A feature representation learning method for temporal datasets

Ward van Breda[1], Mark Hoogendoorn[1], A.E. Eiben[1], Gerhard Andersson[2], Heleen Riper[3],
Jeroen Ruwaard[3], and Kristofer Vernmark[2, 4]
[1]VU University Amsterdam, Dept. of Computer Science
De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands
Email: {w.r.j.van.breda, m.hoogendoorn, a.e.eiben}@vu.nl
[2]Linköping University, Dept. of Behavioural Sciences and Learning
SE-581 83 Linköping, Sweden
Email: gerhard.andersson@liu.se
[3]VU University Amsterdam, Dept. of Clinical Psychology
De Boelelaan 1081, 1081 HV Amsterdam, the Netherlands
Email: {j.j.ruwaard, h.riper}@vu.nl
[4]Psykologpartners Linköping
St.t Larsgatan 30, 582 24 Linköping, Sweden
Email: kristofer.vernmark@psykologpartners.se

Abstract—**Predictive modeling of future health states can greatly contribute to more effective health care. Healthcare professionals can for example act in a more proactive way or predictions can drive more automated ways of therapy. However, the task is very challenging. Future developments likely depend on observations in the (recent) past, but how can we capture this history in features to generate accurate predictive models? And what length of history should we consider? We propose a framework that is able to generate patient tailored features from observations of the recent history that maximize predictive performance. For a case study in the domain of depression we find that using this method new data representations can be generated that increase the predictive performance significantly.**

I. Introduction

E-health is a domain where health care and advancements in electronics, computer science and communication science meet (for more info, see e.g. [6]). Due to the developments in sensor hardware, sensor analytics, and (mobile) applications that measure health related aspects over time seamlessly a wealth of data is becoming available. This data, often of a temporal nature, has potential value for the development of intelligent e-health solutions for physical and mental health. One aspect that can drive these intelligent solutions involves predictive modeling: if we can predict a future state we might be able to intervene before the onset of certain undesired behavior or diseases.

Predictive modeling with this highly temporal data is a difficult task. Approaches exist that are able to take advantage of developments over time. These are however either not powerful enough to represent complex patterns (e.g. ARIMA, see [10], or do not result in insightful models (e.g. recurrent neural networks). Insightful approaches obviously exist in machine learning domain, but these simply look at the data at a certain time point in isolation. Of course this is not optimal. It is very likely that not only the current or previous time point influences a future health state, but also data from a couple of time points ago. For instance, think of being tired, this is most likely the result of an accumulation of several days with too little sleep.

Unfortunately few papers exist focus on the temporal dimension and derivation of useful features to drive more accurate predictions. Interesting work has been reported in our application domain e-mental health (see e.g. [1], [2], [15]) but these show that making accurate predictions is very difficult. Possibly, more extensive exploitation of the temporal domain can be beneficial to improve the predictive scores. Some generic approaches to handle temporal data have been proposed in the domain of temporal data mining (see e.g. [9]) and granularity computing (e.g. [16]). Studying ways to optimize the temporal features that are extracted has not been done in a very rigorous way: how should we aggregate the historical values (e.g. should we use the mean, a trend, etc.), and what history is considered important? When thinking of the e-health case with datasets originating from widely varying patients different choices might be required per patient to optimize temporal features and thus predictive performance.

In this paper we present a feature learning method that enables the automated identification of suitable temporal features for individual patients. It does so by generating a series of aggregated values from a variable historical window of measurements and it optimizes the window of history used per patient. The optimization criterion is the predictive performance of a model generated based on the identified features. We evaluate the method by applying it on a dataset originating from the EU project E-Compared [5], specifically related to depression. We hypothesize that by the proposed feature identification method described, the predictive performance of a dataset can be increased. We use a case study where we predict the mood of a number of depressed individuals based on information from their past.

This paper is organized as follows. In Section II we will put this work in the broader scientific context. Then, in Section III we describe the method itself. In section IV we shortly describe the case study in which we apply the method. We describe the experimental setup in section V and the results are presented in Section VI. We conclude with a discussion in section VII.

II. Related research

There are three fields which include research relevant to this paper, namely feature learning, temporal data mining, and granular computing.

The field of feature learning, or representation learning, is occupied with the goal of finding optimal representations of data (see e.g. [3] for an overview). Such representations increase the explanatory power of the data and subsequently increase the performance of the machine learning techniques that use this data. The two main techniques, or paradigms if you will, used in this field are probabilistic models and deep learning. Generally, two types of learning can be applied, namely supervised learning, where data representations are evaluated using labeled data, such as neural networks, and unsupervised learning where no targets are available. The method described in this paper is evidently a supervised learning method. Although the goal of our research is completely in line with that of representation learning, we focus more on the temporal aspect of the data and a range of known aggregation functions and optimize parameters settings within that search space.

In temporal data mining many methods try to deal with representing temporal data. In [13] a part has been devoted to time-domain continuous representations. It is suggested to leave the data in its original form, ordered by their instant of occurrence. Another possibility suggested is to use so-called change-point detection, where only data is considered where significant change in behavior occurred. There are also possibilities related to transformation based representations, where the original data is transformed into a new domain, where points in this domain are used to represent the original data. For example transforming time-series data into a frequency domain. Other methods are mentioned, such as discretization based methods, which transform time-series data into discretized sequences, and probabilistic generators, which can identify sub sequences in larger sequences. For more information about these examples, see [13]. Such methods can be very interesting, but do not solve the problem of what is the optimal granularity of features in temporal data.

Another field relevant to our work is called granularity computing, which has its roots in the field of fuzzy logic. Granularity computing uses so-called information granules for the purpose of problem solving (see e.g. [16]). The information granules can be groups, classes or clusters of a universe that are derived from a data representation source. In [11] a classification framework of granular time series is described that assumes a representation of a time series called *feature space* in the data. They build features called *granular feature spaces* which are representations of the original feature space in terms of varying granularity. Next, a granular classifier that uses the *granular feature spaces* for the purpose of classification is applied. Using such a method it is possible to uncover new explanatory power that was not explicitly present in the original representation of the data. When considering the domain of granularity computing (cf. [11]) the area is still open to rigorous investigation. The current study is an example of such an investigation, which includes a novel method, and is applied in the domain of mental health.

III. Method

In this section we explain how the algorithm generates alternative feature representations and how to apply it in a representation learning setting. For the purpose of clarity we distinguish between two types of features, namely *basic features*, which are the original attributes, and *aggregated features*, which are new transformed attributes (e.g. the mean value over a certain interval).

A. Feature generation algorithm

The aim of the algorithm is to generate alternative dataset representations by varying the aggregation intervals of each basic feature in the dataset. Subsequently the activity within the interval of each basic feature is represented in four different ways and therefore lead to new aggregated features. The goal of the process is to find aggregated feature representations that have increased explanatory power compared to the basic features.

Given dataset δ, and target feature τ, for any attribute a, time point t, and window size interval k, we can define an aggregated interval $I(a,t,k) \in \mathbb{N}$, where $I(a,t,k) = t-k, t-(k-1), \ldots, (t-1)$.

In Figure 1 an example is displayed, where e.g. for attribute a_2 a window size interval is set of $k = 3$. Given current timepoint $t = 5$, the aggregated interval of a_2 is based on the information over time points $[2,3,4]$.

We assume that the basic features of time point t are used for prediction of the target feature of time point $t+1$, which is often the case in temporal modeling tasks. The aggregated features therefore shift one time step to the left compared to the target feature.

For any aggregated interval $I(a,t,k)$ we can define the following aggregated features:

$$M(a,t,k) = mean\{v \in I(a,t,k)\}$$
$$Sd(a,t,k) = stde\{v \in I(a,t,k)\}$$
$$S(a,t,k) = sum\{v \in I(a,t,k)\}$$
$$C(a,t,k) = coef\{v \in I(a,t,k)\}$$

$M(a,t,k)$ expresses the average value within the aggregated interval; $Sd(a,t,k)$ expresses the average deviation from the mean within the aggregated interval; $S(a,t,k)$ expresses the

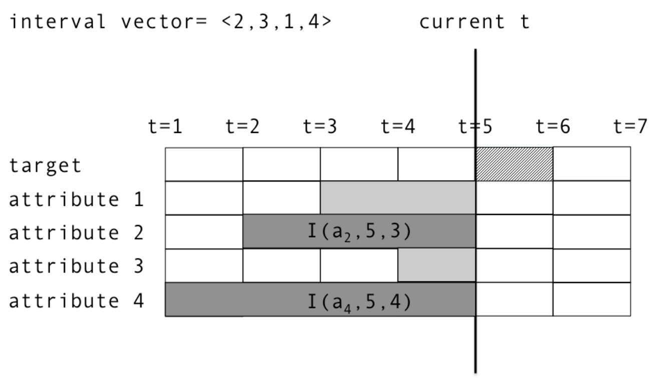

Fig. 1: Example of the process of selecting basic features given a set of aggregation intervals to produce aggregated features. Note that for attribute 2 and 4 the aggregated intervals are specified for exemplary purposes.

total of activity within the aggregated interval; and $C(a, t, k)$ expresses the trend of the activity within the aggregated interval by taking the slope of a linear fit. Note that there are other representations that can be interesting to use, such as with *min* and *max*. For now we will use the aforementioned, but we might add others in future research.

B. *Feature testing strategy*

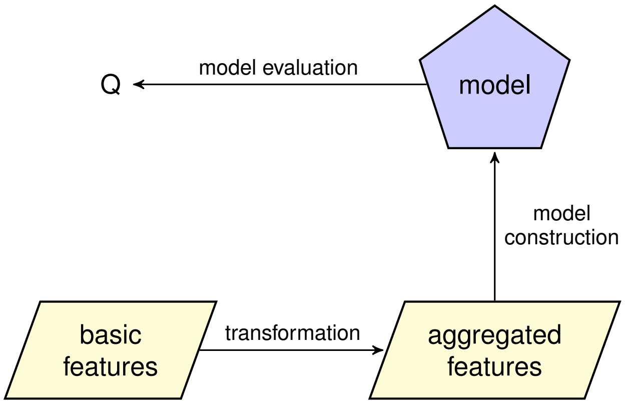

Fig. 2: Overview testing strategy.

As depicted in Figure 2, to evaluate a set of aggregated features a model needs to be constructed using these features. The performance of the model on unseen data can then be used to represent the fitness of the transformed features. Subsequently the challenge is how to find the optimal set of aggregated intervals, which can be solved using well suited technology such as genetic algorithms [8].

IV. Case study

A. *Dataset Description*

We use a dataset from the EU project E-Compared [5]. Specifically, the data consists of ecological momentary assessment measures (EMA, for more information see [14]). For 70 days 49 participants were asked to enter their mood, worry and self-esteem ratings two times a day, and rating related to sleep, activities done, enjoyed activities and social contact once a day on a Likert scale with an interval of [1,10], using an application on their mobile phone. This was part of a trial on blended cognitive behaviour therapy for depression in which psychologist used an online treatment program (available on both smartphone and computer) in conjunction with face-to-face visits. An overview of the EMA measures and their corresponding questions are shown in Table I. These are measured on a regular basis during the therapy. The mood feature is the target feature that needs to be predicted. The EMA data features are expected to have predictive value for describing the mood over time.

TABLE I: EMA measures that are present in the dataset.

Abbreviation	EMA question
Mood	How is your mood right now?
Worry	How much do you worry about things at the moment?
Self-Esteem	How good do you feel about yourself right now?
Sleep	How did you sleep tonight?
Activities done	To what extent have you carried out enjoyable activities today?
Enjoyed activities	How much have you enjoyed the days activities?
Social contact	How much have you been involved in social interactions today?

B. *Exploratory Data Analysis*

Of the total of 13083 questions, 3368 were not answered, and are therefore missing values (25.7%). In Table II an overview is given of the amount of missing data per feature. Mood has the highest number of missing values, but was also one of the questions that was asked most frequently.

In Figure 3 the percentage of missing values over time is depicted. Clearly the further participants came in the trials, the more questions were left unanswered. Because the last 2 days in the dataset show a sharp increase in missing data we decided to exclude this part for further use. In Figure 4 it can be seen that some participants did not actively participate in the trial. We decided not to include the 7 participants that have missing data higher than 60%, resulting in a dataset containing 42 participants with data over 68 days. The missing data that was left in the data was filled by taking the average of the last available data point earlier in time and the first available data point later in time. If only one of these data points was available we used that data point to fill the missing data.

Note that we are considering depressed patients, that show a huge variance in their rating, making the task interesting and challenging. To exemplify this variance, an example of the mood over time of participant 1 is depicted in Figure 5.

V. Experimental setup

In this section, we describe the setup to evaluate our proposed approach using the dataset described above.

TABLE II: Missing data per feature in the dataset.

Feature	% of Missing Data
Mood	31%
Worry	36%
Self-Esteem	36%
Sleep	32%
Activities done	38%
Enjoyed activities	39%
Social contact	39%

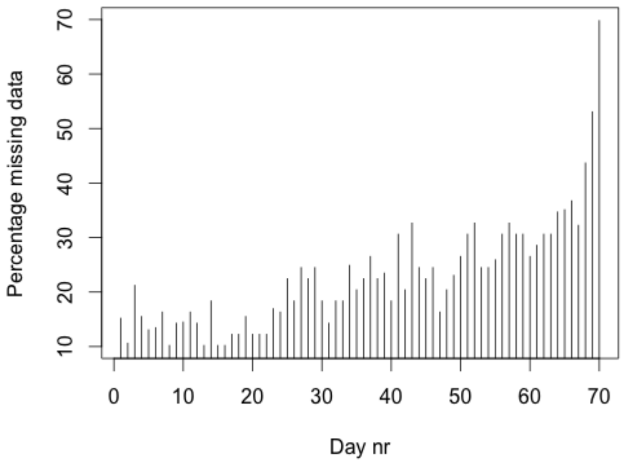

Fig. 3: Percentage of missing data over time.

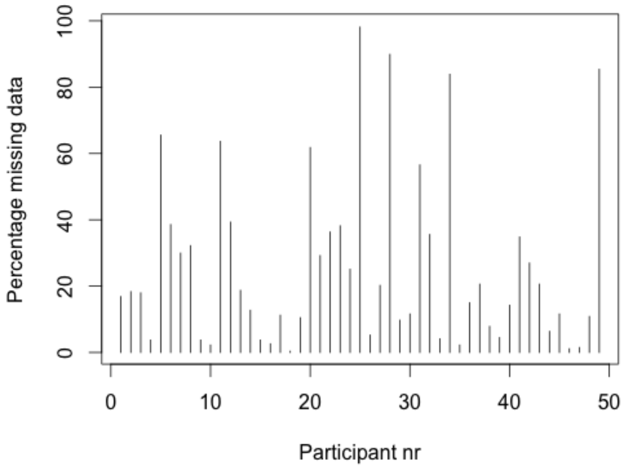

Fig. 4: Percentage of missing data per participant.

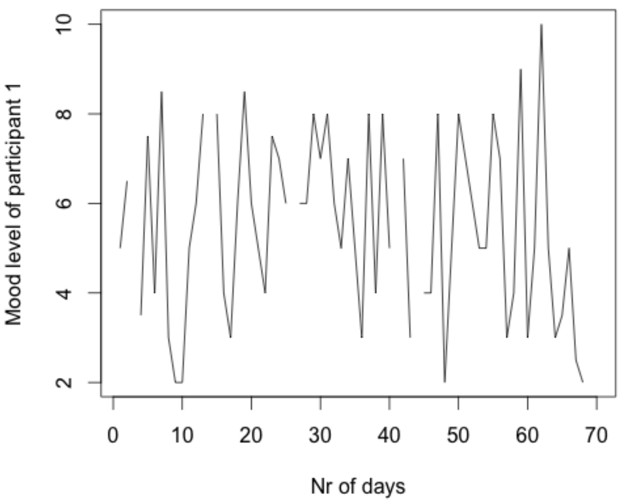

Fig. 5: Mood level of participant 1 over time.

A. General setup

We aim to predict the mood rating at the next time point, and all historical ratings are available to identify features. We identify two conditions, namely our algorithm, referred to as the experimental condition (EC) and a control condition (CC). In the EC we optimize the aggregation intervals per participant, while in the CC we have fixed aggregation intervals over all participants. To be able to compare the results from the CC and the EC we use the following data strategy. For each participant we divide the data into three datasets, namely a training set, a validation set and a test set. We decided to split up the 68 days of data for each participant in 40 days for the training set, 14 days for the validation set and 14 days for the test set. The training set is used by both the CC and the EC to generate models for prediction, where the EC uses the validation set to evaluate the suitability of the aggregated features as part of its optimization algorithm. The test set is put completely aside and is only used to compare the performance of the resulting models in each condition after the optimization has been finished.

The choice for the machine learning algorithm to generate models (given a set of aggregated features) is extremely important. Because we divide the already limited amount of data in three separate parts problems like overfitting are probable to occur. Also, because of the amount of missing data we expect noise in the data, which effect the behaviour and performance of the different conditions, i.e. variance is present. To counter overfitting problems we decided to use a bagging approach (for more info, see e.g. [4]). More specifically, 20 models are fitted using a linear regression approach on 20 random samples of 30 days from the training set. To represent the model fit on the training set an average mean squared error (MSE) over all samples is used. As the bagging approach is stochastic, we ran the bagging approach 10 times. This setup therefore generates 10 averaged MSE's on the training set during the fitting process, and, either 10 related averaged

TABLE III: General setup of conditions per participant.

Condition	Model	Sample Size	Nr Bags	Nr Rounds	Optimized
CC1	LR	30 of 40	20	10	no
CC2	LR	30 of 40	20	10	no
CC3	LR	30 of 40	20	10	no
CC4	LR	30 of 40	20	10	no
EC1	LR	30 of 40	20	10	yes
EC2	LR	30 of 40	20	10	yes

TABLE IV: Conditions specific setup per participant.

Condition	Aggregation Intervals	Solutions per Participant
CC1	$< 1,1,1,1,1,1,1 >$	Fixed
CC2	$< 2,2,2,2,2,2,2 >$	Fixed
CC3	$< 3,3,3,3,3,3,3 >$	Fixed
CC4	$< 4,4,4,4,4,4,4 >$	Fixed
EC1	Random Sampling	Best of 300
EC2	Genetic Algorithm	Best of 300 (10 gen * 30 pop)

MSE's on the validation set, or 10 related averaged MSE's on the test set, during the prediction process. An overview of this setup is displayed in Table III.

We decide to have seven aggregation intervals corresponding to the seven features that can be varied in the experimental conditions. This means that per set of aggregated features types (the mean, the standard deviation, the sum and the slope) there is one aggregation interval. We assume that this level of flexibility should be enough to make a difference between the experimental conditions and control conditions.

For comparing the different conditions we compare each participant's 10 MSE's on the training, validation and test sets. Note that the validation set is not used by the control conditions, but their scores on the validation set might still provide insight, which we will describe in Section VI.

For the CC with fixed aggregation intervals, we select different settings for the intervals. The options are displayed in Table IV. For each CC subsequent we increase the fixed aggregation interval by one day over all features. This way we can also get a general impression about how these transformations influence (or deteriorate) the predictive performance.

B. *Experimental conditions per participant*

As displayed in Table IV, we select two ECs which use an optimization method to find high potential aggregation intervals. For EC1 we use a random sampling method, by generating 300 random solutions per participant. The bandwidth of the random solutions is [1,2,3,4]. We purposely chose to keep the scale small to decrease the chance of overfitting. For EC2 we run a genetic algorithm (see e.g. [8]) using the R package GA [12] for 10 generations with a population size of 30, with a crossover probability of 0.8 and a mutation probability of 0.3. We use a binary representation for the aggregation vectors of length 4. This means that the aggregation interval also has a bandwidth of [1,2,3,4]. For each EC and each participant we select the aggregation interval vector that generates the lowest MSE on the validation set, after which it is used to predict the mood on the test set.

TABLE V: The MSE and SD scores of the control conditions and the experimental conditions on the training set, validation set and test set. The prediction scores displayed here are the averaged over the 10 rounds per participant and averaged over participants.

Condition	Training Set		Validation Set		Test Set	
	MSE	SD	MSE	SD	MSE	SD
CC1	1.30	0.05	6.09	0.73	8.18	1.09
CC2	0.91	0.05	7.25	1.29	9.82	1.87
CC3	0.85	0.04	8.98	1.17	9.75	1.80
CC4	0.78	0.04	10.17	1.36	10.79	2.07
EC1	0.94	0.05	2.22	0.32	6.68	1.40
EC2	0.93	0.05	2.15	0.31	7.60	1.38

VI. Results

In this section we describe the results given the experimental setup described. First, we will study the control and experimental conditions separately, followed a comparison between the two.

A. *Within control conditions*

As described in Section V we have four control conditions for each participant, namely CC1 to CC4 with four setups of fixed aggregation intervals over all participants. As can be seen in the top four rows of Table V the MSE on the test set generally gets worse as the window size increases. The same trend can be seen for the MSE of the CC on the validation set. On the training set the opposite seems to hold: increasing window sizes are better, apparently the lengthier windows tend to overfit the training data more.

When considering the standard deviation (SD) of the MSE scores shown in the same table, the variance in the predictions is quite low, indicating the prediction quality is robust. The bagging approach likely contributed substantially to the found low variance.

In Figure 6 all MSE prediction scores for each of the CC are displayed. For the purpose of comparison the results are sorted by MSE. CC1 especially excels in generating improved low error predictions. For the harder predictions CC1 does not seem to generate better results compared to the other fixed window settings.

B. *Within experimental conditions*

As described in Section V we have two experimental conditions, namely EC1 where we use a random sampling optimizer, and EC2 where we use a genetic algorithm optimizer to find the best predictions on the validation set. In Table V EC1 on average performs better than EC2 on the test set. On the training set and the validation set EC2 seems to have a slightly lower MSE. Again, a low SD is seen. When we look at Figure 7 we see that EC1 specifically performs better on the prediction problems with higher errors, which explains the relatively large difference in mean performance compared to the EC2.

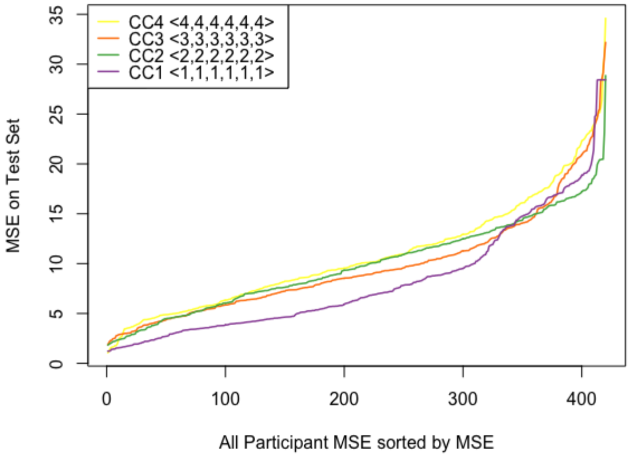

Fig. 6: The MSE prediction scores (42 participants, 10 rounds) of the control conditions on the test set. The scores are sorted by MSE.

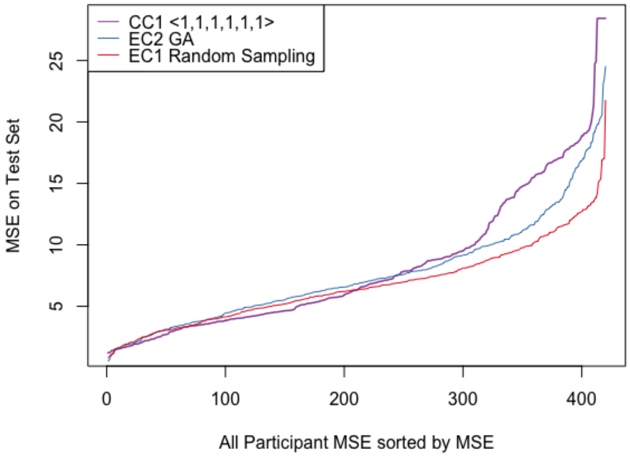

Fig. 7: The MSE prediction scores (42 participants, 10 rounds) of the experimental conditions EC1 and EC2 and the best performing control condition CC1 on the test set. The scores are sorted by MSE.

To see why random sampling performs better, let us consider the evolution of the fitness value with the number of generations shown in Figure 8. We do see that the algorithm seems to converge very fast. Better parameter settings would likely have resulted in better performance. Due to the required computation time, and the fact that this is not the main contribution of the paper, we decided not to optimize the setting further.

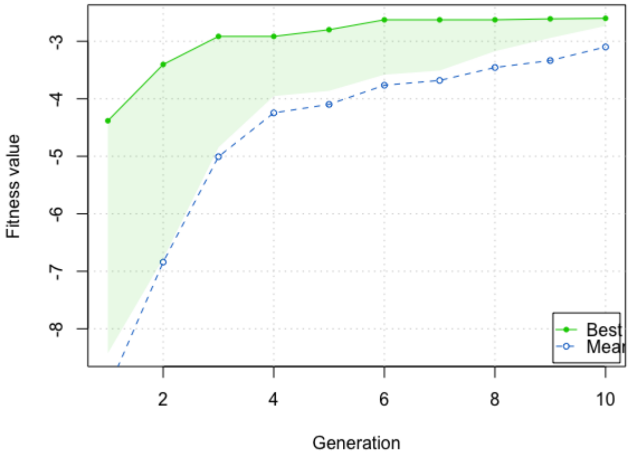

Fig. 8: The evolution of the population fitness for 10 generations within EC2. Example is taken for participant 30. The fitness is expressed in -MSE, because this package only maximises fitness.

In Figure 9 the average and standard deviation of optimal aggregated intervals can be seen for each feature generated by each of the experimental conditions. The average of optimal aggregated intervals that are found for the features seem to be between two and three days for both conditions.

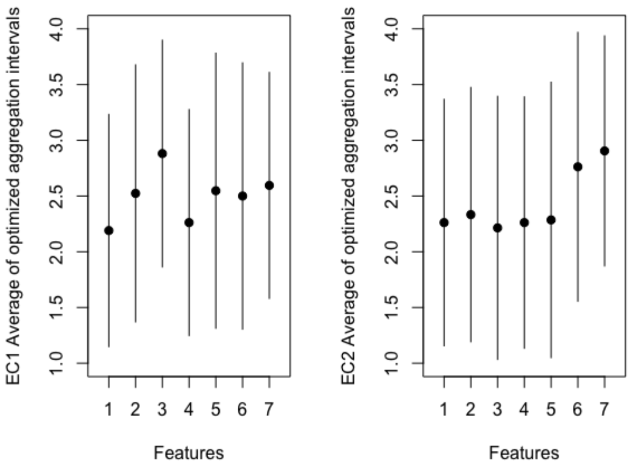

Fig. 9: The average and standard deviation of the optimal aggregation intervals per feature that is found by EC1 (left) and EC2 (right). The features are sleep, worry, self-esteem, mood, enjoyed activities, social contact, and activities done, respectively.

C. Control conditions versus experimental conditions

We want to know if better predictions are generated in the EC compared to the CC given all predictions of all participants (i.e. 42 participants times 10 repetitions). When we compare the scores, the EC1 condition has the best accuracy on the

TABLE VI: The one-sample Kolmogorov-Smirnov Test comparing the condition test set performances using the whole set of predictions of all rounds of all participants, i.e. comparing 420 MSE prediction scores per condition. Displayed is the p-value between each condition, where p-values are rounded if smaller than 1e-06. To have significant difference between conditions we need to satisfy $p - value < 0.05$.

	CC1	CC2	CC3	CC4
EC1	< **1e-06**	< **1e-06**	< **1e-06**	< **1e-06**
EC2	**0.006486**	< **1e-06**	< **1e-06**	< **1e-06**

TABLE VII: The one-sample Kolmogorov-Smirnov Test comparing the condition test set performances per individual set of predictions, i.e. comparing 10 MSE prediction scores for 42 participants. Displayed is the number of cases that for a participant $p - value < 0.05$ between conditions.

	CC1	CC2	CC3	CC4
EC1	**17/42**	**24/42**	**24/42**	**25/42**
EC2	**19/42**	**23/42**	**18/42**	**24/42**

test set, following by EC2. In Table VI the results of a one-sample Kolmogorov-Smirnov test (see [7]) are displayed that compare all predictions of all participants. The results indicate that the differences between the EC and CC are significant. This finding is interesting, but does not yet shed light on the differences between EC and CC on a patient level.

In Table VII one-sample Kolmogorov-Smirnov tests are conducted for each participant. The results show that in 17 to 19 of the 42 participants EC1 and EC2 generated significantly better results than CC1 given an alpha of 0.05.

For CC2 to CC4 the number of significant differences generally increase. The fact that not more significant differences are found is most likely due to the small sample size. Other factors that play a role are the quality of the data and amount of data available per individual.

Based on the results we can conclude that the sets of aggregated features found by the EC have higher predictive capabilities.

D. Individual example predictions

Next to the higher level comparisons between conditions, it is interesting to look at the implications on a practical level. For this purpose we compare CC1 and EC2 for participant 16 and participant 36. These are representative examples. The fit on the training set, the prediction on the validation set and the prediction on the test set for participant 16 are depicted in Figure 10, and for participant 36 are depicted in Figure 11. The examples show that the predictions for the independent test set are quite reasonable for the EC2, especially considering the fact that the problem we are facing in general is known to be notoriously difficult. For the CC1 the models seem to describe the trends less well. A comparison on the validation set is not fair as the EC2 exploits the performance on the validation set to optimize the fitness values.

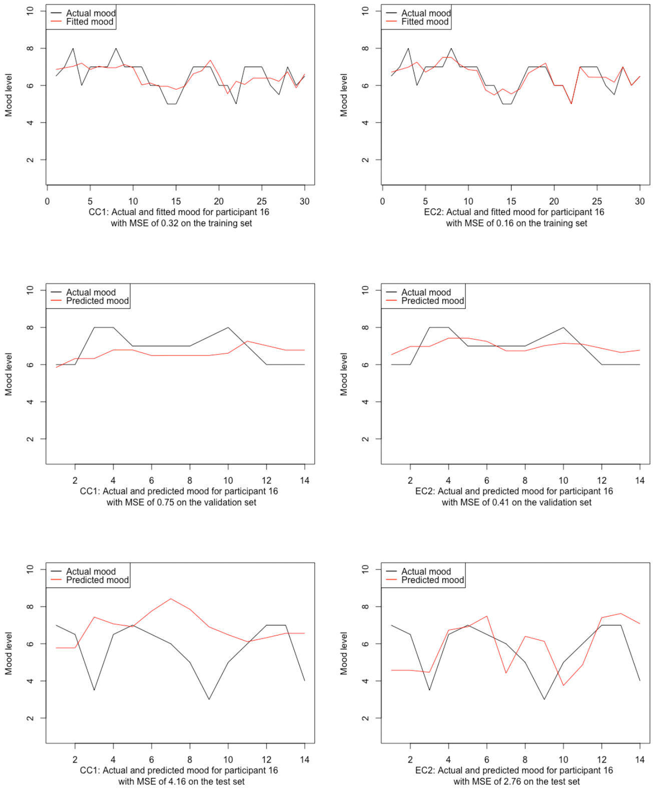

Fig. 10: The fit on the training set, prediction on the validation set, and the prediction on the test set (top to bottom) of CC1 (left) and EC2 (right) for participant 16. Specifically the models in the 6th round generated these fits and predictions.

VII. Discussion

In this paper we proposed and tested a feature learning method for temporal predictive models. We have evaluated the approach using data from the domain of e-health, specifically depression. Significant differences were found on the prediction task between each of the CC and EC: the EC outperforms the CC.

The feature learning method automates a part of the preprocessing stage when using temporal data. In each prediction task many decisions need to be made about how to prepare the data that is fed to the predictive model. Among such choices is the decision about the time window to consider for temporal attributes. Often it is unclear what the right time windows is. Data practitioners therefore go with their intuition. Also, the emphasis is often on other parts of the prediction process, such as which model is best suited. By automating this preprocessing using the proposed method in combination with an optimizer, better representations can be generated, that increase the predictive accuracy.

The method is well suited to be employed in the e-health domain, because new technologies are used that measure a flurry of information in a temporal fashion, such as information related to mental health, physical health, or geographic infor-

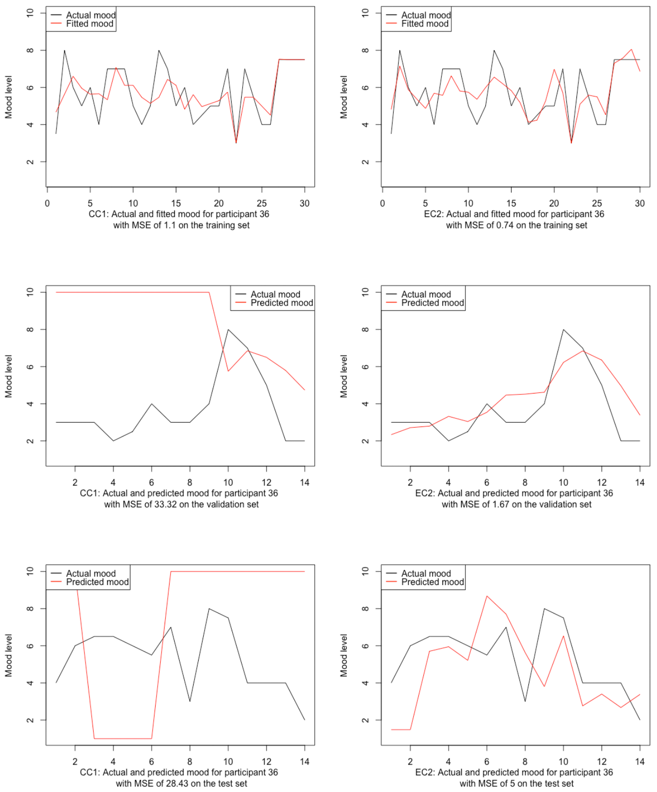

Fig. 11: The fit on the training set, prediction on the validation set, and the prediction on the test set (top to bottom) of CC1 (left) and EC2 (right) for participant 36. Specifically the models in the 2th round generated these fits and predictions.

mation. Little research is done about how such features relate to the target of the prediction. Also, such relations are often highly personal and do not generalize well across patients. Therefore, it is interesting to exploit the proposed method which includes personalization of features.

Of course, the quality of the data and the amount of missing values severely impact the eventual results in terms of predictions. In this case we suffering from a lack of data and ample missing values. This problem even became more severe because we had to divide the data for each participant into three datasets, namely a training-, validation and test set. However, to obtain a proper and solid analysis we did consider it needed. We sampled 30 of the 40 data points of the training set to fit linear regression models on, which is very little for this purpose. In our case, to counter the effects of overfitting, i.e. fitting to noise, we chose to use a bagging approach. In any case, for the purpose of validating the performance of the proposed method, it would be interesting to apply it on more data with higher quality.

Within our experimental conditions we have seen that the genetic algorithm was not able to go through the search space very effectively as random sampling worked better. On forehand we expected the genetic algorithm to outperform the random sampling method. We expect this to be caused either by the lack of parameter optimization caused by the severe computation cost, or by the shape of the fitness landscape.

For future work we would like to apply the method to more data. More data will shed further insight in the added value the method can generate. Also, we want to look more at the aggregated feature types that are generated and how they effect the performance. In this experiment we generated the mean, standard deviation, sum and slope for each basic feature. Possibly other types are interesting as well such as minimum and maximum. Ultimately the aggregation types that effect the predictive power the most, will vary depending on the problem that needs to be solved. From that perspective it might be best to add more aggregation types. Furthermore, we want to increase the parameters, so that each aggregation type has its own aggregation interval. Finally, we would like to explore whether we can improve the optimization algorithms beyond their current performance, and compare the experimental conditions with control conditions that use more training data, to explore if and when the additional training data in the control conditions outperform the experimental conditions.

ACKNOWLEDGMENT

This research has been performed in the context of the EU FP7 project E-COMPARED (project number 603098).

REFERENCES

[1] J. Asselbergs, J. Ruwaard, M. Ejdys, N. Schrader, M. Sijbrandij, and H. Riper. Mobile phone-based unobtrusive ecological momentary assessment of day-to-day mood: An explorative study. *Journal of medical Internet research*, 18(3), 2016.

[2] D. Becker, V. Bremer, B. Funk, J. Asselsbergs, H. Riper, and R. Jeroen. How to Predict Mood? Delving into Features of Smartphone-Based Data. *Americas Conference on Information Systems! AMCIS 2016: San Diego*, 2016.

[3] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[4] L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[5] E-Compared. E-compared. european comparative effectiveness research on internet-based depression treatment, 2016.

[6] G. Eysenbach. What is e-health? *Journal of medical Internet research*, 3(2):e20, 2001.

[7] F. J. Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78, 1951.

[8] M. Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.

[9] T. Mitsa. *Temporal data mining*. CRC Press, 2010.

[10] J. Pastor and W. van Breda. Analyzing and predicting mood of depression patients. 2015.

[11] W. Pedrycz. *Granular computing: analysis and design of intelligent systems*. CRC press, 2013.

[12] L. Scrucca. Ga: a package for genetic algorithms in r. *Journal of Statistical Software*, 53(4):1–37, 2013.

[13] M. Shahnawaz, A. Ranjan, and M. Danish. Temporal data mining: an overview. *International Journal of Engineering and Advanced Technology*, 1(1):2249–8958, 2011.

[14] S. Shiffman, A. A. Stone, and M. R. Hufford. Ecological momentary assessment. *Annu. Rev. Clin. Psychol.*, 4:1–32, 2008.

[15] W. van Breda, J. Pastor, M. Hoogendoorn, J. Ruwaard, J. Asselbergs, and H. Riper. Exploring and comparing machine learning approaches for predicting mood over time. In *Innovation in Medicine and Healthcare 2016*, pages 37–47. Springer, 2016.

[16] Y. Yao. Granular computing: basic issues and possible solutions. In *Proceedings of the 5th joint conference on information sciences*, volume 1, pages 186–189. Citeseer, 2000.