Feeling Moody - A Time Series Analysis to Predict Mood Based on Smartphone Usage

Valentin Buchner (2647413), Paola Feil (2732911), Matilda Knierim (2700374)

Vrije Universiteit Amsterdam

1 Introduction

Mental health studies mostly rely on self-report questionnaires, which tend to be highly biased. Most participants fill out questionnaires regarding their inner state with a tendency of recall bias. In other words, the memory about a certain state or situation is deviating from reality. One method that tries to overcome these biases and make psychological data more reliable is the ecological momentary assessment (EMA), in which psychological phenomena are assessed in a person's natural environment. EMA does not dispense on self-reports completely, but tries to tackle the recall bias by collecting data multiple times a day and specifically asking about a recent time-frame (e.g. the previous 30 minutes). With this method, the state is still in fresh memory and can be reported reliably. However, EMA does not reduce other biases such as social desirability. One psychological state that can be investigated using EMA is mood, as was done by [1]. The authors investigated the influence of smartphone usage on mood with the use of EMA and an app which tracked different variables (such as app usage, screen time, calls, etc.). Given that mood fluctuates during the day and is dependent on different environmental and psychological stimuli, understanding what influences mood is crucial to determine mental health and general well-being. Moreover, understanding the influence that smartphone usage has on mood could be used by tech firms in order to advise their users a healthier smartphone usage. This research aims to investigate the influence of smartphone usage factors on mood based on data assessed with EMA (and smartphone data). Therefore, with a machine learning based approach.

2 Domain Understanding

As previously mentioned, [1] successfully used the EMA method and a self-developed app that collects the usage time of different apps and assesses mood. In their study, the mood and smartphone usage data of 27 Dutch students was assessed over the course of six weeks. Several variables regarding smartphone usages were assessed next to mood. All variables measure in the study and their explanation are displayed in Table 1.

Even though the authors described their results as "sobering", their method of data collection and therefore the assessed data seem to be of high quality.

Variable	Explanation	Mean	SD
mood	The mood scored by the user on a scale of 1-10	6.99	1.03
circumplex.arousal	The arousal scored by the user, on a scale between -2 to 2	-0.1	1.05
circumplex.valence	The valence scored by the user, on a scale between -2 to 2	0.69	0.67
activity	Activity score of the user (number between 0 and 1)	0.12	0.19
screen	Duration of screen activity (time)	75.34	253.82
call	Call made (indicated by a 1)	1.0	0
sms	SMS sent (indicated by a 1)	1.0	0
appCat.builtin	Duration of usage of builtin apps (time)	18.54	415.99
appCat.communication	Duration of usage of communication apps (time)	43.34	128.91
appCat.entertainment	Duration of usage of entertainment apps (time)	37.58	262.96
appCat.finance	Duration of usage of finance apps (time)	21.76	39.22
appCat.game	Duration of usage of game apps (time)	128.39	327.15
appCat.office	Duration of usage of office apps (time)	22.58	449.60
appCat.other	Duration of usage of other apps (time)	25.81	112.78
appCat.social	Duration of usage of social apps (time)	72.40	261.55
appCat.travel	Duration of usage of travel apps (time)	45.73	246.11
appCat.unknown	Duration of usage of unknown apps (time)	45.55	119.40
appCat.utilities	Duration of usage of utilities apps (time)	18.54	60.96
appCat.weather	Duration of usage of weather apps (time)	20.15	24.94

Table 1. Variable Descrpitives

Furthermore, previous research investigating mood highlights influential variables. Screen time was repeatedly found to have a significant association with mood and mental health [2][5]. Especially evening screen time was highlighted to significantly reducing sleep time and daytime vigilance [5]. Based on these findings, this research extends the variables measured by [1] and adds the variables day and evening screen time. Given the effectiveness of EMA, this research re-investigates the data provided by [1]. Moreover, we take two machine learning approaches into account, the random forest algorithm and the long short-term memory (LSTM) model.

3 Data Exploration

The data was provided in a CSV file consisting of participant ID, the timestamp of the assessment, the measured variable, and the recorded value. All included variables can be found in Table 1.

First, to gain a better understanding of the data, descriptive statistics of the variables, their distributions and correlations, as well as the occurrence of influence points and missing values were investigated. The descriptive analysis demonstrated that all variables mentioned in the project description were present in the provided data set. In total, there are records for 27 participants.

Feature Distributions Concerning the variables' distributions, mood was approximately normal, all other variables had non-normal distributions (mostly Poisson). This should not pose an issue, since the chosen predictive models do not require normality of the data, and therefore no transformations are necessary.

Collinearity Regarding pairwise correlations between features, several significant associations were found, of which some have a high strength. Yet, this should not be problematic, since the random forest is more robust to such collinearity than for example linear regression models. This is because the individual trees of the random forest are built on bootstrap samples of the training data and potentially do not include all features. For the LSTM model, a Principal Component Analysis will be conducted as a mean of feature reduction.

Missing Value Detection Moreover, several missing value entries were found. Amongst the *appCat* variables, *call*, *sms*, and *screen* these seem to represent the absence of usage of these phone features rather than a missing value, since no values for these variables were actually set to zero seconds. Yet, several missing values were detected amongst the arousal (N=46) and valence (N=156) variables which represent the absence of measurements.

Influence Points Regarding influence points, several were found in the data and each potential influence point was investigated separately. As a result, the minimum (i.e. -82799) of the variable *appCat.builtin* and two other negative values were deleted given negative seconds cannot be a valid measurement. Further, it was analysed whether all other influence points come from the same participant to exclude the possibility of a measurement malfunction. We concluded that this was not the case and kept all other influence points due to their potential predictive importance for mood.

Mood Trend and Autocorrelation As typical for time series analysis, it was investigated whether mood follows a certain trend over time. Yet, this was not the case. Furthermore, the autocorrelation of mood was examined to find a meaningful time interval/lag for restructuring the data into a supervised learning problem. Upon closer investigation of autocorrelation per participant, it became clear that the optimal time lag would differ per person. Therefore, the average autocorrelation was computed across participants. The highest score resulted from a time lag of one day (score: 0.24), yet, this would not allow to treat this as a time-series problem (as given by the intructions). Therefore, a time-lag of two days was chosen.

After the descriptive analysis, the data was pre-processed for the application of the random forest algorithm and the LSTM.

4 Data Preparation

4.1 Data Aggregation to Day-Level

To prepare the data for further processing, it was divided into one data frame for each participant whilst also dropping the ID column. Subsequently, for all resulting data frames, the data was resorted so that each row represents all measured values of the corresponding participant at one time stamp. Finally, the data for each measured variable was aggregated to day level. Doing so, the values of *app.Cat* variables, *call*, *sms*, *screen*, and *activity* were summed up as they constitute counts of seconds, occasions, or intensity of acceleration. Further, the number of observations of *app.Cat* variables, *screen*, and *activity* were counted, since the number of times someone uses their phone or moves might have a different effect than how long/intense someone uses their phone or moves. For the variables *mood*, *circumplex.arousal*, and *circumplex.valence* values of the same day were averaged since these variables only have a certain range of valid values.

4.2 Additional Features

The interaction of arousal and valence is assumed to be a potentially valid predictor of mood. Thus, their interaction was added as a feature to the used models. Moreover, as mentioned in the section Domain Understanding, four features were added by distinguishing evening screen time (i.e. 9pm-12am) and day screen time, and the count of evening and day screen usages. Including these aggregations, a total of 28 variables was reached.

4.3 Handing Invalid and Missing Values

As the data exploration showed one negative and therefore invalid measurement in *app.Cat.builtin*, this value was removed. The missing values amongst the *app.Cat* variables, were replaced with zeros, since a time seemed to have only been entered when the app was used. Therefore, these were interpreted as no usage of the corresponding app rather than missing measurements. Missing values amongst mood, valence and arousal were handled for each participant individually in the following way: All missing values occurring before the first measurement and missing values existing after the last measurement were deleted. If a value at day t between two valid measurements was missing, the average of this variable at *t-1* and *t+1* was taken. If two values on consecutive days t and *t+1* were missing, the average of the variable at *t-1* and *t+2* was entered at t. Subsequently, the average of this result and *t+2* was entered for *t+1*. If more than two consecutive values at time t and $t+n$ were missing between two valid measurements for a variable, the data for this participant was split into two data frames, the first ending at *t-1* and the second beginning at *t+n+1*.

4.4 Principal Component Analysis

As high correlations were observed between multiple sets of features, it was decided do conduct feature reduction by means of PCA [4]. PCA was conducted on all input features except for the variable mood as mood might be the most relevant feature, and the variability in mood should therefore be preserved. First, PCA was conducted with the maximum number of components, and the Eigenvalues of all components were plotted in a scree plot, shown in Figure 1.

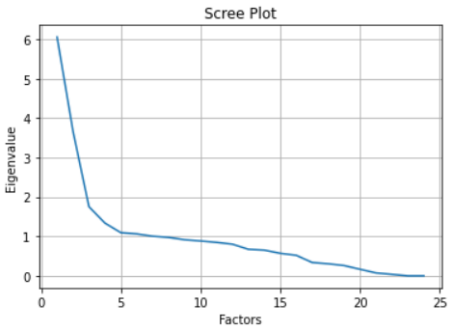

Fig. 1. Scree plot of Eigenvalues

This plot indicates that the first five components explain the most variance in the data. However, as it was noted that these only account for an accumulative variance of 57%, it was decided to include the first ten components, which together explain an accumulative variance of 78%. The original data features were converted to a matrix which was multiplied with the factor loading matrix of the first 10 components to create a new set of 10 features for each observed day and participant. Subsequently, mood was manually added to the 10 features, resulting in each day being represented as a 11-dimensional vector. This 11 dimensional data was further used for fine-tuning and training the LSTM model, yet not for the random forest. Since a relevant benefit of the random forest model is its ability to explain feature importance, it can be considered beneficial to keep the original features when training a random forest. Using the principle components for a random forest would result in losing the advantage of explainability, since it is effortful to reconstruct which variables contribute to a principle component. However, this does not constitute a problem for a LSTM model, since a LSTM model is neither able to explain feature importance when using the original features.

4.5 Data Scaling

Since unscaled data can result in unstable learning of a neural network, the data resulting from the previous step was scaled using min-max normalization, resulting in all values being in the range *[0, 1]*. This was applied to the 11-dimensional

data containing the 10 principle components and mood. As random forests are not sensitive to feature scaling [4], this step was not considered necessary for the original features used to train the random forest.

4.6 Data Restructuring for Supervised Learning

Each participant's data was restructured into two different formats, so that it could be used for a supervised learning task by both, a traditional machine learning algorithm and one that can handle time series data.

Non-Aggregated Format To prepare the data for the LSTM model, each participant's data was restructured so that all feature values, including *mood*, would be grouped together for two-day time intervals, thus constituting one input. The corresponding output would then be the value for *mood* on the next day following the two-day interval. Figure 2 shows a mock-up of how this was implemented for one participant. After each participant's data was restructured, their resulting arrays for the input and output were concatenated vertically.

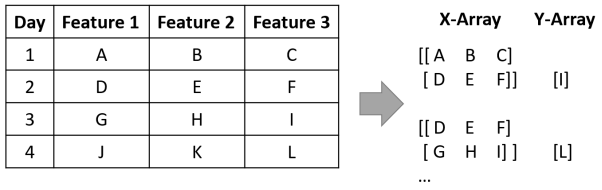

Fig. 2. Data Restructuring for LSTM

Aggregated Format For the random forest algorithm, the data was restructured so that for each two-day time period, feature values would be averaged, including mood. The corresponding output value for these averages is then the measured value for mood on the following day.

Train-Test Set Splitting After restructuring the data, we ended up with 1152 input-output pairs. As the data set was reduced due to the amount of missing values, it was chosen to include 25% of the data in the test set. This ensures sufficient training examples for a machine learning algorithm, while still maintaining an appropriate size for a test set. For hyper-parameter tuning, the training data was also used for five-fold cross-validation in both the random forest and the LSTM model.

5 Data Modelling

To predict the mood score on a given day, two general models were built, with the aim to generalise across all participants rather than training personalised models. Their performances are then compared to that of a simple benchmark model.

5.1 Benchmark Model

To find a baseline for the quality of predictions of the two designed models, a benchmark was implemented. The benchmark model predicts the mood score for any day t by equaling it to the score of the previous day: $mood(t) = mood(t-1)$.

5.2 The Random Forest Algorithm

Model Description To analyse the aggregated version of the data, a random forest regression model was implemented. A random forest is an ensemble machine learning algorithm which builds a collection of decision trees that randomly differ from each other, and eventually averages across the predictions of all trees[4]. The randomness between trees is generated by building each tree on a different bootstrap sample of the data and by choosing random feature subsets for each split. In general, tree-based models bear the advantage of higher explainability of predictions [4]. While a random forest cannot be as easily visualised as decision-trees, the importance of all features for the final prediction of mood can be analysed. This provides more interesting insights compared to the 'black box' nature of many other algorithms. Furthermore, random forests are relatively easy to set up as they tend to work well without heavy parameter tuning and do not require extensive scaling of the data [4]. Finally, since we aim to generalise across people with possibly slightly different mood patterns, a random forest is a suitable choice as it counteracts the tendency of decision trees to overfit by averaging the predictions of the ensemble's trees.

Implementation and Tuning The algorithm was implemented using the *RandomForestRegressor* of the *sklearn* python library. In total, 1000 trees were included in the ensemble. For each split in a tree, the feature selection is based on variance reduction (i.e. parameter *criterion="squared_error"*). The hyperparameters max_features (i.e. the maximum size of the feature subset considered for each split) and max_depth (i.e. the maximum allowed depth of each tree) were then tuned on the training set by applying a grid search with cross-validation including a total of five folds. The former parameter regulates the similarity of the trees with higher values leading to less diverse trees and lower values requiring deeper trees to fit the data well [4]. As the rule of thumb for regression suggests to use the log2 of the number of features, the tested values included following values rounded to integers: [log2(n_features)-3, log2(n_features)-2, ..., log2(n_features)+2, log2(n_features)+3, 10, 15, 20, 25]. The depth parameter was tuned to limit overfitting of each tree. Tested values for this parameter included values from 5 to 205 in a step size of 10, and additionally 250 and 300. The evaluation of the resulting models was based on the R^2 score achieved on the test sets within the folds as this metric's interpretation is more intuitive than the mean-squared-error or mean-absolute-error. The highest scoring model was found using the parameters max_features=log2(n_features)+2 and max_depth=105, which led to an R^2 score of 0.21. However, running the model with its default values where max_features is set to the number of features and

not specifying a maximum depth led to a higher performance with an R^2 score of 0.36. Therefore, the random forest was trained with these parameters on the whole training set.

5.3 The LSTM model

Model Description Artificial recurrent neural networks (RNN) are able to process time-series data by storing the activation of the previous input in a hidden state which is used to calculate the activation of the following inputs. LSTM models constitute a specific architecture of RNN models which contain an additional forget state. This forget state is used to select which input should be remembered in processing the following inputs, and therefore allows the LSTM model to forget irrelevant information. For the purpose of predicting the average mood of day *t+1*, a *tensorflow* LSTM model was trained using data from day *t* and *t-1*. Because the target variable mood is measured on a scale between 1 and 10, all target values were divided by 10, so that they fall within the range *[0, 1]*. The resulting target values m were aggregated to arrays defined as *[1-m, m]*, such that $sum([1-m, m]) = 1$. The array values can also be seen as defining the probability of being in a *[bad, good]* mood. The LSTM model was constructed using one LSTM layer of rectified linear units (ReLU), one LSTM layer of hpyerbolic tangent units (tanh), and one dense layer reducing the output to a two-dimensional vector. Finally, a softmax function is applied to the output of the dense layer, converting it to a probability distribution.

Implementation and Tuning Mean square error (MSE) was chosen as an evaluation metric as it is convenient to interpret and well comparable. Therefore, it was also used to calculate the model loss, which was optimized using the Adam Optimization Algorithm [3]. The model was set to run for 500 epochs. Before running the model, 10% of the training data was put aside for validation, and the remaining 90% of the training data was shuffled before each epoch.The *learning rate* and *number of nodes per hidden layers* were fine-tuned using 5-fold cross-validation using only the training data. This resulted in an optimal learning rate of *0.001* and an optimal number of nodes per hidden layer of *100*. For calculating the performance metric on the test set, only the models predicted probability for a good mood was used and was multiplied by 10 to restore the original scoring on the range *[1, 10]*.

6 Evaluation of Results

The performances of the baseline and the two trained models are displayed in Table 2. 95% Bootstrap Confidence Intervals (CIs) were computed for MSE and R^2 using 10000 samples with replacement.

The random forest performed better than the baseline model, as can be seen above. The results highlight that with an R^2 of 0.36, the model seems to be able to predict mood. The most important feature detected by the random forest

Model	MSE	MSE 95% CI	R^2	R^2 95% CI
Baseline	0.850	[0.494, 1.415]	-0.60	[-1.77, 0.023]
Random Forest	0.43	[0.29, 0.657]	0.362	[0.254, 0.424]
LSTM	0.469	[0.336, 0.636]	0.383	[0.180, 0.509]

Table 2. Performance Metrics on Test Set

algorithm is mood. Moreover, arousal, usage of communication apps, usage of entertainment apps, evening screen time and activity also appear to be relatively important. However, the difference to the other variables is only marginally. The reasonable importance of the feature evening screen time confirmed our choice of adding it to the model.

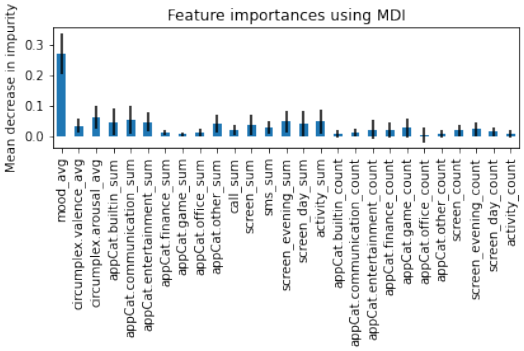

Fig. 3. Variable importance on mood

Similarly, the LSTM model performed better than the baseline model, and its performance can be compared to the Random Forest model. It can be noted that the CIs of both metrics overlap for the random forest and the LSTM model, while this is not the case for the baseline model.

7 Discussion

While both the random forest and the LSTM successfully performed better than the benchmark, this study is not without limitations. Firstly, a lot of the data was missing and therefore had to be imputed, removed or split into several time series. This may well have affected the results of this study. Future research may experiment with different imputation methods than applied in this study. For instance, missing values may be replaced with a K-Nearest-Neighbour algorithm. Secondly, there was not a lot of data per participant, thus a general model rather than a personalised model per participant was created. However, since people have different relationships with their phones as well as varying emotional regulation capabilities, personalised models may be more effective in predicting

mood. Hence, future research projects could collect more data per person and compare whether personalised time series models outperform general models. Thirdly, we applied an LSTM with a time lag of two days. However, the LSTM is particularly suitable for analysing long time sequences. While it was still the best performing model when comparing R^2 values, its full potential could be harnessed when working with longer time lags. Based on participants' individual autocorrelation values, a longer time lag would be recommended than two days. Thus, if future researchers try to build personalised mood prediction models, an LSTM may be still an appropriate choice. Finally, while collinearity amongst features was initially judged as unproblematic due to the randomness induced in the forest model via bootstrapping and feature subset selection, the latter was eventually eliminated from the model. When not choosing a subset of features for each split, the model performed better. Yet, it may be worth investigating whether this performance could be increased when reducing strongly correlated features.

8 Conclusion

This research set out to examine whether people's mood on a given day can be predicted based on self-recorded measures of mood, arousal level, and valence of arousal, as well as measured smart phone usage data. Considering people's measures of the last two days, an LSTM model as well as a more traditional machine learning approach, namely a random forest regressor, were implemented to predict mood on the third day. Subsequently, these models were compared to a benchmark predicting mood to be the same as the day before. The results showed that the LSTM performed the best, yet, not that much better than the random forest which could be considered a simpler and more explainable algorithm when it comes to time series analysis. Under consideration of this study's limitations, one can say that mood on any day is partially explained by a person's mood and also phone usage on the previous two days. For the practical deployment of these models, it is recommended to use the LSTM if the focus lies on performance, while the random forest should be used if more transparency of each measure's contribution is required.

9 Contributions

- **Valentin**:
- **Paola**: Main tasks: Research of implemented machine learning models. Analysis of autocorrelation and data restructuring. Implementation of random forest algorithm. Writing of the report.
- **Matilda**: Analysis of the descriptive values for each variables and analysing all the plots for each variable and person; Pre-processing of the data set; Implementation of the random forest algorithm; Writing the report

References

1. Asselbergs, J., Ruwaard, J., Ejdys, M., Schrader, N., Sijbrandij, M., Riper, H.: Mobile phone-based unobtrusive ecological momentary assessment of day-to-day mood: An explorative study. Journal of Medical Internet Research **18**(3), e72 (2016). https://doi.org/10.2196/jmir.5505
2. Babic, M.J., Smith, J.J., Morgan, P.J., Eather, N., Plotnikoff, R.C., Lubans, D.R.: Longitudinal associations between changes in screen-time and mental health outcomes in adolescents. Mental Health and Physical Activity **12**, 124–131 (2017). https://doi.org/10.1016/j.mhpa.2017.04.001
3. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
4. Müller, A., Guido, S.: Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly Media, 1 edn. (2016)
5. Perrault, A.A., Bayer, L., Peuvrier, M., Afyouni, A., Ghisletta, P., Brockmann, C., Spiridon, M., Hulo Vesely, S., Haller, D.M., Pichon, S., Perrig, S., Schwartz, S., Sterpenich, V.: Reducing the use of screen electronic devices in the evening is associated with improved sleep and daytime vigilance in adolescents. Sleep **42**(9) (2019). https://doi.org/10.1093/sleep/zsz125