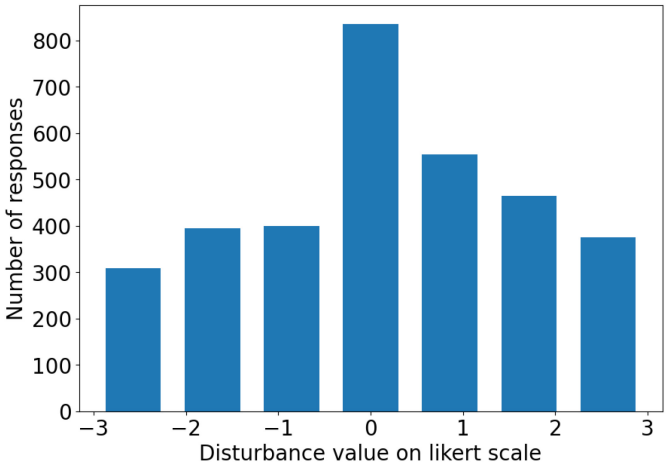

Fig. 3. Distribution of self reported disturbance in Likert-seven scale.

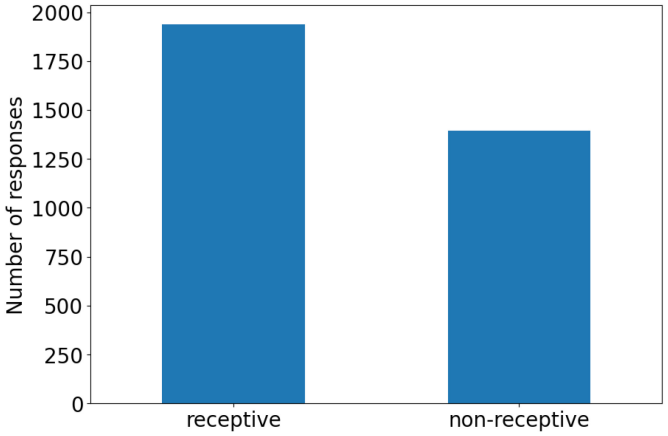

Fig. 4. Distribution of receptivity (binned disturbance).

was zero (neutral). Note that for the application of JIT interventions, it would be logical to consider neutral cases as receptive. Therefore, we considered the disturbance reports of −3, −2, −1, and 0 as receptive cases and the remaining as nonreceptive. The distribution after binning the outcome variable is presented in Fig. 4. It is noteworthy that the distribution of binary receptivity can be changed by modifying the binning threshold. For instance, the neutral disturbance level (zero Likert scale in Fig. 4) can be considered both receptive and nonreceptive.

6) Correlation Between the Response Delay and Perceived Disturbance: In previous studies, in the absence of self-reported disturbances, a response delay was used as an objective measure of user receptivity. Sarker et al. [19] defined availability as the state of an individual wherein they were capable of engaging in an incoming unplanned activity. Subsequently, the inferred moment is receptive if the response delay was small enough (e.g., 10 min). Künzler et al. [14] defined receptivity as the state of an individual wherein they responded to an incoming JIT health intervention within 10 min. Note that KEmoPhone includes both the response delay and self-reported receptivity (i.e., disturbance); therefore, existence of a relationship between them could be visualized and examined. Fig. 9 presents a visualization of

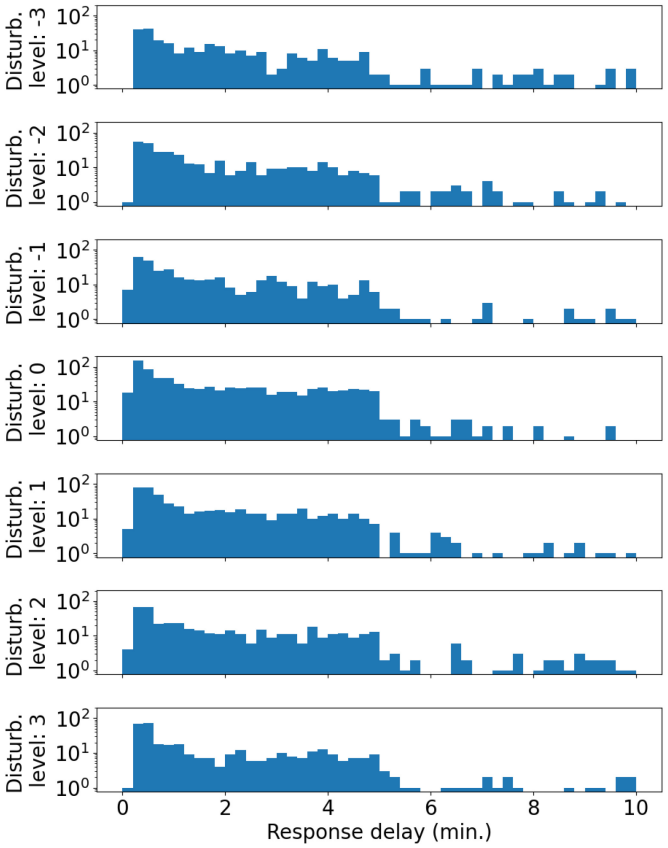

Fig. 5. Correlation between the response delay and self-reported disturbance on log scale.

this correlation, wherein each row represents one disturbance level on the Likert-seven scale. The observations imply that the relationship between the response delay and the self-reported disturbance is extremely weak, with a Pearson correlation coefficient of −0.03. Perhaps relationship is not a linear but a more complex function. Therefore, we plotted the disturbance in log scale in Fig. 5.

V. Methodology

Notably, the key factor for the successful prediction of user receptivity to incoming health interventions is the engineering of good features. Feature engineering is important yet mostly overlooked process for improving the performance of ML-based models [23]. This study developed a novel feature engineering method that exploits the co-occurrence of sensor values to extract good features. This was achieved using the concept of CFFs first introduced by Xu et al. [24] for depression detection. In this section, first, a brief overview of the main components, such as the different kinds of features and underlying concepts, are explained. Next, detailed explanation of the proposed methodology for feature extraction is provided. First, three types of feature-extraction approaches are explained in detail. Subsequently, the rationale behind window size selection is explained, followed by the proposed feature-selection pipeline and proposed context mining procedure.

A high-level explanation of two conventional feature engineering approaches and CFFs is depicted in Fig. 6.

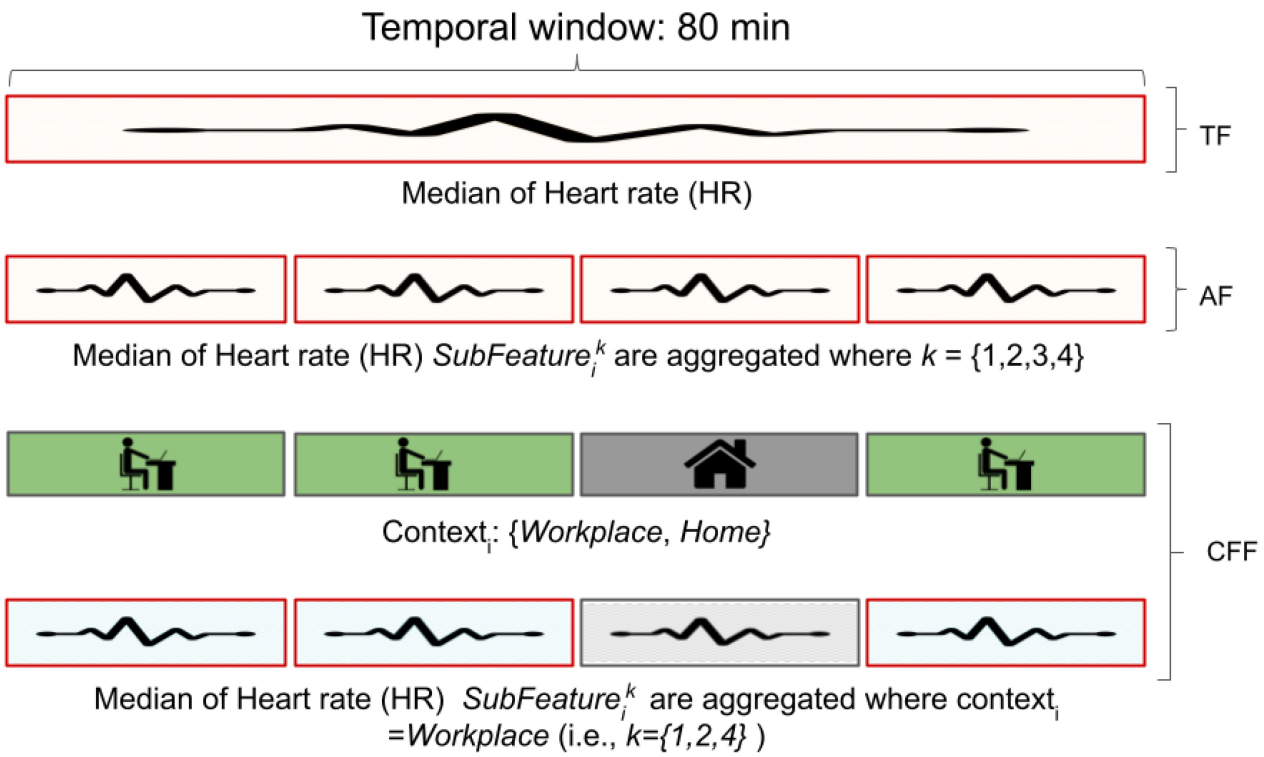

Fig. 6. Types of features. The first and second rows depict conventional TFs and AFs which was recently popularized. Note that an AF is computed by aggregating (i.e., averaging and computing the standard deviation) its subfeatures (can be also considered as sub TFs). CFFs are similar to AFs, except that they do not consider all the subwindow features during aggregation. The parts to be aggregated are decided based on whether the time window meets predetermined context criteria for extracting the given feature. Here, the context for heart rate variability is the user being in the workplace.

Conventionally features are extracted from time series (e.g., the temperature of the phone battery) over a given time window (e.g., 80 min) using statistical moments, such as the mean, standard deviation, and kurtosis. Here, we refer to such features as temporal features (TFs) (first row in the figure) which is the most common feature-extraction approach. For a depression detection studies, another commonly used feature engineering approach is called aggregated features (AFs) [24], [25], wherein a feature was computed for each day (or epoch of the day), and such features were aggregated across all days (or epochs). AFs are obtained from (sub) TFs. In other words, aggregation computes the mean and standard deviation of the TFs. Unlike depression studies, the time windows in this study are much shorter as user receptivity can change with higher frequency than an individual's depressive state. Therefore, instead of epochs, we introduce the concept of subwindows (20 min) as shown in the second row of the figure. Finally, the third and fourth rows in the figure depict the manner in which we obtain CFFs. Intuitively, CFF attempts to mine more specific features where values of different sensors co-occur together. In other words, such co-occurrence windows may provide valuable information that can more accurately describe outcome variables.

A. Data Split

In order to extract CFFs, a context (i.e., co-occuring sensor values) should be mined via the association rule mining (ARM) algorithm. However, if the data set on which rules are mined is the same as the data set on which the ML model is evaluated, the evaluation is said to be biased. This is because the results become optimistically biased due to overfitted models. To prevent bias in results, the data set was divided into RuleGenerateSet (1677 interventions from 36 participants) and TrainEvalSet (1657 interventions from 37 participants). This made sure that the rules found in RuleGenerateSet did not depend on TrainEvalSet, which was used to train and evaluate the performance of the ML models and the features they used.

B. Temporal Window Features

TFs are statistical moments extracted from time-series data. Temporal window features included the median, min, max, entropy, variance, skewness, kurtosis, the absolute sum of changes, lag features, linear trend, and time-series complexity features. They were extracted from an 80-min temporal window, which corresponded to the time right before user intervention. Therefore, we refer to them as TFs. In Fig. 6, the median heart rate is a TF, from which the heart rate is extracted. We have extracted a range of TFs from each timeseries sensor data stream, including information related to the median, statistical extremities (minimum and maximum), binned entropy, and variance of the data set, among other features. These features are computed for each designated time window within the data set. A detailed list of these TFs, along with their respective descriptions, is provided in Table IV. Crucially, the research design does not make any prior assumptions about which sensor data or TFs might prove beneficial

TABLE IV
LIST OF EXTRACTED TFS AND THEIR DESCRIPTIONS

Feature	Description
MED	Median of the data points in the given time window
MIN	Minimum value within the data points
MAX	Maximum value within the data points
BEP	Binned Entropy of the data points computed over 10 bins
AVG	Mean (average) of the data points
VAR	Variance of the data points with degrees of freedom=1
SKW	Skewness of the data points (biased=False)
KUR	Kurtosis of the data points (biased=False)
ASC	Absolute sum of the differences between consecutive data points
MAXLAG	Lag at maximum autocorrelation of data points
MAXLAGVAL	Maximum value of Autocorrelation of data points
MINLAG	Lag at minimum autocorrelation of data points
MINLAGVAL	Minimum value of Autocorrelation of data points
LTS	Slope of the Linear Regression of data points
LTI	Intercept Of the Linear Regression of data points
CID	Time-series complexity measure accounting for the variability in amplitude and frequency over time. Calcualted as the root mean square of power differences between normalized data points

for predicting user receptivity to JIT interventions. Rather, approach used in this work is to systematically consider all available sensor data streams and extract a comprehensive set of TFs from each. To determine which of these features might be most informative for predicting user receptivity, we employ a feature selection pipeline, as described in a later section. This method allows us to leverage the greatest amount of information present in the data set, whilst also tailoring feature selection to the unique characteristics of each data set.

C. Subwindows and Aggregated Features

For a depression detection studied by Xu et al. [24], AFs were used, wherein a feature was computed for each day (or epoch of the day), and such features were then aggregated across all days (and epochs) with the mean and standard deviation. Unlike the foregoing study, the outcome variable considered in this study was user receptivity, which can change within a day. As will be explained later in this section, we extracted the AFs from a considerably smaller window. Particularly, a window size of 80 min with four subwindows was selected empirically. The first four subwindow features (subfeatures), each 20-min long, were extracted for each ESM. In total, this covered 80 min of the entire temporal window before intervention. Second, to obtain the AFs, the mean and standard deviation of each of the four subwindows were computed. Thus, the purpose of the subfeatures was to compute the AFs. Subsequently, the subfeatures were also used to compute the CFFs. The subfeatures had 389 dimensions. This resulted in 778-D AFs (because each subfeature was aggregated by the mean and standard deviation).

D. Contextually Filtered Features

The first row in Fig. 6 indicates the time-series features extracted over a given time window. The second row in the figure depicts the AFs. Here, the AFs attempt to model the trends followed by the features by dividing the (entire) temporal window into multiple subwindows and aggregating the subwindow features with the mean and standard deviation. A CFF is a sophisticated version of an AF, wherein only specified (i.e., qualified) subwindows participate in aggregation. In other words, not all subwindows (there are a total of four subwindows, as shown in the figure) are used during aggregation. Instead, only subwindows with qualified contexts are used. The context in the figure is the semantic location of the user's workplace. Therefore, the median heart rate feature was computed only when the participant had a certain context (location = "workplace," as shown in the figure). We determined the candidate context for computing a given feature using co-occurrence relationships. For instance, if the value of "feature A" was always low when the value of "feature B" was high, this relationship could be exploited during feature engineering. Hence, when extracting through the aggregation of four subwindow features for feature_A, we only consider the subwindows that had a high value for "feature B." We termed the process of identifying such relationships "context mining," and this will be described in the Context Mining section.

The core idea of the proposed feature engineering approach involves leveraging sensor feature co-occurrence. Therefore, unlike Xu et al. [24], we did not use label (outcome variable) information to mine the rules. Instead, rules were mined solely using sensor and user–phone interaction data. The authors exploited label information (outcome variable, /ie, depressed versus nondepressed) also for rule selection. In other words, they selected the rules that distinguished the two categories (depressed versus not depressed). Rule selection metric is much simpler and allows for easy replication of the experiments. In addition, the design for CFF extraction allows to increase the number of data points used for the rule-mining process. For instance, if we were to exploit labels, we would be constrained to using only 1677 transactions, which correspond to the ESM responses collected from 36 participants (i.e., RuleGenerateSplit). The advantage of using the proposed approach is that it allows us to increase the number of data points by exploiting all windows for which sensor and phone–user interaction data are collected. Because data were collected for seven days between 10 A.M. and 10 P.M., we obtained $numtransactions = 36 * 7 * 12 * 60/20 = 9072$ transactions. Here, 36 is the number of people who are taking part, and 20 is the number of minutes for the subwindow.

TABLE V
PERCENTAGE OF MISSING FEATURES IF GIVEN WINDOW SIZE AND NUMBER OF SUBWINDOW COMBINATIONS WERE TO BE CHOSEN

number of subwindows	window size	percent of missing features more than 20% of the time
2	40	13.60
2	80	10.57
2	160	8.99
4	40	16.17
4	80	14.39
4	160	11.84
8	40	18.49
8	80	16.81
8	160	15.97

TABLE VI
PERFORMANCE OF LOGISTIC REGRESSION CLASSIFIER ON COMBINATION OF WINDOW SIZES SETTINGS. AFS ARE USED

Num. of sub-windows	Full window size	Accuracy	Balanced accuracy	F1 score	ROC-AUC
2	40	64.1	62.6	0.621	0.663
2	80	64.6	62.3	0.621	0.670
2	160	63.6	61.2	0.608	0.660
4	40	61.2	59.5	0.589	0.634
4	80	63.9	62.9	0.622	0.676
4	160	63.9	61.9	0.614	0.672
8	40	61.5	60.0	0.593	0.623
8	80	62.5	60.8	0.603	0.644
8	160	64.1	62.3	0.615	0.673

The second important contribution of this study is the proposal of a simple rule-selection algorithm based on lift. This algorithm is different from the complex rule-selection algorithm proposed by Xu et al. [24], which uses multiple criteria and sums their votes using manually selected weights. Additional details on the same will be provided in the context mining section.

E. Window Size Selection

To reduce the number of missing features, the size of the temporal window must be sufficiently large. Simultaneously, the temporal window should be sufficiently small so that subsequent ESMs have the minimum possible temporal overlap. Because the expected time interval between subsequent ESMs was 45 min, we determined that the complete temporal window would be approximately around 90 min. As stated, AFs are computed from multiple subwindows that span a given window size. However, the number of subwindows required for the AFs is an unknown quantity. Therefore, we examined multiple values for the number of subwindows, and the percentage of missing values was calculated for each combination. To identify this difficult tradeoff, the simplest approach involves evaluating the predictive performance of the features for each window size. This is done by building an ML classifier for each of the resulting feature spaces. Another heuristic for choosing the window size was the number of missing values for each combination.

Table V depicts the percentage of missing features for a given window size and number of subwindows. Note that we considered potential window sizes of 40, 80, and 160 min (color-coded in the figure). Moreover, we evaluated 2, 4, and 8 as the number of subwindows for each given temporal window. The figure presents features that are missing in at least 20% of the cases. It can be observed that more features appear to be missing when the number of subwindows is eight. However, no noticeable difference can be noted when the number of subwindows is two or four (first and second subfigures). Thus, we selected four as the number of subwindows (second subfigure). Following this, we selected a complete window size of 80 min because it proved to be better than a window size of 40 min and comparable to a window size of 160 min; moreover, it was close to first heuristic of 90 min. The table indicates that eight subwindows with a window size of 80 min produce 16.81% of the features that are mostly (more than 20% of the time) missing. Furthermore, this number is close to 14.39%; therefore, we chose four subwindows to prevent bias toward CFFs. In other words, if we had more subwindows, the CFFs would most likely perform better than the TFs and AFs, owing to the CFF's capability of computing features based on fine-grained context. We also noted the performance of AFs for different window and subwindow combinations. Additionally, a logistic regression classifier is fitted for each combination, and performance is compared. As can be observed from Table VI, the selected combination of four subwindows and a complete window size of 80 min provided the highest area under the curve (ROC-AUC) score of 0.676. Interestingly, when the subwindow size was 10 min or smaller, the ROC-AUC score dropped significantly to 0.623.

F. Feature Selection

We followed bottom-up feature engineering, wherein there is no prior belief regarding the use of specific features for predicting the outcome variable, receptivity. Initially, we considered all types of features as potentially useful and analyzed them. However, considering all the potential features for ML models is known to lead to the curse of dimensionality, where the data become sparser in a high-dimensional space. In addition, an increase in the number of features leads to overfitted learning models while adding significant storage and data analytics costs [26]. Furthermore, study requires a lower feature dimensionality to prevent an increasing number of frequent itemsets and rules while mining associations for CFFs. Notably, dimensionality reduction techniques can address this issue and improve the performance of the learning algorithms. These techniques can be grouped into feature extraction (e.g., PCA) and feature selection (e.g., LASSO). In this article, we employ feature-selection approaches that preserve interpretability (i.e., original definition of the selected

TABLE VII
SUMMARY OF FEATURE SELECTION STEPS

Removal step	Remaining number of features
Original AFs	778
Elimination of zero variance features	770
Elimination of mostly missing features	754
Elimination of pairwise correlations	416
Lasso	113

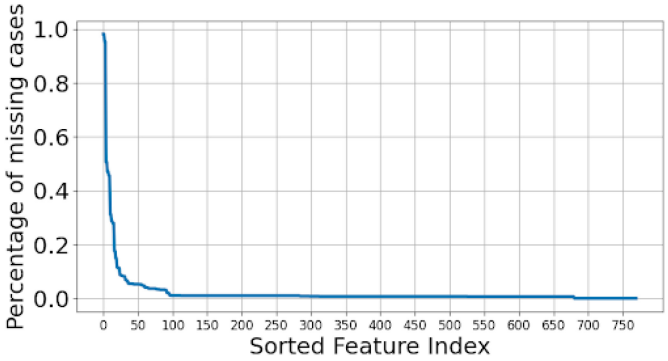

Fig. 7. Percentage of missing cases.

feature). The field of feature selection is vast and can constitute a separate research topic. Here, we provide a brief outline and refer the readers to relevant studies for details [26]. For supervised classification problems, feature selection can be divided into three primary categories. Among these, the most computationally efficient family involves filter-based approaches. Such approaches use data characteristics, pairwise correlation, variance, and correlation with the target variable to filter unwanted features. It is noteworthy that filter-based feature selection uses ground-truth information from the data set. Since proposed pipeline uses a separate RuleGenerateSet for selecting features that is independent of MLTrainEvalSet (data set for ML model evaluation), we could safely employ filter-based feature selection in the analysis. Feature selection pipeline below is performed on AFs extracted for RuleGenerateSet.

1) Feature Selection Pipeline: Given the nature of selected data set, a single-feature selection technique was deemed inadequate. Therefore, we devised the pipeline: Table VII lists the steps involved in the feature-selection pipeline. Notably, of the 778 AFs, 121 were selected.

Elimination of Low-Variance Features: As can be expected, features with zero variance do not offer any useful information to the classification algorithm; however, they can increase the complexity of the learning process. Therefore, we eliminated such features. Furthermore, we used a threshold of 1e-07 instead of an exact zero variance.

Elimination of Mostly Missing Features: As depicted in Fig. 7, certain features are more frequently missing than others. We particularly eliminated a feature if it was missing for more than 20% of the time. The rationale for such elimination was the following: when features are missing most of the time, imputing them may increase the noise to signal ratio. In total, 16 features were removed that were found to be missing more than 20% of the time.

Elimination of Pairwise Correlations: Even after previous steps in the pipeline, several of the features were correlated with each other. Notably, a high correlation between features A and B implies that feature A already carries the information provided by feature B. Therefore, in such cases, we retained only one feature to reduce dimensionality without losing information. To decide which of the correlated features should be retained, we exploited the correlation of the features with the target variable. In particular, features with a higher correlation with the target variable were retained. Herein, we considered features to be pairwise correlated if they exceeded a correlation tolerance threshold of 0.8. Consequently, 416 features were retained after selection.

Logistic Regression Using Lasso: The next step was to fit a logistic regression model with L1 regularization (i.e., lasso). This regularization forced the less predictive features to have an almost zero coefficient. Consequently, features with zero coefficients in the logistic regression classifier were eliminated. The figure indicates that only 113 of the 416 features had nonzero coefficients. Therefore, 113 features were retained after elimination.

G. Context Mining

In the field of ubiquitous computing, context can imply smartphone user's contextual information which could be user-centric and device centric [2], [27], [28]. In this section, we explain how we identified useful context for extracting the given feature. In other words, idea of CFF is to extract features from specific time intervals where desired context is met. Hence, to obtain the CFFs, first, a method to contextualize the feature extraction process had to be devised. We adapted the approach that was originally proposed for depression detection, which had a window size of 4 months. Note that the key difference between the study and that conducted by Xu et al. [24] lies in the manner in which we mined and selected the rules. Instead of ranking rules that differentiated the outcome variable, we mined the rules to seek frequently co-occurring sensor values. Another important difference is that window sizes are drastically different, where Xu et al. used four months, we use an 80-min window size. Similarly, for subwindow size, we use 20 min as opposed to an epoch (3 h). The sensor values recorded in each 20-min window served as a single basket for the ARM algorithm.

Association rules were mined from symbolic or categorical data. Hence, for rule mining, we simply quantized numeric features(sliding window features) as low, medium, and high at the participant level. Features that did not have variance over 20-min window were excluded from rule mining as they are not meaningful. It is noteworthy to mention that to prevent the bias in evaluation, context mining was performed (all procedures in this section) using only the RuleGenerateSet split of the K-EmoPhone data as explained in Data Split.

1) Mapping Subfeatures to Temporal Features: As shown in feature selection section 113 AFs were selected. However, to mine the association rules, normal (temporal) features were

used. Therefore, the 113 AFs (mean and standard deviation of the TFs) were mapped to the original TFs. Upon mapping, 113 AFs were reduced to 97 TFs.

2) Sliding Window Features as ARM Transactions: To mine the association rules, first, the sliding subfeatures were extracted and processed. These features were independent of the outcome variables and were extracted every 20 min (subwindow size) during the data collection period. This collection resulted in 9072 entries for the RuleGenerateSet. Upon feature extraction, the missing values were imputed with the mean values of the participant for a given feature. It is important to note that we did not make any assumptions about sensor data before figuring out the sliding window features.

3) Mining Association Rules: ARM is a widely used technique for the identification of associations in basket analysis, transactions, etc. The sliding window features for 9072 records were recoded as low, medium, and high to be used as transactions in ARM. We mined the association rules that implied co-occurrence between the features. This co-occurrence could then later be used to pair feature–context relationships. To mine the association rules, we used the MLExtend library [29]. In the following, we explain the details of two main steps of rule mining: 1) generation of frequent item sets and 2) generation of the association rules.

Mining Frequent Patterns: The mining of frequent patterns involves specifying a minimum level of support (number of occurrences for the item sets to be considered frequent). However, no clear guideline has been stipulated regarding this value in the literature. While generating frequent itemsets, we used a minimum support threshold of 0.3. This ensured that the found item sets were together at least 30% of the time. Due to its faster mining speed, the FP-growth algorithm was selected for frequent item mining. We constrained the maximum length of the frequent item sets to be five. Consequently, 1 946 843 frequent item sets were mined. In the second step, association rules were mined from the frequent item sets.

Mining Associations: In the next step, we extracted the rules from the given item sets. In total, 51 911 740 rules were mined with a minimum confidence threshold of 0.2. Among them, we retained 9 297 565 rules, which had only one feature as a rule consequence. This is because the CFF-extraction method required the result side of the rule to have only one thing (i.e., a binned feature).

4) Rule Selection: Owing to the large number of rules, extracting CFFs for all the rules significantly increases the dimensionality and computation. This makes the learning of a classifier impractical. Furthermore, the majority of these features exhibit high-pairwise correlations owing to the nature of the rule-mining algorithm. To address this problem, we propose using lift as a rule selection metric.

Rule Interestingness Measure: To determine whether a rule was interesting, we used lift as a metric. Notably, lift is a simple correlation measure, which can be described as follows: the occurrence of item A is independent of the occurrence of item B if $P(A, B) = P(A)P(B)$; otherwise, item A and item B are dependent and correlated as events [30]. This definition can easily be extended to more than two itemsets. The lift between the occurrences of A and B can be computed using

$$L(A, B) = \frac{P(A \cup B)}{P(A)P(B)}. \tag{1}$$

Consequently, the metric lift indicates if the occurrence of an antecedent (LHS) can predict the occurrence of a consequent (RHS). For contextual filtering, a feature (the RHS of the rule) conditioned upon the LHS of the rule's occurrence, the lift is a natural rule-selection measure.

VI. Experimental Settings (Implementation Details)

In this stage, we prepare the environment, methods, and metrics that are needed for the experiments in the next section.

A. Classifiers

For predicting user receptivity with ML, we used many off-the-shelf classifiers, including state-of-the-art gradient boosting algorithms, such as CatBoost [31]. However, to reduce the scope, sometimes a single classifier—a logistic regression—was used as a de facto classifier. We selected this classifier owing to its simplicity and computational efficiency. Later in this section, we benchmark other classifiers to assess the generalization of the findings of this work.

B. Evaluation

1) Cross Validation: For the ESM responses of the participants, each participant provided multiple observations. Therefore, three possible approaches could be adopted for evaluating the model with cross validation (CV). These included the following: 1) leave one subject out of CV–each participant's data are evaluated separately; 2) KFold CV–data are split into K folds, and each fold is evaluated (subject information is ignored); and 3) Group KFold CV–all data are split into K folds, and each fold is evaluated (subject data do not overlap across folds). Similar to [20], we adopted the third option and split the data into fivefold, where participant data could only be in one of the folds.

2) Evaluation Metrics: Notably, the commonly used metrics in interruptibility and user receptivity studies are the F1 score, precision, and recall. We follow the studies that are closely related to ours [11], [14], [20] based on Table I and those that use the F1 score as a primary metric. The F1 score is calculated using equation X

$$F1 = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}. \tag{2}$$

In binary classification problems, three main approaches can be adopted to compute the F1 score: 1) computing the F1 score only for the positive (receptive) class; 2) computing the F1 scores for both receptive and nonreceptive classes and averaging them; and 3) computing the F1 scores for each sample and averaging them. These approaches are referred to as binary, macro, and micro averaging, respectively, in the scikit-learn community. The evaluation metrics should reflect

this because the detection of both OMs and non-OMs is considered important. So, we calculated out the F1 scores for both classes and took the average. This gave us the macro-averaged F1 score, which takes into account that it is important to find both receptive and nonreceptive cases.

Another important metric in binary classification is the ROC-AUC. The ROC-AUC is preferred over other confusion matrix-based metrics, such as the accuracy or F1 score, because it evaluates the classifier's discriminative power at multiple cutoff thresholds. So, in this work, the ROC-AUC metric was used to choose the model, and the F1 score was used to compare it to other methods that have been written about.

C. Baseline Random Strategy

In this study, we used intelligent ML models that employed context features for detecting the opportune nature of a moment (i.e., user is receptive). To evaluate the improvement achieved using these ML models, we, however, needed a baseline random strategy (i.e., dummy classifier) that did not use any input but provided random predictions using a specified strategy. We followed the terminology of the scikit-learn ML community [32] to assess the performance of the random baseline strategy for binary classification. More specifically, dummy classification has the options listed below. For additional information on baseline random classification strategies, we refer readers to [32].

1) *Most Frequent:* The *predict* method always returns the most frequent class label in the observed y argument passed to fit.
2) *Prior:* The *predict* method always returns the most frequent class label in the observed y argument passed to fit (similar to *most frequent*).
3) *Stratified:* The *predict proba* method randomly samples one-hot vectors from a multinomial distribution parameterized by empirical class prior probabilities. The *predict* method returns the class label with a probability of one in the one-hot vector of *predict proba*. Therefore, each sampled row for both methods is independent and identically distributed.
4) *Uniform:* Generates predictions uniformly but in a random manner from the list of unique classes observed in y, i.e., each class has an equal probability.
5) *Constant:* Always predicts a constant label provided by the user. This is useful for metrics that evaluate nonmajority classes.

In line with the literature, for comparison with ML models, we selected the *most frequent* strategy as the random baseline strategy. However, we acknowledge that this setting may be sensitive to class distributions. In other words, depending on the binning threshold used, the resulting binary class distribution can change. This, in turn, can lead to a change in the baseline results for those who use the *most frequent* strategy. However, this can be addressed by employing a stratified random strategy that considers class imbalance. Therefore, we report the baseline results obtained based on both strategies.

D. Operating System Environment

The system used for this study is running on Ubuntu 22.04 as the operating system. The specific characteristics of the OS, including the number of CPU cores, random access memory (RAM) capacity, and solid state drive (SSD) storage, are essential factors that influence the system's performance during data processing and computation.

CPU Cores: The system is equipped with a total of 24 CPU cores, which allows for parallel processing and multitasking, enabling efficient handling of complex computations.

RAM Capacity: The system is equipped with 64 GB of RAM, a crucial resource for storing data that is currently being used by the system. The large RAM capacity enables the system to handle sizable data sets and memory-intensive tasks efficiently.

Operating System Version: The system is running on Ubuntu 22.04, a popular and widely used Linux distribution known for its stability, security, and extensive software support.

SSD Storage: The system is equipped with a 1 Terabyte (1TB) SSD. SSDs provide faster data access and read/write speeds compared to traditional mechanical hard drives, contributing to quicker system boot times and improved overall performance. These OS characteristics play a vital role in ensuring a smooth and efficient computing environment for conducting data analysis, running ML algorithms, and processing large-scale data sets effectively.

VII. Results and Analysis

Note that the experiments discussed in this section are intended to answer the aforementioned research questions. Here, the MLTrainEvalSet was used for all experiments. We compared the performance of the *three different feature extraction approaches*. In Section V, we described the process of obtaining such features, below we briefly highlight their differences.

1) *TFs:* The first approach extracts TFs for a given window (e.g., 80 min). This is the most commonly used approach for sensor data [18].
2) *AFs:* The second approach obtains the AFs, which represent the aggregation (mean and standard deviation) of subwindow features. The window with a size of 80 min has multiple subwindows (4×20 min). This approach is previously widely used in depression detection problems [24].
3) *CFFs:* The third approach is similar to that of the AFs but considers only specific parts of the subwindows for aggregation. A subwindow is considered for aggregation only if it meets a certain condition (i.e., context). As explained in detail in Section V, context is obtained via ARM. For the given feature, certain value (one of low, middle, high) of another feature(s) is (are) considered as a context. Such context determination criteria is not necessarily related to user receptivity. Instead, it finds the co-occurrence of multiple features (sensors) via ARM and assume those cases capture a different aspect of the user which was not captured by TFs and AFs.

TABLE VIII
EFFECT OF FEATURE ENGINEERING ON LOGISTIC REGRESSION CLASSIFIER PERFORMANCE

Feature type	Feature dimension	AUC	F1 score
Contextually Filtered F.	400	0.581	0.531
Aggregated F.	186	0.554	0.518
Temporal(full) F	93	0.562	0.533
Baseline (most_frequent)	N/A	0.5	0.38

TABLE IX
EFFECT OF FEATURE ENGINEERING ON OFF-THE-SHELF CLASSIFIERS. THE NUMBERS INDICATE THE ROC-AUC METRIC VALUES OF THE CLASSIFIER ON THE TEST SET

Classifier	TFs	AFs	CFFs
AdaBoost	0.539	0.537	**0.544**
CatBoost	0.574	0.561	**0.582**
DecisionTree	0.505	0.525	**0.532**
Gaussian Naive Bayes	0.546	0.555	**0.587**
GaussianProcess	0.521	0.527	**0.565**
LogisticRegression	0.562	0.554	**0.581**
Multi-layer Perceptron	0.550	0.539	**0.572**
QuadraticDiscriminantAnalysis	0.546	**0.550**	0.530
RandomForest	0.560	0.566	**0.580**
SVM	0.564	0.566	**0.575**
Avg. of all classifiers	0.547	0.548	**0.565**

A. Effect of Feature Engineering on Classification Performance

We compared three types of feature extraction approaches and their effects on the ML model's performance. In particular, comparisons included the performance of the classifier under different feature extraction settings, feature dimensionality, and performance of the classifier on the test set (i.e., ROC-AUC). A significant difference was noted between the ROC-AUC and F1 score metrics. The F1 score was evaluated on a single cutoff threshold, whereas the ROC-AUC considered all possible thresholds for evaluation. Consequently, the ROC-AUC was considered a more reliable metric. However, for ease of interpretation, we also included the F1 score. The results indicated that the CFF provides promising results. Note that the results displayed in Table VIII are only used for the logistic regression classifier. Because we used default hyperparameters, these results could be biased. Therefore, to confirm the effectiveness of the CFF, we experimented with multiple classifiers, as shown in Table IX. Results show that for all but the RF classifier, CF features are outperforming other kinds of features.

B. Benchmarking Common ML Classifiers on Combined Feature Set

As presented in Tables VIII and IX, the CFFs resulted in the best performance. Further, we examined the performance of the combined features, which include the TF, AF, and CFF. The resulting dimensionality of the features was $400 + 186 + 93 = 679$.

The results for 11 commonly used classifiers are listed in Table X. The performance measures include several confusion matrix-based metrics, such as accuracy, balanced accuracy (to account for data imbalance), and F1 score. The results additionally include the ROC-AUC as a more exhaustive performance metric that considers multiple cutoff thresholds for positive and negative predictive values. It is noteworthy that the results correspond to the default parameter settings of each classification algorithm without any hyperparameter fine-tuning. An evaluation of the results with a focus on the ROC-AUC suggests that the SVM, RandomForest, GaussianNaiveBayes classifiers provided the best results. The decision tree classifier produced an ROC-AUC score of 0.492, which is extremely close to that of a random guess. We posit that this can be attributed to the overfitting nature of the decision tree classifier. It is noteworhty that it is possible to do feature selection after combining three types of features. For simplicity, it is not done in this work.

TABLE X
BENCHMARKING MULTIPLE CLASSIFIERS ON A COMBINED SET OF FEATURES

Classifier	Accuracy	Balanced accuracy	F1 score	AUC
AdaBoost	52.2	49.5	0.484	0.520
CatBoost	**59.0**	54.1	0.518	0.577
DecisionTree	50.6	49.2	0.485	0.492
Gaussian Naive Bayes	56.9	**56.7**	**0.549**	0.587
GaussianProcess	55.2	53.7	0.528	0.522
LogisticRegression	57.8	54.4	0.534	0.579
Multi-layer Perceptron	54.9	52.0	0.516	0.541
QDA	58.0	49.8	0.432	0.498
RandomForest	58.4	54.2	0.523	0.580
SVM	58.4	54.8	0.527	**0.593**

TABLE XI
TIMINGS AND FEATURE DIMENSIONS FOR FEATURE EXTRACTION IN EVALUATION PHASE

Feature Type	Extraction Time	Dimensions
Temporal Features	13 min	93
Aggregated Features	13 min + 59.5 ms	186
Contextually Filtered Features	13 min + 2.27 sec	400

C. Complexity Analysis of the Proposed Method

The complexity of the proposed method for predicting user receptivity to health interventions can be analyzed in both the model preparation/training phase and the evaluation phase. In the model preparation phase, the computational requirements primarily depend on feature extraction and model training. The extraction of TFs and AFs can be performed relatively quickly, with a time complexity that is linear with respect to the data set size. The major bottleneck in the extraction of CFFs is the rule mining procedure. However, this is an offline process and is performed only once.

During the evaluation phase, the computational complexity primarily depends on the feature extraction process and the prediction model's requirements. The feature extraction involves extracting TFs, AFs, and CFFs. Table XI provides the timing information for the feature extraction process, specifically for the extraction of the 95-D TFs. It is important to note that the times provided for the extraction of TFs also apply to the computation of aggregated and CFFs, as

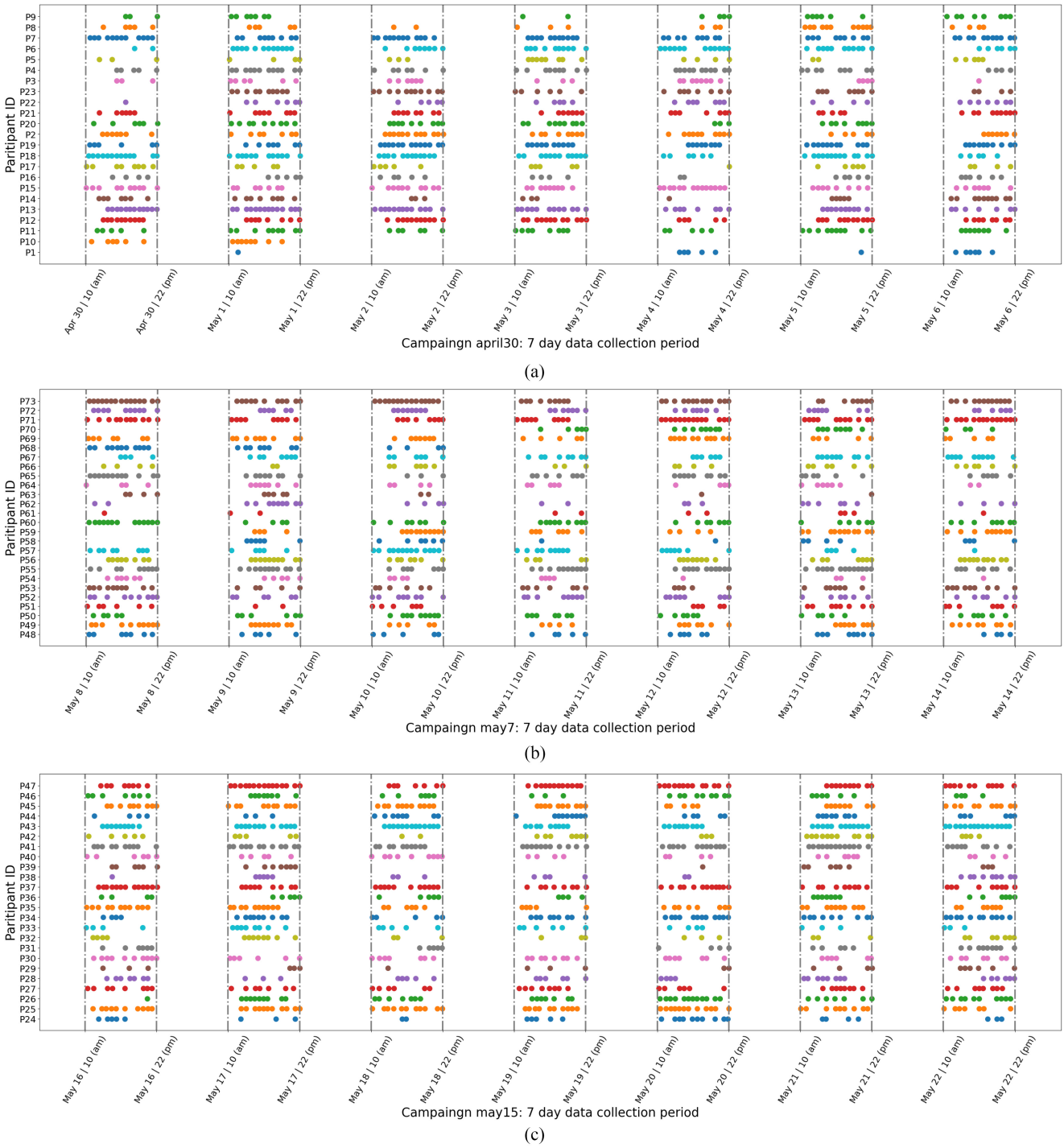

Fig. 8. Self-reported questionnaire collection timeline over seven days. The x-axis depicts the timeline for data collection and the y-axis presents the participant ID. (a)–(c) correspond to different data collections campaigns that started in April 30, May 7, and May 15.

they are derived from the TFs extracted for subwindows. This feature extraction complexity enables the development of the prediction models and their evaluation in real-time deployment scenarios.

D. Discussion

1) RQ1—Can ML Techniques Be Applied Effectively to Enhance Receptivity to Health Interventions?: The results displayed in Table VIII reveal that all three types of features performed better than the random baseline strategy (ROC-AUC of 50%). This implies that receptivity to DTx interventions can be increased by employing contextual factors, such as sensor data and user–phone interactions.

2) RQ2—Can We Improve the Performance of the ML Model With Better Feature Engineering?: The results displayed in Tables VIII–X reveal the significant impact of engineering features on the ML model's performance. This

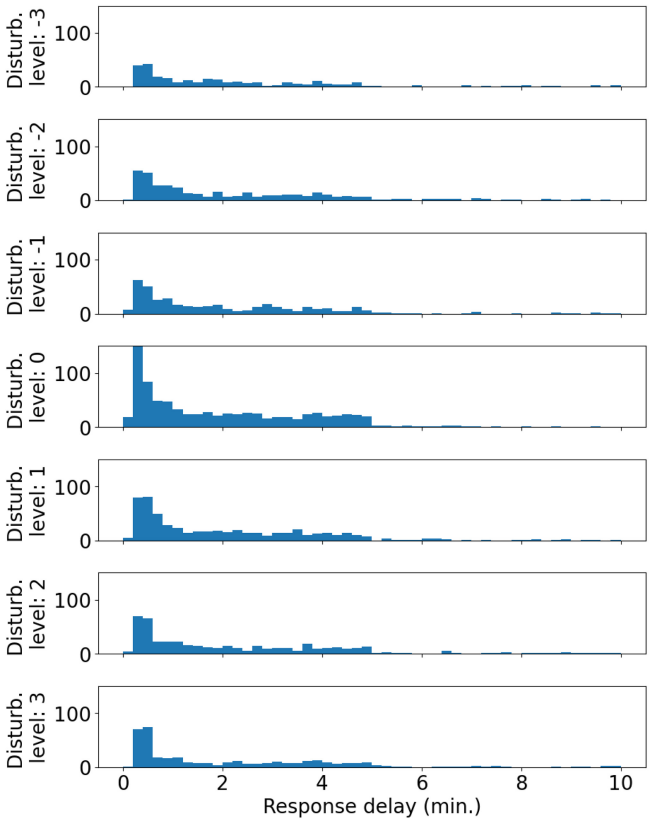

Fig. 9. Correlation between the response delay and self-reported disturbance.

effect is retained even when the choice of the ML model changes, as indicated in Table IX. This implies that in addition to tuning the hyperparameters of classification algorithms, an additional approach for improving the predictive power of receptivity prediction should involve the engineering of more powerful features with rich representations.

3) Limitations of the Proposed Method and Future Directions: A key strength of the proposed receptivity prediction model lies in its use of CFFs which offer a comparatively richer representation. However, acquiring such features can be more intricate compared to other conventional TFs. Despite this, we believe that the complexity of obtaining CFFs is a minor, if not negligible, constraint in light of the benefits offered by their robust representation. Even though the inclusion of CFFs introduces additional steps thereby increasing complexity, this process is confined to the offline phase where rules for contextually filtering the process are identified. In the deployment (or online) phase, the complexity involved in extracting CFFs aligns closely with common procedures for TF or AF extraction. The topic of association-rule mining is an extensively investigated area. The findings underscore the superior potential of CFFs in predicting user receptivity. Nevertheless, there is a multitude of options for rule selection that could enhance the performance of CFFs even further. As such, an intriguing avenue for future research might focus on optimizing rule-selection methods. For instance, during the investigation, we discerned that the parameters of rule mining significantly influenced the final performance of CFF-based ML models.

Another promising research path could involve direct mining of association rules from raw data sources, as opposed to mining from preprocessed TFs extracted over a 20-min (sub) windows. The rationale for this is that organizing raw data into bins can make the data more insightful. For example, it might be more effective to categorize heart rate sensor readings as low, median, and high, rather than labeling time-series complex features as such. Moreover, these rules tend to be more meaningful and interpretable.

Furthermore, fine-tuning the model's hyperparameters presents substantial opportunities for enhancing the performance of the ML model.

VIII. Conclusion

This study demonstrated the effect of feature engineering on the receptivity prediction performance of an ML model. In other words, the results revealed that the same ML model could perform better when richer features were engineered (average ROC-AUC of 0.565 versus 0.547 for CFFs and TFs). This was achieved based on the proposed CFF-extraction approach. Although locating useful features that can identify a subject's receptivity is often complicated, we achieved the same using a novel automated feature engineering approach.

Appendix

See Figs. 8 and 9.

References

[1] U. Lee et al., "Toward data-driven digital therapeutics analytics: Literature review and research directions," 2022, *arXiv:2205.01851*.
[2] M. N. Burns et al., "Harnessing context sensing to develop a mobile intervention for depression," *J. Med. Internet Res.*, vol. 13, no. 3, 2011, Art. no. e1838.
[3] B. Y. Laing et al., "Effectiveness of a smartphone application for weight loss compared with usual care in overweight primary care patients: A randomized, controlled trial," *Ann. Internal Med.*, vol. 161, no. 10S, pp. S5–S12, 2014.
[4] P. Klasnja et al., "Microrandomized trials: An experimental design for developing just-in-time adaptive interventions," *Health Psychol.*, vol. 34, p. 1220, Dec. 2015.
[5] J. Linardon, P. Cuijpers, P. Carlbring, M. Messer, and M. Fuller-Tyszkiewicz, "The efficacy of app-supported smartphone interventions for mental health problems: A meta-analysis of randomized controlled trials," *World Psychiat.*, vol. 18, no. 3, pp. 325–336, 2019.
[6] L. M. Collins and K. C. Kugler, *Optimization of Behavioral, Biobehavioral, and Biomedical Interventions: Advanced Topics*. New York, NY, USA: Springer, 2018.
[7] O. Inan et al., "Digitizing clinical trials," *NPJ Digit. Med.*, vol. 3, no. 1, pp. 1–7, 2020.
[8] P. Liao et al., "Just-in-time but not too much: Determining treatment timing in mobile health," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 2, no. 4, pp. 1–21, 2018.
[9] P. Liao, K. Greenewald, P. Klasnja, and S. Murphy, "Personalized heartsteps: A reinforcement learning algorithm for optimizing physical activity," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 4, no. 1, pp. 1–22, 2020.
[10] V. Pejovic and M. Musolesi, "InterruptMe: Designing intelligent prompting mechanisms for pervasive applications," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2014, pp. 897–908.
[11] M. Pielot, B. Cardoso, K. Katevas, J. Serrà, A. Matic, and N. Oliver, "Beyond interruptibility: Predicting opportune moments to engage mobile phone users," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 3, pp. 1–25, 2017.
[12] N. Kern, S. Antifakos, B. Schiele, and A. Schwaninger, "A model for human interruptability: Experimental evaluation and automatic estimation from wearable sensors," in *Proc. 8th Int. Symp. Wearable Comput.*, vol. 1, 2004, pp. 158–165.

[13] I. Nahum-Shani et al., "Just-in-time adaptive interventions (JITAIs) in mobile health: Key components and design principles for ongoing health behavior support," *Ann. Behav. Med.*, vol. 52, no. 6, pp. 446–462, 2018.
[14] F. Künzler, V. Mishra, J.-N. Kramer, D. Kotz, E. Fleisch, and T. Kowatsch, "Exploring the state-of-receptivity for mHealth interventions," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 3, no. 4, pp. 1–27, 2019.
[15] W. Choi, S. Park, D. Kim, Y.-K. Lim, and U. Lee, "Multi-stage receptivity model for mobile just-in-time health intervention," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 3, no. 2, pp. 1–26, 2019.
[16] J. Fogarty, S. E. Hudson, and J. Lai, "Examining the robustness of sensor-based statistical models of human interruptibility," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2004, pp. 207–214.
[17] B. Poppinga, W. Heuten, and S. Boll, "Sensor-based identification of opportune moments for triggering notifications," *IEEE Pervasive Comput.*, vol. 13, no. 1, pp. 22–29, Jan.–Mar. 2014.
[18] M. Choy, D. Kim, J.-G. Lee, H. Kim, and H. Motoda, "Looking back on the current day: Interruptibility prediction using daily behavioral features," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2016, pp. 1004–1015.
[19] H. Sarker et al., "Assessing the availability of users to engage in just-in-time intervention in the natural environment," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2014, pp. 909–920.
[20] V. Mishra, F. Künzler, J.-N. Kramer, E. Fleisch, T. Kowatsch, and D. Kotz, "Detecting receptivity for mHealth interventions in the natural environment," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 5, no. 2, pp. 1–24, 2021.
[21] A. Mehrotra, V. Pejovic, J. Vermeulen, R. Hendley, and M. Musolesi, "My phone and me: Understanding people's receptivity to mobile notifications," in *Proc. Conf. Human Factors Comput. Syst.*, 2016, pp. 1021–1032.
[22] S. Kang et al., "K-EmoPhone: A mobile and wearable dataset with in-situ emotion, stress, and attention labels," *Sci. Data*, vol. 10, no. 1, p. 351, 2023.
[23] A. Zheng and A. Casari, *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. Sebastopol, CA, USA: O'Reilly Media, Inc., 2018.
[24] X. Xu et al., "Leveraging routine behavior and contextually-filtered features for depression detection among college students," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 3, no. 3, pp. 1–33, 2019.
[25] R. Wang et al., "StudentLife: Assessing mental health, academic performance and behavioral trends of college students using smartphones," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2014, pp. 3–14.
[26] J. Li et al., "Feature selection: A data perspective," *ACM Comput. Surveys*, vol. 50, no. 6, pp. 1–45, 2017.
[27] Y. Vaizman, K. Ellis, and G. Lanckriet, "Recognizing detailed human context in the wild from smartphones and smartwatches," *IEEE Pervasive Comput.*, vol. 16, no. 4, pp. 62–74, Oct.–Dec. 2017.
[28] V. Pejovic and M. Musolesi, "Anticipatory mobile computing: A survey of the state of the art and research challenges," *ACM Comput. Surveys*, vol. 47, no. 3, pp. 1–29, 2015.
[29] S. Raschka, "MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack," *J. Open Source Softw.*, vol. 3, no. 24, p. 638, Apr. 2018. [Online]. Available: http://joss.theoj.org/papers/10.21105/joss.00638
[30] J. Han, J. Pei, and H. Tong, *Data Mining: Concepts and Techniques*. Burlington, MA, USA: Morgan Kaufmann, 2022.
[31] A. V. Dorogush, V. Ershov, and A. Gulin, "CatBoost: Gradient boosting with categorical features support," 2018, *arXiv:1810.11363*.
[32] "SKLearn dummy classifier." Accessed: Oct. 25, 2022. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyClassifier.html

Jumabek Alikhanov (Student Member, IEEE) received the B.S. degree from Tashkent University of Information Technology, Tashkent, Uzbekistan, in 2014, and the M.E. degree from Inha University, Incheon, South Korea, in 2017, where he is currently pursuing the Ph.D. degree in electrical and computer engineering.

He is currently devoted to the development of lightweight deep learning models, emphasizing the design of resource-efficient AI computing machines that offer speed and reduced carbon footprint. His research spans machine learning, with a focus on applications in computer vision, sensor data analytics, and natural language processing. He is focused on designing resource efficient AI computing machines.

Panyu Zhang (Student Member, IEEE) received the B.S. degree in industrial engineering from the School of Mechanical Engineering, Beijing Institute of Technology, Beijing, China, in 2019, and the M.S. degree from the Department of Industrial and Systems Engineering, Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2023, where he is currently pursuing the Ph.D. degree with the Graduate School of Data Science.

His research is in the broad area of machine learning for health and health behavior change. He is currently devoted specifically to the field of reproducible mobile stress detection and employing advanced deep learning models to push the limit of real-world mobile stress detection.

Youngtae Noh (Member, IEEE) received the B.S. degree in computer science from Chosun University, Gwangju, South Korea, in 2005, the M.S. in information and communication from Gwangju Institute of Science and Technology, Gwangju, in 2007, and the Ph.D. degree in computer science from the University of California at Los Angeles, Los Angeles, CA, USA, in 2012.

He is currently an Associate Professor with Energy AI, Korean Institute of Energy and Technology, Naju, South Korea. His previous appointments include an Associate Professorship with Inha University (Yonghyeon Campus), Incheon, South Korea, from 2015 to 2022, and a Staff position with Cisco Systems, San Jose, CA, USA, from 2012 to 2014. His research interests are in mobile and pervasive computing, mobile systems, mobile data science, mobile-HCI, cloud computing, data center networking, wireless networking, and future Internet architectures.

Hakil Kim (Member, IEEE) received the M.Sc. and Ph.D. degrees in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 1985 and 1990, respectively.

In 1990, he was with the College of Engineering, Inha University, Incheon, South Korea, where he is currently a Full Professor with the Department of Information and Communication Engineering. In order to retain the balance between academic research and commercial development, he founded Vision Inc., Blackwood, NJ, USA, in 2014. His research interests include biometrics, intelligent video surveillance, and embedded vision for autonomous vehicles.

Prof. Kim has been actively involved as a Project Editor of the International Standardization of Biometrics at ISO/IEC JTC1/SC37, since 2003.