Smolak et al. *EPJ Data Science* (2022) 11:45
https://doi.org/10.1140/epjds/s13688-022-00356-4

EPJ Data Science
a SpringerOpen Journal

EPJ.org

REGULAR ARTICLE Open Access

Explaining human mobility predictions through a pattern matching algorithm

Kamil Smolak[1*], Witold Rohm[1] and Katarzyna Sila-Nowicka[1,2,3]

*Correspondence:
kamil.smolak@upwr.edu.pl
[1]Institute of Geodesy and Geoinformatics, Wrocław University of Environmental and Life Sciences, Norwida 25, 50-375 Wrocław, Poland
Full list of author information is available at the end of the article

Abstract

Understanding what impacts the predictability of human movement is a key element for the further improvement of mobility prediction models. Up to this day, such analyses have been conducted using the upper bound of predictability of human mobility. However, later works indicated discrepancies between the upper bound of predictability and accuracy of actual predictions suggesting that the predictability estimation is not accurate. In this work, we confirm these discrepancies and, instead of predictability measure, we focus on explaining what impacts the actual accuracy of human mobility predictions. We show that the accuracy of predictions is dependent on the similarity of transitions observed in the training and test sets derived from the mobility data. We propose and evaluate five pattern matching based-measures, which allow us to quickly estimate the potential prediction accuracy of human mobility. As a result, we find that our metrics can explain up to 90% of its variability. We also find that measures that were proved to explain the variability of predictability measure, fail to explain the variability of predictions accuracy. This suggests that predictability measure and accuracy of predictions should not be compared. Our metrics can be used to quickly assess how predictable the data will be for prediction algorithms. We share developed metrics as a part of HuMobi, the open-source Python library.

Keywords: Human mobility; Prediction; Predictability; Sequence alignment; Global alignment; Sequence matching

1 Introduction

The possibility of gathering precise individual movement data in large populations resulted in a plethora of studies explaining human movement and applying gathered knowledge in many fields, such as traffic forecasting, urban planning, disease spread modelling and disaster response [1–4]. For many of these applications, future locations of people are essential information, enabling them to deliver accurate results. Hence, improving the accuracy of human mobility predictions is a crucial challenge that has to be addressed to develop yet better technologies. In this paper, we aim to improve our understanding of human mobility predictability. For that, we deliver a set of novel metrics, which explain what impacts the accuracy of human mobility predictions, which in turn can help design better mobility prediction algorithms.

Springer

Human mobility predictions are based on mobility sequences that correspond to a series of symbols, where each symbol represents a different location [5]. These sequences are extracted from one person's raw movement trajectory, which includes the positions of individuals recorded over time. Locations that are included in the movement sequence are determined as places where a person spends a significant amount of time. These places are considered important in an individual's daily mobility [6]. This approach enables treating human mobility prediction as any sequence prediction task. The general goal of these predictions is to predict the next symbol or symbols in the sequence [7]. Despite some degree of randomness and spontaneity, human mobility is predictable [8]. Regardless of that, predicting human mobility is challenging and has resulted in many different algorithms, including Markov-based, compression-based, time-series and machine learning methods [9]. The accuracy of predictions is measured as the number of correctly predicted symbols [7]. However, since prediction algorithms are tested on different datasets, their accuracy cannot be directly compared.

Incomparability of prediction results has been addressed by Song et al. [8], who provided a methodology for estimation of the limit of predictability of human movement sequences. The predictability estimation method calculates the entropy of each sequence, which is then converted into the limit of predictability by solving a limiting case of Fano's inequality (which is related to the average information loss in a message obtained over a noisy channel) [10]. Predictability limit serves as a reference for mobility prediction algorithms to which their performance can be compared [11]. Song et al. [8] quantified the predictability of individual mobility using a mobile phone location dataset collected from 45,000 people. The authors have processed the dataset into movement sequences by determining a person's location at regular time intervals (this approach is called the *next time-bin* approach). Their work reported the upper bound of predictability to be 93%. However, later works used the same method and found predictability to range from 43% to 95% as a result of different input data used for its estimation [5]. This large difference has driven researchers to investigate the factors influencing mobility predictability. Their identification is crucial for movement prediction and will enable a deeper understanding of human mobility behaviour. The methodology of predictability estimation suffers from low interpretability, as it is based on a complex Lempel–Ziv data compression algorithm [12]. Therefore, finding what impacts movement predictability is difficult and this issue has not been fully resolved yet [13].

Identifying the factors impacting mobility predictability have been attempted multiple times. The biggest impact on predictability have data characteristics and processing methods. Specifically, changes in movement sequences directly impact predictability, among which the number of unique locations (unique symbols) in the sequence and its length have been identified as the most influential [14, 15]. Furthermore, Kulkarni et al. [14] identified the existence of long-range structural correlations in movement sequences, finding the number of interacting symbols (symbols that are co-occurring in a specific pattern) and the distance between them to be other important factors impacting movement predictability. However, the changes are introduced into sequences indirectly through mobility data processing, therefore, it is important to know how the data processing influences extracted sequences and hence, predictability.

Predictability varies with spatio-temporal data resolution [5]. A decrease in spatial data resolution increases predictability and the dependence between temporal resolution of

the data and predictability is irregular. In these cases, predictability variations still stem from changes in extracted movement sequences, but they are caused by the variations of data spatio-temporal resolution. For example, data of higher spatial resolution will have a higher number of unique locations in the sequence and thus lower predictability.

Another important factor was noted simultaneously by Ikanovic & Mollgaard [16] and Cuttone, Lehmann & Gonzalez [17]. The original approach, used in the work of Song et al. [8], to present the predictability concept was to extract movement sequences using the *next time-bin* approach. Ikanovic & Mollgaard [16] and Cuttone, Lehmann & Gonzalez [17] noticed that this sequence extraction method artificially raises mobility predictability through the introduction of many situations when a person at the current and next time interval is in the same location (the next symbol in the sequence is identical to the previous one). These repetitions are called *self-transitions*. For such a sequence, even a naïve algorithm, guessing the next location to be the same as the previous one, would achieve high accuracy. To eliminate self-transitions, the authors suggested recording only transitions between distinct locations in the movement sequences, arguing that such an event is the most important and difficult to predict in an individual's mobility pattern. This approach is called the *next-place* approach. Sequences extracted using the next-place approach were found to have significantly lower predictability (for example in the work of Cuttone, Lehmann & Gonzalez [17] predictability decreased from 95% for next time-bin sequences to around 70% for next-place sequences).

Some works raised concerns regarding the theory behind predictability estimation methodology. Moreover, contradictory results of actual predictions surpassing the theoretical limit were reported, suggesting that this limit is underestimated. Kulkarni et al. [14] noted that sophisticated prediction algorithms surpass the predictability limit, which is caused by the existence of long-range structural correlations in movement sequences. The existence of these correlations is not considered by the predictability limit estimation method. Lu et al. [7] found that the prediction accuracy of prediction algorithms surpasses the predictability limit when a sequence has non-stationary characteristics, that is when the unconditional joint probability distribution of a sequence varies across its span [18]. This aligns with the fact that the Lempel–Ziv algorithm, which is used to estimate sequences entropy, provides accurate estimations only for the stationary trajectories [13]. It is highly likely that sequences extracted using the next time-bin and, especially, the next-place approaches are non-stationary [19], hence the predictability estimation method is not suitable for this type of data.

Discrepancies between predictability limit and prediction accuracy suggest that factors that were found to impact the predictability limit are not influencing the accuracy of the predictions in the same way. First of all, because the Lempel–Ziv algorithm is suitable only for stationary trajectories, the predictability estimation method may not be suitable for movement sequences and yield incorrect values. Another concern is related to the interpretation of this limit. Predictability is measured for the whole sequence and corresponds to the general predictability of the movement, while the accuracy of prediction algorithms is measured over a part of the movement sequence, usually referred to as a test set. Prediction algorithms require some part of the sequence for training, which has to be excluded from the prediction. Therefore, predictability is measured on a different sequence than accuracy and they cannot be directly compared.

In summary, discrepancies identified between predictability estimation theory and actual predictions are

- Prediction accuracy values are surpassing the theoretical limit;
- Lempel–Ziv estimator is suitable only for stationary sequences, while movement sequences can be non-stationary;
- Predictability cannot be related to a prediction task.

Hence, in this paper, we propose an alternative approach to explain the predictability of human mobility. This work is inspired by the two simple measures of *stationarity* and *regularity*, proposed by Teixeira et al. [20], which can explain a large portion of predictability variations. However, instead of studying factors influencing the predictability limit, we focus on the accuracy of the actual predictions and explain what impacts them. We propose a more complex approach based on a pattern matching algorithm, which quantifies the similarity of information contained in the training and test sequences. We show that this similarity is strongly related to the maximum accuracy that algorithms can achieve.

We validate our method using sequence prediction algorithms, deep neural networks, ensemble decision trees, Markov chains, and a naïve approach. First, to validate our assumptions, the proposed approach is tested on generated sequences with known properties. Then, we use a human mobility dataset from 500 mobile phones users, where the data are processed to extract mobility sequences on various levels of spatial and temporal aggregation. The contributions of this work are:

- We present a novel pattern matching-based approach explaining what impacts the accuracy of actual predictions made on movement sequences;
- We validate the discrepancies between the predictability and predictions, presenting the relationship between them;
- We compare the accuracy values of different sequence prediction algorithms on various types of movement sequences.

2 Data and methods

This section presents the methods and data used in the study. The workflow is presented in Fig. 1. In this research, we will use two datasets: synthetic sequences and real human mobility data. First, we introduce synthetic sequences generated for this experiment. Then we present a human mobility dataset, which was first preprocessed using techniques described in the data processing section. Further, we describe prediction methods that were used to obtain prediction accuracy values. At the end of this section, we present all the metrics used in this study. These are predictability metric proposed by Song et al. [8], metrics proposed by Teixeira et al. [21], and our proposed pattern matching-based metrics. Metrics and prediction accuracy were used to study their correlations and functional dependencies, for which results are presented in the Results section.

2.1 Synthetic sequences

In order to provide a theoretical analysis of our approach, we generated three types of artificial sequences: random, Markovian and non-stationary. The sequence is a series of symbols

$$X = [x_1, x_2, \ldots, x_m], \tag{1}$$

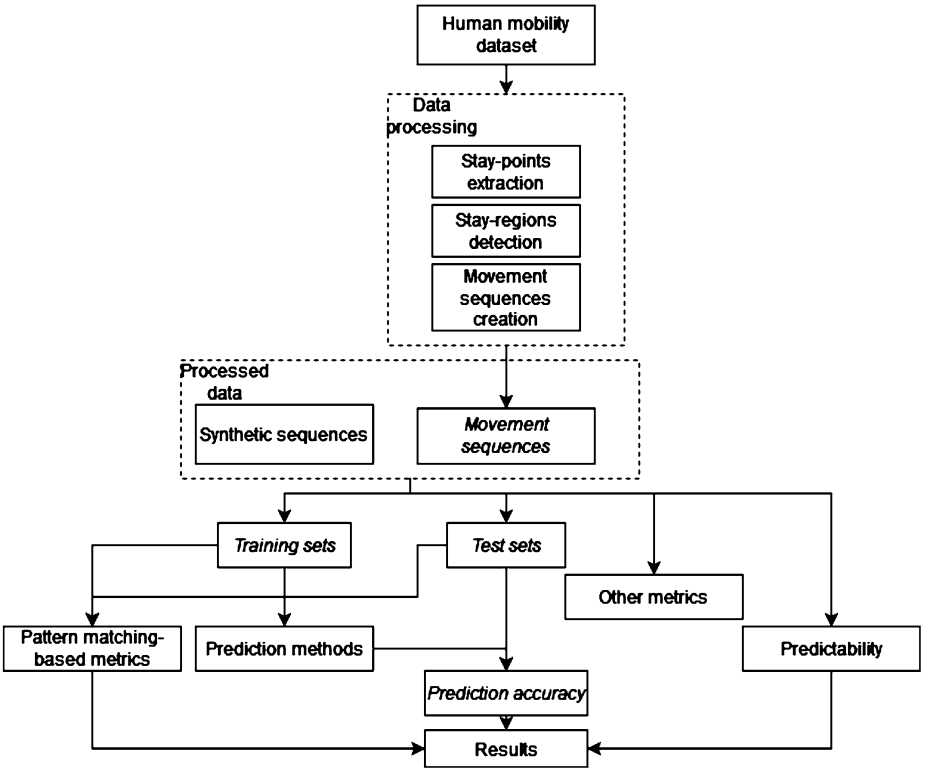

Figure 1 Workflow adopted in this study. Each block's name (apart from those written in italic font) corresponds to the particular subsection of the workflow description. First, data were processed into movement sequences, after which predictability, stationarity, regularity and proposed metrics were calculated. Results were compared against the accuracy of mobility predictions

where each symbol x_m represents a different location (as in mobility sequences). The symbols can repeat within the sequence. For each type of sequence, we generate 100 instances. Each instance is generated using sequence type-specific parameters, which are selected randomly from a given subspace (see sequence types description below for details). The lower bound of the subspace of possible parameters is set so the generated sequences are stable (have to be long enough to obtain stable results) and can be analysed (have at least two symbols).

In random sequences, every symbol x_m in a sequence X is selected from a uniform distribution of m possible symbols. The number of possible symbols m (from 2 to 20) and the length of the sequence n (from 100 to 500 symbols) are randomly selected for each generated sequence. This kind of sequence corresponds to the case of low predictability, as the information shared between training and test sequences should be minimal. The predictability of such sequences decreases with the increase of m possible symbols.

In Markovian sequences, each symbol is following a deterministic sequence $x_1 \to x_2 \to \cdots \to x_m \to x_1 \to \cdots$ with probability p. With probability $1 - p$, the next symbol is randomly selected from the m possible symbols. The p value, sequence length n, and the number of m symbols are selected randomly for each generated sequence and are in the range of 0.1 to 0.9, 100 to 500, and 2 to 20, respectively. Markovian sequences are repetitive and should be easy to predict for most of the prediction algorithms, however, with larger values of p the predictability will decrease.

Non-stationary sequences are generated by a mixture of states, where each state has a different symbol generation process [18]. The state is selected for every generated symbol separately from a distribution P_s, where each state has been assigned a corresponding probability. These probabilities are selected randomly from a uniform distribution and they are normalised together. Each state generates a symbol x_m, from the set of m possible symbols, using probability distribution P_m, which is created using an identical approach as for the creation of the P_s distribution. The number of m possible symbols (from 2 to 20), sequence length n (from 100 to 500), and the number of states s (from 2 to 12) are selected randomly for each generation. Such a generation routine ensures the creation of non-stationary sequences, for which the Lempel–Ziv estimator should fail [20].

2.2 Human mobility dataset

We validate our approach on a large human mobility dataset of a high spatio-temporal resolution. It was collected using mobile phone built-in Global Navigation Satellite Systems (GNSS) receivers and shared for the purpose of this study by the UberMedia company. It contains mobility data from 500 mobile devices of people living in London, UK. In this dataset, location data are collected through applications installed on mobile devices and stored in a database using an advertising identifier. The length of the collected movement trajectories varies from 28 to 31 days. The median fraction of missing records q, expressed in hour-long intervals [8], is $q = 0.04$, which can be considered as complete data with almost no data gaps [19].

It is worth noting that devices with a low value of q can be owned by a specific socio-demographic group, which tends to use mobile internet often. Therefore, this data might not be representative of people who do not use it often. However, we do not study mobility in the spatio-temporal context, such as land use, therefore this potential bias should not impact our findings. Although to mitigate this type of selection bias, the sample was randomly chosen from the subset of trajectories that had $q < 0.4$.

2.3 Data processing

Movement sequences X of individuals represent a series of locations, where each location has been assigned a corresponding symbol in a sequence. Therefore, to extract movement sequences from mobility data, locations visited by an individual have to be detected. This process has three steps: stay-points extraction, stay-regions detection, and movement sequences creation.

2.3.1 Stay-points extraction

First, a stay-point detection algorithm, based on two parameters δ and τ, searches for places where an individual stayed for a significant amount of time in one location. Let the movement trajectory

$$T = \{t_1, t_2, \ldots, t_n\} \tag{2}$$

be a sequence of data points recorded by a single device. This algorithm iterates through each data point in a movement trajectory in temporally ascending order. Each data point is a triplet (x_t, y_t, t) of two coordinates x_t and y_t recorded at time t. Starting from the first

point in the movement trajectory, the algorithm calculates a distance between each iterated data point and the first point. If that distance is lower than δ, that data point is assigned to a currently processed stay-point and the algorithm moves to the next data point. If the distance is higher than a threshold δ, the algorithm calculates the time interval between the first point and the last point within the δ distance. If that time interval is larger than τ, then all data points within the distance threshold are recorded as a stay-point, otherwise, they are discarded. The stay point is recorded as a quadruple $(x_n, y_n, start_n, end_n)$, where x_n and y_n are geographical coordinates of a visited stay-point centre between $start_n$, and the end_n time. After that, the process repeats starting from the first point that was not assigned to the previous stay-point. The process is repeated until the last point in the movement trajectory is processed. δ and τ have to be set to the values ensuring that unimportant stops, such as traffic lights stop [16], are not considered as stay-points. The level of δ also has to account for a GNSS positioning error. In this work we set these values following the guidelines from the work of Jiang et al. [22], that are $\delta = 300$ metres and $\tau = 10$ min.

2.3.2 Stay-regions detection

In the second step of the process, all the stay-points are spatially aggregated into stay-regions, where each stay-region corresponds to a location that was repeatedly visited by an individual. When a location was visited more than once, nearby stay-points probably represent the same location, therefore they can be assigned the same symbol. For this step, we use the density-based spatial clustering of applications with noise (DBSCAN) [16, 17, 23] algorithm. DBSCAN clusters stay-points based on a distance parameter ϵ. That is, if a stay-point is closer to another stay-point in a cluster than ϵ they are considered a single stay-region. After this process, each stay-point is assigned a label of a cluster to which it has been allocated. To simulate various spatial resolutions of data we process data with ϵ equal to 33, 204 and 1688 metres, which approximately corresponds to the scale of buildings, streets and districts, respectively [24].

2.3.3 Movement sequences creation

Finally, the detected stay-regions are processed into the movement sequences. So far, in the predictability studies, two types of movement sequences, *next time-bin* and *next place* have been used. Therefore, in our experiment, we process our data into these types of sequences.

To create the *next time-bin* sequences, we record positions of an individual at regular time intervals (time-bins) Δt. For each time-bin, we check the currently visited stay-region and assign its label to a sequence. If more than one location was visited in a time-bin, then the location visited for a longer period is recorded. If none of the stay-regions was visited during the selected time interval, an empty value is assigned, creating a gap in a sequence. This process creates a temporally ordered movement sequence consisting of symbols representing stay-regions. We use Δt equal to 30 min and 1 h to simulate different temporal resolutions of data often used in human mobility studies [8, 15–17]. Our resulting *next time-bin* sequences have an average length of 697 symbols for $\Delta t = 1h$ and twice more for $\Delta t = 30$ min. The number of unique symbols in the sequences decrease with spatial resolution, starting from 25 unique symbols at an average for $\epsilon = 33m$, through 20 unique symbols for $\epsilon = 204m$, to 9 unique symbols for $\epsilon = 1688m$.

The *next time-bin* approach tends to create many self-transitions, that is situations when symbols are consecutively repeated in a sequence. The idea behind the *next-place* approach is to eliminate these self-transitions, therefore, the *next-place* sequence is created by temporally ordering visited stay-regions. In the *next-place* sequences the temporal dimension is lost, as visited locations are not evenly spread on a time scale. Resulting *next-place* sequences have an average length of 76 symbols. The average number of unique symbols is slightly higher in these sequences, being 27, 20, and 10 unique symbols for $\epsilon = 33m$, $\epsilon = 204m$, $\epsilon = 1688m$, respectively.

2.4 Prediction methods

In this work, we focus on explaining what impacts the accuracy of movement sequences predictions. To ensure the best prediction accuracy, we simultaneously assign the same prediction task, that is predicting the next symbol in a sequence, to various methods. These are deep neural networks, ensemble decision trees, and Markov chains. They represent three groups of the most commonly used algorithms for human mobility predictions [9], that is deep learning and shallow learning algorithms, and Markov-based models. Using different approaches, we are able to select the best predictor for each kind of sequence for further analyses.

It is important to note that each sequence, representing a movement of an individual, is subject to a separate prediction. We use the same approach to prepare input data for machine learning algorithms. First, sequences are transformed into chunks through a windowing algorithm (see Fig. 2). The window extracts $W + 1$ symbols, where W is the size of the window. Next W first symbols are kept as input data and the last symbol is a target value for the prediction. We set $W = 10$, as using larger windows did not result in improvement in predictions, while for lower values we observed a drop of accuracy. At each step, the window moves by one position towards the end of the sequence. Then, the extracted chunks are divided into training and test sets, where the training set contains 80% of the data. A training set is transformed into training-validation pairs using 5-fold cross-validation, each time leaving a fifth part of the data for validation. The prediction accuracy is measured as the number of correctly predicted symbols in a test sequence [7].

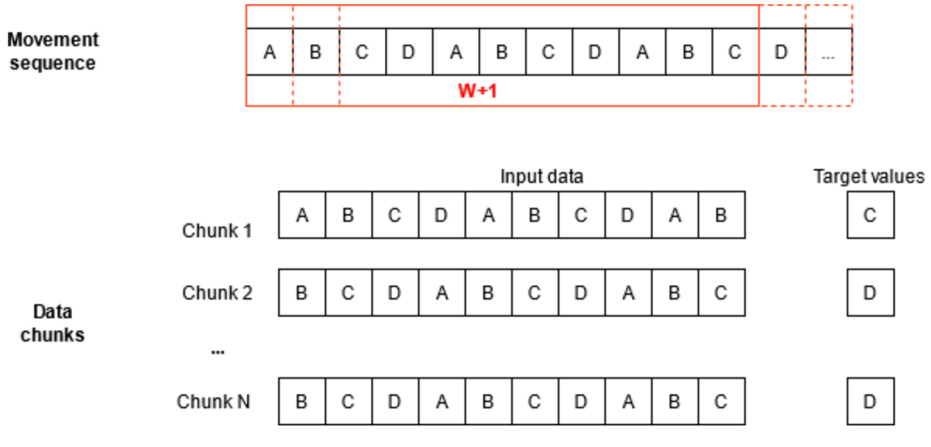

Figure 2 The scheme of windowing algorithm. A window of size $W + 1$, moving over a movement sequence, cuts sequence into chunks of size W and a target value at each step

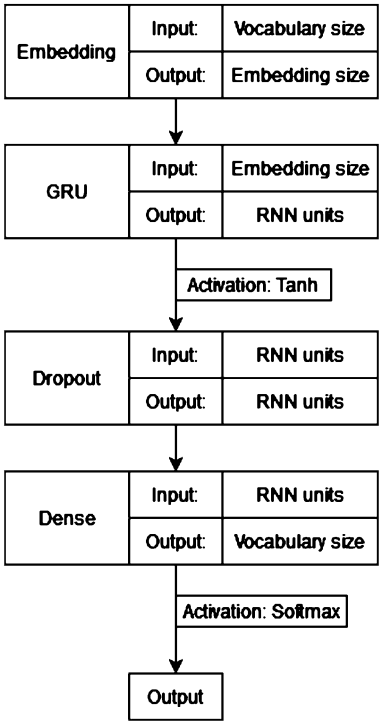

Figure 3 The scheme of a GRU neural network. Each block represents a layer and contains layer's name, the name of a parameter representing the input, and output size of a layer. Arrows indicate the direction of the forward propagation of data through the network

2.4.1 Deep learning network

We implement a deep neural network as a state-of-the-art sequence prediction method. Our solution is based on the gated recurrent unit (GRU) which is a type of recurrent neural networks (RNNs). GRUs are one of the proposed solutions to the vanishing gradient problem, which prevents the neural network from effective training through limited weights adjustment. GRUs are extended by update and reset gates which decide whether the information should be passed to the output. That way, noise is removed during training and important information is kept in the training cycle for longer. The architecture of our network is similar to the next character prediction networks used in language modelling and is presented in Fig. 3. First, we use an embedding layer to transform chunks extracted from sequences into a dense vector. It is then fed to a single GRU layer with a hyperbolic tangent activation function, allowing to scale learned weights into a range from -1 to 1. The GRU layer is connected to the dropout layer, which randomly resets weights to prevent a network from overfitting. Finally, the information is fed to a dense layer activated using the softmax function, which outputs a categorical probability distribution. The distribution represents how likely it is for each symbol to be the next one in the sequence. Using it we draw the next symbol.

To select the embedding layer output size and the number of units in the GRU layer, each time during network training we conduct a series of tests. Using the cross-validation approach, we aim to reach the best prediction accuracy. We test all the combinations of values from a set of powers of 2, ranging from 2^7 to 2^{11}. The input and the output sizes (vocabulary size) of the network are fixed and equal to the number of stay-regions in a processed sequence. We train each network for 30 epochs, however, we implement an early stopping mechanism preventing the network from further training when accuracy on the validation set drops at the two next epochs. In all of the cases, the upper limit of 30

epochs is never reached as networks can be effectively trained within a lower number of epochs.

Movement sequences, especially for high levels of temporal aggregation, can be short. This impacts neural network performance, and they are known to be highly data demanding. Therefore, we decided to use GRUs as they perform better on shorter sequences with less data than other RNNs architectures [25]. Moreover, we apply a cross-validation process to network training which is a rather unusual technique but in our case improves accuracy. Using each data fold, we repeat the training process of a network, which incrementally improves its performance. As a result we noticed around 30% of accuracy improvement.

2.4.2 Ensemble decision trees

As mentioned earlier, the length of movement sequences may be low for which deep-learning-based methods will have limited prediction capabilities. To mitigate the impact of limited sequences length on prediction accuracy, we apply a less data-demanding approach, that is Random Forest (RF), a tree-based ensemble method. This type of method is known for being robust to overfitting problems and to effectively handle small sample sizes [26].

During training, RF constructs a set of trees, each being a separate predictor. These predictors are trained by applying a bootstrap aggregation, that is each tree learns from a randomly selected chunk of data fed to the RF. Bootstrapping ensures that they are uncorrelated, which enables maximising the amount of captured relationships in the data. The final result is derived through the majority vote rule applied to the output of each tree.

We approach the sequence prediction problem with a classification variant of the RF algorithm. The node split is based on the Gini impurity metric, which expresses the likelihood of observation misclassification. Using the cross-validation each time when RF is trained (i. e. for every predicted sequence) we conduct an exhaustive search to select the number of predictors trained within the model. Using a small training sample we found that number of estimators ranging from 500 to 2000 trees gives the best prediction results. We used that search subspace for the RF training.

2.4.3 Markov chains

Markov chains (MCs) have been often used in mobility prediction [7, 9, 13, 20, 27, 28]. MCs are based on probabilities determining which state (symbol) will follow a finite number of symbols preceding it. The number of previous symbols k considered in the MC is called a chain order. For example, an MC of second-order considers the current and the previous symbol when predicting the next symbol. Probabilities are determined using learning data. When predicting, depending on the k last symbols of a sequence corresponding probabilities are selected and used to draw the next symbol. In our experiment, we consider MCs orders from one to six. Research shows that the increase in order does not usually result in an increase in algorithm accuracy [7, 14].

2.4.4 Naïve predictor

For reference, we use a naïve predictor from the work of Cuttone et al. [17], called *toploc*. During prediction the algorithm repeats the symbol which most often appears in a training sequence, hence guessing that the next location will be the one that was most often visited.

This method was found to perform well for next time-bin sequences where a large number of self-transitions appears [17].

2.5 Predictability

We compare our metrics to the actual predictability estimations based on the work of Song et al. [8]. The measure of predictability $\Pi_{\max}$ expresses a theoretical upper bound of predictability that a theoretically perfect (infallible) prediction algorithm can reach. Predictability is estimated separately for each sequence and the whole sequence is taken into consideration. To estimate predictability, first, we measure an actual entropy of a sequence as

$$S_i = -\sum_{X_i' \subset X_i} P(X_i') \log_2\left[P(X_i')\right], \tag{3}$$

where $P(X_i')$ is the probability of finding a particular time-ordered subsequence X_i' in the X_i sequence. Then, entropy is converted into predictability by solving a Fano's inequality [10] which is

$$\Pi_i \leq \Pi_i^{\text{Fano}}(E, m), \tag{4}$$

where E is the measured entropy, and m is the number of unique symbols in the sequence. Π_i^{Fano} is given by

$$\begin{aligned} E = &-\Pi_i^{\text{Fano}} \log_2\left(\Pi_i^{\text{Fano}}\right) - \left(1 - \Pi_i^{\text{Fano}}\right) \log_2\left(1 - \Pi_i^{\text{Fano}}\right) \\ &+ \left(1 - \Pi_i^{\text{Fano}}\right) \log_2(m-1). \end{aligned} \tag{5}$$

By substituting E by S_i we are able to calculate an upper bound of predictability $\Pi_{\max}$.

Direct calculation of entropy is computationally demanding, thus, Song et al. proposed to use the Lempel–Ziv estimator [12]. An actual entropy can be calculated as

$$S^{\text{est}} = \left(\frac{1}{n}\sum_j \Lambda_j\right)^{-1} \log_2 n, \tag{6}$$

where n is the length of the sequence and Λ_j is the length of the shortest substring starting at position j in the sequence, which does not appear from position 1 to $j-1$.

Since the publication of the predictability estimation theory, some researchers noted that a vague description of calculation methodology led to implementation inconsistencies [11]. These include unmatched logarithm bases in equations (5) and (6) and incorrect values of Λ_j in positions where unique substring could not be found (for details refer to [11]).

There are two other major issues worth mentioning. First, the Lempel–Ziv estimator was proved to provide accurate estimates only for stationary sequences, while movement sequences might have non-stationary characteristics [13, 18]. Second and the most important issue is that the predictability and accuracy of predictions should not be compared because of the fundamental differences in their definitions. These measures are calculated using different sequences, which leads to discrepancies.

2.6 Pattern matching-based measures

Given the discrepancies between predictability estimation theory and actual predictions, we propose alternative approaches to explain the variations in mobility predictions accuracy. In this section, we present our pattern matching-based metrics which can be used to estimate the potential accuracy of the prediction of movement sequences.

We base our measures on two types of sequence matching methods, which are the longest common subsequence (LCS) and Needleman–Wunsch [29] algorithms. Originally, these methods are widely used in the bioinformatics field for nucleotide or protein sequences alignment [29–31]. Converting movement sequences into a series of symbols enables the application of those methods to search for the best match between the training and test sets on which movement prediction algorithms were used earlier. A large overlap between the training and test sequences should indicate that a test sequence is highly predictable, while the low number of matched symbols should indicate that the movement will be predicted with low accuracy. We extend sequence matching algorithms to adjust them to sequence prediction problems and derive novel metrics which will help us understand movement predictability.

First, we define a general pattern matching problem. The goal of the pattern matching algorithm is to find matching series of symbols in two movement sequences. In our case, these are training X_{tr} and test X_{ts} subsequences extracted from a movement sequence of an individual. Intuitively, if a series of symbols to predict is also present in a training subsequence, it should be possible to predict it as it was already encountered by an algorithm. Moreover, the longer the matching pattern is the easier it should be to predict.

The idea is to measure the similarity of training and test sequences using a score based on the LCS or Needleman–Wunsch algorithms. This score is calculated differently for each metric, for which details are presented below. However, each metric is normalised using the same approach to eliminate the effect of the differences in sequences lengths. This normalisation is based on the number of transitions observed in the test sequence, where a transition is a pair of symbols. Therefore, the number of transitions in the test sequence is its length minus one. The motivation for the use of transitions as a normalisation factor is that prediction algorithms learn to forecast the next symbol based on the symbols preceding it. Calculating the number of identical transitions present in the training and test sequence gives a better overview of the similarity of these sequences. After normalisation, metrics express the ratio of matched transitions to the number of transitions that could be matched in the test sequence. For example, a score of one indicates that all transitions in the test sequence were found in the training sequence. Each metric can be generally expressed as:

$$\frac{S}{T} = \frac{S}{n-1}, \tag{7}$$

where S is a score, T is the number of transitions, and n is a sequence length.

2.6.1 LCS-based metrics

The goal of the LCS algorithm is to find the longest subsequence shared by the pair of sequences. Matching subsequences have to appear in the sequences, from which they have been derived, in the same order. They do not necessarily have to appear at the same positions and can be separated by a number of mismatched symbols, called gaps. Gaps can

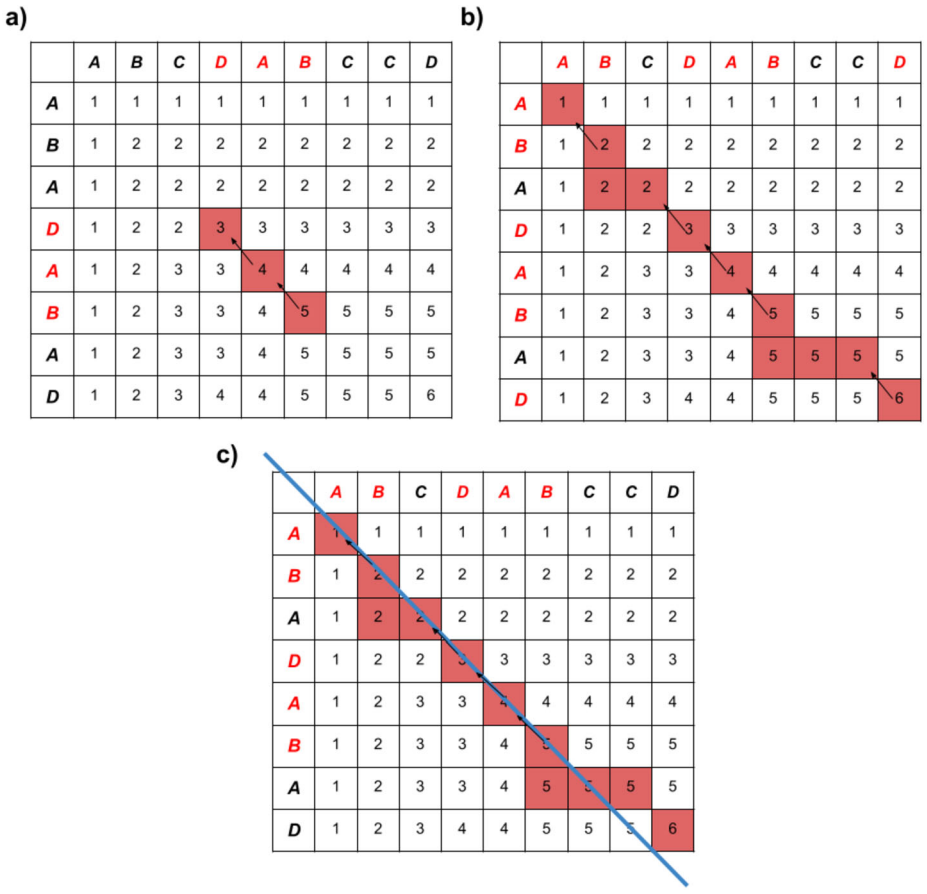

Figure 4 Visualisation of three variants of the longest common subsequence problem. Three score matrices represent scoring for the same pair of sequences. The longest common subsequence is chosen differently in each variant. Picture (**a**) shows the *dense repeatability metric*, picture (**b**) represents the *sparse repeatability* metric, and picture (**c**) shows the *equally sparse repeatability metric*, where the blue line indicates the diagonal with the highest score

have different sizes, including the size of gaps separating the same symbols matched in two sequences.

We propose three LCS-based metrics, denoted *dense repeatability (DR)*, *sparse repeatability (SR)*, and *equally sparse repeatability (ESR)*. Each of these metrics expresses the normalised length of the longest matching subsequence. The length of the sequence is measured using different variants of the LCS algorithm, which are depicted in Fig. 4. In the presented example, a sequence was divided into training $X_{\text{tr}} = [A, B, C, D, A, B, C, C, D]$ and test $X_{\text{ts}} = [A, B, A, D, A, B, A, D]$ sequences. First, let A be a matrix of scores (presented for each metric in Fig. 4) created for a pair of matched sequences, where (i, j) is an element keeping a score for the $i-th$ element of X_{ts} and $j-th$ element of X_{tr}. The algorithm iterates through the matrix, and for each matched pair of symbols an element at (i, j) positions is given a score of $(i-1, j-1)+1$. When symbols are not matching, element at (i, j) is given the higher of $(i-1, j)$ and $(i, j-1)$ elements. The score matrix A is created identically for all three metrics.

Next, the longest common subsequence is found using a traceback approach. The algorithm starts from any element of matrix A and searches for the path. When being at

element (i,j) element at $(i-1,j-1)$ is smaller, the path is drawn between those two elements. When the above is not the case, then the path can be drawn between element (i,j) and $(i-1,j)$ or (i,j) and $(i,j-1)$ if they are equal. This produces a series of possible paths, which are the basis for the three suggested metrics. The difference between three of proposed metrics is in how the longest path is chosen.

In Fig. 4(a) a path for the *dense repeatability (DR)* is presented. This approach assumes that no gaps are allowed in a matched subsequence. That variation of LCS is known as the longest common substring problem [32]. This is an equivalent to the longest path, where all the moves between elements are diagonal, resulting in $[D,A,B]$ with a score equal to two (two matched transitions). Figure 4(b) presents the *sparse repeatability (SR)*. In this case, the longest possible path is chosen, but only those elements which are positioned diagonally to each other are the matching pairs of symbols. Therefore the result is $[A,B,D,A,B,D]$ and the score is five. Figure 4(c) presents the *equally sparse repeatability (ESR)* metric. Here, gaps are allowed but the additional constraint is that the corresponding gaps in the matched subsequences have to be of identical size, but the size of these gaps can vary across the matched pattern. To enforce that, the path with the highest number of elements on the single diagonal is chosen. In our example, the result of such function would be $[A,B,D,A,B]$, because the length of a gap between the last two symbols is different.

2.6.2 Needleman–Wunsch algorithm-based metrics

We base two other measures of similarity on the Needleman–Wunsch algorithm [29], which finds the optimal global alignment between two sequences. In comparison to the LCS algorithm, the Needleman–Wunsch algorithm tries to match the whole sequence, rather than find the longest matching subsequence. Usually, the Needleman–Wunsch algorithm is given a reward for every matched symbol and a penalty for symbols that could not be matched. The overall goal of the algorithm is to maximise the overall score. Additionally, it is allowed to deliberately introduce gaps in a sequence to increase the number of matched symbols, however, the algorithm may be penalized for each introduced gap as well this penalty can be higher if the gap is larger. The algorithm tries various alignments, including variants where some gaps are introduced into both sequences (if algorithm is allowed to introduce them), and calculate the score for each of them. In our example, one potential alignment might be

ABCDABCCD

ABADABAD -,

but also another might be

ABCDABCCD - -

ABADAB - - - AD.

For the details on how these candidate alignments are chosen, see the original work [29].

We set up the Needleman–Wunsch algorithm to be rewarded and penalized identically for each matched or mismatched symbol and introduced gap. We do not penalize the

algorithm for a gap size, as larger spacing between matched information does not affect the predictability of the sequence. Also, we do not penalize the algorithm for introducing gaps at the beginning and the end of any of the sequences. In that case, in our example, the best global alignment would be

ABCDABCCD–

ABADAB—AD,

with five matches, one mismatch and one gap introduced. We calculate the score for our metric as the number of matched transitions reduced by a penalty for each gap and mismatched transition. This score, when normalised, yields a value of another proposed metric called *global alignment (GA)*.

The last proposed metric is *iterative global alignment (IGA)*. In some cases, to find the best alignment, parts of X_{ts} subsequence are not matched and are left out of the matching process. However, we are also interested if these parts can be matched with the X_{tr} subsequence, that is if they are predictable. Therefore, we propose a modification to the Needleman–Wunsch algorithm, where the alignment process is repeated until all parts of X_{ts} subsequence are subject to matching. In our example, we would have two iterations, where the best global alignment would be

ABCDABCCD–

ABADAB—AD

in the first iteration, and

ABCDABCCD

A - - D - - - - -

in the second iteration. To calculate the score for *iterative global alignment (IGA)*, the scores from all the iterations are summed.

2.7 Stationarity and regularity

This work is inspired by the measures proposed by Teixeria et al. [20] in the attempt to explain what impacts the predictability of movement sequences. They proposed a measure of *regularity*, which along with *stationarity* explains a large portion of predictability variations. Stationarity is defined as

$$Stationarity = \frac{ST}{n}, \tag{8}$$

where ST is the number of observations when a person stays in the same location (self-transitions), that is situations when the next symbol in the sequence is the same as the previous one. Regularity is defined as

$$Regularity = \frac{n}{UQ}, \tag{9}$$

where *UQ* is the number of unique symbols in the sequence. It is important to note, that in the original work these measures were compared to predictability, while in our experiment we compare them to the accuracy of actual predictions.

We propose a modified version of stationarity measure which is calculated as the number of *self-transitions* divided by the total number of transitions in a sequence. The motivation for developing that metric is to have a stationarity measure based on the same principles as the pattern-based metrics. We name that measure *normalised stationarity (NStationarity)*.

3 Results

This section summarises our findings and presents obtained results that compose the three major contributions mentioned in the introduction. Specifically, we compare the accuracy of prediction algorithms, validate the discrepancies between predictability limit and predictions, and examine the ability of proposed metrics to explain the variability of predictions accuracy.

Ensemble decision trees and deep learning networks yield the best prediction results for our datasets, with a slight advantage of the decision trees. Although on average the predictability limit is not violated, every algorithm, including the naïve approach, surpasses the theoretical limit in a several cases. Generally, the best performing algorithm surpasses this limit in a higher number of cases than other prediction algorithms. This shows that the predictability limit cannot be compared to the accuracy of predictions.

As an alternative approach to explain variability of predictions accuracy, we propose and evaluate five candidate metrics. We base our solutions on the sequence matching and alignment algorithms, which purpose is to quantify the similarity of training and test sets used for mobility predictions. Using the R-squared (R^2) metric, we measure which of the metrics explains the most of the predictions accuracy variability. The highest values are reached for the IGA and ESR metrics. The IGA metric reaches up to $R^2 = 89.67\%$ for the next time-bin sequences and is further improved when combined with NStationarity, reaching up to $R^2 = 90.33\%$. The ESR metric performs best on the next-place sequences reaching $R^2 = 61.09\%$ of explained variability. At the same time, we show that R^2 values of regularity and stationarity are low for the accuracy of predictions, proving their inability to explain prediction accuracy variations.

3.1 Predictions accuracy and the upper bound of predictability

We start with verifying the existing discrepancies noted in the literature. Specifically, we compare the accuracy of our predictors against the theoretical limit of predictability. Table 1 presents algorithms' accuracy values obtained using the synthesised datasets. The accuracy on random sequences is almost identical in all of the algorithms, which is around 20%, showing no clear advantage of any approach. A similar situation can be observed in non-stationary sequences, as their generation process is close to random. Markovian sequences, which are less random, can be predicted with higher accuracy. It is shown by the superiority of machine learning-based methods. In all the cases, higher-order MCs do not yield better prediction accuracy than lower-order variants of this algorithm and sometimes their accuracy is even worse.

Although the average accuracy of any algorithm does not surpass the average theoretical limit, there are situations when this happens (see Table 2). Comparison of the results in

Table 1 An average accuracy of four prediction algorithms and theoretical upper bound of predictability calculated for the three types of synthetic sequences

	Random [%]	Markovian [%]	Non-stationary [%]
GRU	19.11 ± 18.6	34.46 ± 26.2	23.40 ± 15.2
RF	18.93 ± 18.9	**41.01** ± 27.7	25.22 ± 15.5
MC	19.49 ± 18.0	28.29 ± 21.7	23.12 ± 15.2
Toploc	**19.54** ± 19.7	13.97 ± 12.4	**27.90** ± 16.5
Π_{max}	40.04 ± 8.6	52.10 ± 15.7	44.05 ± 5.1

GRU is the deep neural network algorithm, RF is an ensemble decision trees method, MC is the Markov chains-based prediction algorithm, and Toploc is a baseline predictor. Bold values indicate the best result for each sequence type. Values given after the plus-minus signs are corresponding standard deviations.

Table 2 A relative number of cases when prediction accuracy of four different prediction algorithms surpasses the predictability limit calculated for the three types of synthetic sequences

	Random [%]	Markovian [%]	Non-stationary [%]
GRU	8.00	12.00	9.00
RF	7.00	**31.00**	11.00
MC	**16.00**	8.00	14.00
Toploc	8.00	2.00	**15.00**

GRU is the deep neural network algorithm, RF is an ensemble decision trees method, MC is the Markov chains-based prediction algorithm, and Toploc is a baseline predictor. Bold values indicate the highest result for each sequence type.

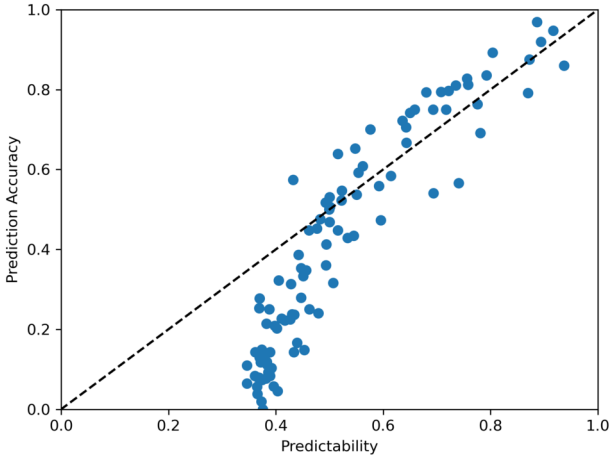

Figure 5 Predictability of sequences in comparison to the accuracy of the best predictor. The black dashed line represents the $x = y$ line, where predictability is equal to the accuracy of predictions

Tables 2 and 1 shows that algorithms that perform better on a particular type of sequence also tend to surpass the limit more often than other prediction methods. Interestingly, we find that most of the cases when algorithms surpass the limit occur when the limit value is over 55%. An example of that on Markovian sequences can be found in Fig. 5. In the case of non-stationary sequences, the fraction of predictions surpassing the limit is relatively high for all the algorithms.

Next, we verify the accuracy of predictions made on the actual mobility dataset. The dataset was processed using the next time-bin and next-place approaches into different spatial and temporal resolutions. In Table 3 we present prediction accuracy values for the processed dataset. The best performing algorithm is RF, with a deep learning-based

Table 3 An average accuracy of four prediction algorithms and theoretical upper bound of predictability calculated for the two types of mobility sequences of various spatio-temporal resolution

	Next-place [%]		
	1688 m	204m	33 m
GRU	58.58 ± 22.2	43.95 ± 19.1	33.12 ± 18.6
RF	**60.60** ± 22.8	**47.68** ± 18.6	**40.08** ± 17.9
MC	55.88 ± 22.5	39.23 ± 17.8	28.56 ± 16.3
Toploc	45.11 ± 11.2	38.26 ± 10.1	36.30 ± 10.3
Π_{max}	72.93 ± 9.6	68.28 ± 8.6	59.90 ± 10.8

	Next time-bin [%]				
	1688 m	204m	33 m	1-hour	30-minutes
GRU	94.63 ± 5.1	90.32 ± 7.3	88.73 ± 7.6	89.65 ± 7.7	92.81 ± 5.7
RF	**94.80** ± 5.0	**90.90** ± 6.0	**89.28** ± 7.0	**90.48** ± 6.6	**92.84** ± 5.4
MC	92.07 ± 6.0	87.09 ± 6.7	85.63 ± 7.5	85.93 ± 7.7	90.60 ± 5.8
Toploc	84.45 ± 17.2	78.99 ± 16.9	78.14 ± 17.22	80.22 ± 17.1	80.84 ± 17.1
Π_{max}	97.29 ± 1.5	95.89 ± 1.7	95.71 ± 1.8	95.30 ± 2.1	97.29 ± 1.2

GRU is the deep neural network algorithm, RF is an ensemble decision trees method, MC is the Markov chains-based prediction algorithm, and Toploc is a baseline predictor. Bold values indicate the best result for each sequence type. Values given after the plus-minus signs are corresponding standard deviations.

method yielding very close results, especially for the next time-bin sequences. The deep learning-based method is performing slightly worse for the next-place sequences. Other algorithms have noticeably worse accuracy. Similarly to the results of the experiment obtained on the synthetic sequences, the order of MCs is not correlated with the accuracy of predictions. This means that all the MCs models perform with almost identical accuracy. The maximum difference between the mean accuracy of evaluated MCs models is lower than 0.01%.

Although by looking at the average performance of prediction algorithms and the predictability of the dataset the limit seems not to be violated, conducting a detailed investigation of results reveals the same situation as in the case of the synthetic sequences. The ratio of predictions violating the predictability limit of the next-place sequences is positively correlated with the spatial resolution of these sequences. At a resolution of 33 metres (in the next-place sequences), predictions reaching over 40% of accuracy surpass the limit, while for the resolution of 1688 metres (in the next-place sequences) only predictions over 95% violate the limit of predictability. In the case of the next time-bin sequences, predictions reaching over 90% are surpassing the limit. However, it is important to note that for these sequences accuracy is on average higher than for the next-place sequences. As the comparison of the results in Table 4 and Table 3 shows, the fraction of predictions surpassing the limits is higher for prediction algorithms which performed better, confirming that the predictability limit is violated more often when this limit is relatively high.

3.2 Relationship between metrics and predictions accuracy

To measure the relationship between pattern matching-based measures and predictions accuracy we calculate Spearman's rank correlation, which expresses the strength of the monotonic relationship between variables. Then, we determine the level up to which prediction accuracy can be explained by these measures using R^2 metric for the best regression model fit. As a reference, we conduct the same tests using proposed in the literature

Table 4 A relative number of cases when prediction accuracy of four different prediction algorithms surpasses the predictability limit calculated for the two types of mobility sequences of various spatio-temporal resolution

	Next-place [%]			Next time-bin [%]				
	1688 m	204m	33 m	1688 m	204m	33 m	1-hour	30-minutes
GRU	**25.67**	5.33	3.33	26.50	10.33	8.00	16.11	13.78
RF	25.30	**6.33**	**6.67**	**27.00**	**11.11**	**8.50**	**17.44**	13.67
MC	18.00	6.00	5.33	18.17	3.33	3.50	8.44	**16.00**
Toploc	18.00	0.67	4.33	21.83	8.33	8.33	8.22	10.00

GRU is the deep neural network, RF is an ensemble decision trees method, MC is the Markov chains-based prediction algorithm, and Toploc is a baseline predictor. Bold values indicate the highest result for each sequence type.

Table 5 Spearman's correlation of the evaluated metrics and accuracy of predictions calculated on the three types of synthetic sequences

	Random [%]	Markovian [%]	Non-stationary [%]
ESR	88.21	**95.76**	**86.67**
SR	90.14	74.80	82.34
DR	79.50	88.76	73.13
Regularity	64.88	33.01	62.30
Stationarity	87.99	–20.78	82.66
NStationarity	87.99	–20.78	82.66
GA	**90.60**	85.51	85.95
IGA	88.04	84.97	83.56
Π_{max}	60.56	93.29	77.04

ESR is an *equally sparse repeatability*, SR is a *sparse repeatability*, DR is a *dense repeatability*, GA is a *global alignment* measure, and IGA is an *iterative global alignment* measure. All the correlations are significant at the level of $p < 0.001$ (significance of correlation between stationarity and accuracy of markovian sequences is $p < 0.03$). Bold values indicate the best result for each sequence type.

metrics, which are stationarity and regularity. These metrics were originally used to explain predictability variability.

Table 5 presents Spearman's correlation values observed on the synthetic sequences. The accuracy of predictions is strongly correlated with all the proposed measures. The most correlated measures on average are ESR, GA, and IGA (in that order). The superiority of the ESR metric is clearly seen when applied to Markovian sequences. In other cases (for random and non-stationary sequences), correlation values associated with GA and IGA metrics are similar to the correlation of ESR. Stationarity also has a large impact on the predictability of random and non-stationary sequences, however, not on Markovian sequences which have a small number of self-transitions. Regularity is on average the least correlated measure. Π_{max} seems to be correlated strongly only with Markovian sequences, while in other cases this correlation is relatively low.

Spearman's correlation between predictions accuracy and metrics calculated on real mobility data are presented in Table 6. Tests were conducted on the two types of sequences of various spatio-temporal resolution. For the next-place sequences, Spearman's correlation value for the ESR metric is the highest (75% on average) for all the spatial resolutions of data. The correlation value slightly decreases with the spatial resolution increase, which is caused by the higher number of unique symbols present in the sequence. GA metric has the second-highest correlation, while regularity is the least correlated metric. By definition, all self-transitions are removed in the next-place sequences, therefore stationarity is not correlated with this type of movement sequence (it is always equal to one). For the

Table 6 Spearman's correlation of the evaluated metrics and accuracy of predictions calculated on the two types of mobility sequences of various spatio-temporal resolution

	Next-place [%]			Next time-bin [%]				
	1688 m	204m	33 m	1688 m	204m	33 m	1-hour	30-minutes
ESR	**75.38**	**75.26**	**74.35**	76.42	68.24	71.55	73.85	69.25
SR	52.92	62.17	66.25	74.80	79.73	86.35	76.42	83.04
DR	48.83	56.41	55.49	80.93	74.28	73.60	75.89	72.65
Regularity	51.48	46.18	57.93	54.87	53.26	56.15	53.04	56.45
Stationarity	–	–	–	76.55	68.00	65.30	69.94	66.06
NStationarity	–	–	–	76.55	68.00	65.30	69.94	66.06
GA	68.18	71.79	70.39	90.62	89.61	92.50	89.70	91.53
IGA	67.12	69.32	69.85	**92.19**	**91.34**	**94.50**	**90.52**	**93.86**
Π_{max}	63.16	62.64	62.05	74.89	68.08	64.96	70.93	67.70

ESR is an *equally sparse repeatability*, SR is a *sparse repeatability*, DR is a *dense repeatability*, GA is a *global alignment* measure, and IGA is an *iterative global alignment* measure. All the correlations are significant at the level of $p < 0.001$. Bold values indicate the best result for each sequence type.

next time-bin sequences, Spearman's correlation values of IGA and GA metrics are the highest by a large margin, almost 10% over the third-highest correlated SR metric. Other LCS-based metrics and stationarity are strongly correlated with prediction accuracy on an average level of around 70%. Similarly to the next-place sequences, regularity is the least correlated metric on average. The average correlation value across all spatio-temporal resolutions and sequence types is the highest for the IGA metric, reaching over 83%. In all of the cases, correlation between Π_{max} and prediction accuracy is smaller than for ESR, IGA, and GA.

To validate the extent up to which these metrics explain predictions accuracy variability, we fit various regression functions to the data and calculate the coefficient of determination R^2 for each of these fits. First, we determine the type of functional dependency between metrics and predictions accuracy. We find that the relationship between accuracy and all the LCS-based metrics is exponential, regularity and stationarity have a logarithmic relationship, and the GA and IGA metrics are linearly dependent on the accuracy variable. Next, we fit a regression model modelling those functional relationships and calculate the R^2 for each fit. To avoid overfitting, R-squared is calculated using a 5-fold cross-validation approach. Moreover, because NStationarity was weakly correlated with the other metrics, we combine all the metrics with NStationarity by fitting a multivariate regression model. For those combinations, adjusted R-squared is calculated. The results are presented in Table 7 and Table 8.

Tests conducted on the synthetic sequences reveal that the ESR metric, together with NStationarity, explains the accuracy of predictions made on Markovian sequences best, reaching $R^2 > 90\%$. The accuracy of predictions on random sequences is explained well by all of the metrics, especially when NStationarity is involved. Predictions made on Markovian sequences appear to be more difficult to explain than random sequences because only ESR and GA metrics combined with NStationarity reach $R^2 > 80\%$. Interestingly, regularity combined with stationarity is unable to explain the accuracy of predictions made on Markovian sequences. Similarly to Markovian sequences, the accuracy of predictions made on non-stationary sequences is explained well by ESR, GA, and IGA metrics, as well as all the combinations where NStationarity is involved. The Π_{max} metric, in most cases, performs worse than ESR, IGA, and GA metrics, but it yields good results for highly predictable Markovian sequences.

Table 7 An average proportion of variation of accuracy of predictions made on the synthetic sequences explained by the regression models fit to the evaluated metrics

	Random [%]	Markovian [%]	Non-stationary [%]
ESR	91.11	89.98	**85.08**
SR	78.81	45.81	42.98
DR	78.09	73.85	59.14
ESR + NStationarity	93.03	**91.53**	83.94
SR + NStationarity	92.08	58.87	81.16
DR + NStationarity	91.55	74.71	81.40
Regularity + Stationarity	91.32	1.25	73.83
GA	91.18	67.41	79.17
GA + NStationarity	**93.04**	81.56	82.38
IGA	88.12	65.24	75.54
IGA + NStationarity	92.35	77.61	81.78
Π_{max}	80.76	89.22	76.44

ESR is an *equally sparse repeatability*, SR is a *sparse repeatability*, DR is a *dense repeatability*, GA is a *global alignment* measure, and IGA is an *iterative global alignment* measure. All the R^2 values are significant at the level of $p < 0.001$. Bold values indicate the best result for each sequence type.

Table 8 An average proportion of accuracy variation explained by the regression models fit to the evaluated metrics presented for the two types of mobility sequences of varied spatio-temporal resolutions

	Next-place [%]			Next time-bin [%]				
	1688 m	204m	33 m	1688 m	204m	33 m	1-hour	30-minutes
ESR	**57.71**	**61.09**	56.97	38.78	27.02	35.69	38.18	33.20
SR	26.45	41.65	47.63	47.61	29.51	46.80	42.51	43.69
DR	19.34	30.95	32.06	50.57	44.87	44.28	50.21	40.92
ESR + NStationarity	–	–	–	47.92	45.03	47.14	52.88	42.95
SR + NStationarity	–	–	–	75.63	48.92	67.64	72.68	62.60
DR + NStationarity	–	–	–	48.91	52.28	47.99	57.46	47.10
Regularity + Stationarity	23.79	24.36	31.89	43.28	38.12	38.49	40.62	36.44
GA	55.27	55.32	**56.98**	75.69	51.84	61.50	64.91	57.09
GA + NStationarity	–	–	–	80.55	65.48	66.49	74.56	66.41
IGA	52.97	52.74	55.95	86.23	85.44	89.67	84.44	88.03
IGA + NStationarity	–	–	–	**88.55**	**86.49**	**90.33**	**87.53**	**89.23**
Π_{max}	38.32	43.39	42.10	48.56	48.57	47.70	52.92	43.63

ESR is an *equally sparse repeatability*, SR is a *sparse repeatability*, DR is a *dense repeatability*, GA is a *global alignment* measure, and IGA is an *iterative global alignment* measure. All the R^2 values are significant at the level of $p < 0.001$. Bold values indicate the best result for each sequence type.

Predictions made on the real movement sequences proved to be less explainable than predictions made using synthetic sequences. Predictions made on the next-place sequences are best explained by the ESR metric, with an average value reaching $R^2 > 58\%$ (see Fig. 6 for the example). The GA and IGA are closely following, reaching an average value of $R^2 > 53\%$. Predictions made on the next-time bin sequences are best explained by the IGA metric, outperforming other metrics by a large margin. Interestingly, the GA metric is performing worse than expected given its strong correlation with the accuracy variable (see GA correlations in Table 6). Combining metrics with NStationarity improved their ability to explain the variability of predictions accuracy, especially in the case of the SR metric. However, NStationarity increased R^2 for the IGA metric only slightly, which means that IGA already incorporates the majority of information delivered by the stationarity-based metric. The combination of regularity and stationarity is the worst performing metric in most cases. Performance of the Π_{max} metric on human mobility sequences seems to be poor, which proves that Π_{max} should not be compared

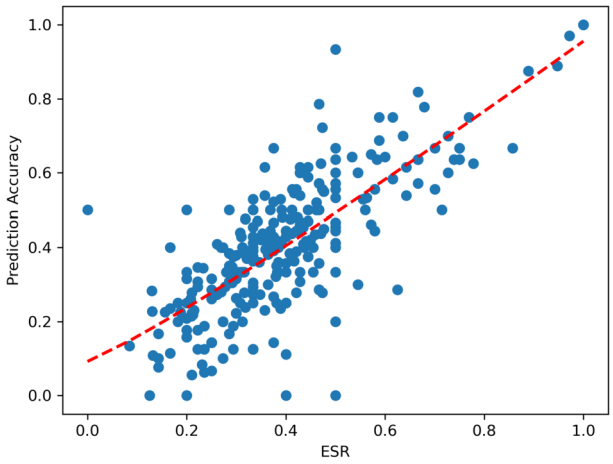

Figure 6 *The equally sparse repeatability* metric in comparison to the accuracy of the best predictor. The red dashed line represents a regression model fit to the data

Table 9 An average proportion of predictability variations explained by the regression models fit to the evaluated metrics calculated for the two types of mobility sequences

	Next-place [%]	Next time-bin [%]
ESR	43.23	40.21
SR	42.56	4.61
DR	45.76	51.50
ESR + NStationarity	–	82.23
SR + NStationarity	–	80.03
DR + NStationarity	–	83.32
Regularity + Stationarity	58.91	81.40
GA	**64.50**	32.67
GA + NStationarity	–	**83.13**
IGA	62.70	41.26
IGA + NStationarity	–	80.53

ESR is an *equally sparse repeatability*, SR is a *sparse repeatability*, DR is a *dense repeatability*, GA is a *global alignment* measure, and IGA is an *iterative global alignment* measure. All the R^2 values are significant at the level of $p < 0.001$. Bold values indicate the best result for each sequence type.

with the accuracy of predictions. The average R^2 value across all spatio-temporal resolutions and sequence types is the largest for the IGA metric.

3.3 Relationship between metrics and predictability

The combination of regularity and stationarity was originally intended to explain fluctuations of predictability in mobility data, rather than the accuracy of predictions. Therefore, we check the R^2 values for all the metrics presented above in relation to the predictability of movement sequences. Table 9 presents the average value of R^2 for all the types of sequences. The results confirm the findings of previous works, that regularity combined with stationarity explains predictability well in both types of movement sequences. Although the R^2 values of regularity and stationarity are not the highest, they are close to the highest values. The results suggest that the stationarity is uplifting the R^2 value in the case of next time-bin sequences and explains the majority of predictability variations.

4 Discussion

Since the publication of Song et al. [8], predictability of human mobility has been subject to intensive studies which aimed to improve our understanding of human mobility behaviour and quantify the degree of randomness in the movement. Only a few times, the outcomes of these estimations were compared to actual predictions [7, 14], yielding a surprising result of prediction accuracy surpassing the theoretical upper bound. We investigated that phenomenon in more detail and observed the same result on the synthetic and real mobility sequences. Specifically, we found that predictions accuracy surpasses the theoretical upper bound of predictability when both of these values are high, that is when a sequence is highly predictable and the algorithm is able to capture the complex structure of the sequence. This finding aligns well with the results reported by Kulkarni et al. [14], who found that sophisticated algorithms, such as deep neural networks, are surpassing predictability values. Algorithms of higher overall accuracy were surpassing the upper bound of predictability more often than algorithms of lower prediction accuracy. Such results suggest the unsuitability of the predictability estimation theory to human movement sequences. Few authors raised concerns regarding the calculation process, specifically the Lempel–Ziv algorithm. Kulkarni et al. [14] found that this algorithm does not capture long-range structural correlations present in movement sequences, which in result decreases the predictability estimation. Moreover, Lu et al. [7] observed that predictions made on non-stationary sequences surpass the predictability limit, which aligns with the fact that the Lempel–Ziv algorithm is proved to work accurately only on stationary sequences [13].

However, we argue that the accuracy of predictions should not be compared to the predictability limit at all, as these values are based on different parts of movement sequences. Predictability is calculated as a single metric for the whole sequence, while accuracy is obtained only on a part of the data which had to be split to provide the learning material for the prediction algorithm. For example, in our experiment outcomes of such situations can be observed when even the naïve algorithm was able to surpass the predictability limit. In these cases, the test set was consisting of only one symbol which is easy to predict and results in perfect accuracy, while the training set had additional, unique symbols in it, which decreased the predictability estimation. Also, it is important to note that predictability estimated only on a test dataset would also not be comparable to prediction results as prediction algorithms use information from training sets which, in that case, would be omitted by the predictability estimator.

In their work, Teixeira et al. [20] attempted to explain predictability fluctuations through other, easier to interpret, metrics. As we confirm (see Table 9), their two simple metrics are able to explain the majority of predictability variability. Although, stationarity is not applicable to the next-place sequences (because by the definition it is always equal to one), regularity alone is able to explain almost 60% of the predictability variability. However, these metrics are poorly explaining the variability of predictions accuracy, which is another argument for the incomparability of predictability and predictions.

As an alternative, we proposed a set of metrics based on pattern-matching algorithms. These algorithms were modified to search for identical transitions (pairs of symbols), instead of reoccurring symbols. This increased R^2 values in all the cases. This shows that repeatability of transitions is another important factor influencing the accuracy of predictions. We applied our metrics on the two types of datasets: synthetically generated se-

quences and actual mobility data processed into the two types of sequences (next time-bin and next-place) aggregated to various spatio-temporal scales. The IGA metric, which is based on global sequence alignment, explained on average over 88% of variability in predictions accuracy in the next time-bin sequences. The ESR metric, based on the longest common subsequence matching, was able to explain almost 59% of variability in the accuracy of predictions for the next-place sequences.

Through the analysis of the correlations between the accuracy of predictions and various metrics, we found that stationarity is strongly correlated with predictions (see Table 5 and 6) and usually weakly correlated with other metrics, therefore, we decided to combine its modified variant with our metrics in the regression models. Stationarity is useful when analysing next time-bin sequences where the number of self-transitions is high. Multiple works found the cause of the uplifted predictability of next time-bin sequences in the high number of self-transitions [11, 17, 20], and as expected it raised R^2 value in our multivariate regression models. Therefore, stationarity is an important factor influencing also the accuracy of predictions in the next time-bin sequences.

Among all the proposed metrics, IGA combined with stationarity was performing the best on average, but IGA alone was also able to explain a large portion of the variability in prediction results. The Needleman–Wunsch algorithm is able to align transitions, including self-transitions, between sequences, hence, adding stationarity to the model did not result in the large increase of R^2 value. Also, we found that applying a penalty to the IGA and GA metrics scoring, for every gap or mismatched transition, further increases R^2 values. Among the LCS-based measures, the ESR metric was performing best and was the overall best performing metric on the next-place sequences. The reason for that performance was the constraint imposed on the ESR metric forcing matched transitions to be identical (identically spaced), which can be observed as a superiority of ESR over SR, especially in the next-place sequences. Such transition, present in a training set, should be predictable for the algorithm in a test set.

We investigated in detail sequences in which ESR performed poorly and found that ESR is underestimating predictability in situations when the longest matched pattern is much shorter than the test sequence. This causes ESR not to capture all the reoccurring transitions which contribute to the increased predictability of a sequence. These transitions are usually overlapping, which makes that task non-trivial, hence our attempts to merge these detected transitions in a single sequence failed. On the other hand, sequence alignment used in the IGA metric is free of such problems, as the whole sequence is subject to optimal matching. However, in contrast to the ESR metric, the IGA metric is unable to capture reoccurring transitions that are separated by other symbols. We found that in the next-place sequences, which are short and where all self-transitions are removed, such transitions are often the only transitions that could be matched between sequences and which were predicted by the prediction algorithm. In such cases, IGA underestimated predictability.

This work can be further extended by developing even more robust metrics based on our current findings. One solution may be to merge the best performing metrics, which are ESR and IGA, which should help to overcome their identified limitations. Although the primary goal of this work was to identify factors influencing the accuracy of predictions made on human movement sequences, our solution has other applications which we plan to pursue in the future. First of all, a quick estimation of the potential predictability of a

sequence may serve as a reference value during the data preprocessing and filtering stage. Assessment of potential predictability of movement sequence, in combination with quantification of information loss (due to data preprocessing), may be used to find optimal data preprocessing methods that maximise retained information and data predictability. This would minimise the bias introduced into the data through inattentive processing, such as accidental split of stay-regions, which significantly influences the outcomes of analyses and modelled mobility. Also, transition detection algorithms developed during this experiment may be used to construct a new sequence prediction algorithm. Such an algorithm would scan the training set in search of repeated transitions (for example using the approach from the ESR metric), which could be later used at the prediction stage when a similar series of transitions appear. In contrast to many machine learning algorithms, such a method would be much more transparent.

5 Conclusions

In this work, we evaluate and confirm the discrepancies between the theoretical limit of predictability of human mobility and the results of the actual predictions. In response, we attempt to develop a pattern matching-based metric that will help quickly evaluate the actual predictability of movement sequences and serve as an alternative solution to the predictability estimation theory. We propose five candidate metrics and evaluate them on the results of actual predictions. The key findings of this work are:

- We find that the accuracy of sophisticated prediction models surpasses the theoretical upper bound of predictability;
- We confirm that the best of the proposed metrics, that are IGA and ESR, explain on average over 88% of variability in predictions accuracy in the next time-bin sequences and almost 59% of variability in the accuracy of predictions for the next-place sequences;
- We find the regularity and stationarity metrics proposed by Teixeira et al. [20] explain the accuracy of predictions worse than any of our metrics. On the other hand, we confirm that regularity and stationarity are able to explain a major portion of the variability of the predictability measure proposed by Song et al. [8], demonstrating that the predictability and accuracy of predictions should not be compared.

A good performance of our metrics implicates that similarity of transitions present in the training and test set highly impacts the predictability of the movement sequence. Moreover, relative spacing (the number of other symbols separating the transition) of these transitions is important. We confirm that stationarity is a significant factor impacting the predictability of the next time-bin sequences.

We identify shortcomings of our metrics and for future works, we propose merging the IGA and ESR metrics into a single measure able to perform best for all types of movement sequences. Their abilities are complementary, and we expect their combination to improve our results. Our metrics can be applied for a quick estimation of the potential predictability of movement sequences, which combined with quantification of information loss caused by data preprocessing, might be used to optimise data preprocessing algorithms. This would help to maximise the amount of information retained in the data and avoid potential biases caused by inattentive data processing.

Although, we focus on the human mobility studies from which predictability theory stems, our findings can be applied beyond that area. Our solution can be applied to mea-

sure the predictability of any type of series of symbols and possibly can be expanded to work with discrete sequences.

Supplementary information

Supplementary information accompanies this paper at https://doi.org/10.1140/epjds/s13688-022-00356-4.

Additional file 1. A complete results of evaluations. A package of csv files containing the full results: accuracy of predictions and metrics values for every sequence in the dataset. It can be accessed here: 10.5281/zenodo.5788701.
Additional file 2. A Python code used to calculate presented results. All the experiment was implemented in Python as a part of HuMobi programming library developed at the Institute of Geodesy and Geoinformatics at Wrocław University of Environmental and Life Sciences. This library is available at GitHub repository: 10.5281/zenodo.6817269.

Acknowledgements
The authors would like to thank the Spyrosoft S.A. company for sharing the data for this study.

Funding
The research was financed under the National Science Centre, Poland research grant "Explanation and mitigation of the bias in human mobility predictions" no. 2021/41/N/HS4/03084, the National Centre for Research and Development research Project No. POIR.01.01.01-00-0641/20, and the Leading Research Groups support project from the subsidy increased for the period 2020–2025 in the amount of 2% of the subsidy referred to Art. 387 (3) of the Law of 20 July 2018 on Higher Education and Science, obtained in 2019.

Availability of data and materials
The human mobility datasets analysed during the current study are not publicly available due to privacy restrictions. Other datasets and Python code are available in Additional files 1 and 2.

Declarations

Competing interests
The authors declare that they have no competing interests.

Author contribution
KS conceived the experiment, conducted the experiment, analysed the results, and wrote the research article. KSN and WR analysed the results. All authors reviewed the manuscript. All authors read and approved the final manuscript.

Author details
[1]Institute of Geodesy and Geoinformatics, Wrocław University of Environmental and Life Sciences, Norwida 25, 50-375 Wrocław, Poland. [2]School of Environment, The University of Auckland, Aukcland CBD, 1010 Auckland, New Zealand. [3]Urban Big Data Centre, University of Glasgow, 7 Lilybank Gardens, G12 8RZ Glasgow, Scotland, UK.

Publisher's Note

Received: 17 December 2021 Accepted: 14 July 2022 Published online: 30 July 2022

References

1. Barbosa-Filho H, Barthelemy M, Ghoshal G, James CR, Lenormand M, Louail T, Menezes R, Ramasco JJ, Simini F, Tomasini M (2018) Human mobility: models and applications. Phys Rep 734(29):1–74. https://doi.org/10.1016/j.physrep.2018.01.001. arXiv:1710.00004
2. Lu X, Bengtsson L, Holme P (2012) Predictability of population displacement after the 2010 Haiti earthquake. Proc Natl Acad Sci USA 109(29):11576–11581. https://doi.org/10.1073/pnas.1203882109
3. Frias-Martinez E, Williamson G, Frias-Martinez V (2011) An agent-based model of epidemic spread using human mobility and social network information BT. In: Proceedings of the international conference on social computing (SocialCom)
4. Calabrese F, Ferrari L, Blondel VD (2014) Urban sensing using mobile phone network data: a survey of research. ACM Comput Surv 47(2):1–20. https://doi.org/10.1145/2655691
5. Smolak K, Siła-Nowicka K, Delvenne JC, Wierzbiński M, Rohm W (2021) The impact of human mobility data scales and processing on movement predictability. Sci Rep 11(1):1–10. https://doi.org/10.1038/s41598-021-94102-x
6. Ahas R, Silm S, Järv O, Saluveer E, Tiru M (2010) Using mobile positioning data to model locations meaningful to users of mobile phones. J Urban Technol 17(1):3–27. https://doi.org/10.1080/10630731003597306
7. Lu X, Wetter E, Bharti N, Tatem AJ, Bengtsson L (2013) Approaching the limit of predictability in human mobility. Sci Rep 3:1–9. https://doi.org/10.1038/srep02923
8. Song C, Qu Z, Blumm N, Barabási AL (2010) Limits of predictability in human mobility. Science 327(5968):1018–1021. https://doi.org/10.1126/science.1177170. 0307014

9. Wang J, Kong X, Xia F, Sun L (2019) Urban human mobility: data-driven modeling and prediction. ACM SIGKDD Explor Newsl 21(1):1–19
10. Fano RM (1961) Transmission of information: a statistical theory of communications. Am J Phys 29(11):793–794
11. Xu P, Yin L, Yue Z, Zhou T (2019) On predictability of time series. Phys A, Stat Mech Appl 523:345–351. https://doi.org/10.1016/j.physa.2019.02.006
12. Ziv J, Lempel A (1978) Compression of individual sequences via variable-rate coding. IEEE Trans Inf Theory 24(5):530–536. https://doi.org/10.1109/TIT.1978.1055934
13. Teixeira D, Almeida J, Viana AC, Teixeira D, Almeida J, Carneiro A, Understanding V, Teixeira D, Almeida JM, Viana AC (2021) Understanding routine impact on the predictability estimation of human mobility To cite this version: HAL Id: hal-03128624 Understanding routine impact on the predictability estimation of human mobility
14. Kulkarni V, Mahalunkar A, Garbinato B, Kelleher JD (2019) Examining the limits of predictability of human mobility. Entropy 21(4):1–27. https://doi.org/10.3390/e21040432
15. Smith G, Wieser R, Goulding J, Barrack D (2014) A refined limit on the predictability of human mobility. In: 2014 IEEE international conference on pervasive computing and communications, PerCom 2014, pp 88–94. https://doi.org/10.1109/PerCom.2014.6813948
16. Ikanovic EL, Mollgaard A (2017) An alternative approach to the limits of predictability in human mobility. EPJ Data Sci 6(1). https://doi.org/10.1140/epjds/s13688-017-0107-7. arXiv:1608.06419
17. Cuttone A, Lehmann S, González MC (2018) Understanding predictability and exploration in human mobility. EPJ Data Sci 7(1). https://doi.org/10.1140/epjds/s13688-017-0129-1. arXiv:1608.01939
18. Liu LF, Hu HP, Deng YS, Ding ND (2014) An entropy measure of non-stationary processes. Entropy 16(3):1493–1500. https://doi.org/10.3390/e16031493
19. Lin M, Hsu WJ, Lee ZQ (2012) Predictability of individuals' mobility with high-resolution positioning data. In: UbiComp'12 – proceedings of the 2012 ACM conference on ubiquitous computing, pp 381–390. https://doi.org/10.1145/2370216.2370274
20. Do Couto Teixeira D, Viana AC, Alvim MS, Almeida JM (2019) Deciphering predictability limits in human mobility. In: GIS: proceedings of the ACM international symposium on advances in geographic information systems, pp 52–61. https://doi.org/10.1145/3347146.3359093
21. Teixeira D, Viana AC, Alvim M, Almeida J, Teixeira D, Viana AC, Alvim M, Almeida J (2019) Deciphering Predictability Limits in Human Mobility To cite this version: HAL Id. hal-02286128
22. Jiang S, Fiore GA, Yang Y, Jiang S, Ferreira J, Frazzoli E, González MC A review of urban computing for mobile phone traces terms of use a review of urban computing for mobile phone traces: current methods. Challenges and Opportunities, 1–9. https://doi.org/10.1145/2505821.2505828
23. Hahsler M, Piekenbrock M, Doran D (2019) dbscan: fast density-based clustering with R. J Stat Softw 91(1):1–30. https://doi.org/10.18637/jss.v091.i01
24. Alessandretti L, Aslak U, Lehmann S (2020) The scales of human mobility. Nature 587(7834):402–407. https://doi.org/10.1038/s41586-020-2909-1
25. Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint. arXiv:1412.3555
26. Tyralis H, Papacharalampous G, Langousis A (2019) A brief review of random forests for water scientists and practitioners and their recent history inwater resources. Water 11(5):910. https://doi.org/10.3390/w11050910
27. Qiao Y, Si Z, Zhang Y, Ben F, Zhang X (2018) Neurocomputing a hybrid Markov-based model for human mobility prediction. Neurocomputing 278:99–109. https://doi.org/10.1016/j.neucom.2017.05.101
28. Schreckenberger C, Beckmann S, Bartelt C (2018) Next place prediction: a systematic literature review. In: Proceedings of the 2nd ACM SIGSPATIAL international workshop on prediction of human mobility, PredictGIS 2018, pp 37–45. https://doi.org/10.1145/3283590.3283596
29. Needleman SB, Wunsch CD (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 48(3):443–453
30. Chen Y, Wan A, Liu W (2006) A fast parallel algorithm for finding the longest common sequence of multiple biosequences. BMC Bioinform 7(4):1–12
31. Li D, Becchi M (2012) Multiple pairwise sequence alignments with the needleman-wunsch algorithm on gpu. In: 2012 SC companion: high performance computing, networking storage and analysis. IEEE Comput. Soc., Los Alamitos, pp 1471–1472
32. Kociumaka T, Radoszewski J, Starikovskaya T (2019) Longest common substring with approximately k mismatches. Algorithmica 81(6):2633–2652