RELAPSE PREDICTION FROM LONG-TERM WEARABLE DATA USING SELF-SUPERVISED LEARNING AND SURVIVAL ANALYSIS

E. Fekas, A. Zlatintsi, P. P. Filhtisis, C. Garoufis, N. Efthymiou and P. Maragos
School of ECE, National Technical University of Athens, 15773 Athens, Greece
fekas.evangelos@gmail.com, {nzlat, maragos}@cs.ntua.gr, {nefthymiou, filby}@central.ntua.gr, cgaroufis@mail.ntua.gr

ABSTRACT
The introduction of biometric signal analysis in psychiatry could potentially reshape the field by making it more accurate, proactive and personalized. Such biosignals usually acquired from wearables encompass the quantification of human behavior and traits. In this study, we use long-term data acquired from commercial smartwatches, including kinetic and physiological signals, to extract information-thick descriptors that are used for the prediction of subsequent relapses in patients in the psychotic spectrum. Specifically, we propose a novel combination of methods based on Self-Supervised Learning and Survival Analysis that operates on unlabeled and censored data. When combined with other static features that describe the past course of the patient’s health, the proposed methodology yields promising predictive results in terms of two standard survival analysis metrics.

Index Terms — Self-Supervised Learning, Psychotic Disorders, Smartwatch Wearables, Survival Analysis, Relapse Prediction

1. INTRODUCTION
The current state of psychiatric practice relies primarily on brief clinical interactions focused on history taking, symptom rating and clinical judgments, and less on measuring emotion, cognition, or behavior with standard, validated tools [1]. As a result, therapists usually struggle to detect patients’ relapses in time [2]. One way to improve the precision of the diagnosis is the moment-by-moment quantification of the individual’s behavioral and cognitive state using personal digital devices. Wearable consumer products such as smartwatches are the most promising sources for obtaining such information, as they offer a reliable, unobtrusive, and remote personalized collection of numerous physiological data through their sensors. Using such signals to develop a feedback system that alerts therapists when symptoms recur [3]. Mostly, though, it could help reduce the severity of relapses or even prevent them from occurring [4].

Modern approaches such as deep learning are difficult to employ when handling physiological signals, as they rely heavily on vast amounts of carefully annotated data, which is uncommon and impractical in such low-label data regimes. Indeed, although collecting large amounts of unlabeled biosignals is easy as they are passively recorded, their labels are often difficult to obtain, requiring expert knowledge and hours of manual annotation [5]. Thus, automatically learning valuable representations could drastically reduce the cost and time required to process them, while leveraging the existing oversupply of unlabeled data.

The field trying to overcome this problem is Self-Supervised Learning (SSL). The idea behind SSL is to utilize the structure of unlabeled data to automatically generate labels from them. Together with the original unlabeled signal, these labels form a supervised problem called pretext task. Early examples of pretext tasks were: solving jigsaw puzzles [6] and predicting image rotations [7]. However, their representations could not generalize well because their respective encoders were overfitting the task’s transformations [8]. An alternative to this is contrastive learning [9, 10, 11], which instead of trying to predict a task’s transformations, learns representations that are invariant to them. Contrastive methods are based on creating positive and negative samples, forcing the former closer in the embedding space, and pushing the latter away. The representations learned from the pretext task can then be reused in a downstream task, potentially reducing the required number of labels.

An interesting family of downstream tasks is Time-to-Event Prediction, which is adequately handled by Survival Analysis, a collection of statistical procedures with applications in medicine [12], engineering [13], and economics [14]. The main difference between survival analysis and regression methods is that the former can operate on partially observed (censored) data. One such case is right censoring, where all that is known about a subset of the data is that no event has occurred for at least some known time.

In our work, we propose a novel way to predict subsequent relapses in patients with psychotic disorders, based on SSL and Survival Analysis. To this end, we use long-term data acquired within the e-Prevention project (http://e-prevention.gr), which aims to provide innovative e-health services and effective monitoring for patients with mental disorders [15]. To the best of our knowledge, the combination of these methods has never been used before for such a relapse prediction task. Specifically, we perform Self-Supervised pretraining to learn representations from fully unlabeled, long-term, continuous recordings of biometric signals collected through commercial smartwatches. We then provide these representations as inputs to survival analysis models to predict subsequent relapses on the data subset containing relapse labels. When combined with handcrafted features, this method yields promising results.

The rest of this paper is organized as follows: In Sec. 2, we present data collection, preprocessing, and the final datasets used for learning and evaluating the task of relapse prediction. In Sec. 3, our proposed methodology is described; while Sec. 4 presents a thorough analysis of our method’s performance. Finally, Sec. 5 concludes our work and gives future directions.

2. DATA PREPROCESSING AND DATASET CREATION
The raw e-Prevention dataset, used in this work, consists of data acquired from a Samsung Gear S3 Frontier smartwatch, recorded continuously (24/7 - except from ca. 2 hours/day when the smartwatch...
was charging; during charging all data were uploaded to a cloud-based platform [16]). Data collection began in 11/2019 and continues to this date. Sixty-four (64) people participated in the project (26 controls and 38 patients); the detailed recruitment protocol is presented in [15]. Briefly, all patients undergo monthly assessments by the project’s clinicians, who label the patient’s condition as healthy or relapsing. Twenty (20) out of 38 patients had actually experienced one or more relapses, resulting in a total of 37 relapsing incidents of varying duration and severity, as shown in Fig. 1. The measurements that we use in this work are: 3-axis linear acceleration and angular velocity (accelerometer and gyroscope sensors, respectively; both sampled at 20 Hz), heart beats per minute, and RR-intervals (obtained via Photoplethysmography; sampled at 5 Hz).

Data Preprocessing: In order to disregard outliers, we excluded data points exceeding the kinetic sensors’ limit values, identical consecutive RR intervals, and intervals longer than 2000 ms or shorter than 300 ms. We then imputed the excluded values with linear interpolation while keeping only 1-hour recordings, where the heart rate sequence summed up to at least 54 minutes (an empirical threshold corresponding to 90% of valid data [17]). Finally, we applied an 1-minute moving average filter. This way, we can have samples covering more extensive time windows while maintaining a relatively small sample size. This operation corresponds to low-pass filtering, in other words noise reduction, since the smartwatch’s noise mainly lies in high frequencies as found in [18].

Dataset Creation: Intuitively, the interval used to assess whether the user is close to a relapse should be long enough so that the model can detect unusual behaviors compared to normal ones. On the other hand, the longer the interval, the more likely we are to have gaps, either due to smartwatch charging or errors during data collection. Concatenation or interpolation of such gaps would introduce noise. Since the models we use exploit time series dynamics, they may classify such noise as an anomaly. Given the above constraints, we experimented with three datasets consisting of consecutive measurements of 4, 8, and 12-hour samples covering more extensive time windows while maintaining a relatively small sample size. This operation corresponds to low-pass filtering, in other words noise reduction, since the smartwatch’s noise mainly lies in high frequencies as found in [18].

<table>
<thead>
<tr>
<th>Sample size</th>
<th>Pretext Dataset ( (S,C,L) )</th>
<th>Downstream Dataset ( (S,C,L) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 hours</td>
<td>(19126, 10, 240)</td>
<td>(7193, 10, 240)</td>
</tr>
<tr>
<td>8 hours</td>
<td>(13909, 10, 480)</td>
<td>(5925, 10, 480)</td>
</tr>
<tr>
<td>12 hours</td>
<td>(10576, 10, 720)</td>
<td>(4584, 10, 720)</td>
</tr>
</tbody>
</table>

Table 1. Datasets’ # samples \( (S) \), # variables \( (C) \), and sample lengths \( (L) \). \( C \) consists of 3 channels for the accelerometer, 3 for the gyroscope, 2 for heart rate, and 2 for the sine/cosine variables.

3. METHODOLOGY

In order to accurately predict the time until a subsequent relapse, we propose a methodology based on SSL and survival analysis. The benefits of combining these two approaches are twofold: 1) By using SSL, we exploit our entire unlabeled dataset during representation learning, in contrast to a supervised method that would only use the labeled subset. 2) With survival analysis, we use samples with no relapses in their future, which contains valuable information that the user is free of relapses for at least \( t \) time.

Self Supervised Learning: As a first step, we pretrain three state-of-the-art contrastive SSL methods for time series on our Pretext Dataset. The first method is Mixing-Up [19], which proposes an augmentation scheme, where new samples are generated by mixing two data samples with a mixing component. The pretext task is correctly predicting the mixing proportion of two time-series samples and a fully convolutional network (FCN) with three convolutional layers is used as a backbone.

The second method is ‘Time-Series representation learning framework via Temporal and Contextual Contrasting’ (TS-TCC) [20], where two separate data views are created, using weak and strong augmentations, encoded by a three-layer convolutional network and summarized into a context vector using a transformer model. The pretext task uses the context vector of the strong augmentations to predict future timesteps of the weak ones and vice versa. In order to learn discriminative representations, a contextual contrasting module is also built upon the context vectors.

Lastly, we used a method, which proposes Time-Frequency Consistency (TFC) [21], according to which for every time series sample, there exists a latent time-frequency space, where time- and frequency-based representations (both encoded via 3-layer 1-D ResNets [22]) of the same sample together with their local augmentations, are close to each other.
In order to apply the above methods to a new dataset like e-Prevention, we must first perform hyperparameter tuning. In our case, we optimized the hyperparameters on a proxy downstream task: person identification i.e., predicting the unique identifier of the smartwatch user. This task can provide valuable insights into patients with psychiatric disorders by identifying the characteristic behavior of these individuals [23]; as such, representations succeeding in the person identification task could prove useful in analyzing their condition. Since the downstream dataset’s user-ID distribution is severely imbalanced (some users contain more than 700 recordings, while others contain less than 40) we filter the dataset for this proxy task, using only 14 of 20 users with more than 100 instances.

Survival Analysis: After pretraining we use the learned representations on the downstream dataset to predict the time until the subsequent relapse. Figure 1 shows the time intervals of relapses and their severity, for each user. The first relapsing episode’s first date is 2019-11-20 for User 3, while the ending date of the last episode is 2021-12-26 for User 18. We mark every sample with a dot, i.e., days with a preceding relapse event are denoted in black, otherwise they are marked in grey. To predict the time before the appearance of relapses, one could pick the black dots and regress the time interval from a specific black dot to the beginning of the next event. This results to loss of the information that the gray dots provide for at least t event-free periods, where t is the interval from the dot to the user’s last sample. These samples are called ‘right censored’ and are handled properly with survival analysis methods.

In survival analysis [24, 25], instead of predicting a single number as the time until an event, we are predicting a function: either the survival or hazard function. Let T denote a continuous non-negative random variable representing the time until an event occurs. The survival function \( S(t) \) is the probability that the event of interest has not occurred by some time \( t \): \( S(t) = \Pr[T \geq t] \). Similarly, the hazard function \( h(t) \) denotes an approximate conditional probability, that the event will occur within \( [t, t+dt] \), given that it has not occurred before: \( h(t) = \frac{dS(t)}{dt} \), and the cumulative hazard function is the integral over the interval \([0, t] \) of the hazard function: \( H(t) = \int_0^t h(u) \, du \). Finally, by subdividing the time axis of the predicted cumulative hazard function in \( J \) parts, we can calculate the risk score of each sample \( x \), such that: \( r(x) = \sum_{j=1}^{J} H(t_j, x) \).

We employed four survival-regression models that can handle non-linearities in their input covariates (the pretrained embeddings in our case). The first two are extensions of Random Forests to right-censored data: Conditional Survival Forest [26] and Extremely Randomized Survival Trees [27]. The latter two are based on Deep Neural Networks: Neural MTLR [28] and DeepSurv [29].

Finally, we evaluated their performance using two standard survival analysis metrics: C-index [30] and Brier-Score (BS) [31]. The first is a discrimination metric, which measures the model’s ability to correctly provide an accurate ranking of survival times: C-index = \( \frac{\sum_{i,j} 1 (T_j < T_i \land T_j > r_i \delta_j)}{\sum_{i,j} 1 (T_j < T_i \land T_j > r_i)} \), where \( i, j \) iterate over the whole dataset and \( \delta_k \in \{0, 1\} \), indicates whether sample \( k \) is right censored or not. The BS, unlike the C-index, is both a discrimination and a calibration metric and is calculated as: \( BS(t) = \frac{1}{N} \sum_{i=1}^{N} \left( \frac{1 - S(t, x_i)}{\hat{G}(t)} \right)^2 1(t \leq T_i \land r_i = 1) + \left( \frac{S(t, x_i)}{\hat{G}(t)} \right)^2 1(T_i > t) + \frac{1}{\hat{G}(t)} \), where \( \hat{G}(t) = \Pr[C > t] \) is the estimator of the conditional survival function of the censoring times, and \( C \) is the censoring time. In a well-calibrated model the predicted event risk, for each time \( t \), should match the actual frequency of occurred events in the data. A non-random predictor should have C-index > 0.5 and BS(t) < 0.25, for the given times \( t \) that we are interested. To summarize the model performance with a single metric over the whole analysis period, the Integrated Brier Score is used: \( IBS(t_{max}) = \frac{1}{t_{max}} \int_{0}^{t_{max}} BS(t) \, dt \).

However, in this work, we use \( -\log \)(IBS) instead of IBS to convert IBS into a ‘higher is better’ metric.

4. EXPERIMENTAL EXPLORATION

Hyperparameter tuning in Person Identification Task: During hyperparameter-tuning, we pretrained the three SSL models mentioned in Sec. 3 on the entire 4-hour pretext dataset. We then trained a 14-class linear classifier, on top of the pretrained embeddings, predicting the user’s-ID of each recording. After maximizing the macro-F1 score in the above task, the embeddings’ dimensions are 128, 256, and 1408 for Mixing-Up, TFC, and TS-TCC, respectively.

Although we do not expect a linear classifier to solve such a challenging 14-class problem completely, comparing the relative performance of the embeddings on the person identification task is reasonable. Indeed, if users are linearly separable in the learned embeddings space, then the embeddings contain information about the user’s ‘normal’ behavior [23], which is a good starting point for the upcoming time-to-relapse problem. We assessed the embeddings’ performance through a two-step, 5-fold cross-validation: First, we pretrained the SSL models with five different initializations on the whole pretext dataset using the tuned hyperparameters described above. We also fitted MiniRocket [32], a state-of-the-art feature extractor, with five different seeds, as a baseline. Thus, we got five embeddings for each model (one for each fold), which were used in the second step as input to a 14-class classifier predicting the downstream dataset’s user IDs. A different 75/15/10% train/val/test split was used in the downstream dataset in each of the five iterations. Since the label distribution is highly imbalanced, we focus on macro-averaged F1 and Average Precision (AUPRC) scores. The final aggregated results, are shown in Table 2; Mixing-Up achieves the best results with mean-F1 = 0.77, while TS-TCC the worst.

Survival analysis: After tuning, we proceed to our main task, predicting the time until the subsequent relapse. We utilized the embeddings that achieved the highest F1-score in the ID task for every model out of the five iterations. If we think of each embedding as the ‘summary’ of its respective interval for each user, we aim to predict if this user is approaching a relapse. Intuitively, each user exhibits different variations from their stable behavior, so we concatenated the one-hot-encoded user-ID to the pretrained embeddings. We saw that this additional feature boosted the performance, so we are only considering the concatenated embeddings for the rest of this section. Afterward, we split the downstream datasets at a 60/40/40% train/test ratio and trained four survival models on top of these embeddings.

In Fig. 2, the \( x \)-axis shows the input pretrained embeddings, and the \( y \)-axis the score distribution over the evaluated survival models. The three colors represent the duration of the three datasets. We observe that TFC embeddings obtain the best results, with C-index = 0.754 and \( -\log \)(IBS) = 2.012 when coupled with DeepSurv model on the 12-hour dataset. We also notice that the 8 and 12-hour datasets offer better discrimination (higher C-index) while the 4-hour offers calibration (higher \( -\log \)(IBS)).
that change only after an event occurs, such as the number of pre-compliance, or the same diagnosis. Similarly, we can add features from sample to sample. However, it only 'dynamic' features, i.e., features that describe one user’s sample/day, which also change from sample to sample. However, it would be helpful to add static features (st) so that the survival-model could associate similar users, i.e., users with the same age, treatment compliance, or the same diagnosis. Similarly, we can add features that change only after an event occurs, such as the number of previous episodes (p_ep), the severity level of the last episode (ll), and the severity level of the last episode (ll), and the severity level of the last episode (ll).

Next, we compare the different survival models. We chose the 8-hour dataset (highest C-index, thus discrimination, which also applies to the 12-hour dataset), as we are interested in correctly distinguishing users with high risk of relapse from those with low risk. Using the 12-hour dataset is also impractical due to watch charging errors. In Fig. 3, we see the results for the 8-hour dataset for different combinations of pretrained embeddings and survival-models. The two deep-learning based methods: Neural MTLR and DeepSurv have more capacity and thus obtain better results in our context (where we have enough labeled samples). In terms of the pretrained embeddings and for the DeepSurv survival model that yields the best results, TS-TCC gives the worst results with C-index = 0.708, while Mixing-up the best with C-index = 0.753.

Feature importance and static features: So far, we have used only 'dynamic' features, i.e., features that describe one user’s sample/day, which also change from sample to sample. However, it would be helpful to add static features (st) so that the survival-model could associate similar users, i.e., users with the same age, treatment compliance, or the same diagnosis. Similarly, we can add features that change only after an event occurs, such as the number of previous episodes (p_ep), the severity level of the last episode (ll), and

### Table 3. Relative change (%) of scores, when we remove one feature at a time, compared to when all features are active.

<table>
<thead>
<tr>
<th>(+): included</th>
<th>C-index rel. change (%)</th>
<th>− log (IBS) rel. change (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>user's-ID</td>
<td>-23.74</td>
<td>-19.24</td>
</tr>
<tr>
<td>heart rate</td>
<td>-1.66</td>
<td>-1.30</td>
</tr>
<tr>
<td>hour-of-day</td>
<td>-1.15</td>
<td>-0.01</td>
</tr>
<tr>
<td>accelerometer</td>
<td>-0.03</td>
<td>+0.22</td>
</tr>
<tr>
<td>gyroscope</td>
<td>+1.94</td>
<td>+0.14</td>
</tr>
<tr>
<td>+ st</td>
<td>-1.39</td>
<td>-2.67</td>
</tr>
<tr>
<td>+ ps, p_ep, ll</td>
<td>+9.56</td>
<td>+14.17</td>
</tr>
<tr>
<td>+ st, ps, p_ep, ll</td>
<td>+11.60</td>
<td>+15.90</td>
</tr>
</tbody>
</table>

In this paper, we used large amounts of unlabeled kinetic and physiological data acquired from wearable devices, during the e-Prevention project, to learn valuable representations for the task of relapse prediction. To this end, we used a novel combination of Self-Supervised Learning and Survival Analysis. We utilized the smartwatch’s unique identifiers to solve the person identification problem in order to monitor the self-supervised training procedure. Our proposed methodology achieves promising results, especially when combined with additional static attributes that describe the past course of the patient’s condition, especially concerning past relapse episodes. In the future, we would like to explore how this methodology would perform when we fuse wearable’s data with other modalities, such as vision and/or audio.

5. CONCLUSIONS
6. REFERENCES


