Synthesis of Longitudinal Human Location Sequences: Balancing Utility and Privacy

MAYA BENAROUS, ERAN TOCH, and IRAD BEN-GAL, Department of Industrial Engineering, Tel Aviv University

People's location data are continuously tracked from various devices and sensors, enabling an ongoing analysis of sensitive information that can violate people's privacy and reveal confidential information. Synthetic data have been used to generate representative location sequences yet to maintain the users' privacy. Nonetheless, the privacy-accuracy tradeoff between these two measures has not been addressed systematically. In this article, we analyze the use of different synthetic data generation models for long location sequences, including extended short-term memory networks (LSTMs), Markov Chains (MC), and variable-order Markov models (VMMs). We employ different performance measures, such as data similarity and privacy, and discuss the inherent tradeoff. Furthermore, we introduce other measurements to quantify each of these measures. Based on the anonymous data of 300 thousand cellular-phone users, our work offers a road map for developing policies for synthetic data generation processes. We propose a framework for building data generation models and evaluating their effectiveness regarding those accuracy and privacy measures.

CCS Concepts: • **Security and privacy → Data anonymization and sanitization**; • **Computing methodologies** → *Neural networks*; • **Information systems → Location based services;**

Additional Key Words and Phrases: Synthetic data, location sequences, privacy, long short term memory network (LSTM)

ACM Reference format:
Maya Benarous, Eran Toch, and Irad Ben-Gal. 2022. Synthesis of Longitudinal Human Location Sequences: Balancing Utility and Privacy. *ACM Trans. Knowl. Discov. Data.* 16, 6, Article 118 (July 2022), 27 pages.
https://doi.org/10.1145/3529260

1 INTRODUCTION

Location data are collected on a massive scale by mobile apps, cellular service providers, public infrastructure providers, and connected cars. The accumulation of location data allows detailed and longitudinal location sequences that reflect people's schedules and whereabouts for long periods. Long location sequences were shown to be very beneficial for scientific and industrial applications: from urban planning [34, 48, 50, 52, 52] to traffic and mobility planning [47, 47, 51]. At the same time, location sequences can pose a threat to privacy [14, 39]. Location sequence data are sensitive

This work was partially supported by the Israel Ministry of Science (grant number 3-12460) and the Koret Fund for Digital Living 2030.
Authors' address: M. Benarous, E. Toch, and I. Ben-Gal, Department of Industrial Engineering, Tel Aviv University, Tel Aviv 69978, Israel; emails: rdt.maya@gmail.com, erant@tauex.tau.ac.il, and bengal@tau.ac.il.

1556-4681/2022/07-ART118 $15.00
https://doi.org/10.1145/3529260

because people's whereabouts might reveal confidential personal information or allow for the reidentification of individuals in a database [12, 24, 40]. For example, it has been shown that just four spatiotemporal points can be enough to uniquely identify 95% of individuals in a location sequence dataset [14]. Aside from the user's identity, data collectors can create an entire user profile, including the user's routine, acquaintances, favorite locations, and much more. This opens a gate to many privacy threats that can be more intruding than just identifying a user [50].

Using synthetic data to allow privacy-preserving location analysis was used in the past for generating data in the form of location trajectories [26, 30, 45], which are concise representations of behaviors of moving objects as sequences of regions frequently visited within a typical travel time. The difference between generating trajectories and location sequences is whether the generation model needs to process the timestamp. The added specificity and details of long location sequences add additional complexity and may require new generation methods than the ones used in the literature. Since we are generating sequence-based data, we use models that have temporal properties. We use a **long short-term memory** (**LSTM**) network to generate synthetic data. LSTM is a neural network-based model with loops that allows information to persist. Since the model is built from a neural network, it can learn complex patterns in data, enabling it to achieve high-quality synthetic data. Another sets of models we employ are Markov-based models since they are the most used models in mobility modeling and analysis. These models create good representations of the transition probabilities, and we use them to simulate mobility patterns. We use the **position weight matrix** (**PWM**), **Markov chains** (**MC**), and **variable-order Markov** (**VMM**) chains , which have proven to yield positive results [5, 20, 36].

We use a varied set of measures to assess the performance of the different models: privacy, statistical similarity, per-instance similarity, and diversity.[1] Where we could, we used known standards, such as the framework of tracking attacks [43] to measure privacy. In some cases, such as for diversity, there are not many known measures, so we created a new estimate based on **Bilingual Evaluation Understudy** (**BLEU**) from **Natural Language Processing** (**NLP**) [19, 32, 37]. We show how each performance measure contributes to our understanding of the synthetic data and the performance of the generation model. We use a Pareto analysis for the performance measures to decide which generation model is best suited for a specific objective. Additionally, we use a weighted average of the most relevant performance measures to choose the best model for the current purpose. While the weighted average gives a definitive score for each model, the Pareto analysis helps analyze if the score accurately indicates the model's success. We explore how these evaluation methods are needed to decide which model should be used [41].

2 BACKGROUND

2.1 Location Data Synthesis

Synthetic data generation can provide valuable insights about human mobility while preserving confidentiality. Unlike obfuscation [3] and differential privacy [16], the data do not contain any actual information about any real user. Studies that were focused on generating synthetic location data (as shown in Table 5) were based on datasets of trajectories: vectors of tuples wherein each tuple is a location and the corresponding timestamp. Short time trajectories are mostly for traffic planning and similar applications [26, 30, 45, 53]. Location sequences are a longitudinal sequence of locations, which are mostly used in behavioral applications that analyze people's routines [50]. For example, Ben Gal et al. [8] clustered human long location sequences to identify lifestyle choices.

[1]The code for performing the evaluation of the methods can be found at https://github.com/iWitLab/evaluating_synthetic_data.

There are several models for data synthesis that have been suggested in the literature. One of the first data generation models that were proposed is the PWM [23]. The PWM is a basic model that measures the distribution of users over different locations. It is mostly used in the field of genetics [13, 29, 44]. On the other hand, the model does not measure sequential information, which is the main quality of our data, for example, the conditional probability of being in a location given a history of locations. That is why this model is often referred to as a zero-order Markov model. MC models with higher order memory can help in modeling sequential information. In this model, an object can move between different states, and the probability of the current state depends only on the previous K states [46]. In location data, each state is a location, and each object is a user. The probability of the next location would depend on the last K locations. However, this type of model suffers from the exponential growth of the state space when using a high order (large K). Therefore, a Markovian model that aims at addressing this problem is the VMM, which, on one hand, can model sequential data of considerable complexity [5, 8]. However, on the other hand, the computation of the algorithm still takes considerable time. These models have been shown to yield good results, yet such techniques also ignore the presence of long-range dependencies inherent to human mobility, including non-Markovian characters.

The continuous development of deep learning has led many researchers to use deep models to generate synthetic data. Many studies have used **generative adversarial networks (GANs)** [31, 42, 45]. When dealing with sequential data, the most prominent model in deep learning is the LSTM network. The model was first suggested in 1997 by Hochreiter, and Schmidhuber [25], and since then, it has been used in many applications, such as generating trajectories [1, 2, 42], and text [18, 49]. LSTM has feedback connections and can process single data points (such as images) and entire sequences of data (such as speech or video). A standard LSTM unit comprises a cell, an input gate, an output gate, and a forget gate. The cell remembers values over arbitrary time intervals, and the three gates regulate the flow of information into and out of the cell. These gates can learn which data in a sequence are essential to keep or throw away. By doing so, it can pass relevant information down the long chain of sequences to make predictions.

2.2 Evaluating Synthetic Data

Evaluating the quality of synthetic data is a challenging task for both theoretical and practical reasons. When generating synthetic images, there is a clear set of measures for evaluating the quality of a generated image, such as the inception score and maximum mean discrepancy [10]. These measures are used repeatedly and are a common standard when comparing generation models since they have proven similar results for human tests. However, it is not always clear how to evaluate synthetic data in our field, and every article uses different measures to do so. As Table 5 shows, most studies use up to two performance measures to evaluate the quality of synthetic data, and the performance measures are usually used without any justification from prior works. However, each of these performance measures serves a specific purpose, and overlooking one can lead to misleading results.

The first performance measure considered in the literature is statistical similarity. Similarity can be measured through aggregated statistics, such as comparing the distributions of the most visited locations, travel time, and work hours [9, 22]. However, the similarity of the statistics between the synthetic and original data does not guarantee that each location sequence in the generated data will be similar to the original location sequences. A visual test sometimes accompanies the statistical similarity, simply going over the synthetic data with a human eye to see if the generated data looks similar to the original data. [38, 45, 53].

Other studies focused on the performance measures of per-instance similarity and diversity [26]. Per-instance similarity means that each location sequence is similar to the location sequences

in the original data. Diversity means that the distribution of the synthetic data is the same as that of the original data. However, there can be a tradeoff between per-instance similarity and diversity: a dataset could have a high similarity score by repeating one good instance over and over again, which, in turn, would wield a shallow diversity score. Some studies focused on the performance measure of privacy [31, 42], comparing it to the similarity of the data. Previous work had shown the tradeoff between per-instance similarity and privacy, and this is also supported in our work [31, 42]. Synthetic data with high per-instance similarity may be a copy of the original data, which would not conserve any privacy. Since no study has used all the above performance measures, the understanding of the results is partial. Missing one or more of the performance measures might lead to a misunderstanding of the performances of the proposed models. Most studies use up to two performance measures to evaluate the quality of synthetic data. However, each of these performance measures serves a specific purpose, and overlooking one can lead to misleading results. For example, if one does not look at the diversity of the given synthetic data, the model can generate the same instance many times. The synthetic data will receive very high per-instance similarity scores if an instance is similar to the original data. Therefore, the evaluation will show very positive results, while in actuality, the synthetic dataset will not be helpful.

Since most studies used measures relevant only for short trajectories, we use known evaluation measures originating from the mobility and NLP fields to evaluate our synthetic data. Text data have similar temporal characteristics to location sequences, so some of their measures are very well suited for our study. This is a relatively new approach to evaluating synthetic mobility data. To measure the per-instance similarity, a widely used measure from the NLP field, the BLEU is suggested [19, 32, 37]. To use measures from the text generation field, we consider a diary to be a sentence and each location in the diary to be a word in a sentence; therefore, a set of diaries can be considered as a corpus (a language resource consisting of a large and structured set of texts). The BLEU evaluation measure counts the number of overlapping n-grams in the generated and original text, divided by the total number of n-grams. The measure itself is straightforward to calculate and understand.

A popular measurement method for per-instance similarity that also originates from the text generation field is based on a practice in NLP where, to tell if a text generation model is good, another model (usually a neural network) is created to distinguish the generated text from the real text [4, 21]. This evaluation model is customized for the dataset that is being tested and therefore gives a score that matches the problem at hand.

The most commonly used evaluation measure of per-instance similarity in the mobility field is the Markov log-likelihood [36]. The likelihood computes the probability of a sample being drawn from the same distribution as that of the original data. If the sample is of good quality, the probability is high. To measure the diversity of the synthetic data, we use the **Kullback–Leibler** (**KL**) divergence measure, as demonstrated by Huang, D. et al. [26]. For privacy, Kulkarni et al. [31] used the framework of tracking attacks [43]. Schematically, the adversary specifies a subset of users, regions, and time instants and asks for information related to these subsets. If the adversary's objective is to determine the whole sequence (or a partial subsequence) of the events in a user's trace, the attack is called a tracking attack. These measures have not yet been used in the field of synthetic mobility data as evaluation measures; in this study, we utilize them for our evaluation process.

3 METHOD

This section discusses our methodology for processing the original data, creating data generation models, and evaluating these models. The method with which we devise our models starts with building the model; next comes the training of the model, followed by the model generating the

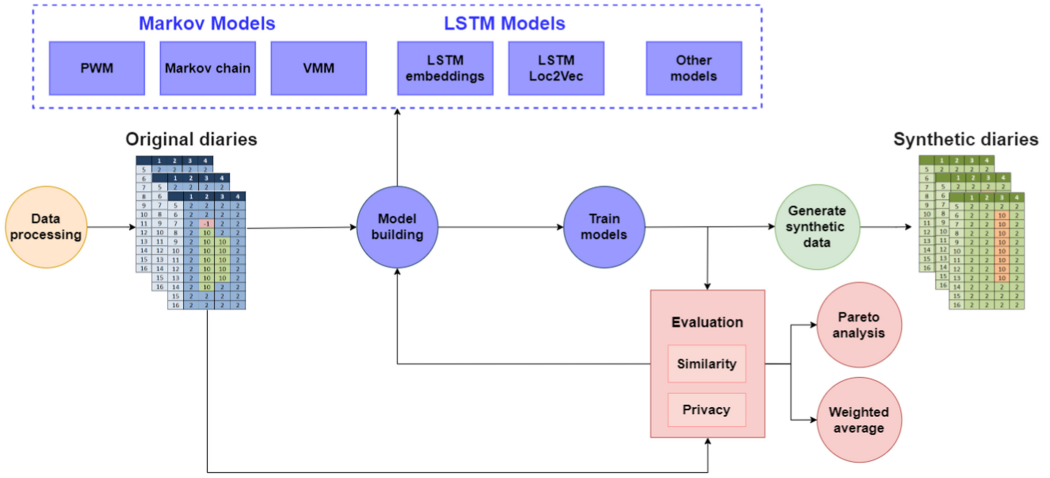


Fig. 1. The flow chart describes the process of generating synthetic data. On the left, we start with the data processing step, which constructs the original data in a diary format. The original data are used to build and train our models. After the models are trained, they are used to generate synthetic data. Finally, the synthetic data are evaluated against the original data to decide which model to use for a specific purpose. This evaluation is done using various measures (some were developed as part of this study), which are later used in a pareto analysis and weighted average.

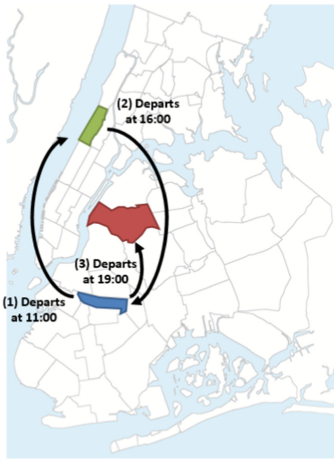
synthetic data, as seen in the general flow diagram in Figure 1. Furthermore, each model needs the data to be adjusted to process it, and this will be discussed with each model's structure. After generating synthetic data, the data are evaluated against the original data using different performance measures. Using these measures, we can compare the models to decide which model to use.

3.1 Location Sequence Data

3.1.1 Data Definitions. First, we formally define the format of our original data. Let G denote a finite set of geographical locations in our data. Then, a location sequence is a vector $\mathbf{X} \in G^n$, where n is the number of time slots in which the location data are recorded. For example, considering Figure 2, G contains Crown Heights, Columbia University, and Williamsburg, where n is 13. In our data, $n = hours\ per\ day \cdot week\ days$. This definition was used in a few other studies, such as [6, 8].

A trajectory is formally defined as a column vector $\mathbf{X} = (G, T)^m$, where G denotes a finite set of geographical locations, T is a timestamp, and m is the number of timestamps for which the location data are recorded. An example of a location sequence and a trajectory can be seen in Figure 2. Using location sequences instead of trajectories has several advantages. First, the visual representation of the data makes it more compatible with human visual testing. The second issue with using trajectories is that many trajectories can have different lengths for the same time period. We have used a padding method to reach similar lengths: inserting additional unnamed data points so that all data sequences are of the same size. In location sequences, all instances in the data are of the same size and therefore do not need any alterations. Another issue is that most models in the mobility field are sequential models.

3.1.2 Data Processing. We start the process with location data and use data processing methods to transform them to a location sequences. The diaries are created in three steps: finding the users' frequent locations, discretizing the location coordinates, and dividing the data into evenly sized



Trajectory

Time	Location
9:00	Crown Heights, Brooklyn
11:00	Columbia University
13:00	Columbia University
15:00	Columbia University
17:00	Crown Heights, Brooklyn
19:00	Long Island, Queens

Location sequence

Time	Location
8:00 - 9:00	Crown Heights, Brooklyn
9:00 - 10:00	Crown Heights, Brooklyn
10:00 - 11:00	Crown Heights, Brooklyn
11:00 - 12:00	Columbia University
12:00 - 13:00	Columbia University
13:00 - 14:00	Columbia University
14:00 - 15:00	Columbia University
15:00 - 16:00	Columbia University
16:00 - 17:00	Crown Heights, Brooklyn
17:00 - 18:00	Crown Heights, Brooklyn
18:00 - 19:00	Crown Heights, Brooklyn
19:00 - 20:00	Long Island, Queens
20:00 - 21:00	Long Island, Queens

Fig. 2. Same day for a user represented as both a location sequence and a trajectory. The trajectory contains locations that were recorded, and therefore, the length of the trajectory can vary depending on the number of recorded terms. In this case, the trajectory was recorded every 2 hours.

time periods. The frequent locations are found using DBSCAN clustering, as shown by Burkhard et al. [11]. Since we did not have any additional data types to utilize in this case, we removed the time tags from our data during the clustering process. We used only the location data, which can be considered continuous and thus could fit DBSCAN to find the frequent locations per user. DBSCAN was selected since it is known to be an efficient clustering scheme over big data for arbitrary shapes and not-necessarily smoothed clusters (locations such as malls, junctions, stations, etc, often have arbitrary shapes). DBSCAN was a good fit also since it is based on batch processing, thus it does not require an entire rerun of the algorithm when new points are added to the dataset. However, as indicated in the text, the clustering module is independent and therefore the DBSCAN can be replaced by another clustering method as a module in the process. Since every user has a different number of frequent locations, a clustering algorithm that does not require prespecifying the desired number of clusters is needed. DBSCAN was found to be the best suited algorithm to fulfill that requirement, and it offers two more benefits: robustness to noise and flag points that do not belong to any cluster (tagged as "−1"). For mobility data, a "−1" location symbolizes traveling between locations. After clustering the data, we need to discretize them to reduce their complexity. To do that, we use statistical areas defined by the Central Bureau of Statistics. A statistical area is the smallest statistical-geographical unit of a municipality for which census data are available. Each statistical area contains between 3,000 and 5,000 residents, resulting in this case in 3,071 areas. Statistical areas are used in many articles that study the differences between various populations [17, 28, 33]. Let us note that, in general, if census data are available on smaller areas the proposed method could be applied also to such cases. Our method could work for any type of discretization of the coordinates, e.g., based on a grid, octagons, and so on, as long as the number of areas is not too large. To create the final diaries, the data are aggregated the following way: for each hour and weekday, the most prominent location is chosen. Night hours are, for the most part, uninformative and therefore removed. Final diary examples are presented in Figure 3.

3.2 Markovian Models

As discussed in Section 2, we use the following baseline models for comparison purposes:

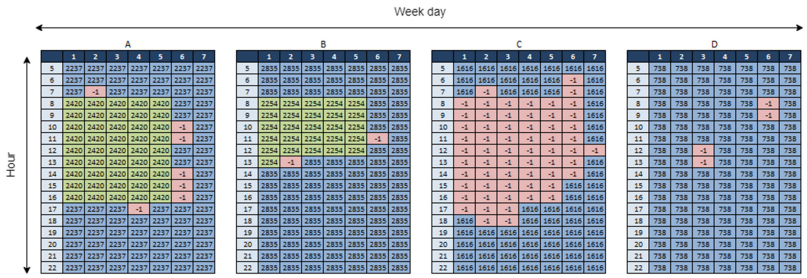


Fig. 3. Examples of five original diaries. The columns represent the day of the week (1 - Sunday, . . . , 7 - Saturday), and rows represent the hour of the day (5:00–22:00). The location number is the statistical area ID, and the cells marked in blue represent home, green represents work, and pink represents traveling. These diaries represent different lifestyles of different users. Users A and B are working people with different working hours. User C is a user that travels for most of the day. User D is a user who mainly stays at home.

Position Weight Matrix (PWM)—This is a basic model that measures the distribution of users over different locations. The PWM is an occurrence table, where each value is a weight representing the propensity of a given symbol to occur at a given position. Given a set X of N aligned sequences of length L, the elements of the PWM matrix ($M_{k,l}$) are calculated as follows:

$$M_{k,l} = \frac{1}{N} \sum_{i=1}^{N} I(X_{i,l} = k), \tag{1}$$

where $i \in (1, \ldots, N)$, $l \in (1, \ldots, L)$, and $k \in K$, where K is a finite set of symbols. Even though the PWM is a very simple model, it is important for signals in biological sequences. [29]. In DNA sequences, the PWM models nucleotides as symbols and therefore calculates the probability of each nucleotide being in each location in the sequence. In our case, the sequences are diaries, and thus, the locations are symbols. Using the PWM in this way provides a good representation of the distribution of locations in time. Another possible use of the PWM could be to address semantic locations (home, work, etc.), and doing so provides a representation of the distribution of different semantic locations (which we do not use in this study). Looking at Figure 2, the symbols are Crown Heights, Columbia University, and Williamsburg, and the PWM algorithm calculates the probability of being at one of these locations in each time slot. This model is also viewed as an MC of order 0, and it has been widely used due to its ability to decide which locations and time slots the users are distributed in. The main limitation of the PWM is that it cannot model correlations between symbols. If there is a common sequence of symbols, the model will not be able to learn it.

Markov Chains of High Order (MC)—To solve the issue of finding the dependencies between locations, a Markov process with an order higher than 0 is needed. A Markov process is a stochastic process that satisfies the Markov property. A stochastic process has the Markov property of order m, if the conditional probability distribution of future process states depends only upon the last m states. This can be seen in the following equation:

$$p(r_t) = p(r_t | r_{t-m}, \ldots, r_{t-1}), \tag{2}$$

where r_t is the location variable r at time t and m is the order of the MC. We define the states of the MC to be the locations and the corresponding times that create a nonhomogeneous Markov model. In this way, the transition probability is affected by location, the day of the week, and the time of the day. Differentiating between days, weekends, hours, and so on, is more accurate in terms of capturing the diaries at those slots, but will result in a significantly smaller amount of

training data. MC are straightforward models. However, they involve several modeling challenges. For one, when the order increases, so does the complexity of the model. The higher the order is, the higher the computation time and memory needed. Another issue is that for low Markov orders, the model underfits and cannot represent highly complex trajectories. Furthermore, for high Markov orders, the model overfits with smaller amounts of data. This implies that when the model does not have enough variant patterns to learn from, it will repeat the same patterns as those in the original data.

Variable-Order Markov Model (VMM)—The VMM is an extension of general Markov models, and the states are defined similarly to their definitions in the MC model. In contrast to fixed-order Markov models, where the orders are the same for all positions and all contexts, in VMMs, the order may vary for each position based on its context. For example, the conditional distribution of a set of users' locations at 6:00, given the locations at 5:00, does not necessarily depend on the locations at 4:00 (e.g., in most cases, if a user was at home at 5:00, he/she was there at 4:00 as well). Thus, at 6:00, the required memory order can be limited to only 1 hour. However, the conditional distribution of a set of users' locations at 6 pm, given the locations at 5 pm, also usually depends on the locations at 4 pm. Therefore, a model order of 2 or higher is required for this time slot. Thus, VMMs provide the means for capturing both large and small orders. A VMM reduces the memory needed to store the model (compared to MC of high memory orders); however, it requires increased computation time. This model was used before on locations sequences by Ben-Gal and Weinstock et al. [8] and has shown promising results.

3.3 LSTM Models

Our proposed models are based on an LSTM network. The ability of LSTMs to remember past behavior for future predictions has proven to be valuable in the mobility field [1, 2, 30] and motivates us to use them for generating user mobility diaries. Similar to those of MC, the model's input is a sequence of locations, and the model's output is the next location. Therefore, we need to create a training set where the training input is a set of location sequences and the labels are the next locations of the sequences. To use diaries as inputs for a neural network, we need to convert the diaries to sequences and vectorize each sequence (embeddings). An embedding is a mapping of a discrete variable to a vector of continuous numbers. We use two different methods for embeddings: creating the embeddings before building the model and building the embeddings as part of the model. On one hand, embedding the data before inputting them into the model is a much faster process and has proven to yield good results. On the other hand, embeddings that are created within the model are optimized for the specific problem the model is facing but take much more time to compute. See a comparison of the algorithms in Figure 4.

Location2Vector LSTM (Loc2Vec)—This model requires embedding the diaries before inputting them into the neural network. To do that, we use the algorithm of Word2Vec [35], specifically the **Continuous Bag-of-Words** (**CBOW**). Word2Vec creates word embeddings so that words that share common contexts in the corpus are located near one another in the vector space. Word2Vec originates from the NLP field, which means that to use it, we need to convert our data to a text format. We use the definition we showed in Section 2, where we define a document to be a list of all the diaries, each diary is a sentence in the document, and each location is a word in a sentence. The resulting vocabulary is a list of all unique locations. Using Word2Vec on our data converts all locations in the diaries to vectors of size v, resulting in diaries of size $(n \times v)$, where n is the number of time slots at which the location data are recorded. At this point, the data are well formatted and can be used as inputs for the LSTM network. The network's input sequence size is set to be u so that the final input size for the network is $u \times v$. Aside from the location vector, another input for the network is the start time of the sequence $t \in (1, \ldots, n)$. The first layer of

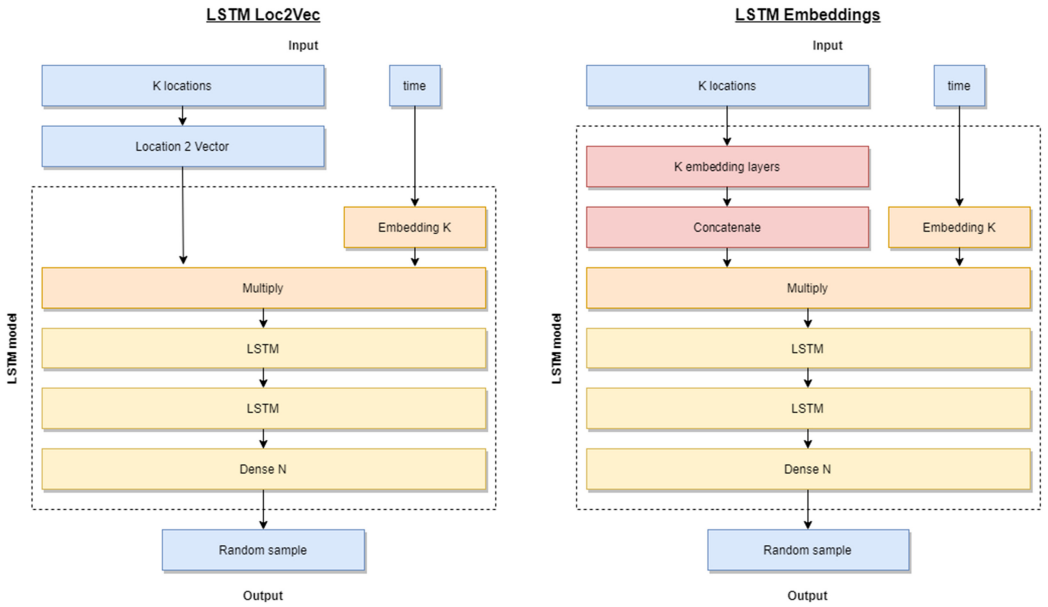


Fig. 4. Proposed LSTM models architectures.

the network is an embedding layer for the time input, and an embedding converts the time to a vector of size $n \times v$. Then, the location vector is multiplied by the time vector, and it goes through two LSTM layers and an additional dense output layer of the vocabulary sizes. The loss function is the categorical cross-entropy loss, which means that the output is a vector of probabilities for each location. In this way, we can randomly choose the next location based on these probabilities and create varying diaries for each initial input.

LSTM with Embedding Layers—In this model, we learn the embeddings as part of the neural network model, which in theory creates more custom embeddings for the model than learning the embeddings separately. As before, we use an input length of u, but here, the model's inputs are raw location data. Each location in the sequence goes through an embedding layer that turns the location into a z-length vector. The embedding vectors are initialized randomly and are trained to minimize the final loss function during model training. These low-dimensional dense embedding vectors are then concatenated to create a vector of length $m \times z$ and, as before, they are multiplied by the embedded time vector. Following this, the multiplied vector goes through the LSTM layers and an additional dense layer. The benefit of this method is its ability to learn the embeddings while optimizing the neural network. On one hand, this should provide the optimal embeddings for this problem. On the other hand, the chance of overfitting increases.

In Figure 5, we can see the synthetic diaries generated by the different models. The PWM diary does not resemble any possible diary in the original data. While this algorithm ensures that many of the statistical qualities of the original data are kept, it does not guarantee that the outcome will be similar to the original data. The other models manage to generate very realistic diaries representing different working day hours for the users. For further examples of generated diaries see Figures 10–14.

3.4 Evaluation

The generated data are evaluated using four performance measures: statistical similarity, per-instance similarity, diversity, and privacy. Each of these performance measures is analytically

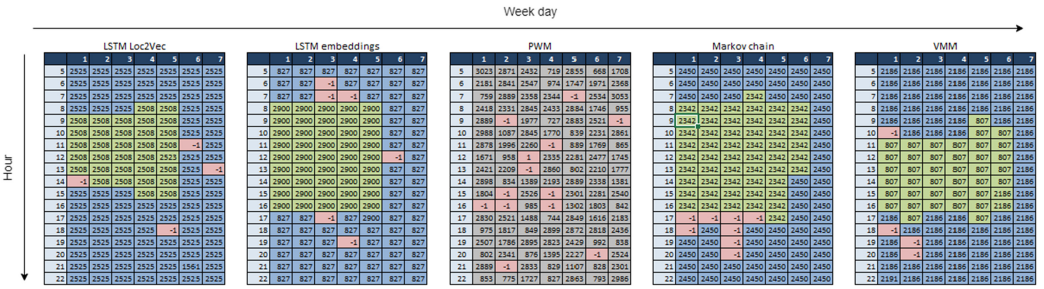


Fig. 5. Examples of four diaries generated by the different models. We can see that the models learn different patterns of user behaviors, such as being at home in the evenings and on weekends, going to work, and so on.

calculated and assigned a numerical value for comparison against those of the other models. The performance measures are also calculated on an original subsample dataset, which is a sample of the original data with the same size as that of the generated synthetic data. This is done to account for the effect of using a small dataset. A desired result would be considered a generated dataset that yields similar results to those achieved by the original subsample in terms of the statistical similarity, per-instance similarity and diversity performance measures, while preserving more privacy. To calculate the evaluation measures for each of the models, we use the following procedure: we generate 50 datasets, each containing 20,000 diaries, for each model. Each evaluation measure is then calculated per dataset, resulting in 50 scores. The mean score is then calculated and assigned as the evaluation measure score for each model accordingly.

After calculating the evaluation measures for each model, one use a weighted average of the measures to assess the usefulness of each model for a specific objective. Each measure is put on a scale between 0 and 1 and receives a weight of how important it is for the objective at hand. For example, if the objective is to release a dataset for public use, the privacy measure will receive the highest weight, while the similarity measures will receive lower weights. If the objective is to create a dataset for human mobility analysis, statistical similarity and per-instance similarity will receive the highest weights, while the privacy measure will receive the lowest weight. The final weighted and averaged score indicates which of the tested models best suits the desired objective. This is a powerful tool for application developers.

3.4.1 Privacy. When creating synthetic data, our main purpose is to synthesize data that would not reveal the identities of users from the original data. This means that it should not be possible to use the synthetic data to reidentify users or extract further information about them. For this purpose, we use the following measures:

— **Reidentification probability**—This measure uses the framework of tracking attacks [43]: The adversary specifies a group of users and a subset from a location sequence for each user in the group, and the objective of the adversary is to extrapolate additional information about the users in the group. The idea behind this measure is that given a set of known locations, it is possible to identify the rest of the user's diary and therefore identify the user. The known locations are n locations with timestamps from a user's original diary. They are used to calculate the probability of inferring the rest of the original diary by using the synthetic data. This reveals how probably it is to extract information about a user from the synthetic diaries while having partial prior knowledge on the user from the original diaries. Ideally, if the synthetic data preserve privacy well, the reidentification probability should be low. If the probability is high, then the synthetic data are a copy of the original data and

hence do not preserve privacy at all. Given a diary d from the original data and a subsample d_n of d, where $|d_n| = n$, we set S to be the group of all synthetic diaries that match d_n. The reidentification score is defined as

$$ReID_n = \frac{1}{T}\sum_{t=0}^{T}\frac{1}{|S|}\sum_{s\in S} I(s(t) = d(t)), \tag{3}$$

where T is the time span of diary d, $d(t)$ is the location of diary d at time t, and $s(t)$ is the location of a synthetic diary s in S at time t. If the synthetic dataset is too similar to the original dataset, then the reidentification of the diaries should be very accurate. This means that the lower the reidentification score is, the less privacy we preserve.

— **Proportion of identical diaries**—Given a set of diaries, how many of them are completely identical to an original data diary? This measure highlights the extent to which the model creates direct copies instead of creating unique diaries. If an adversary can find the diaries that are based on real people, it is very easy to identify the users from these diaries. The measure is defined as

$$IdenticalDiaries(S) = \frac{1}{|S|}\sum_{s\in S} I\,(s \in D), \tag{4}$$

where S is the set of generated diaries, s is a diary in S, and D is the set of original diaries.

3.4.2 Statistical Similarity. When generating synthetic data, we expect them to keep the statistical properties of the original data [22, 27]. For example, we want users to have the same working hours as the original users and the same location distribution. To understand if the synthetic data have the same statistical qualities, we use different statistics that were used in previous articles as evaluation measures. These statistics are of value to different applications and reflect the users' mobility behaviors in the dataset:

— **Statistical area distribution**—This is the user distribution across various statistical areas calculated per time slot. The purpose of this statistic is to ensure that synthetic users are present in the same locations as those of the original users, and it is the most popular statistical measure according to [9, 22]. For each statistical area a and time slot t, we calculate the following probability:

$$StatAreaDist(a, t) = \frac{1}{|S|}\sum_{s\in S} I(s(t) = a), \tag{5}$$

where S is the set of generated diaries, s is a diary in S, and $s(t)$ is the location of diary s at time t. The statistical area distribution is the set of $StatAreaDist(a, t)$ for all a in the set of statistical areas A and $t = 1, \ldots, T$.

— **Transition matrix**—This is the share of users that move between every combination of two different locations calculated per time slot. Many organizations use transition tables to understand the flows of people between areas. These kinds of statistics help improve traffic planning. For two statistical areas $a1$ and $a2$ and a time slot t, we calculate the following probability:

$$TransMatrix(a1, a2, t) = \frac{1}{|S|}\sum_{s\in S} I(s(t) = a1\ \&\ s(t+1) = a2), \tag{6}$$

where S is the set of generated diaries, s is a diary in S, and $s(t)$ is the location of diary s at time t. The statistical area distribution is the set of $TransMatrix(a, t)$ for all $a1$ and $a2$ in the set of statistical areas A and $t = 1, \ldots, T-1$.

—**Task hours distribution**—This is the distribution of daily hours spent on a relevant task (work, travel, home, etc.). This measure helps understand users' daily routines, and it was used by a previous article [9]. For a task b and number of hours n, we calculate the following probability:

$$TaskHoursDist(b,n) = \frac{1}{|S|}\sum_{s\in S} I\left(\sum_{t=1}^{T} I(d(t) = a(b)) = n\right), \tag{7}$$

where S is the set of generated diaries, s is a diary in S, $s(t)$ is the location of diary s at time t, and $a(b)$ is the statistical area that represents task b. The task hours distribution is the set of $TaskHoursDist(b,n)$ for $n = 1,\ldots,N$.

We measure each statistical similarity measure by calculating the distance between the statistics of the synthetic and original data. This distance is calculated by the KL divergence , as seen in Bindschaedler et al. [9]. The KL divergence is a natural way to compare distributions: It returns a nonnegative real number, where a larger value denotes a greater distance between two discrete probability distributions. Given P and Q, which are defined on the same probability space X, the KL divergence from Q to P is defined as

$$D_{KL}(P||Q) = \sum_{x\in X} P(x) log\left(\frac{P(x)}{Q(x)}\right). \tag{8}$$

Given a random variable X, P is the distribution of the synthetic data over X, while Q is the distribution of the original data over X. For example, if the statistic used is the statistical area distribution, then $x \in X$ is a statistical area from the statistical areas list.

3.4.3 Per-instance Similarity. An important performance measure is per-instance similarity. Each diary in the synthetic data should be similar to the original diaries. This means that each generated diary, when evaluated, should have a score close to the original subsample score. For example, in the field of image analysis, a similarity test could be similar to a visual test if a tester looked at a generated image and could not tell if it was generated or real. While various studies use different methods to measure per-instance similarity [26, 31, 42], we use the following measures:

—**Log Likelihood**—This measures the "goodness of fit" of a model to a sample of data. This is the log of the probability that a set of synthetic data is from the same distribution as the original data. This measure is based on the work of Ohler et al. [36] and has been used in many studies [7, 8, 50]. It assumes that the original data behave according to a MC, and using the transition matrix from the MC model, the likelihood is calculated by applying the law of total probability, that is, multiplying the conditional probabilities:

$$L(\theta) = -LOG\left[\sum_{i=1}^{T} P(r_i|r_{i-1},\ldots,r_{i-15})\right], \tag{9}$$

where T is the sequence's length, r_i is the location at time i, and $P(r_i|r_{i-1},\ldots,r_{i-15})$ is the conditional probability that the user is in location r_i given that he/she followed a trajectory specified by the sequence $r_{i-1},\ldots,r_{i-15}$. If the diary contains frequent sequences from the original data, its log likelihood score should be close to 0.

—**Time-based BLEU**—The main objective of this measure is to determine how many sequences of words are repeated in the generated text and the original text. The BLEU measure was proposed by Papineni et al. [37] and was originally used to evaluate the quality of

generated text [19, 32, 37]. The BLUE measure is defined as

$$BLEU_N = \frac{Matched(N)}{S(N)}, \tag{10}$$

where $Matched(N)$ is the number of overlapping n-grams in the original data and synthetic data, and $S(N)$ is the number of n-grams in the synthetic data. Overall, it measures the collective synthetic dataset against the original dataset. In this way, we can understand whether a pattern in the synthetic data exists in the original data. Traditionally, the BLEU measurement would use n-grams of words that would translate to n-grams of the locations in location sequences. However, in location sequences, when looking at the locations, it is also important to know when the user was there. This is why we tweak the BLEU measure by adding timestamps to the n-grams

$$TimeBLEU_N = \frac{\sum_{t=0}^{T} Matched(N,t)}{\sum_{t=0}^{T} S(N,t)}, \tag{11}$$

where $Matched(N,t)$ is the number of overlapping n-grams in the original data and the synthetic data at time t, and $S(N,t)$ is the number of n-grams in the synthetic data at time t.

—**Discriminator**—This measure is based on a deep neural network assigning scores to diaries. It is based on a common practice in NLP where, to find if a text generation model is good, another model is created to distinguish the generated text from the real text [4, 21]: Let L be the language and L^* be the generated text. Additionally, let X be a piece of text obtained from either L or L^*. Let $y = h(X)$ such that $y = 1$ if $X \in L$ and $y = 0$ if $X \in L^*$, where h is the hypothesis function (a classifier like AdaBoost, SVM, etc). If $P[y = h(x)]$ is found to be sufficiently high, X is very similar to L.

In our study, we score generated diaries on a range between diaries of random sequences and diaries from the original data. In particular, we create a dataset of fake diaries that are assigned random locations in sequences and then tagged as "fake" and original diaries that are tagged as "real". A fully connected neural network with embedding layers for each location in a diary is trained on this dataset. After the neural network is fully trained, it evaluates the generated diaries from the synthetic data. Each diary is then scored between 0 (false) and 1 (real). The higher the score is, the more the model "believes" that the diary belongs to the original data. The total score for this evaluation measure is the average of all diary scores.

This measure differs from the log likelihood and the time-based BLEU measure by looking at the diary as a whole in contrast to looking at sequences from the diary. The time-based BLEU measure will give a good score to a synthetic diary as long as every sequence in that diary was in some other diary from the original diary, even if the resulting synthetic diary does not resemble any plausible original diary. For example, if we see a diary that shows a user starting the day at location "1", going to work at location "2", and finishing the day at location "3", the time-based BLEU score will be high if there is a user that lives in location "1" and works at location "2" and another user that works at location "2" and lives in location "3". This means that the two described sequences can be found in two diaries of two original users; however, the synthetic diary itself does not show any actual patterns that users follow.

3.4.4 Diversity. Synthetic data also need to have diversity, and the generated diaries should have the same variety of diaries as the original diaries. For example, a common issue with generative models is that the models enter a state of "mode collapse" and generate the same sample over and over again. This might not be detected in the per-instance similarity if the repeated diary is nearly identical to an original diary, nor will it be detected in the statistical similarity if it copies the statistics of the original diaries. For example, if the statistic we are measuring with statistical

similarity is the average working hours, we can create a dataset with one repeated diary that has a user with the exact same working hours as the average working hours of the original data. Should that diary be an exact diary from the original dataset, it would also receive a high score in the per-instance similarity measure. We want our diversity measure to indicate if such issues arise.

— **Wasserstein Likelihood (WS)**—This measure quantifies if the dataset has the same distribution of likelihood scores (described in terms of per-instance similarity) as the original data. We use the Wasserstein distance to calculate the similarity between the distributions of scores, as shown by Del Barrio et al. [15]. The Wasserstein distance is defined as

$$W_p(\mu, \nu) = (infE[d(X, Y)^p])^{1/p}, \tag{12}$$

where $p > 1$ denotes the moment, $\mathbf{E}[Z]$ denotes the expected value of a random variable Z, and the infimum is taken over all joint distributions d of random variables X and Y with marginals μ and ν, respectively. A successfully generated dataset would have a likelihood WS score that is as low as possible.

— **Reverse BLEU**—To test diversity, we want to measure how many of the unique n-grams of the original data appear in the synthetic data. The reason behind this is that we want the synthetic data to have as many different patterns as in the original data. If the synthetic data are not diverse, they will have very few patterns that match the original data patterns. Similar to the time-based BLEU measure in the discussion on per-instance similarity performance measures, the location n-grams are given corresponding timestamps. The reverse BLEU is defined as

$$RevBLEU_N = \frac{\sum_{t=0}^{T} Matched(N, t)}{\sum_{t=0}^{T} O(N, t)}, \tag{13}$$

where $Matched(N, t)$ is the number of matched n-grams in the original data and synthetic data at time t and $O(N, t)$ is the number of n-grams in the original data at time t.

— **Wasserstein Discriminator (WS)**—We want to measure if the synthetic data have the same discriminator scores (based on the per-instance similarity discriminator measure) as those of the original data. To calculate the distance between the synthetic discriminator scores and the original discriminator scores, the Wasserstein distance is used to calculate the distance between these distributions. This measure is calculated using the same Wasserstein distance as that defined in Equation (12).

4 EXPERIMENTS

In this section, we apply the method discussed above to a real example dataset to evaluate the quality of the generation models. The data we use are anonymous cellular data. We show examples of diaries and the generated diaries from the different models trained on these data. Finally, we go over the proposed measures and evaluate the performances of the different models.

4.1 Data

This study uses anonymized data from a cellular service provider covering a central city and its surrounding area as its raw data. This data consists of 1.8 Million cellular phone users in the December 2012 – January 2013 time period. The data are anonymous: users' **International Mobile Subscriber Identifiers (IMSIs)** have been removed and replaced with randomly assigned IDs. The raw cellular data are a compound of five features that the cellular operator saves and analyzes, as used in previous studies [8].

Generally, the collection of location data occurs during any type of data transfer, such as phone calls, streaming, interactive media, background, and registration signals (correspondence between

the device and the cellular network) or other network signals. The data available per user, in most cases, are not continuous, and there are many blank time slots in users' data. Blank time slots are time frames in which no cellular signal was available, resulting in no location data being available during the specific days/hours/minutes. This could happen when the user leaves an area covered by the **Radio Network Controller** (**RNC**), turns off his/her cellular device or enters an area with no reception. Thus, there is no geographical continuity per user, even in the central city and surrounding area. For that, We remove users that do not have enough data. Days with less than 10 hours are removed reducing data from 1.8 M users to only 400 K users, afterward, users with less than four active days were removed reducing the total number of users further to 300 K users. The remaining data goes through the proposed processing method (shown in Section 3.1) results in a 7-day diary of size 7×18, and an example of such a diary is shown in Figure 3 in Section 3.

To train the LSTM models successfully, we need to apply one more data preprocessing step. When looking at different diaries, we can see that most people stay at a location for more than an hour. This means that when looking at the last k locations, the best guess for most hours is that a user will stay at the same location as the last location he/she has been at. Training the LSTM models on the original diaries, as described in Section 3.1, will give problematic results since the data will teach them to use only the last location to predict the next location. Therefore, we use an undersampling of 50% for all the subsequences where the label is the same as that of the last location.

4.2 Results

A good data generation model creates synthetic data that are fairly similar to the original data while preserving a higher level of privacy. Therefore, we want the synthetic data to perform sufficiently well in all performance measures. In this section, we evaluate all of the performance measures as described above, looking at each one individually and all of them as a whole.

4.2.1 Privacy. We first look at the privacy measures, as shown in Figure 6. We use the original subsample as a baseline to see if the different models preserve privacy better than the original data. It is expected that the original subsample will have a relatively bad privacy score since it does not preserve any privacy by definition. (A bad score means that the adversary can easily identify users from the data.) Models with better scores can be considered privacy-preserving models.

The Markov-based models (the MC and VMM) achieve approximately the same reidentification probability scores as that of the original subsample. We can see by the proportion of identical diaries that the Markov-based models reflect some of the original data and hence are able to achieve a reidentification score similar to that of the original subsample. According to the privacy performance measures, the Markov-based models do not offer any advantages over using the original data for privacy-preserving purposes. The LSTM embedding model also obtains similar scores as those of the original subsample in terms of the reidentification probability and a similar score to the Markov-based models in terms of the proportion of identical diaries measure. This can happen if the model produces very realistic diaries that are too similar to the original data or if the model is overfitted and hence simply memorizes the original data. The privacy scores of the PWM model are nearly a constant 0 (the best possible privacy score), because the diaries generated by the PWM model are not similar at all to the original diaries, hence preserving maximum privacy. The next best model after the PWM model, is the **Location2Vector LSTM** (**LSTM Loc2Vec**). The LSTM Loc2Vec score for the proportion of identical diaries is close to 0, and the reidentification probability scores are consistently 10% better than those of the rest of the models. Additionally, only 0.3% of the diaries are identical to any original diary, meaning that identifying users is a much harder task with this model compared to the other models, where 3.3% of the diaries are identical. Taking

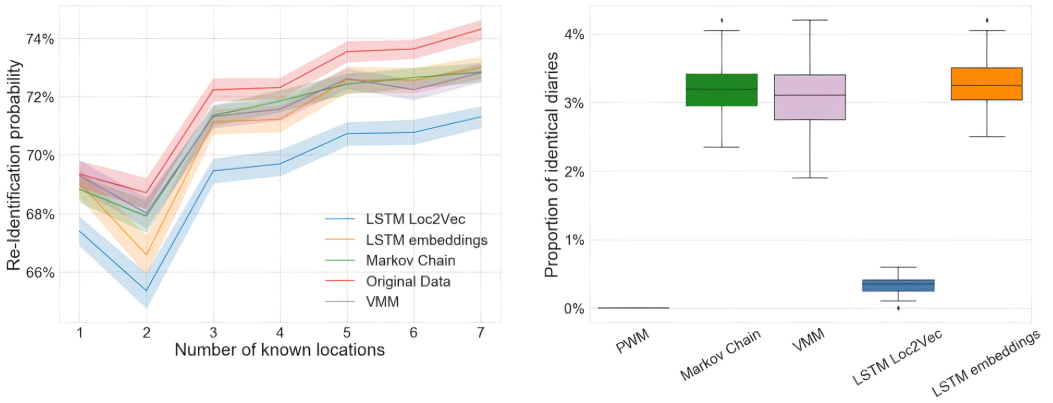


Fig. 6. The left graph shows the reidentification probability measure of each model and the number of known locations. The PWM model is not shown since the results are almost 0 for every number of known locations, and this would distort the graphs. The right graph shows the proportion of identical diaries measured by each model. The original subsample results are not shown in this graph since they are a constant 1 - all of the diaries in the original subsample exist in the original diaries.

all the above factors into consideration, it can be concluded that LSTM Loc2Vec shows the most potential as a generation model for privacy protection.

4.2.2 Statistical Similarity. The first statistic we look at is the **statistical area distribution**. When comparing the models using this statistic, the PWM and Markov-based models (the MC and VMM) obtain the best results, as shown in Table 1. The statistical area distribution scores of the Markov-based models are only 7% higher than those of the original subsample, while those of the LSTM-based models are 50%–80% higher than those of the original subsample.

The second statistic we look at is the **transition matrix**, which details the hourly transitions between statistical areas. Looking at Figure 7, all models except for the PWM have generally similar transition matrices to that of the original subsample. The PWM transition matrix looks like random noise, and accordingly, its score is also the worst. The MC and VMM obtain the closest scores to that of the original subsample, while the LSTM models have slightly worse results (2%–5% higher than the original subsample).

The last statistic is the **task hours distribution**; for our research, we choose to focus on the task of going to work. To calculate the working hours distribution, users are first divided into three categories: working users, nonworking users, and travelers (users that travel between locations most of the time). Nonworking users and travelers are assigned zero work hours. For working users, we find the work location of each user by finding the second most prominent location in the user's location sequence (assuming their home is the most prominent location). Using the work location of each user, we count the number of working hours and average it by dividing by the working days. This measure helps understand users' daily routines and how many users actually work, which is very important for government agencies, for example. To calculate the distribution traveling hours, we count the number of hours traveled per day. This statistic helps understand how much time people spend on the road and helps with traffic and transportation planning. As seen for the other statistics, the Markov-based models have the closest values to those of the original subsample. If we look at the LSTM-based models, LSTM Loc2Vec has better results than the LSTM embedding model for the working hour distribution. However, for the traveling hour distribution, the LSTM embedding model has better results. This shows that the models learn

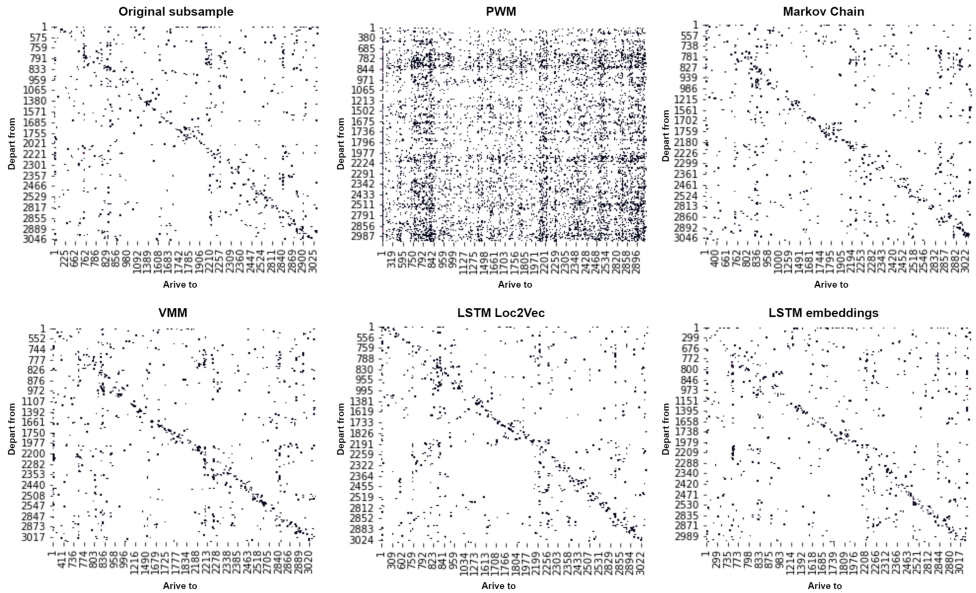


Fig. 7. The transition matrix at 8:00 on Sunday for the original data and the models' synthetic data. The y-axis shows the location IDs from which the users departed, and the x-axis shows the locations at which the users arrive.

Table 1. The Table Shows the Statistical Similarity Measures Obtained by Each Model as a Deviation Percentage from the Original Subsample

Measure	Original subsample	PWM	MC	VMM	LSTM Loc2Vec	LSTM embeddings
Statistical areas distribution	0%	7.4%	7.4%	3.7%	81.4%	59.2%
Home hours distribution	0%	inf	37%	37%	66%	80%
Work hours distribution	0%	inf	37%	37%	66%	80%
Traveling hour distribution	0%	inf	60%	59%	74%	53%

Each row represent a different statistical similarity measure, and each column represent a different model. All measures and models are tested by a T-test with the original subsample and are found to be statistically significant with a confidence level of 90%. This means they are all significantly different from the original data, which is not what we desire.

different patterns: the LSTM Loc2Vec model better learns how to create realistic working days, while the LSTM embedding model better learns how to create traveling hours.

When looking at all the statistical similarity results in Table 1, we observe that the MC model and VMM achieve the best statistical similarity scores. This tends to happen since the Markov model consumes a large amount of data for training the model and refining the transition probabilities, and then in the generation stage, it generates patterns that are often very realistic and actually repeat real patterns in the dataset. This is consistent with the results from the privacy performance measure. Therefore, while Markov-based diaries are statistically similar to the original data, they are of low quality in terms of privacy.

4.2.3 Per-Instance Similarity. When looking at the per-instance similarity scores in Table 2, the Markov-based models have the closest scores to those of the original subsample. This is expected

Table 2. The Table Shows the Per-Instance Similarity Measure Achieved by Each Model as a Deviation Percentage from the Original Subsample

Measure	Original subsample	PWM	MC	VMM	LSTM Loc2Vec	LSTM embeddings
Log Likelihood	0%	−inf	−13.9%	−14.2%	−31.6%	6.9%
Time based BLEU 4	0%	−100.0%	−1.3%	−1.3%	−10.5%	2.6%
Discriminator	0%	−99.9%	−0.4%	−0.3%	−1.2%	−0.7%

Each row represents a different statistical similarity measure, and each column represents a different model. All measures and models are tested by a T-test with the original subsample and are found to be statistically significant with a confidence level of 90%. This means they are all significantly different from the original data, which is not what we desire.

Table 3. The Table Shows the Diversity Measures of Each Model as a Deviation Percentage from the Original Subsample

Measure	Original subsample	PWM	Markov Chain	VMM	LSTM Loc2Vec	LSTM embeddings
WS Likelihood	0%	9.5%	8.1%	8.2%	9.2%	4.1%
Reverse BLEU 4	0%	15.2%	−0.1%	0.0%	0.8%	0.7%
WS Discriminator	0%	inf%	16%	10%	68%	23%

Each row represents a different statistical similarity measure, and each column represents a different model. All measures and models are tested by a T-test with the original subsample and are found to be statistically significant with a confidence level of 90%. This means they are all significantly different from the original data, which is not what we desire.

since they are very similar to the original data. The LSTM embedding model also consistently achieves good results, 2.6% different than the original subsample in terms of the time-based BLEU measure, and only 0.7% different in terms of the discriminator measure. This shows that using embedding layers in the neural network helps the model tackle the problem effectively. It can further be seen that the LSTM Loc2Vec model scores only 1.2% lower than the original subsample in terms of the discriminator measure. This shows that the discriminator model believes that its synthetic data could very well be from the original dataset. The PWM model achieves very different scores than those the original subsample in terms of all of the per-instance similarity measures, further strengthening our understanding that this model simply generates noise, specifically when we look for longer patterns and not only at the distribution over the different areas.

4.2.4 Diversity. The Markov models have the best scores when looking at the diversity measures (Table 3). This is consistent with the fact that we know that the Markov-based models generate very similar data to the original data. Therefore, not only is each diary in the data similar to a diary in the original data, but also the distribution of the diaries is similar to the original data distribution. The PWM model has the worst diversity, confirming the suspicion that the model creates unrealistic diaries. The LSTM embedding model has better diversity scores than the LSTM Loc2Vec model, showing again that the embedding layers give the model an edge over using static embeddings.

4.2.5 Pareto Analysis. Assuming that the objective of an application is to share data with other parties, the most important performance measure would be the privacy conservation of the data while still producing realistic diaries. Figure 8 shows the Pareto analysis of the measures that could be the most important ones for this objective. When comparing privacy to per-instance similarity and statistical similarity (the top graphs in Figure 8), it is apparent that the LSTM Loc2Vec model

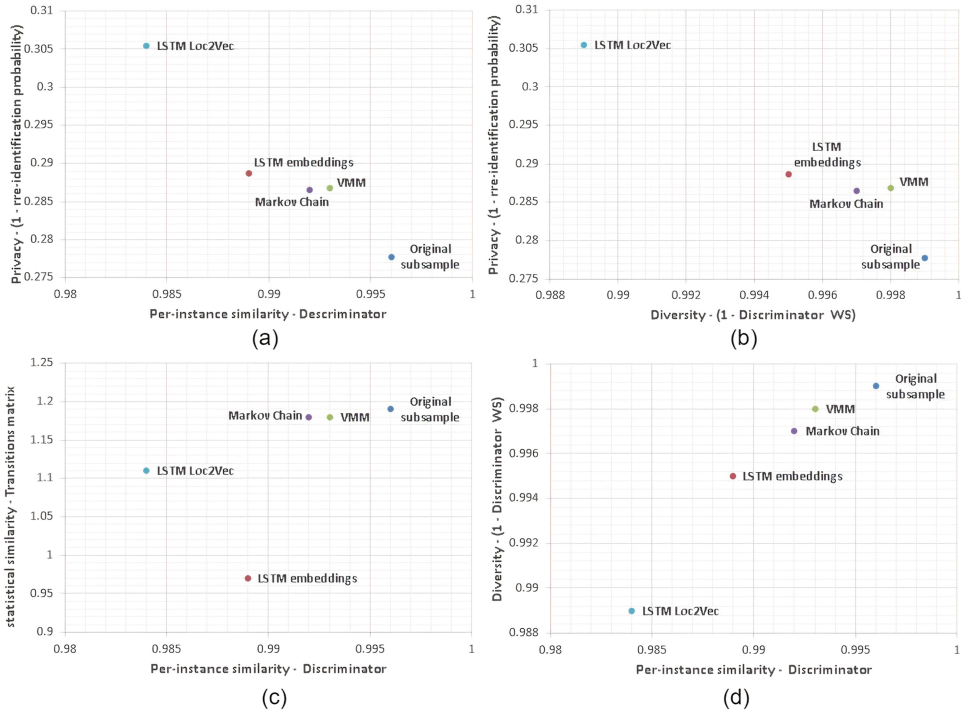


Fig. 8. Pareto analysis of different performance measures. In graphs (a) and (b), we compare privacy (measured by the reidentification probability) with per-instance similarity (measured with the discriminator) and statistical similarity (measured by the transition matrix), respectively. In graphs (b) and (c), we compare per-instance similarity (measured by the discriminator) with statistical similarity (measured by the transition score) and diversity (measured by the WS discriminator), respectively.

preserves more privacy, while the Markov-based models' diaries are more similar to the original data. We use a weighted average of

$$0.8 \times Privacy + 0.1 \times PerInstanceSimilarity + 0.05 \times StatisticalSimilarity + 0.05 \times Diversity, \quad (14)$$

where privacy is measured using the reidentification measure, per-instance similarity is measured by the discriminator measure, statistical similarity is measured by the transition matrix, and diversity is measured by the WS discriminator. In the top row of Table 4, we can see the results of the models using this weighted score, and we see that the best model is the PWM. Taking the fact that all similarity measures show that this model does not produce usable diaries into account, the next best model is LSTM Loc2Vec. This shows that it is important to look at each measure individually before aggregating the measure into one score. If the objective of an application is to create data for mobility analysis, the most important performance measures would be the similarity measures. When comparing both statistical similarity and diversity to per-instance similarity (the bottom graphs in Figure 8), the VMM model seems to dominate the other models in terms of all performance measures. For this objective, we use a weighted average of

$$0.2 \times Privacy + 0.3 \times PerInstanceSimilarity + 0.3 \times StatisticalSimilarity + 0.2 \times Diversity, \quad (15)$$

where privacy is measured using the reidentification measure, per-instance similarity is measured by the discriminator measure, statistical similarity is measured by the transition matrix, and

Table 4. The Table Shows the Weighted Average Scores for Each Model based on Different Objectives

Measure	PWM	MC	VMM	LSTM Loc2Vec	LSTM embeddings
Sharing data Equation (14)	0.8	0.19	0.2	0.21	0.14
Mobility pattern analysis Equation (15)	0.2	0.79	0.8	0.72	0.56

Each row represents a different task, and each column represents a different model.

diversity is measured by the WS discriminator. In the bottom row of Table 4, we can see that the VMM model is considered the best for this objective.

5 DISCUSSION

Finding the balance between similarity and privacy is the ultimate goal when building a model for generating synthetic data. The PWM model generates data that preserve privacy by creating unrealistic diaries. The Markov-based models create almost identical datasets to the original data, resulting in high similarity scores (8% better than the LSTM Loc2Vec model and 1% better than the LSTM embedding model) and low privacy scores. The LSTM Loc2Vec model might not have the best results in any performance measures. Still, when examining its overall performance, it is clear that it preserves the most privacy, 10% better than the other models while generating credible diaries. The Markov-based models are not relevant for general public use, as they are major privacy liabilities. However, we could use them for many other applications, such as the analysis of user behaviors or for artificially enlarging datasets. The LSTM Loc2Vec model could potentially be used to generate data that can be utilized by researchers, application developers, and government agencies to reduce privacy risks.

In contrast to other articles in this field that trajectories as their data format, we generate location sequences, which are used extensively in lifestyle studies, from improving city functionality (e.g., transportation and construction) [48, 50, 52] to real estate pricing evaluations [34]. This further drives us to create a framework for using location sequence data as inputs for our tested models. The main difference between using location sequences versus trajectories is that the times between locations are known in location sequences, while in trajectories, the generation model needs to predict the next location and the time at which it occurs. This increases the complexity of the generation model. This drives us to explore LSTM models and different MC-based models for generating synthetic diaries. These models are very appropriate for temporal data and hence are widely used in the field of NLP.

Our work offers a road map for developing policies for synthetic data generation processes. We define the framework of building generation models and evaluate various models to find the best models for different purposes. First, it is essential to determine the objective of the given generation model, and this indicates which performance measure to focus on. The data type also influences the process: is the data trajectory-based or location sequence-based, and what is the period of the data? After training the models and using different measures to quantify the performance measures of each model, we can make an educated decision as to which model would best suit the defined objective using both a Pareto analysis and a weighted average that gives a definitive score for each model. Both the Pareto analysis and the weighted average are essential for understanding the final result since looking at just one evaluation measure can hide the crucial deficiencies of some models.

Our work highlights the importance of diverse evaluation measures for synthetic data. Existing works used a maximum of two performance measures (see Section 5), but ignoring any one of

these performance measures could lead to irrelevant synthetic data. If diversity is ignored, it could lead to a dataset comprised one repeated diary, while ignoring privacy could lead to copying the original data identically. Observing all performance measures gives a proper understanding of the synthetic data's quality and helps match the best model to the synthetic data's problem.

To understand which data generation model is best suited for a specific application, we compare five models using two performance measures: similarity and privacy. These performance measures have an inherent tradeoff; the more similar the synthetic data are to the original data, the more privacy leakage they present. Another contribution our study offers is comparing models using an efficient Pareto frontier for analyzing both privacy and quality together; in this way, a person can select a model that best fits the given application's requirements. In our results, the PWM obtains good scores in some statistical similarity measures, but receives the worst scores for all other quality measurements. The Markov-based models (MC and VMM) have very high similarity scores (statistical similarity and per-instance similarity) and diversity scores while achieving low privacy scores. These models partially replicate the original data, meaning that when releasing datasets for public use, these models risk users' information. However, the models' parameters can be saved instead of the original data itself to reduce memory usage, thereby saving much storage space. Furthermore, the LSTM models present good similarity scores, though slightly less than those of the Markov-based models, while presenting the best privacy scores, far better than those of any of the other models. This shows that LSTM models can actually represent the original data while producing nonidentical synthetic data. These results demonstrate how important it is to evaluate synthetic datasets using different performance measures to understand their overall quality fully.

Given more data types, such as the user's activity or his environment, the considered process could be converted to a multidimensional one. In such a case, each model would need to predict not only the next location but also the other considered data types. While the Markov-based models could be used also in the new process, given that the Markovian states could be redefined to include the additional data types, the LSTM model would need to rely on a new architecture. In particular, depending on the type of data added, embedding or dense layers would need to be added as well as the output layer would need to be changed. As for the evaluation measures, with a few changes, they could also be used to evaluate the new generated data types. In most cases, we could just change the evaluation measures to consider the extra data types, for example, in the Log Likelihood or the reidentification probability. In the statistical measures, we would need to add statistical measures for the extra data types, for example, activity distribution or environment distribution. For the discriminator, the model used for this measure would need to be modified to include the extra data types (similarly to the LSTM model). It is important to mention that adding information on the users would increase the challenge of keeping their privacy. More information means that even with less data it would be easier for an adversary to get information on users. Of course, adding additional data types can help achieve more interesting conclusions on user behavior.

It is also important to highlight some of the limitations of our work. First, we convert the location coordinates to a list of places between which the users move, using statistical areas as the basis for analysis. This is a decision made in many studies [17, 28, 33], but it also means that our results may not be suitable for more precise locations. For some purposes, the coordinates themselves, which our models do not generate, are needed. Also, we use DBSCAN for location clustering, which may have some implications about the way locations selected. However, as we use multiple locations and statistical areas for identifying diary slots, we believe that the impact on the outcomes of the generators is minimal. Another limitation is the period. We work for one week and generate location sequences for one week. The models suggested here will need to be further explored to use the same method for a longer period (e.g., one month). However, the evaluation methods suggested

in this study should be applicable for any period. Another limitation is that the evaluation process depends on the application and is not independent of it. This means that comparing models without an application in mind is problematic.

6 CONCLUSIONS

Working with human location data is increasingly challenging because of the potential privacy leakage it can inflict. In this article, we develop synthesis methods for longitudinal location sequences and evaluate for utility and privacy. Our experiments show that different models are compatible with various applications. On one hand, Markov-based models achieve the highest similarity scores relative to the original data, but they do not preserve any privacy, meaning they are best for applications such as lifestyle analysis and memory power reduction. On the other hand, the LSTM models preserve the most privacy while having good similarity scores, meaning they are most useful for releasing data for public or research uses.

Further work can investigate the difference between generating location sequences and trajectories. We argue that LSTM models are best suited for working with location sequences, but other articles have used LSTM models with trajectories. We want to test different generation models on trajectories and location sequences to evaluate the best results for a specific problem. Another validation needed is to compare the LSTM models to GAN models, which many studies use. Moreover, since the importance of evaluating performance measures is proven in this study, we suggest that future work further explore more sophisticated measures of performing synthetic data evaluations.

APPENDICES

A COMPARISON OF LOCATION DATA SYNTHESIS

Table 5. Previous Studies on Synthetic Mobility Data

Reference	Data	Data type	Model	Stat	Insta	Divers	Privacy	Vis
Song [46]	Dartmouth College students Wi-Fi data	Location changes	MC					
Geyik [22]	cab data San Francisco, CA	Short-term	Probabilistic context-free grammars	V				
Bapierre [5]	Received Signal Strength (RSS) information	Short-term	VMM					
Gambs [20]	Geolife dataset	Location changes	MC					
Bindschaedler [9]	Nokia mobile dataset	Long-term	Aggregate model	V			V	
Zheng [53]	Tracking data from professional basketball games	Short-term	Hierarchical networks					V
Kulkarni [30]	Nokia Mobile Dataset	Short-term	RNN					V
Kulkarni [31]	Nokia mobile dataset	Short-term	GAN	V	V		V	
Huang [26]	navigation GPS	Short-term	Auto encoders		V	V		
Song [45]	N/A	Short-term	GAN					V
Rao [42]	Foursquare weekly trajectory New York City (NYC)	Long-term	GAN		V		V	

The first four columns describe the data the articles used and the generation models. The last five columns represent the five performance measures (statistical similarity, per-instance similarity, diversity, privacy, and visual test), where the measure https://www.overleaf.com/project/60b23aaa5d2bc6a3481f7c7ba given article used is marked.

B ORIGINAL DIARIES EXAMPLES

This appendix presents examples of various original diaries post processing (as described in Section 3) are shown in Figure 9.

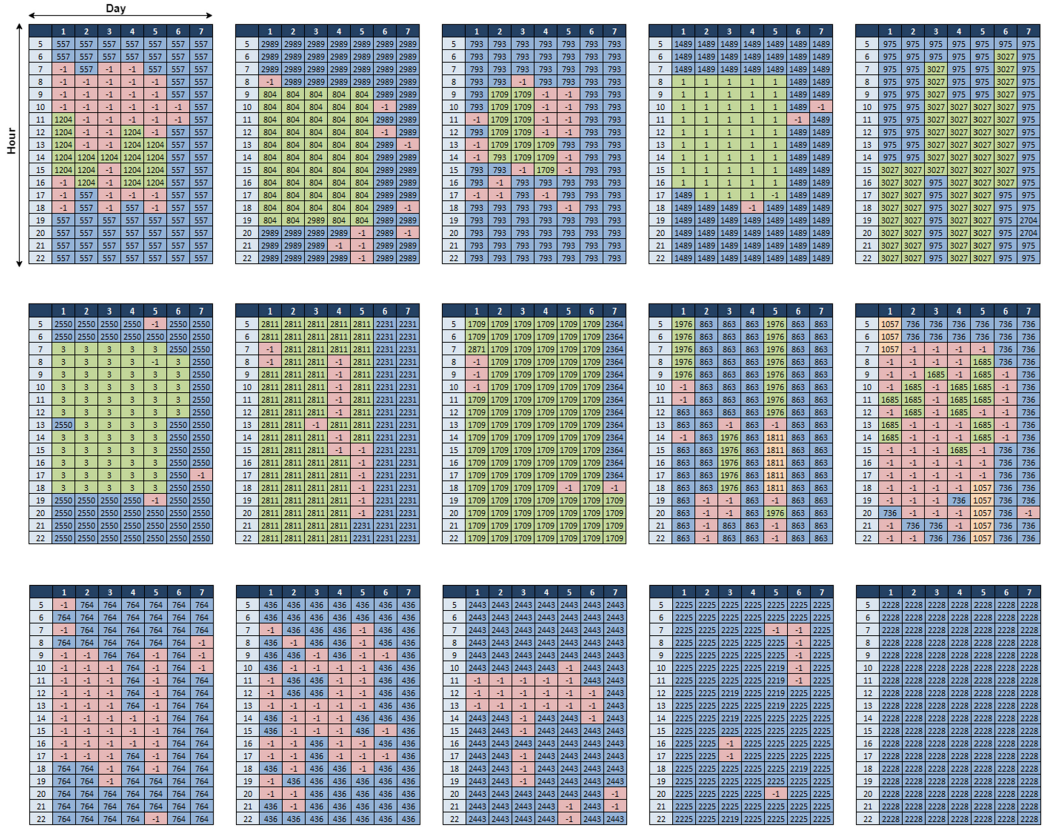


Fig. 9. Examples of original diaries. The columns represent the day of the week (1 - Sunday, . . . , 7 - Saturday) and rows represent the hour of the day (5:00–22:00). The location number is the statistical area ID and the cells marked in blue represents home, green represents work, and pink represents traveling.

C GENERATED DIARIES EXAMPLES

This appendix presents examples of generated diaries from the generation models described in Section 3. In every figure, the columns represent the day of the week (1 - Sunday, …, 7 - Saturday) and rows represent the hour of the day (5:00–22:00). The location number is the statistical area ID and the cells marked in blue represents home, green represents work, and pink represents traveling. For the PWM model diaries (as shown in Figure 10), only one gray color was used due to the amount of different statistical area IDs.

	1	2	3	4	5	6	7
5	1	3047	770	2183	575	2819	555
6	2541	2264	2425	784	2878	756	3023
7	-1	-1	2474	-1	2356	1747	172
8	2851	1	2822	2467	-1	2335	2195
9	-1	1731	757	420	2509	575	759
10	-1	1498	807	777	1614	1489	-1
11	775	847	760	-1	796	2335	2463
12	-1	1608	3048	2830	2367	2879	2351
13	1215	1	-1	2031	2550	2844	331
14	2222	-1	2986	-1	-1	1702	2508
15	2860	2255	3019	794	986	880	-1
16	397	-1	1380	777	1617	789	1492
17	2366	1	1732	2281	555	1675	1617
18	1710	-1	2475	2986	2880	1682	2336
19	2880	2188	2382	1735	-1	2365	1
20	-1	1697	1778	1	2825	778	2469
21	2024	2433	803	222	2895	2891	968
22	1778	1747	2508	1632	1215	2512	1683

	1	2	3	4	5	6	7
5	2454	2881	2419	2331	2331	1762	1632
6	738	738	204	2880	835	769	2848
7	783	-1	827	758	1489	231	-1
8	780	2306	996	2708	2	-1	738
9	-1	-1	2509	-1	2548	-1	2894
10	-1	805	2849	-1	419	1608	2472
11	833	836	2871	-1	3021	850	837
12	1779	744	-1	2890	-1	-1	851
13	841	777	2878	1834	950	2282	1700
14	1680	2210	2984	2985	-1	2813	2884
15	994	2344	1742	2468	798	3018	2817
16	2855	1776	785	2347	2506	-1	2324
17	1277	224	981	834	-1	2338	993
18	964	736	2324	738	1487	2886	2864
19	2182	2509	-1	753	-1	1	1
20	2476	1	2445	2520	779	2280	776
21	736	2708	2854	1685	2341	2309	2464
22	2459	2819	2255	-1	805	791	111

	1	2	3	4	5	6	7
5	1	2506	3020	2526	2423	1	2524
6	1	2022	1211	2433	1487	782	555
7	-1	995	2426	2352	2873	1682	807
8	2837	2988	2	2258	-1	961	2818
9	982	3022	3060	1617	-1	2351	2236
10	557	1561	2822	-1	1779	-1	1977
11	2813	-1	2878	2279	2547	1740	2256
12	-1	-1	-1	1674	807	829	-1
13	3053	-1	2857	1728	-1	-1	2350
14	1614	787	1385	3023	2474	794	1386
15	790	1502	1034	1830	1976	1690	2793
16	-1	2989	2348	846	1	959	2830
17	-1	2813	1738	865	779	2829	776
18	3021	2886	826	-1	2872	1972	2225
19	-1	1744	770	2235	2534	1561	1680
20	-1	827	2448	2866	2296	1387	-1
21	2876	2834	2203	3024	-1	1396	2517
22	2352	758	2540	2322	1396	-1	839

	1	2	3	4	5	6	7
5	2848	1747	1	2852	172	2859	2257
6	3020	984	2343	-1	-1	771	792
7	1	3025	1729	1	2450	2880	2532
8	-1	3048	2838	966	1681	2545	966
9	-1	-1	3060	557	2848	786	791
10	1051	-1	2515	2322	2818	-1	2822
11	-1	1736	797	951	2524	990	2342
12	831	-1	1741	-1	2540	2880	2023
13	-1	2359	801	2864	-1	2541	1388
14	2467	2338	1738	2200	2868	1785	2880
15	2896	2256	-1	555	662	756	1498
16	764	1734	1619	3	-1	2881	1785
17	-1	2473	2210	-1	2821	1071	2857
18	2188	1489	801	2336	2282	2	1778
19	1	1628	-1	781	769	1904	2849
20	2891	1743	2880	958	-1	-1	2859
21	1	1661	2895	2464	2282	2893	2547
22	1	171	2521	2450	2456	1487	3016

	1	2	3	4	5	6	7
5	2471	3048	2511	2	2261	3015	1725
6	2361	2898	2284	2332	3	791	1
7	2	-1	802	1713	2896	-1	2985
8	2023	989	2878	170	1735	825	1487
9	-1	2326	2876	3048	967	2199	757
10	575	2315	838	2313	2	2265	1000
11	-1	1733	-1	827	2321	-1	2321
12	2515	2342	2220	-1	-1	2866	1725
13	-1	2863	1681	2255	2888	825	3022
14	766	1	2876	738	-1	2829	-1
15	1833	2270	-1	2878	1819	2182	1746
16	2023	780	-1	-1	959	2872	-1
17	950	2	-1	2814	1632	1387	2524
18	2434	2871	-1	878	-1	754	2383
19	-1	-1	2522	2534	1040	2227	2200
20	2887	801	1616	994	2353	2524	985
21	210	-1	2513	2548	1739	1980	2353
22	2506	1	2332	827	-1	2186	3055

Fig. 10. Examples of generated diaries from the PWM model.

Diaries generated from the MC model:

	1	2	3	4	5	6	7
5	2340	2340	2340	2340	2340	2340	2340
6	2340	2340	2340	-1	2340	2340	2340
7	2340	2340	2340	2187	2340	2340	2340
8	2344	2344	2340	2187	-1	2340	2340
9	2344	2344	2340	2187	-1	2340	2340
10	2344	2344	2340	2187	-1	2340	2340
11	2344	2344	2340	2187	-1	2340	2340
12	2344	2344	2340	2187	2340	2340	2340
13	2344	2340	2340	2187	-1	2340	2340
14	2340	2340	2340	2187	-1	2340	2340
15	2340	2340	2340	2187	-1	2340	2340
16	2340	2340	2340	2187	-1	2340	2340
17	2340	2340	2340	-1	-1	2340	2340
18	2340	2340	2340	2340	-1	2340	2340
19	2340	2340	2340	2340	-1	2340	2340
20	2340	2340	2340	2340	2340	2340	2340
21	2340	2340	2340	2340	2340	2340	2340
22	2340	2340	2340	2340	2340	2340	2340

	1	2	3	4	5	6	7
5	1094	1094	1094	1094	1094	1094	1094
6	1094	1094	1094	1094	1094	1094	1094
7	1094	1094	1094	1094	1094	1094	1094
8	1094	2534	1094	1094	2534	1094	1094
9	2534	2534	2534	2534	2534	1094	1094
10	2534	2534	2534	2534	2534	1094	1094
11	2534	2534	2534	2534	2534	1094	1094
12	2534	2534	2534	2534	2534	1094	1094
13	2534	2534	2534	2534	2534	1094	1094
14	2534	2534	2534	2534	2534	1094	1094
15	2534	2534	2534	2534	2534	1094	1094
16	2534	2534	2534	2534	2534	1094	1094
17	2534	2534	2534	-1	2534	1094	1094
18	2534	2534	2534	-1	1094	1094	1094
19	1094	2534	2534	-1	1094	1094	1094
20	1094	1094	1094	1094	1094	1094	1094
21	1094	1094	1094	1094	1094	1094	1094
22	1094	1094	1094	1094	1094	1094	1094

	1	2	3	4	5	6	7
5	847	847	847	847	847	847	847
6	847	847	847	847	847	847	847
7	847	847	847	847	847	847	847
8	847	847	847	847	847	-1	847
9	847	847	847	847	847	-1	847
10	-1	847	847	847	847	847	847
11	-1	847	847	847	847	847	847
12	-1	847	847	847	847	847	847
13	-1	847	847	847	847	-1	847
14	847	847	847	847	847	847	847
15	847	847	847	847	847	847	847
16	847	847	847	847	847	847	-1
17	847	847	847	847	847	-1	847
18	847	847	847	847	847	-1	847
19	847	847	847	847	847	-1	847
20	847	847	847	847	847	847	847
21	847	847	847	847	847	847	847
22	-1	847	847	847	847	847	-1

	1	2	3	4	5	6	7
5	987	987	987	987	-1	-1	987
6	-1	987	987	-1	-1	-1	987
7	-1	-1	987	-1	-1	-1	987
8	-1	987	987	-1	-1	-1	987
9	-1	987	987	-1	-1	-1	987
10	-1	-1	987	-1	-1	3026	987
11	-1	987	987	987	-1	3026	987
12	987	987	987	987	987	3026	987
13	987	987	987	987	987	3026	987
14	987	-1	987	987	987	3026	987
15	3026	3026	3026	3026	3026	3026	987
16	3026	3026	3026	3026	3026	987	987
17	3026	3026	3026	3026	3026	987	987
18	3026	3026	3026	3026	3026	987	987
19	3026	3026	3026	3026	3026	987	987
20	3026	987	987	3026	3026	987	-1
21	3026	987	987	987	3026	987	-1
22	987	987	987	987	3026	987	-1

	1	2	3	4	5	6	7
5	1996	1996	1996	1996	1996	1996	1996
6	1996	1996	1996	1996	1996	1996	1996
7	1996	1996	1996	1996	1996	1996	1996
8	-1	-1	-1	-1	-1	1996	1996
9	1661	1661	1661	1661	1661	1996	1996
10	1661	1661	1661	1661	1661	1996	1996
11	1661	1661	1661	1661	1661	1996	1996
12	1661	1661	1661	1661	1661	1996	1996
13	1661	1661	1661	1661	1661	1996	1996
14	1661	1661	1661	1661	1661	1996	1996
15	-1	-1	-1	1661	1661	1996	1996
16	1996	1996	1996	1996	1996	1996	1996
17	1996	1996	1996	1996	1996	1996	1996
18	1996	1996	1996	1996	1996	1996	1996
19	1996	1996	1996	1996	1996	1996	1996
20	1996	1996	1996	1996	1996	1996	1996
21	1996	1996	1996	1996	1996	1996	1996
22	1996	1996	1996	1996	1996	1996	1996

Fig. 11. Examples of generated diaries from the MC model.

Diaries generated from the VMM model:

	1	2	3	4	5	6	7
5	2891	2891	2891	2891	2891	2891	2891
6	2891	2891	2891	2891	2891	2891	2891
7	2891	2891	2891	2891	2891	2891	2891
8	2891	2891	2891	-1	2891	2891	2891
9	2891	2891	2891	-1	2891	2891	2891
10	2891	2891	2891	-1	2842	2891	2891
11	2891	2891	2891	2842	2842	-1	2891
12	2891	2891	2891	2842	2842	2891	2891
13	2891	2891	2891	2842	2842	2891	2891
14	2891	2891	2891	2842	2842	2891	2891
15	2891	2891	2891	2842	2842	2891	2891
16	2891	2891	-1	2842	-1	2891	2891
17	2891	2891	2891	2891	2891	2891	2891
18	2891	2891	2891	2891	2891	2891	2891
19	2891	2891	2891	2891	2891	2891	2891
20	2891	2891	2891	2891	2891	2891	2891
21	2891	2891	2891	2891	2891	2891	2891
22	2891	2891	2891	2891	2891	2891	2891

	1	2	3	4	5	6	7
5	799	799	799	799	799	799	799
6	799	795	795	799	799	795	799
7	799	799	799	799	799	795	799
8	799	799	799	799	799	799	799
9	795	799	799	799	799	799	799
10	795	795	799	795	799	799	799
11	795	795	799	795	795	799	799
12	795	795	795	795	795	799	799
13	795	795	795	795	795	799	799
14	795	795	795	795	795	799	-1
15	-1	795	795	795	795	799	799
16	795	795	795	795	795	799	799
17	795	795	795	795	795	799	799
18	795	795	795	795	795	799	799
19	799	795	795	795	795	799	799
20	799	799	799	795	799	799	799
21	799	799	799	795	799	799	799
22	799	799	799	799	799	799	799

	1	2	3	4	5	6	7
5	2334	2334	2334	2334	2334	2334	2334
6	2334	2334	2334	2334	2334	2334	2334
7	2334	2334	2334	2334	2334	2334	2334
8	2334	2334	2334	2334	2334	2334	2334
9	2334	2334	2334	2334	2334	2334	2334
10	2334	2334	2334	2334	2334	2334	2334
11	2334	2334	2334	2334	2334	2334	2334
12	2334	2334	2334	2334	2334	2334	2334
13	2334	2334	2334	2334	2334	2334	2334
14	2334	2334	2334	2334	2334	2334	2334
15	2334	2334	2334	2334	2334	2334	2334
16	2334	2334	2334	2334	2334	2334	2334
17	2334	2334	2334	2334	2334	2334	2334
18	2334	2334	2334	2334	2334	2334	2334
19	2334	2334	2334	2334	2334	2334	2334
20	2334	2334	2334	2334	2334	2334	2334
21	2334	-1	2334	2334	2334	2334	2334
22	2334	2334	2334	2334	2334	2334	2334

	1	2	3	4	5	6	7
5	763	763	763	763	763	763	763
6	763	763	763	763	763	763	763
7	763	763	763	763	763	763	763
8	763	763	763	763	763	763	763
9	763	763	763	763	763	763	763
10	763	763	763	763	763	763	763
11	763	763	763	763	763	763	763
12	763	808	808	763	808	763	763
13	808	808	808	808	808	763	763
14	808	808	808	808	808	763	763
15	808	808	808	808	808	763	763
16	808	808	808	808	808	763	763
17	808	808	808	808	808	763	763
18	808	808	808	808	808	763	763
19	808	808	808	808	808	763	763
20	808	808	808	808	808	763	763
21	808	763	808	808	808	763	763
22	808	763	763	763	808	763	763

	1	2	3	4	5	6	7
5	2474	2474	2474	2474	2474	2474	2474
6	2474	2474	2474	2474	2474	2474	2474
7	2474	2474	2474	2474	2474	2474	2474
8	2474	2474	2474	2474	2474	2474	2474
9	2474	2474	2474	-1	-1	2474	2474
10	2474	2474	2474	-1	-1	2474	2474
11	2474	2474	2474	-1	-1	2474	2474
12	2474	2474	-1	-1	-1	2474	2474
13	2474	2474	-1	-1	-1	-1	2474
14	2474	2474	-1	-1	-1	2474	2474
15	2474	2474	-1	-1	-1	2474	2474
16	2474	2474	2474	2474	-1	2474	2474
17	2474	2474	2474	-1	2474	2474	2474
18	2474	2474	2474	2474	2474	2474	2474
19	2474	2474	2474	2474	2474	2474	2474
20	2474	2474	2474	2474	2474	2474	2474
21	2474	2474	2474	2474	2474	2474	2474
22	2474	2474	2474	2474	2474	2474	2474

Fig. 12. Examples of generated diaries from the VMM model.

Diaries generated from the LSTM Loc2Vec model:

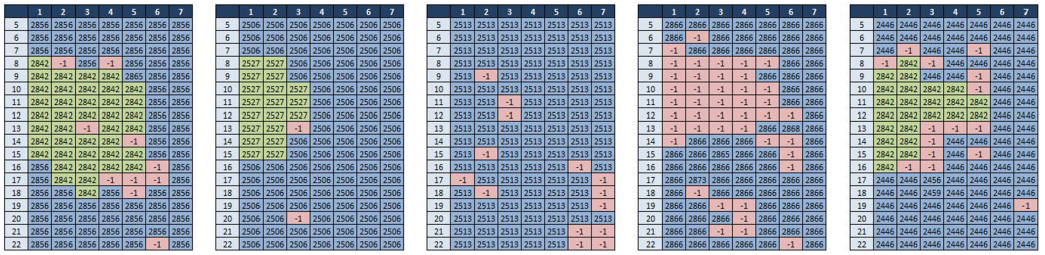


Fig. 13. Examples of generated diaries from the LSTM Loc2Vec model.

Diaries generated from the LSTM Embeddings model:

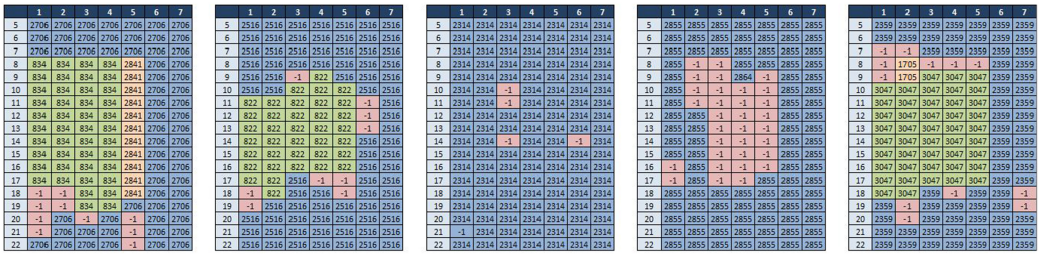


Fig. 14. Examples of generated diaries from the LSTM Embeddings model.

REFERENCES

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. 2016. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 961–971.

[2] Florent Altché and Arnaud de La Fortelle. 2017. An LSTM network for highway trajectory prediction. In *Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems*. IEEE, 353–359.

[3] Claudio Agostino Ardagna, Marco Cremonini, Ernesto Damiani, S De Capitani Di Vimercati, and Pierangela Samarati. 2007. Location privacy protection through obfuscation-based techniques. In *Proceedings of the IFIP Annual Conference on Data and Applications Security and Privacy*. Springer, 47–60.

[4] Sameer Badaskar, Sachin Agarwal, and Shilpa Arora. 2008. Identifying real or fake articles: Towards better language modeling. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing: Volume-II*.

[5] Halgurt Bapierre, Georg Groh, and Stefan Theiner. 2011. A variable order markov model approach for mobility prediction. *Pervasive Computing* (2011), 8–16.

[6] Omer Barak, Gabriella Cohen, and Eran Toch. 2016. Anonymizing mobility data using semantic cloaking. *Pervasive and Mobile Computing* 28, 1 (2016), 102–112.

[7] Irad Ben-Gal, Ayala Shani, Andre Gohr, Jan Grau, Sigal Arviv, Armin Shmilovici, Stefan Posch, and Ivo Grosse. 2005. Identification of transcription factor binding sites with variable-order Bayesian networks. *Bioinformatics* 21, 11 (2005), 2657–2666.

[8] I. Ben-Gal, S. Weinstock, G. Singer, and N. Bambos. 2019. Clustering users by their mobility behavioral patterns. *ACM Transactions on Knowledge Discovery from Data* 13, 4 (2019), 1–28.

[9] V. Bindschaedler and R. Shokri. 2016. Synthesizing plausible privacy-preserving location traces. *In Proceedings of the 2016 IEEE Symposium on Security and Privacy*. 546–563.

[10] Ali Borji. 2019. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding* 179, 1 (2019), 41–65.

[11] Oliver Burkhard, Rein Ahas, Erki Saluveer, and Robert Weibel. 2017. Extracting regular mobility patterns from sparse CDR data without a priori assumptions. *Journal of Location Based Services* 11, 2 (2017), 78–97.

[12] C. Y. Chow and M. F. Mokbel. 2011. Trajectory privacy in location-based services and data publication. *ACM Sigkdd Explorations Newsletter* 13, 1 (2011), 19–29.
[13] Jean-Michel Claverie and Stephane Audic. 1996. The statistical significance of nucleotide position-weight matrix matches. *Bioinformatics* 12, 5 (1996), 431–439.
[14] Y. A. De Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel. 2013. Unique in the crowd: The privacy bounds of human mobility. *Scientific Reports* 3, 1 (2013), 1376.
[15] Eustasio Del Barrio, Juan A. Cuesta-Albertos, Carlos Matrán, and Jesús M. Rodríguez-Rodríguez. 1999. Tests of goodness of fit based on the L2-Wasserstein distance. *Annals of Statistics* 27, 4 (1999), 1230–1239.
[16] Ehab ElSalamouny and Sébastien Gambs. 2016. Differential privacy models for location-based services. *Transactions on Data Privacy* 9, 1 (2016), 15–48. https://hal.inria.fr/hal-01418136/file/tdp.a220a15.pdf.
[17] Ghazi Falah. 1996. Living together apart: Residential segregation in mixed Arab-Jewish cities in Israel. *Urban Studies* 33, 6 (1996), 823–857.
[18] Yuchen Fan, Yao Qian, Feng-Long Xie, and Frank K. Soong. 2014. TTS synthesis with bidirectional LSTM based recurrent neural networks. In *Proceedings of the 15th Annual Conference of the International Speech Communication Association*.
[19] William Fedus, Ian Goodfellow, and Andrew M. Dai. 2018. MaskGAN: Better text generation via filling in the_. arXiv:1801.07736. Retrieved from https://arxiv.org/abs/1801.07736.
[20] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. 2012. Next place prediction using mobility markov chains. In *Proceedings of the 1st Workshop on Measurement, Privacy, and Mobility*. 1–6.
[21] Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. 2019. Gltr: Statistical detection and visualization of generated text. arXiv:1906.04043. Retrieved from https://arxiv.org/abs/1906.04043.
[22] S. C. Geyik, E. Bulut, and B. K. Szymanski. 2010. PCFG based synthetic mobility trace generation. In *Proceedings of the 2010 IEEE Global Telecommunications Conference*. 1–5.
[23] Takashi Gojobori, Li Wen, Hsiung, and Graur Dan. 1982. Patterns of nucleotide substitution in pseudogenes and functional genes. *Journal of Molecular Evolution* 18, 5 (1982), 360–369.
[24] Marco Gruteser and Baik Hoh. 2005. On the anonymity of periodic location samples. In *Proceedings of the International Conference on Security in Pervasive Computing*. Springer, 179–192.
[25] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.
[26] D. Huang, X. Song, Z. Fan, R. Jiang, R. Shibasaki, Y. Zhang, and Y. Kato. 2019. A variational autoencoder based generative model of urban human mobility. In *Proceedings of the 2019 IEEE Conference on Multimedia Information Processing and Retrieval*. 425–430.
[27] K. Ouyang, R. Shokri, D. S. Rosenblum, and W. Yang 2018. A non-parametric generative model for human trajectories. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 3812–3817.
[28] B. A. Kipnis and I. Schnell. 1978. Changes in the distribution of Arabs in mixed Jewish-Arab cities in Israel. *Economic Geography* 54, 2 (1978), 168–180.
[29] Janne Korhonen, Petri Martinmäki, Cinzia Pizzi, Pasi Rastas, and Esko Ukkonen. 2009. MOODS: fast search for position weight matrix matches in DNA sequences. *Bioinformatics* 25, 23 (2009), 3181–3182.
[30] Vaibhav Kulkarni and Benoît Garbinato. 2017. Generating synthetic mobility traffic using rnns. In *Proceedings of the 1st Workshop on Artificial Intelligence and Deep Learning for Geographic Knowledge Discovery*. 1–4.
[31] Vaibhav Kulkarni, Natasa Tagasovska, Thibault Vatter, and Benoit Garbinato. 2018. Generative models for simulating mobility trajectories. arXiv:1811.12801. Retrieved from https://arxiv.org/abs/1811.12801.
[32] Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. arXiv:1603.07771. Retrieved from https://arxiv.org/abs/1603.07771.
[33] H. Litwin and E. V. Sapir. 2008. 3.4 Israel: Diversity among population groups. *3The SHARE Respondents* 1, 1 (2008), 93.
[34] Yanchi Liu, Chuanren Liu, Xinjiang Lu, Mingfei Teng, Hengshu Zhu, and Hui Xiong. 2017. Point-of-interest demand modeling with human mobility patterns. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 947–955.
[35] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. arXiv:1301.3781. Retrieved from https://arxiv.org/abs/1301.3781.
[36] Uwe Ohler, Stefan Harbeck, Heinrich Niemann, Elmar Noth, and Martin G. Reese. 1999. Interpolated markov chains for eukaryotic promoter recognition. *Bioinformatics (Oxford, England)* 15, 5 (1999), 362–369.
[37] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. 311–318.
[38] N. Pelekis, C. Ntrigkogias, P. Tampakis, S. Sideridis, and Y. Theodoridis. 2013. Hermoupolis: A trajectory generator for simulating generalized mobility patterns. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. 659–662.

[39] Roberto Pellungrini, Luca Pappalardo, Francesca Pratesi, and Anna Monreale. 2017. A data mining approach to assess privacy risk in human mobility data. *ACM Transactions on Intelligent Systems and Technology* 9, 3 (2017), 1–27.
[40] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro. 2017. What does the crowd say about you? evaluating aggregation-based location privacy. *Proceedings on Privacy Enhancing Technologies* 2017, 4 (2017), 156–176.
[41] Youyang Qu, Shui Yu, Wanlei Zhou, and Yonghong Tian. 2020. Gan-driven personalized spatial-temporal private data sharing in cyber-physical social systems. *IEEE Transactions on Network Science and Engineering* 7, 4 (2020), 2576–2586.
[42] Jinmeng Rao, Song Gao, Yuhao Kang, and Qunying Huang. 2020. LSTM-TrajGAN: A deep learning approach to trajectory privacy protection. arXiv:2006.10521. Retrieved from https://arxiv.org/abs/2006.10521.
[43] Reza Shokri, George Theodorakopoulos, Jean-Yves Le Boudec, and Jean-Pierre Hubaux. 2011. Quantifying location privacy. In *Proceedings of the 2011 IEEE Symposium on Security and Privacy*. IEEE, 247–262.
[44] Rahul Siddharthan. 2010. Dinucleotide weight matrices for predicting transcription factor binding sites: generalizing the position weight matrix. *PloS One* 5, 3 (2010), e9722.
[45] Ha Yoon Song, Moo Sang Baek, and Minsuk Sung. 2019. Generating human mobility route based on generative adversarial network. In *Proceedings of the 2019 Federated Conference on Computer Science and Information Systems*. IEEE, 91–99.
[46] Libo Song, David Kotz, Ravi Jain, and Xiaoning He. 2004. Evaluating location predictors with extensive Wi-Fi mobility data. In *IEEE Annual Joint Conference: INFOCOM, IEEE Computer and Communications Societies*. IEEE, 1414–1424.
[47] Junkai Sun, Junbo Zhang, Qiaofei Li, Xiuwen Yi, Yuxuan Liang, and Yu Zheng. 2020. Predicting citywide crowd flows in irregular regions using multi-view graph convolutional networks. *IEEE Transactions on Knowledge and Data Engineering* 34, 5 (2020), 2348–2359.
[48] L. Sun and K. W. Axhausen. 2016. Understanding urban mobility patterns with a probabilistic tensor factorization framework. In *Proceedings of the Transportation Research Part B: Methodological*. 511–524.
[49] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. LSTM neural networks for language modeling. In *Proceedings of the 13th Annual Conference of the International Speech Communication Association*.
[50] E. Toch, B. Lerner, E. Ben-Zion, and I. Ben-Gal. 2019. Analyzing large-scale human mobility data: A survey of machine learning methods and applications. *Knowledge and Information Systems* 58, 3 (2019), 501–523.
[51] Zijun Yao, Yanjie Fu, Bin Liu, Wangsu Hu, and Hui Xiong. 2018. Representing urban functions through zone embedding with human mobility patterns. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 3919–3925.
[52] Jing Yuan, Yu Zheng, and Xing Xie. 2012. Discovering regions of different functions in a city using human mobility and POIs. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 186–194.
[53] S. Zheng, Y. Yue, and J. Hobbs. 2016. Generating long-term trajectories using deep hierarchical networks. In *Proceedings of the Advances in Neural Information Processing Systems*. 1543–1551.

Received November 2021; accepted March 2022