Using Autoencoders to Automatically Extract Mobility Features for Predicting Depressive States

ABHINAV MEHROTRA, University College London, United Kingdom
MIRCO MUSOLESI, University College London and The Alan Turing Institute, United Kingdom

Recent studies have shown the potential of exploiting GPS data for passively inferring people's mental health conditions. However, feature extraction for characterizing human mobility remains a heuristic process that relies on the domain knowledge of the condition under consideration. Moreover, we do not have guarantees that these “hand-crafted" metrics are able to effectively capture mobility behavior of users. Indeed, informative emerging patterns in the data might not be characterized by them. This is also a complex and often time-consuming task, since it usually consists of a lengthy trial-and-error process.

In this paper, we investigate the potential of using autoencoders for automatically extracting features from the raw input data. Through a series of experiments we show the effectiveness of autoencoder-based features for predicting depressive states of individuals compared to “hand-crafted" ones. Our results show that automatically extracted features lead to an improvement of the performance of the prediction models, while, at the same time, reducing the complexity of the feature design task. Moreover, through an extensive experimental performance analysis, we demonstrate the optimal configuration of the key parameters at the basis of the proposed approach.

CCS Concepts: • **Human-centered computing** → *HCI design and evaluation methods*; *Empirical studies in ubiquitous and mobile computing*;

Additional Key Words and Phrases: Mobile Sensing, Notifications, Application Usage, Context-aware Computing

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

1 INTRODUCTION

Depression is the most common mental health condition. More than 300 million people suffer from depression worldwide [1]. Depression has also a strong impact not only on the life of the individuals affected by it, but also on their families and social circles. It can also have a severe negative impact on work and school performance and, therefore, a non-negligible economic cost is associated to it. Moreover, depression at its worse could lead a person to suicide. Indeed, in high-income countries mental disorders, and especially depression, are one of the major causes of suicide [35]. The World Health Organization has estimated that by the year 2020 depression will be the second largest cause for lost years of healthy life worldwide [1].

Around the world, only less than half of the people affected by this condition are able to receive effective treatments when necessary. This number is less than 10% in many countries [1]. The key reasons are lack of

Authors' addresses: Abhinav Mehrotra, University College London, Pearson Building, Gower Street, London, WC1E 6BT, United Kingdom, a.mehrotra@ucl.ac.uk; Mirco Musolesi, University College London and The Alan Turing Institute, Pearson Building, Gower Street, London, WC1E 6BT, United Kingdom, m.musolesi@ucl.ac.uk.

2474-9567/2018/9-ART127 $15.00
https://doi.org/10.1145/3264937

resources and inaccurate assessment. Currently, psychologists rely mainly on self-assessment questionnaires (such as PHQ-8 [26] and PHQ-9 [24]) for diagnosing a depressive condition. However, this approach is prone to errors as it often relies on the person's recollection ability [31] and it might be influenced by biased self-representation given the persistent social stigma associated with mental health disorders.

At the same time, interview-based studies have shown that depression leads to a reduction of mobility and activity levels [39]. Moreover, a few recent studies have proposed different approaches to exploit mobility data for monitoring depression [9, 30, 41]. For this reason, we focus on the GPS data, which can be reliably and unobtrusively collected through mobile phones in a robust way, to investigate the potential of exploiting people's mobility characteristics to predict their depressive state. The authors of these studies have proposed a variety of mobility features, such as the total distance covered by individuals, the number of places they visit, their regularity and so on, for characterizing human mobility behavior and predicting people's depressive states. However, these features are in a sense "hard-wired" and pre-defined by researchers, who usually derived them from qualitative observations contained in the existing literature on mental health disorders.

In this paper, we discuss an approach for the *automatic* extraction of mobility features from raw movement data (e.g., GPS traces) without manually engineering them. In our opinion, this is a fundamental problem for a variety of reasons. First of all, we do not have guarantees that the existing features are able to capture effectively mobility behaviour. Indeed, informative emerging patterns in the data might not be captured by them. Moreover, these features are usually dependent on the selection and tuning of sets of parameters. This is a complex and often time-consuming task, since it usually consists of a lengthy trial-and-error process. It might also be strongly dependent on the population under observation.

More specifically, we investigate the use of *unsupervised deep autoencoders* [7, 16, 17] for automatic extraction of mobility features. We then examine their potential for predicting depressive states measured using standard psychological tests. An autoencoder is an artificial neural network that is designed with the goal of copying its inputs to its outputs in an efficient way. By doing so, the underlying network is able to learn useful properties of the data, which can then be used for example for efficient coding. Researchers and practitioners in a variety of domains, including, for example, activity recognition [37], computer vision [23], and speech recognition [29] have exploited autoencoders to extract novel features from raw data. However, this powerful technique has not been applied to the analysis of mobility data yet.

In order to validate our approach, we use the Trajectories of Depression dataset collected by the authors of [9], who presented a set of "hard-wired" features for monitoring depression. The results of our analysis demonstrate that the models trained by using autoencoder-based features could achieve 90% specificity and 75% sensitivity. By comparing our results with the performance of hand-crafted features-based model on the same dataset (as presented by the authors of [9]), we show how the proposed approach is able to provide a significant improvement (i.e., around 8.5% specificity and 10.5% sensitivity) with respect to the hand-crafted features. Even if the improvement provided by these features is in a sense limited, from a practical point of view it is worth underlying its simplicity also with respect to designing and tuning features. Indeed, we believe that this is the major advantage in using autoencoders for this class of problems.

The key contributions of this paper can be summarized as follows:

- We propose an approach for using autoencoders to automatically extract features for characterizing users' mobility behavior using GPS traces collected by means of mobile phones.
- We demonstrate that these automatically extracted features can be used as inputs to machine learning algorithms for constructing personalized models that are able to achieve better performance than those based on complex hand-crafted features.
- We propose different inputs representation of autoencoders and quantify their effectiveness in predicting users' depressive states.

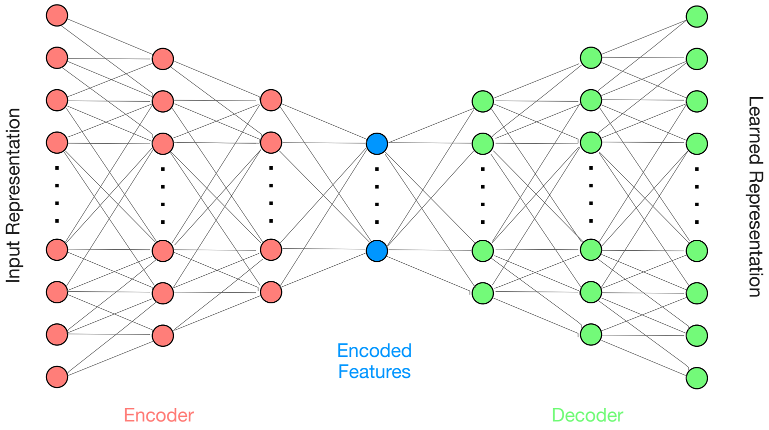


Fig. 1. General architecture of an autoencoder.

Additionally, we also study the impact of a series of key parameters on the performance of our models. In particular, we analyze the use of different autoencoder's network configurations, number of days for which mobility features should be used, and the time interval for which GPS traces should be sampled in order to improve models' performance. Our results show that using more complex networks (i.e., increasing the size) does not improve the performance, there is no significant difference in the results by using different activation functions, and increasing the dropout rate over 10% results in significant drop in the prediction performance.

2 CHARACTERIZING MOBILITY BEHAVIOR USING AUTOENCODERS

In this section we discuss the potential of using autoencoders for the automatic extraction of features for characterizing mobility behavior. We also present the key elements of the design of autoencoders [16], including approaches for selecting effective input representations and for avoiding data overfitting.

2.1 Why Should We Use Autoencoders?

The goal of an autoencoder is to learn a compressed or decompressed representation of input data that can be used to reconstruct the original one with a significantly small error. A few years ago, Hinton et al. demonstrated the potential of autoencoders for automatic and unsupervised learning-based discovery of generic features [16]. As shown in Figure 1, an autoencoder consists of two components: an encoder and a decoder. The objective of an encoder is to compress the input representation. This compressed representation is decompressed by the decoder to obtain the learned representation with the dimensions equal to the original ones.

An autoencoder comprises of one input layer, one output layer and an odd number of hidden layers. The middle layer of the network should have the lowest number of nodes so that the transmission of input data through this bottleneck can be used to obtain meaningful encoded representations. In other words, the outputs of the encoder will be the extracted compressed input representations, which can be seen as discovered emergent features. Once the autoencoder is trained, the *encoder* is taken out from this network and used as a model for encoding the given input representations. In simple words, the encoder is used to compute features from the given raw inputs. We believe that the human mobility behavior is complex and it might be difficult to capture with manually engineered functions. Therefore, autoencoders seem an ideal choice for characterizing generic mobility behavior of users.

2.2 Input Representations

Mobility patterns differ from person to person. They reflect the lifestyle of individuals influenced by the geography of the places visited by them. For example, the distances covered by two people to go from home to work can be very different depending on the length of their commuting. For this reason, thefi rst step is to *normalize* the mobility trajectories in order to make them comparable and to allow for the extraction of common patterns.

The use of raw data could make impossible for an autoencoder to learn a common compressed representation, since it would instead capture the differences in people's movement due to the geography of the places. To better understand this, let us consider an example where two users living in different geographic locations have travelled from home to work and then back home. Let us assume that, for one of them, the commute consists of a 50-mile car trip, whereas for the second one, it consists of 2-mile cycle ride. Now, by simply using the GPS data it would be difficult to translate, compare andfi nd common patterns from these trajectories. Therefore, we have to transform GPS traces into representations that can be generalized. More specifically, we transform them in three ways:

- *Displacement Representation (DR)*: to obtain this transformation we compute a vector of distances between all pairs of adjacent GPS points. In order to make autoencoders less sensitive to the differences in people's mobility behavior, we normalize these distance vectors of each user by using a *MinMax* normalization function. The formal process of transforming GPS points to the displacement representation is described below.
  Given a set $\mathbf{P}^u = \{p_1^u, p_2^u, \ldots, p_N^u\}$ of N GPS points of a user u. We define the displacement vector for user u as:
  $$\mathbf{D}^u = \{d_1^u, d_2^u, \ldots, d_i^u, \ldots, d_{N-1}^u\} \tag{1}$$
  with
  $$d_i^u = dist(p_i^u, p_{i+1}^u) \tag{2}$$
  where *dist()* is a function that computes Haversine distance between two points.

  Then we compute the normalized displacement vector D_{MinMax}^u as follows:
  $$\mathbf{D}_{MinMax}^u = MinMax(\mathbf{D}^u) \tag{3}$$
  where $MinMax()$ is the minmax scaling function applied to each element of D^u. $\mathbf{D}_{MinMax}^u$ is used as input representation of the network.

- *Change in Displacement Representation (CDR)*: we compute a vector containing ratios of distances for all pairs of adjacent GPS points with the distance for the preceding pairs. In other words, wefi rst calculate the distances between all pairs of adjacent GPS points and then take the ratio of each distance value with its preceding distance value. A more formal description of the transformation of GPS points to compute the changes in displacement representation is described below.
  Given a set of GPS points $\mathbf{P}^u$, we define the changes in displacement vector for user u as:
  $$\mathbf{C}^u = \{c_1^u, c_2^u, \ldots, c_i^u, \ldots, c_{N-2}^u\} \tag{4}$$
  with
  $$c_i^u = 1 - d_{i+1}^u / d_i^u \tag{5}$$

where d_i^u is the distance between GPS point p_i^u and p_{i+1}^u as described in Equation 11. $\mathbf{C}^u$ is used as an input representation of the network.

- *Significant Place Representation (SPR)*: in order to obtain this transformation, we compute the time spent at the top S significant places.
  More formally, given a set of GPS points $\mathbf{P}^u$, we first cluster them into significant places using the approach presented in [9] for clustering location points. This results in a set $\mathbf{L}^u = \{l_1^u, l_2^u, \ldots, l_i^u, \ldots, l_{M^u}^u\}$ of M^u places [1]. Then based on the overall time a user u spent during the period of data collection at each place in $\mathbf{L}^u$, we find the top k significant places (sorted in a decreasing order of time spent). This results in a set $\mathbf{S}^u = \{s_1^u, s_2^u, \ldots, s_k^u\}$ of k significant places for user u.
  We then compute the time spent by the user u at all places in $\mathbf{S}^u$ for a given day as follows:
  $$\mathbf{T}_{day}^u = \{t_{day,s_1^u}^u, t_{day,s_2^u}^u, \ldots, t_{day,s_k^u}^u\} \tag{6}$$
  Additionally, all GPS points that are not member of any significant place in $\mathbf{S}^u$ are used to calculate the time spent at non-significant places for each day. Finally, this value (i.e., time spent in non-significant places) is appended to the vector $\mathbf{T}_{day}^u$. By doing so, we obtain a final vector of $\mathbf{T}_{day\diamond}^u$ composed of $S + 1$ elements. $\mathbf{T}_{day\diamond}^u$ is used as the significant place representation.

Note that all of these representations are computed for each day's GPS data for all users iteratively. Moreover, GPS traces are not always sampled at equal intervals but the input layer of an autoencoder has a fixed number of nodes: for this reason, we first transform GPS traces of each day into an equally spaced time series. The process of selecting of optimal time window (τ_{window}) is discussed in Section 4.3.

2.3 Regularization Approach to Prevent Autoencoders from Overfitting

One of the design goals of machine learning algorithms is to be robust to overfitting [14]. A predictive model is said to be overfitted when its performance on the training data is significantly higher than the performance on validation data. In such situations the model adjusts its parameters to capture very specific characteristics of the training data and, for this reason, it does not generalize well to test datasets with different characteristics. This is a common issue with most neural network-based models when they are trained with a limited amount of training data [34].

Numerous methods have been proposed for addressing this issue. These include early stopping of training when the performance on the validation data starts to get worse and soft weight sharing [34]. A most commonly adopted method is "dropout" – a technique that performs model averaging to reduce overfitting in neural networks [44]. It is a method in which nodes (from both hidden and visible layers) in a neural network are temporarily dropped along with all their incoming and outgoing connections to other nodes. More specifically, certain number of nodes are randomly picked to be dropped for an iteration (i.e., completion of a batch). This guarantees that the weight of the network will not adapt to a very specific set of inputs. The number of nodes to be dropped can be selected as a percentage of available nodes in a layer and it is usually referred to as dropout rate. The process of selecting of optimal dropout rate is discussed in Section 4.3.

3 BUILDING PREDICTION MODELS

3.1 Computing PHQ Scores

In this study we use the PHQ-8 test that is a widely adopted and extensively studied 8-item questionnaire for assessing and monitoring depression severity [26]. In this test each question is associated to a score between 0

[1] We use M^u as the number of significant places may vary for each user.

and 3. The PHQ score is computed by adding the contributions of all questions and it lies between 0 and 24, where 0 indicates no depression and 24 indicates severe depression. Note that the PHQ score of a user can be computed at a daily, weekly or monthly frequency (based on the requirements) by asking the user to respond to PHQ-8 tests accordingly. Since the goal of our work is to predict the daily depressive states, we would need to ask users to provide their responses to PHQ-8 tests every day. It is worth noting that our approach is flexible and can also be adapted for the prediction of weekly or monthly depressive states. However, since it is not trivial to engage users in such a study for a longer period, we consider the task of predicting daily depressive states that gives us a statistically significant number of samples (i.e., PHQ-8 scores) for each user to build their personalized models.

One possible approach for predicting users' depressive states is to model this problem using regression, i.e., prediction of scores from 0 to 24 for each day. However, as also reported in the literature [9], building personalized regression models (i.e., performing separate regression task for each user) leads to poor prediction results given the noise in the data (i.e., both in the GPS traces and labels from questionnaires) and the sparse nature of the PHQ-8 scores (i.e., the limited amount of observations per score). For this reason, the prediction task that we consider is modelled as a binary classification task as in the Trajectories of Depression project [9] that we use as a comparator. More specifically, in order to convert this into a classification problem we compute a binary label, which indicates absence (by '0') and presence (by '1') of depressed mood on that day, for each day and for each user. This transformation is done in two steps: (i) we first calculate the mean and standard deviation for the PHQ-8 scores of each user from their daily PHQ-8 scores; (ii) we then assign label '1' (i.e., the presence of depressed mood) if the PHQ-8 score of a day is larger than the mean PHQ-8 score of that user plus one standard deviation, otherwise the label '0' (i.e., the absence of depressed mood) is assigned. It is worth underlining that the mean and standard deviation of PHQ-8 scores are computed only on users' training data.

3.2 Architecture and Implementation

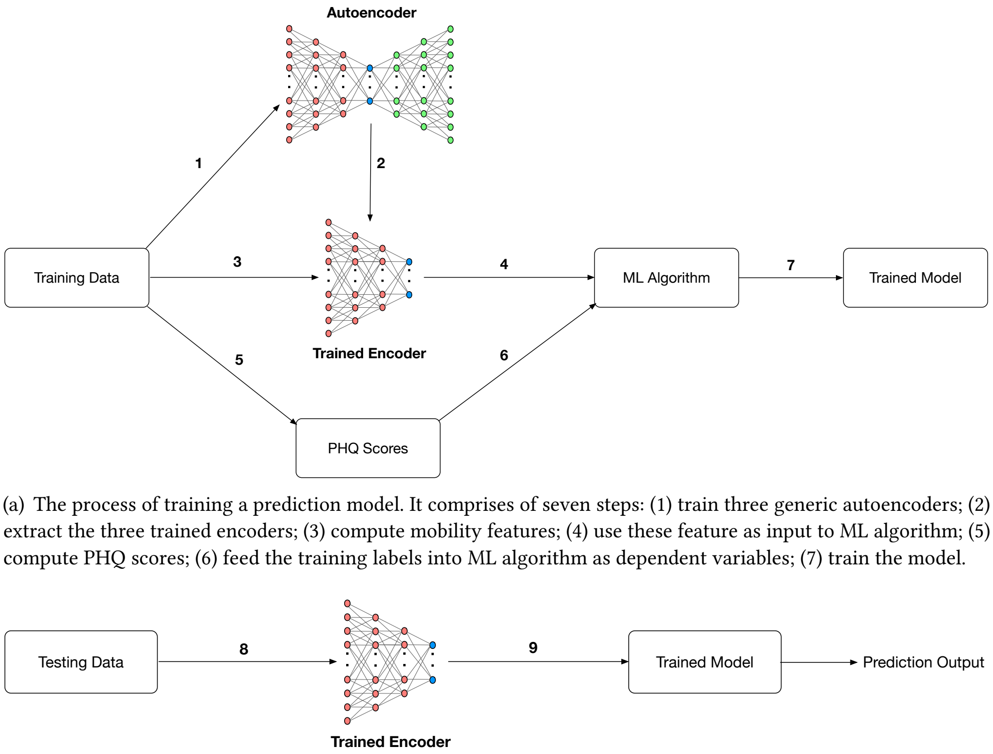
Figure 2 presents the architecture of the proposed prediction mechanism consisting of two processes: (a) training of prediction model, and (b) making predictions. The process of training prediction models consists of the following seven steps:

Step 1: train three generic autoencoders (i.e., autoencoders that are trained with all users' data and exploited to discover features for all users) by using each of the three input representations obtained from the training data of all users;
Step 2: once the autoencoders are trained, extract the three *trained encoders* from these autoencoders;
Step 3: compute mobility features for the training data by using mobility traces (with different input representations) as input for all the trained encoders;
Step 4: use the computed mobility features as input to machine learning (ML) algorithm;
Step 5: compute training labels from the *PHQ-8 scores*;
Step 6: feed the training labels into ML algorithm as dependent variables;
Step 7: once both dependent variables and mobility features are ready, train the ML algorithm to obtain the *trained model.*

Once the models are trained, they can be used to make predictions. As presented in Figure 2(b), this process consists of the following two steps:

Step 8: compute mobility features for the testing data by using mobility traces as input for all the trained encoders;
Step 9: feed the computed features (on testing data) to the trained model for obtaining prediction outputs.

It is worth noting that training and testing datasets are obtained by splitting each users' data into the portions of 80% and 20% respectively. Moreover, we build personalized prediction models using three different machine learning (ML) algorithms: (i) support vector machine (SVM) [11] with Gaussian kernel, (ii) random forest (RF) [8],



(a) The process of training a prediction model. It comprises of seven steps: (1) train three generic autoencoders; (2) extract the three trained encoders; (3) compute mobility features; (4) use these feature as input to ML algorithm; (5) compute PHQ scores; (6) feed the training labels into ML algorithm as dependent variables; (7) train the model.

(b) The process of making predictions using the trained encoder and prediction model. It comprises of two steps: (8) compute mobility features using trained encoders; and (9) use computed features to make prediction.

Fig. 2. Architecture of our prediction mechanism.

and (iii) XGBoost [10]. Note that we could not use neural network-based algorithms because of the limited amount of data available for constructing personalized models for each user.

Additionally, to evaluate the importance of the proposed input representations for predicting the depressive states, we build seven different prediction models by using seven combinations (i.e., all possible combinations) of encoded features obtained through the input representations (discussed in Section 2.2). More specifically, the combinations are: (i) DR, (ii) CDR, (iii) SPR, (iv) DR + CDR, (v) DR + SPR, (vi) CDR + SPR, and (vii) DR + CDR + SPR[2] .

4 EVALUATION SETTINGS

4.1 Dataset

In order to conduct our analysis, we use the Trajectories of Depression dataset that was collected using the MoodTraces application by the authors of [9]. MoodTraces employed a mixed method approach of passive mobile sensing and questionnaire responses from users. Participants were asked to respond to a modified version of the

[2] In order to ensure reproducibility of these results and to make available these tools for the community, the code can be downloaded from https://github.com/AbhinavMehrotra/Mobility_Autoencoder.

Table 1. Version of the PHQ-8 test used in the study.

Have you been bothered by any of the following in the last day?
Little interest or pleasure in doing things.
Feeling down, depressed, or hopeless.
Trouble falling or staying asleep, or sleeping too much.
Feeling tired or having little energy.
Poor appetite or overeating.
Feeling bad about yourself, or that you are a failure, or have let yourself or your family down.
Trouble concentrating on things, such as reading the newspaper or watching television.
Moving or speaking so slowly that other people could have noticed.

PHQ-8 questionnaire [26] that comprised of the eight items listed in Table 1. The questionnaire asked how they felt in the last day but the PHQ-8 score is computed based on how frequently they felt it over a period of last 14 day. Therefore, for each user we computed the PHQ score on a given day x by using their responses from day $x - 13$ until day x and count how many times each depressive symptom occurred in this time interval. Then, for each question a score of 0, 1, 2 and 3 if given when the corresponding symptom has been reported to be occurred for 0-1, 2-6, 7-11, 12-14 days respectively. Finally, we sum up the score for all eight questions to obtain a PHQ-8 score between 0-24. It is worth noting that this method of collecting PHQ-8 data has been widely adopted by the psychiatrists [25].

Since studies have shown that data collected through ESM questionnaires may contain responses that were not reported correctly and were responded for the sake of completion. Therefore, we filter out the questionnaires that were responded too quickly by using the Speeder Index approach [40]. Finally, these remaining responses to questionnaires are used as the ground-truth values of participants' depressive states.

At the same time, the mobile application also passively collected users' mobility traces (i.e., GPS data) [3] through an adaptive sampling approach. In this sensing approach, the location data is collected only when there is a significant change in the user's location, hence the sampling rate is not constant.

Since some of the GPS data can be noisy (for example, in the case when a user is inside a building), we filtered out the GPS points that are not very accurate. To do this, we used the same approach as taken by the authors of the Trajectories of Depression study [9], i.e., we remove points that have accuracy error of more than 200 meters.

Between September 3, 2014 and June 14, 2015, 184 people participated in the study by installing the app. Since many of them participated for a very short duration, we considered only 28 users (15 male and 13 female) who used the app for the minimum duration of 71 days. The average age of all users is 31 years and they are linked to different occupations, such as students, academics, artists, and retired people.

4.2 Selecting Baselines

In the Trajectories of Depression study [9], the authors proposed a variety of novel mobility features for predicting the depressive state of individuals. These features include total distance covered, maximum distance between two locations, radius of gyration, standard deviation of the displacements, maximum distance from home, number of different places visited, number of different significant places visited, and a routine index capturing the repetition of visits of a user over time to the same place at the same hour. By using the same dataset we are able to compare the performance of predictive models built using features from autoencoders against those relying on manually engineered ones.

[3] Note that the application collected additional sensor data such as users' physical activities, application usage and communication logs. This information is not analyzed directly in this work. However, our method could be applied to any other sensor data modalities by designing the corresponding input representation.

4.3 Selecting and Tuning of the Experimental Parameters

We now discuss the selection and tuning of the parameters used in the performance evaluation of our approach.

4.3.1 Autoencoder Network Topology. Network topology (i.e., the optimal number of layers and their nodes) is one of the key parameters of an autoencoder. In order to explore the impact of the network structure on the performance of our models to predict depressive states, we optimize the number of hidden layers (denoted by h in this paper) to construct autoencoders. More specifically, we construct autoencoders with the different number of hidden layers (i.e., excluding the input and output layers) such that $h \in [1, 3, 5, 7]$. We use 1 as a lower bound because that is a minimum number of hidden layer an autoencoder should have. We aim to extend the network to examine whether it can derive features at successive levels of abstraction. Therefore, we construct additional networks with 3, 5 and 7 hidden layers apart from a network with 1 hidden layer.

In the structure of the encoder, we set consecutive layers with half the number of nodes as in the preceding layer, starting from the second layer (i.e., the hidden layer next to the input layer). On the other hand, in the structure of a decoder, we set consecutive layers with half the number of nodes as in the succeeding layer, starting from the second last layer (i.e., the hidden layer preceding the output layer). Note that we restrict any layer to contain less than f nodes and the centre layer to always have f nodes as it is the number of features we want to extract.

Considering the number of features to be extracted is denoted by f, the number of hidden layers in an autoencoder is denoted by h and the number of nodes in the input layer (which is same as in output layer) is denoted by n_{IO}. The number of nodes in j^{th} hidden layer (denoted by n^j_{hidden}) can be computed as:

$$n^j_{hidden} = \begin{cases} f, & \text{if } j = \frac{h-1}{2} + 1 \\ n^j_{encoder}, & \text{if } j < \frac{h-1}{2} + 1 \\ n^j_{decoder}, & \text{if } j > \frac{h-1}{2} + 1 \end{cases} \tag{7}$$

where $n^j_{encoder}$ and $n^j_{decoder}$ are functions to compute the number of nodes for the encoder and the decoder respectively. These functions are defined as:

$$n^j_{encoder} = \begin{cases} f, & \text{if } f > \frac{n_{IO}}{j+1} \\ \frac{n_{IO}}{j+1}, & \text{otherwise} \end{cases} \tag{8}$$

$$n^j_{decoder} = \begin{cases} f, & \text{if } f > \frac{n_{IO}}{h-j+1} \\ \frac{n_{IO}}{h-j+1}, & \text{otherwise} \end{cases} \tag{9}$$

To understand this more clearly, let us consider an example where we have an autoencoder with $h = 5$ hidden layers, $n_{IO} = 48$ input nodes and we want to encode the input to $f = 5$ features. Now, based on the Equation 7 the structure of this network would be 48-24-12-5-12-24-48 (i.e., thefi rst layer will be composed by 48 nodes, the second one by 24 and so on).

Note that the value of the number of nodes in the input/output layer (n_{IO} is based on τ_{hist} and τ_{window} (discussed in next subsection). Formally, the number of nodes in the input/output layer can be computed as:

$$n_{IO} = \tau_{hist} * \frac{24 * 60}{\tau_{window}} \tag{10}$$

Additionally, in order to regularize, the autoencoders we tune the value of dropout rate between 0 and 90% with 10% steps. Note that we do not consider 100% as this would mean that all the nodes of each layer are dropped. Moreover, we use MSE as loss function and ADAM optimizer [20] to train the autoencoders.

It is worth noting that the focus of this work is to investigate the feasibility of an autoencoder based solution for automatically extracting features from raw data in contrast with existing approaches that rely on hand-crafted features. The aim of this paper is to explore the parameter space associated it. For this reason, we selected the simplest autoencoder. The proposed approach is independent from the type of autoencoder used and can be applied to other more complex variations of autoencoders.

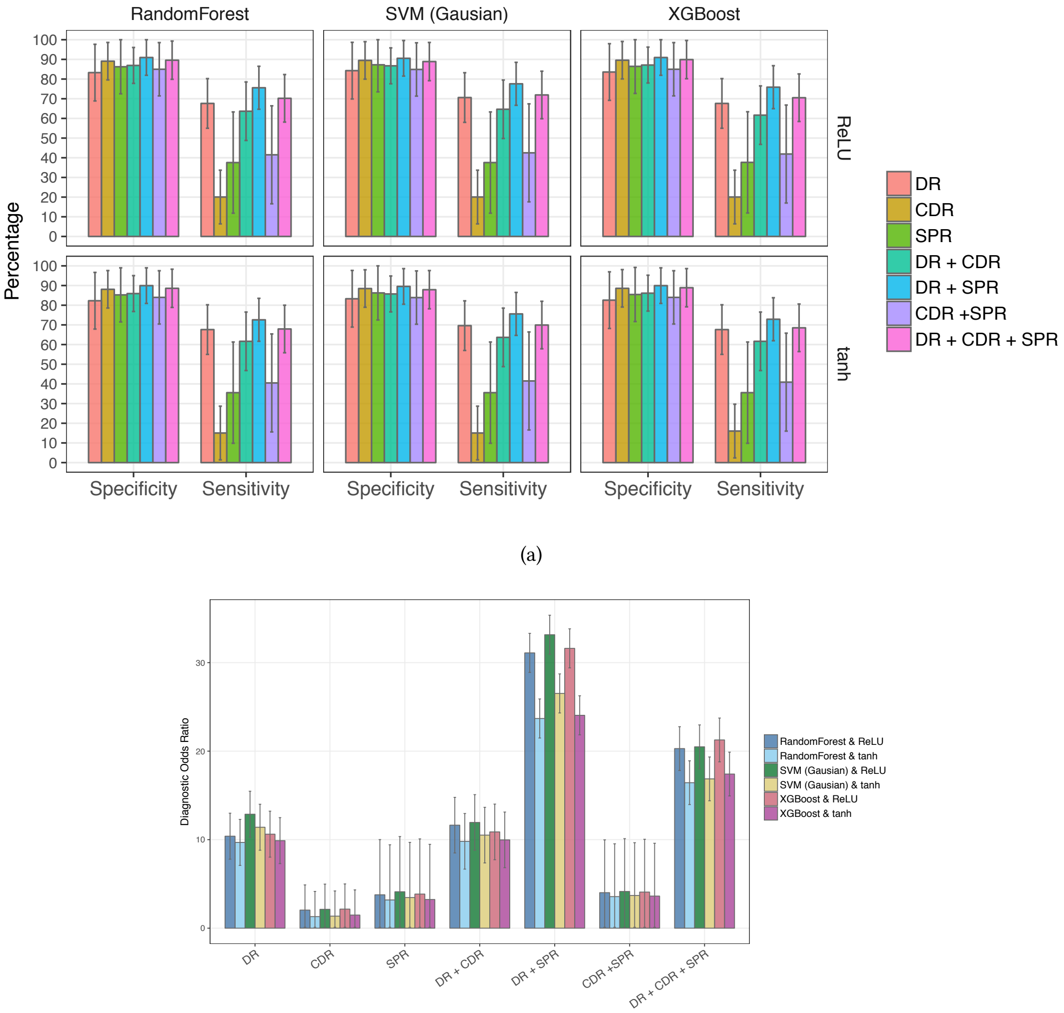
4.3.2 Length of Period for the Extraction of the Mobility Features. Since the PHQ-8 test considers the condition of users for the past 14 days, should we also use the mobility features of these 14 days to build the model or fewer days will suffice? In order to answer this question, we aim to find the optimal number of days (τ_{hist}) for which the mobility features should be considered for the prediction task. We optimize the values of $\tau_{hist} \in [1, 14]$ with steps of 1 day. It is worth noting that τ_{hist} equal to 1 indicates the use of features for the current day. It is worth noting that the focus of this work is on associating changes in depressive states with variations in mobility patterns. Moreover, depressive states usually last for several days and, therefore, we consider a time period of more than one day in our analysis. In other words, we do not consider intra-day mobility variations.

4.3.3 Time Window for Input Representations. Mobility trajectories differ from person to person as they are influenced by the geography of the places visited by them. For this reason, the first step is to transform the mobility trajectories such that they become comparable and common patterns can be extracted as discussed in Section 2.2. However, to perform this transformation we first need to convert the raw GPS traces into equally spaced time series because as discussed above (see Section 4.1) the GPS traces are sampled at uneven intervals (i.e., when there was a significant change in the users' location). In order to do so, the optimal time window (τ_{window}) for this conversion has to be calculated. Since people do not change their location very frequently we use 10m time window as lower bound for computing the input representations. On the other hand, by increasing the time window we will end up aggregating the available data points and thus reducing the size of the input layer in our network. For example, by having the time window of 1 hour we will have only 24 nodes containing distance travelled by the user in each hour. Having a higher value of time window will make it even difficult to extract the features. Therefore, we aim to optimize the value of time window as $\tau_{window} \in [10, 60]$ with the steps of 10 minutes.

4.3.4 Number of Top Places for the Significant Places Representation. In order to compute the significant places representation, we need to set the number of top significant places, which we indicated with k. At the same time, we need to set the value of k greater than F (i.e., the number of features extracted from an autoencoder). However, if we consider high value of k we might not find the same number of significant places for all users. Therefore, it is not reasonable to optimize this parameter. For this reason, we chose the value of k equal to the lowest number of significant places visited by a user in our dataset.

5 PREDICTION RESULTS

We now present the performance of our approach for predicting depressive states using mobility features of each individuals discovered by autoencoders by exploiting different input representations. The selection of input representations is a key step for designing effective autoencoders. Since some combinations might lead to non-optimal results. For this reason, an extensive performance evaluation has to be carried out to investigate all combinations of input representations. We quantified the performance of our models in terms of specificity and sensitivity (presented in Figure 3.a). Moreover, for the ease of comparing the performance of different models, we also computed the diagnostic odds ratio (DOR), which is presented in Figure 3.b. DOR is widely used in clinical studies as a measure for assessing performance of binary classification models [13], which is computed as:



(a)

(b)

Fig. 3. Prediction results (a. in terms of specificity and sensitivity; and b. in terms of DOR) for different combinations of input representations computed by using three different classifiers and two activation functions. These results were obtained by optimizing dropout rate, f, h, τ_{hist} and τ_{window} with the grid search approach.

$$DOR = \frac{specificity * sensitivity}{(1 - specificity) * (1 - sensitivity)} \tag{11}$$

Our results demonstrate that models trained with DR and SPR (i.e., *Displacement Representation* and *Significant Places Representation* respectively) outperform those trained with the other six combinations of input representations. More specifically, this analysis shows that using DR and SPR as input representation, ReLU as activation,

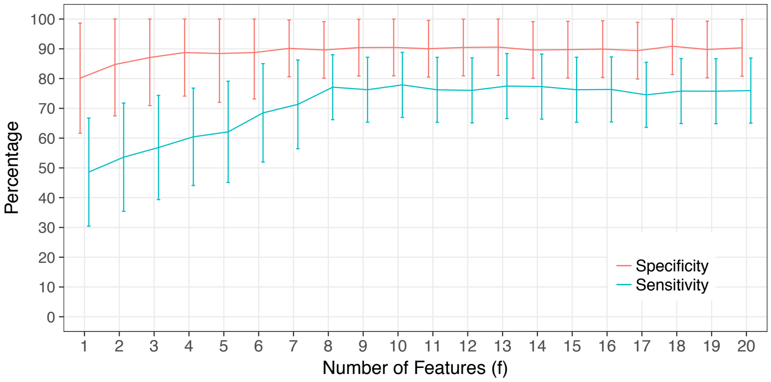


Fig. 4. Prediction performance by using different values of encoded features (f).

and SVM as machine learning algorithm, our approach is able to achieve 91% specificity and 77% sensitivity (i.e., DOR equal to 34).

A comparison with the results obtained using the technique described in the Trajectories of Depression study [9] reveals that our model successfully outperforms the performance of hand-crafted features-based model (which achieved specificity of 82% and sensitivity of 67%) with a significant improvement (i.e., around 9% specificity and 10% sensitivity). In other words, the mobility features extracted by means of autoencoders are more effective for predicting depressive states compared to hand-crafted features. It is worth noting that this study focusses on the analysis of mobility data for predicting users' depressive states; therefore, we could only compare our results with [9], which is the only existing predictive method for the prediction of depressive states based on the analysis of mobility data. However, a detailed comparison of our approach with other techniques that rely on different types of data sources (and potential applications of our approach to alternative or additional data sources) are discussed in the Related Work section (i.e., Section 7).

In Figure 3 we also present the prediction results obtained by using three different machine learning algorithms: SVM (Gaussian), Random Forest, and XGBoost. Our results show that the best prediction results are obtained with SVM. However, there is no difference in the performance of models trained with Random Forest and XGBoost algorithms. More specifically, the performance of the model trained with DR and SPR input representation shows an improvement of 1% specificity and 2% sensitivity by using SVM compared to the other algorithms taken into consideration.

Furthermore, we also compare the use of two types of activation functions in the autoencoder: (i) hyperbolic tangent ($tanh$) [2] and (ii) Rectified Linear Unit ($ReLU$) [33]. Our results show that the models trained with the mobility features obtained through autoencoders relying on $ReLU$ lead to an improved accuracy over the other models relying on $tanh$.

It is worth noting that for these analyses, we optimize the value of dropout rate, number of features f, number of hidden layers (h), τ_{hist}, and τ_{window}. In order to tune the model we employed the grid search approach that trains the model for each combination of the given parameters and evaluates their performance on a held-out validation set [6]. For the best model (discussed earlier) the optimal values for dropout rate, f, h, τ_{hist}, and τ_{window} were 10%, 8, 3, 14 and 10, respectively.

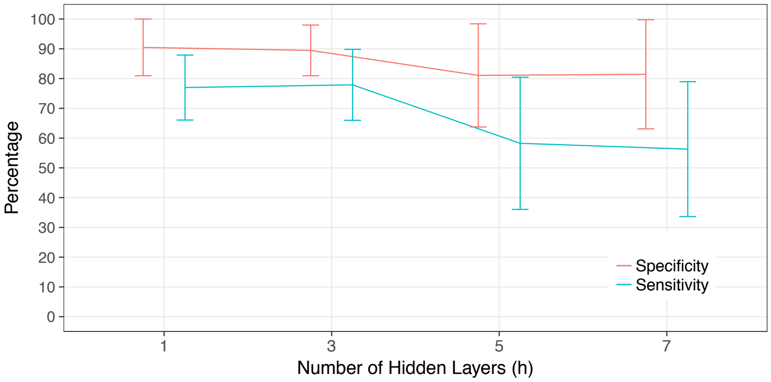


Fig. 5. Prediction performance by using different values of hidden layers (h).

5.1 Impact of f on Prediction Performance

In this subsection we analyze the impact of f (i.e., the number of features encoded) on the prediction performance of our approach. In order to conduct this analysis, we use a combination of *Displacement* and *Significant Places* representations (i.e., DR and SPR) for extracting mobility features. Moreover, we set the activation function, dropout rate, h, τ_{hist} and τ_{window} as ReLU, 10%, 3, 14 and 10 respectively, which are the optimal values discussed in the previous analysis. As shown in Figure 4, our results suggest that there is indeed an impact of the value of f on the performance of our model. As the value of f increases, the prediction performance improves. However, the performance of the model stabilizes when f reaches the value of 8. This indicates that the optimal number of features extracted by the autoencoders is 8. A larger number of features in general might require a larger training set.

5.2 Impact of h on Prediction Performance

In this subsection we examine the impact of h (i.e., the number of hidden layers used in the autoencoder) on the performance of our approach. We perform this analysis by using the combination of *Displacement* and *Significant Places* representations (i.e., DR and SPR) for extracting mobility features in a similar way to the analysis conducted for understanding the impact of the value of f. Moreover, we set the activation function, dropout rate, f, τ_{hist} and τ_{window} as ReLU, 10%, 8, 14 and 10 respectively, which are the optimal values discussed in thefi rst analysis of this section. As shown in Figure 5 our results demonstrate that there is a significant impact of h on the performance of our model. The best performance of the model can be achieved when h is 3. However, the improvement of the performance of the model trained with h equal to 3 compared to that trained with h equal 1 is negligible. At the same time, the model shows worse performance when h is equal to 5 and 7. This could be due to the fact that noise in the input data (i.e., mobility representations) could make it difficult for the network to generalize and this might potentially lead to overfitting. Moreover, it is worth noting that these kinds of dataset are also limited in terms of size, which potentially hinders the training process of network.

5.3 Impact of Dropout Rate on Prediction Performance

In this subsection we analyze the impact of the value of the dropout rate (i.e., the number of nodes to be dropped in each layer to make the autoencoder generalizable) on the performance of the proposed approach. In order to perform this analysis we configure an autoencoder with 3 hidden layers, ReLU activation function and use

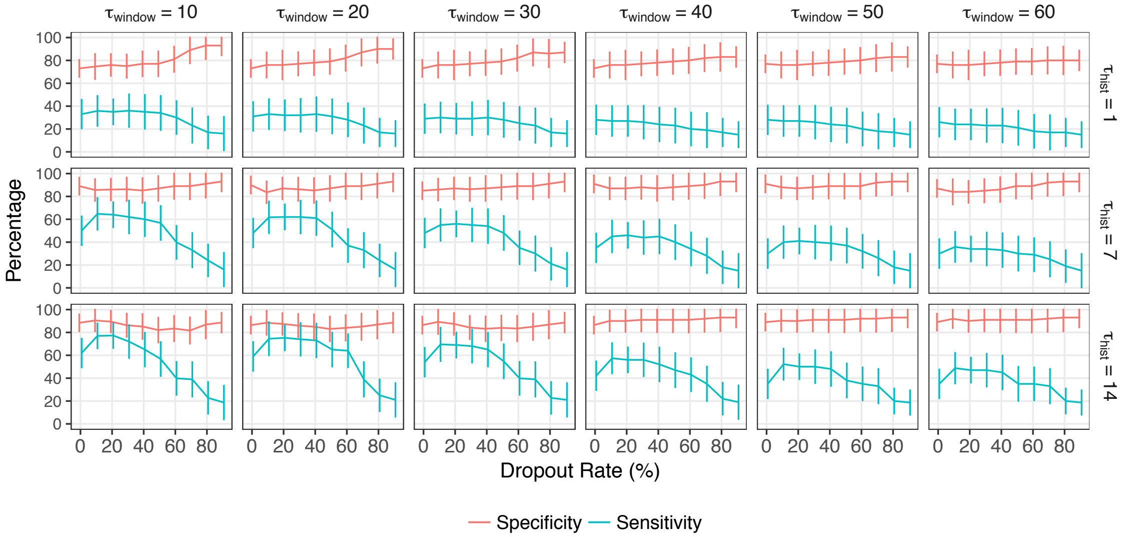


Fig. 6. Prediction performance by using different values of dropout rates.

the combination of *Displacement* and *Significant Places* representations (i.e., DR and SPR) for extracting 8 (i.e., optimal value of f) mobility features.

We then examine the impact of the dropout for all values of $\tau_{hist} \in [1, 7, 14]$ and $\tau_{window} \in [10, 20, 30, 40, 50, 60]$. As shown in Figure 6, the results of this analysis demonstrate that for the optimal values of τ_{hist} (i.e., 14) and τ_{window} (i.e., 10), the performance of our models improves with 10% dropout rate compared to 0% and stay stable until 30% dropout rate. However, there is a significant reduction of performance with the further increase of dropout rate. On the other hand, with other configurations of τ_{hist} and τ_{window} (such as $\tau_{hist} \in [1, 7]$ and $\tau_{window} \geq 20$) the performance stabilizes between 20-40% dropout rate. Thesefi ndings partially contradict the empirical results of Warde et al. [49] that the performance stabilizes between 20-60% dropout rate. However, it is worth noting that the characteristics of our dataset are rather different. Indeed, a possible reason for the drop of performance with increase in the dropout rate could be related to the fact that our dataset is limited in size (which is the problem for most studies in this area) and, therefore, it becomes difficult for the autoencoder to extract effective features with high dropout rate.

5.4 Impact of τ_{window} on Prediction Performance

In this subsection we analyze the impact of τ_{window} (i.e., time window for transforming unevenly spaced time series of GPS traces into equally spaced ones for computing input representations) on the effectiveness of features extracted from autoencoders to predict the depressive state of an individual. In order to perform this analysis we configure an autoencoder with 3 hidden layers, ReLU as activation function, 10% dropout rate and use the combination of *Displacement* and *Significant Places* representations (i.e., DR and SPR) for extracting 8 (i.e., optimal value of f) mobility features.

We then examine the impact of τ_{window} for all values of $\tau_{hist} \in [1, 7, 14]$. As shown in Figure 7, our results demonstrate that as τ_{window} increases, the performance of our model decreases. Moreover, this pattern is present for all values of τ_{hist}. However, there is a considerably small difference in the prediction performance of models

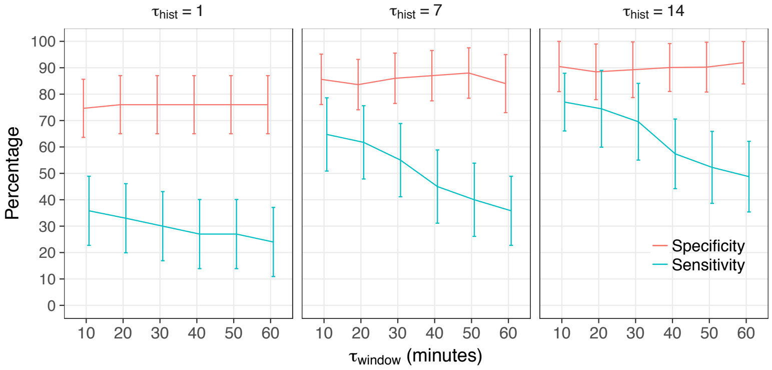


Fig. 7. Prediction results for different values of τ_{window}.

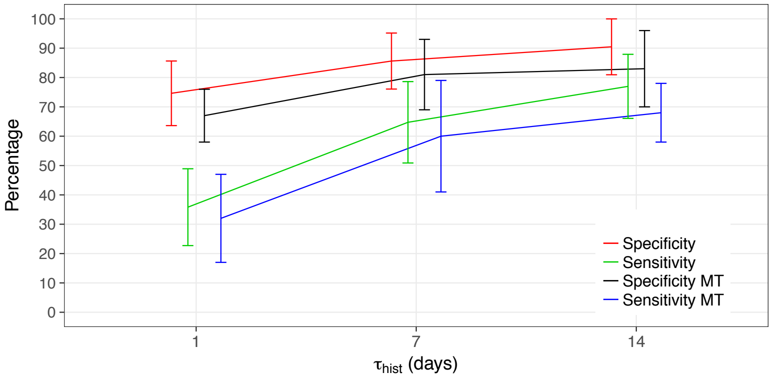


Fig. 8. Prediction results for different values of τ_{hist}.

using τ_{window} equal to 10 and 20. This indicates that the autoencoders require less aggregate information in order to extract useful features.

5.5 Impact of τ_{hist} on Prediction Performance

In this subsection we analyze the impact of τ_{hist} (i.e., the number of days used for the calculation of the mobility metrics) on the performance of our approach for predicting the depressive states of an individual. In order to perform this analysis we configure the autoencoder with 3 hidden layers, ReLU activation function and use the combination of *Displacement* and *Significant Places* representations for extracting mobility features. Moreover, we set the dropout rate, f, h and τ_{window} with their optimal values.

As shown in Figure 8, our results demonstrate that as τ_{hist} increases the performance of models also increases (i.e., sensitivity and specificity increase). With the lowest value of τ_{hist} (i.e., 1), our model could achieve only 75% specificity and 36% sensitivity. On the other hand, with highest value of τ_{hist} (i.e., 14) our model is able to achieve 91% specificity and 77% sensitivity. Additionally, in Figure 8 we also show that our model's performance always remains closely above the performance of models based on hand-crafted features (proposed in [9]).

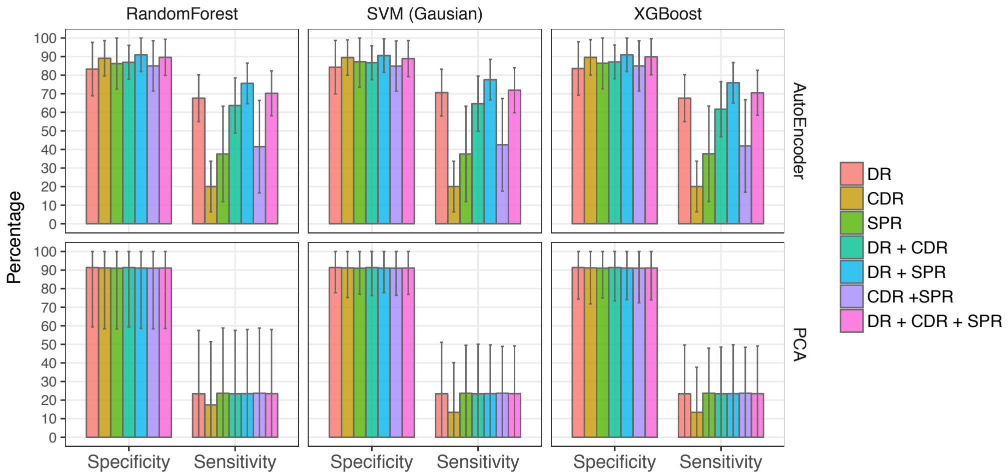


Fig. 9. Comparing prediction performance of our approach by using autoencoders and PCA.

5.6 Why AutoEncoder over Other Types of Feature Extraction Methods?

As discussed in Section 2.1, an autoencoder is the state-of-the-art approach to learn a compressed representation of input data (i.e., to extract features) with an unsupervised learning technique. We now compare our approach with a traditional algorithm for automatic feature compression, namely principal component analysis (PCA) [19]. Algorithms like PCA are comparatively much less sophisticated, but, at the same time, easier to train.

In order to investigate the use of PCA in our application scenario, we simply replace autoencoders with a Gaussian kernel-based PCA in our system architecture (discussed in Section 3.2). We then train three PCAs for the corresponding three input representations and use the combinations of the features extracted by these PCAs to build seven models. This process is similar to that we discussed in Section 3.2. We set the values τ_{hist} and τ_{window} to 14 and 10 as by using these parameter values that lead to the highest number of input features with the lowest amount of data aggregation. Finally, to compute the performance of our models relying on PCAs, we optimize the number of features to be extracted by the PCA.

We compare the performance of our models relying on PCA against the results discussed in Section 5 (i.e., the performance of our models based on autoencoders). As shown in Figure 9, the results demonstrate that autoencoders outperform PCA with a significant difference in terms of sensitivity. More specifically, PCA could only achieve 90% specificity and 23% sensitivity by using most of the combinations for input representations compared to the performance of the autoencoder based model that obtained 91% specificity and 77% sensitivity.

6 LIMITATIONS AND FUTURE WORK

In this paper we have presented a novel approach for predicting depressive states by exploiting mobility features that are automatically extracted from GPS traces. To the best of our knowledge, this is thefi rst study that investigate the potential of autoencoders for extraction of features characterizing complex human mobility behavior. We show that our approach does not only reduce the burden of designing new features for the characterization of human mobility behavior but it could also achieve a significant improvement compared to prediction performed by exploiting “hand-crafted" features.

However, we have identified a few minor limitations in the proposed approach that can investigated in the future studies. One of the limitations is that the current implementation of our approach does not completely explore the temporal aspect of mobility behavior. For instance, there might be some weekly mobility patterns that could not be discovered through a feed forward network-based encoder. Instead, we need to construct the autoencoder using a recurrent neural network (RNN) [32] that is capable of capturing dynamic temporal behavior. However, for evaluating such an approach we would need longitudinal data for several months and possibly years. There is indeed a need of conducting longitudinal studies for exploring the potential of RNNs in this context.

Another limitation is related to the ecological validity of this study. We believe that the key contribution of this work is methodological and, indeed, our results indicate the potential of the application of the proposed methodology for predicting depressive states. *However, it is difficult to make a strong claim in terms of generalizability of the method. We believe that this study should be replicated in order to verify its validity, for example, on different demographics. In our opinion it would also be useful to repeat the study in a clinical setting using different diagnostic methods, in particular one-to-one interviews for collecting a variety of ground-truth data and for validating the robustness of the proposed solution.*

Moreover, while making a selection between hand-crafted and automatically extracted features, there is a trade-off between the prediction performance and interpretability. Our experiments show that the autoencoders have the potential to automatically discover features that are more effective, compared to the hand-crafted ones, for predicting users' depressive states. However, the use of autoencoders affects the explainability of the predictions. In general, the problem of explainability of deep learning algorithms is an open challenge for the machine learning community [12].

At the same time, we believe that this work demonstrates that autoencoders are a powerful tool for the automatic extraction of features for characterizing different aspects of human behavior, not only for well-being applications but for a variety of other anticipatory mobile apps [36].

7 RELATED WORK

In this section we review the related work in two key areas, namely the studies about monitoring mood and well-being of users through the analysis of their mobile sensor data, and those about using autoencoders for extracting features from raw data.

7.1 Exploiting Mobile Sensor Data to Infer Users' Mood and Well-being

The recent advances in context sensing have made mobile phones a unique platform for building effective mental health monitoring systems. Many studies have shown the potential of exploiting mobile sensing for passively inferring users' mental health and well-being [4, 9, 28, 30, 38, 42, 45, 46]. Whilst sounding simple, inferring users' emotional states by exploiting their physical contextual information is a complex task.

In thefi rst study of this area, i.e., EmotionSense [38], the authors used the audio samples to train predictive models running locally on the phone for identifying speakers and inferring their emotion. Their results demonstrate that speech alone can be used to detect emotions with an average accuracy of 71%. Later, many other studies have investigated the use of alternate and less intruding sensors (such as GPS, activity, sms and call logs) for predicting users' emotional states. In [28] Likamwa et al. proposed to exploit the mobile interaction logs (such as SMS, email, phone call, application usage, and web browsing) together with the contextual information obtained through mobile sensors for predicting users' daily average mood. The authors evaluated their approach with 32 participants over two months and demonstrated that their system could predict 93.1% of the daily pleasure averages and 92.7% of the activeness averages with less than 0.25 MSE (mean squared error). Similarly, in [3], Alvarez-Lozano at al. examined the potential of exploiting the mobile app usage logs for predicting the bipolar state of users. In particular, the author proposed to quantify the changes in app usage behavior and exploit them to predict variations in self-reported bipolar states. In other to evaluate their approach, they conducted a study with 18 patients with bipolar disorder for over 5 months. Their results demonstrate that users' app usage patterns have a significantly strong correlation with different aspects of their bipolar states.

Recent studies have also shown that mobility traces (obtained through the GPS sensor) contains valuable information for modeling users' depressive states. In [9] Canzian and Musolesi investigated the potential of exploiting mobility traces for predicting depressive states. The authors proposed a series of novel metrics for characterizing human mobility patterns and used these metrics to build models for predicting depressive states of users. They demonstrated that their approach could be used to infer changes in depressive states from users' average depressive state with the sensitivity and specificity of 83% and 68%. In a similar study [41], Saeb et al. also shown the potential of exploiting human mobility traces for predicting depressive states. In this work, the authors proposed a different set of metrics for characterizing human mobility patterns. Their focus was on correlation analysis: their results show that their metrics also have a strong association with users' depressive states.

All of these studies show the potential of using mobile sensor data for inferring emotional states of users in real-time. In particular, information on human mobility behavior derived from GPS data has been shown to be an invaluable source for passively inferring users' mental health and well-being [5, 9, 41, 42]. However, these studies rely on hand-crafted features in order to build predictive models. Using the approach proposed in this work, we are able to automatically extract features that capture emerging behavioral patterns that are present in the raw data.

7.2 Automatic Extraction of Features Using Autoencoders

Indeed, most machine learning algorithms require humans to be in the loop for designing features that could be used as input of the learning task since designing features requires an in-depth knowledge of the domain. Autoencoders have been recently used to address this problem.

This powerful technique has been employed in numerous fields in computer science, such as vision [23], natural language processing (NLP) [27], speech recognition [29] and activity recognition [37]. It is apparent that this powerful technique has essentially revolutionized the field of computer vision in terms of dimensionality reduction and feature extraction [23]. Researchers and practitioners have exploited different variations of autoencoders to solve several problems in the area. Some of the examples include the design of denoising autoencoders to recover an original undistorted image from a partially corrupted image [48], sparse autoencoders for image reconstruction [22] and variational autoencoders for generating similar images [21].

As discussed above, another application area is Natural Language Processing: various approaches that rely on autoencoders for document clustering [18], learning representations for words and phrases [27], analyzing sentiments [47], and detecting paraphrases [43] have been proposed. Similarly, autoencoders have been used for the extraction of extremely sophisticated features in other fields such as speech recognition [15] and activity recognition [37].

Compared to this body of work, to the best of our knowledge, this is the first study that investigates the effectiveness of autoencoder-based feature extraction for characterizing human behavior and in particular mobility from GPS data and to show its potential for a variety of applications not only in digital mental health.

8 CONCLUSIONS

In this paper we have proposed an approach for predicting the depressive states in users by exploiting a set of mobility features that are automatically extracted from a deep autoencoder. Through a series of extensive experiments we have demonstrated that our approach could successfully achieve performance that provides an improvement compared to the models that are trained on hand-crafted mobility features. We have explored the use of different input representations and their effectiveness in predicting future depressive states of an individual. Finally, through a series of experimental analyses we have demonstrated the optimal configuration of the key parameters used in our solution.

We believe that the approach presented in this work can be applied to a variety of problems that rely on the automatic extraction of features describing human behavior, not restricted to the digital health domain. Indeed, the definition of hand-crafted features might not just be possible in certain domains given the lack of background knowledge. The possibility of exploiting emergent patterns in behavioral data is an extremely powerful technique, which provides an effective solution to this problem.

ACKNOWLEDGMENTS

The authors would like to thank Dr Luca Canzian (the author of Trajectories of Depression study) for useful discussions and for collecting the dataset in thefi rst place. This work was supported through the EPSRC grants EP/L006340/1 and EP/P016278/1 at UCL and by The Alan Turing Institute under the EPSRC grant EP/N510129/1.

REFERENCES

[1] 2017. *WHO Depression Report*. http://www.who.int/mediacentre/factsheets/fs369/en/.

[2] Milton Abramowitz and Irene A Stegun. 1964. *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*. Courier Corporation.

[3] Jorge Alvarez-Lozano, Venet Osmani, Oscar Mayora, Mads Frost, Jakob Bardram, Maria Faurholt-Jepsen, and Lars Vedel Kessing. 2014. Tell me your apps and I will tell you your mood: correlation of apps usage with bipolar disorder state. In *PETRA'14*.

[4] Jakob E Bardram, Mads Frost, Károly Szántó, and Gabriela Marcu. 2012. The MONARCA self-assessment system: a persuasive personal monitoring system for bipolar patients. In *IHI'12*.

[5] Gianni Barlacchi, Christos Perentis, Abhinav Mehrotra, Mirco Musolesi, and Bruno Lepri. 2017. Are you getting sick? Predicting influenza-like symptoms using human mobility behaviors. *EPJ Data Science* 6, 1 (2017), 27.

[6] James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *JMLR* 13, Feb (2012), 281–305.

[7] H. Bourlard and Y. Kamp. 1988. Auto-association by Multilayer Perceptrons and Singular Value Decomposition. *Biological Cybernetics* 59, 4 (1988), 291–294.

[8] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[9] Luca Canzian and Mirco Musolesi. 2015. Trajectories of depression: unobtrusive monitoring of depressive states by means of smartphone mobility traces analysis. In *UbiComp'15*.

[10] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *KDD'16*.

[11] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning* 20, 3 (1995), 273–297.

[12] Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv:1702.08608v2* (2017).

[13] Afina S Glas, Jeroen G Lijmer, Martin H Prins, Gouke J Bonsel, and Patrick MM Bossuyt. 2003. The diagnostic odds ratio: a single indicator of test performance. *Journal of Clinical Epidemiology* 56, 11 (2003), 1129–1135.

[14] Douglas M Hawkins. 2004. The problem of overfitting. *Journal of Chemical Information and Computer Sciences* 44, 1 (2004), 1–12.

[15] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, and others. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine* 29, 6 (2012), 82–97.

[16] Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science* 313, 5786 (2006), 504–507.

[17] Geoffrey E Hinton and Richard S Zemel. 1994. Autoencoders, Minimum Description Length and Helmholtz Free Energy. In *NIPS'94*.

[18] Anil K Jain. 2010. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters* 31, 8 (2010), 651–666.

[19] Ian T Jolliffe. 1986. Principal component analysis and factor analysis. In *Principal Component Analysis*. Springer, 115–128.

[20] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICML'15*.

[21] Diederik P Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. *Stat* 1050 (2014), 1.

[22] Alex Krizhevsky and Geoffrey E Hinton. 2011. Using very deep autoencoders for content-based image retrieval. In *ESANN'11*.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS'12*.

[24] Kurt Kroenke, Robert L Spitzer, and Janet BW Williams. 2001. The PHQ-9. *Journal of General Internal Medicine* 16, 9 (2001), 606–613.

[25] Kurt Kroenke, Robert L Spitzer, Janet BW Williams, and Bernd Löwe. 2010. The patient health questionnaire somatic, anxiety, and depressive symptom scales: a systematic review. *General Hospital Psychiatry* 32, 4 (2010), 345–359.

[26] Kurt Kroenke, Tara W. Strine, Robert L Spitzer, Janet B. W. Williams, Joyce T. Berry, and Ali H. Mokdad. 2009. The PHQ-8 as a measure of current depression in the general population. *Journal of Affective Disorders* 114, 1 (2009), 163–173.

[27] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML'14*.

[28] Robert LiKamWa, Yunxin Liu, Nicholas D Lane, and Lin Zhong. 2013. Moodscope: Building a mood sensor from smartphone usage patterns. In *MobiSys'13*.

[29] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori. 2013. Speech enhancement based on deep denoising autoencoder. In *INTERSPEECH'13*.

[30] Abhinav Mehrotra, Robert Hendley, and Mirco Musolesi. 2016. Towards Multi-modal Anticipatory Monitoring of Depressive States through the Analysis of Human-Smartphone. In *Adjunct UbiComp'16*. Heidelberg, Germany.

[31] Abhinav Mehrotra, Jo Vermeulen, Veljko Pejovic, and Mirco Musolesi. 2015. Ask, But Don't Interrupt: The Case for Interruptibility-Aware Mobile Experience Sampling. In *UbiComp'15 Adjunct*.

[32] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, JanČ ernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH'10*.

[33] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted Boltzmann machines. In *ICML'10*.

[34] Steven J Nowlan and Geoffrey E Hinton. 1992. Simplifying neural networks by soft weight-sharing. *Neural Computation* 4, 4 (1992), 473–493.

[35] J Olesen, A Gustavsson, Mikael Svensson, H-U Wittchen, and B Jönsson. 2012. The economic cost of brain disorders in Europe. *European Journal of Neurology* 19, 1 (2012), 155–162.

[36] Veljko Pejovic and Mirco Musolesi. 2015. Anticipatory mobile computing: A survey of the state of the art and research challenges. *Comput. Surveys* 47, 3 (2015), 1–47.

[37] Thomas Plötz, Nils Y Hammerla, and Patrick Olivier. 2011. Feature learning for activity recognition in ubiquitous computing. In *IJCAI'11*.

[38] Kiran K. Rachuri, Mirco Musolesi, Cecilia Mascolo, Jason Rentfrow, Chris Longworth, and Andrius Aucinas. 2010. EmotionSense: A Mobile Phones based Adaptive Platform for Experimental Social Psychology Research. In *UbiComp'10*.

[39] Babak Roshanaei-Moghaddam, Wayne J Katon, and Joan Russo. 2009. The longitudinal effects of depression on physical activity. *General Hospital Psychiatry* 31, 4 (2009), 306–315.

[40] Joss Roßmann. 2010. Data quality in web surveys of the German longitudinal election study 2009. In *3rd ECPR Graduate Conference*.

[41] Sohrab Saeb, Mi Zhang, Christopher J Karr, Stephen M Schueller, Marya E Corden, Konrad P Kording, and David C Mohr. 2015. Mobile phone sensor correlates of depressive symptom severity in daily-life behavior: an exploratory study. *Journal of Medical Internet Research* 17, 7 (2015), e175.

[42] Sandra Servia-Rodríguez, Kiran K Rachuri, Cecilia Mascolo, Peter J Rentfrow, Neal Lathia, and Gillian M Sandstrom. 2017. Mobile Sensing at the Service of Mental Well-being: a Large-scale Longitudinal Study. In *WWW'17*.

[43] Richard Socher, Eric H Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Y Ng. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *NIPS'11*.

[44] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *JMLR* 15, 1 (2014), 1929–1958.

[45] Yoshihiko Suhara, Yinzhan Xu, and Alex'Sandy' Pentland. 2017. DeepMood: Forecasting Depressed Mood Based on Self-Reported Histories via Recurrent Neural Networks. In *WWW'17*.

[46] Sara Ann Taylor, Natasha Jaques, Ehimwenma Nosakhare, Akane Sano, and Rosalind Picard. 2017. Personalized Multitask Learning for Predicting Tomorrow's Mood, Stress, and Health. *IEEE Transactions on Affective Computing* 99, 1 (2017), 17–33.

[47] Peter D Turney. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *ACL'02*.

[48] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research* 11, 12 (2010), 3371–3408.

[49] David Warde-Farley, Ian J Goodfellow, Aaron Courville, and Yoshua Bengio. 2013. An empirical analysis of dropout in piecewise linear networks. In *ICLR'14*.

Received February 2018; revised May 2018; accepted September 2018