Modeling and Predicting Individual Transitions within the Homelessness System

Khandker Sadia Rahman[1*†] and Charalampos Chelmis[1†]

[1]Department of Computer Science, University at Albany, SUNY, 1400 Washington Ave, Albany, 12222, NY, USA.

*Corresponding author(s). E-mail(s): srahman2@albany.edu;
Contributing authors: cchelmis@albany.edu;
†These authors contributed equally to this work.

Abstract

This study focuses on how individuals navigate the homelessness system over time, with the ultimate goal of securing stable housing. Administrative data collected by homeless service providers are used to infer the unobserved underlying network of homeless services. A similarity score between the ordered sequences of services that individuals receive is proposed. The score leverages the structure of the inferred network in addition to historical observations to identify individuals with similar trajectories. In doing so, the service an individual will be assigned to next can be predicted. Extensive experiments show that the proposed approach performs well not only on predicting exit from the system, or simply guessing high frequency services (as most baselines), but is also successful in less frequent scenarios. Building a model that learns to replicate the dynamics of the existing system is the first step towards developing computational methods to maximize outcomes (i.e., ensuring that as many homeless individuals as possible secure stable housing).

Keywords: Complex systems, network inference, similarity, trajectory prediction

1 Introduction

The U.S. Department of Housing and Urban Development (HUD) defines homelessness as a situation where an individual experiences lack of fixed, regular, and adequate nighttime residence [1]. Homelessness poses a long-standing problem to the society with more than 582,000 people having experienced homelessness on a single night in 2022. Among which, 30% of the individuals experience prolonged homelessness for at least 12 months, or repeated homelessness over a period of three years (i.e., chronic homeless [2]) [3]. Numerous methods have been proposed to predict reentry [4–6] and chronic homelessness risk prediction [7, 8]. Most such works are formulated as a binary–classification task that doesn't capture the complexities of the homeless service system as a whole. On the other hand, [9] explored the utility of machine learning models (eg., decision tree, random forest, multi-class AdaBoost) to predict the service allocation upon entry to the homeless system.

This work focuses on individuals experiencing chronic homelessness, broadly defined here as individuals entering the homeless system two or more times. Viewing the history of each individual as a sequence of services and time of stay within each service, the goal is to learn a model that can be used to accurately estimate the next service an individual will be assigned to within the homeless system in the future. To address this problem, we propose an approach that, given the history of an individual, identifies individuals with similar sequences of homeless services, based on which it predicts the next service the given individual will be assigned to. To model the overall behavior of individuals within the homeless system, we represent the homeless system as a network of interconnected services which individuals traverse over time. Our comprehensive experimental evaluation demonstrates the ability of the proposed approach to accurately model the dynamics of this complex sociotechnical system. Our key contributions can be summarized as follows:

- We infer the network of homeless services from administrative data collected by homeless service providers.
- We define a similarity score between ordered sequences of services that are visited by the homeless as they traverse the network of services.
- We propose a method that, given the history of an individual, can predict the service she will be assigned to next.
- To ensure the reproducibility of the work, the source code is available at https://github.com/IDIASLab/TRACE.

The remaining paper is organized as follows. Section 2 summarizes related work. Section 3 introduces the problem statement. Section 4 describes the process used to infer the network of homeless services from administrative data. Section 5 introduces the proposed similarity score. Section 6 discusses the proposed trajectory similarity estimation and probabilistic prediction method and analyzes its computational complexity, whereas Section 7 explains the tiered model for decision-making, MetaTier. Section 8 describes the data, baselines, and metrics used to evaluate the performance of the proposed approach,

whereas Section 9 provides a detailed analysis of the experimental results. Finally, Section 10 concludes with a discussion of the limitations of this study, and potential future research directions.

2 Related work

Prior research that is most related to this study can be separated into three main themes as follows.

The first theme comprises approaches that model reentry and chronic homelessness prediction as binary classification (e.g., [4, 5, 8]). Unlike such methods, this work addresses the more challenging problem of determining whether an individual will exit the homeless system, or the exact program she will be assigned to next. Furthermore, this work is the first to leverage the history of an individual (both as a sequence of events, and her trajectory over the unobserved network of services) to learn an accurate model.

The second theme of prior work focuses on similarity measures for time–series data [10], including but not limited to Euclidean distance [11], Dynamic Time Warping based measures [12], shapelets [13], and information theoretic measures [14]. While suited for numerical (time–series) data, our study involves trails of timestamped categorical data. Prior research on sequence analysis includes optimal matching [15] and principle of minimal shared time [16]. Such methods are often used to determine a common subsequence between two categorical sequences. However, data points in a sequence are typically assumed to be independent and identically distributed. In contrast, the approach described here uncovers the network that generates the observed sequences, and leverages this knowledge, along with temporal overlap, to compute a novel similarity score between two categorical trajectories.

The third theme of relevant work focuses on network inference from data [17]. Such methods either focus solely on inferring the network structure (e.g., [18]), or infer transmission rates in addition to structure from observed traces of diffusion processes (e.g., [19]). Different to this study, existing work relies on unordered node activations, assumes that diffusion traces are directed acyclic graphs (DAG) and that the transition probability for one node to another is fixed and same for all edges, or infers pairwise interactions for pairs of nodes that are expected to be directly connected via an edge. In this study, the temporal chain of events is observed (leading to an easier inference problem) while at the same time, observed trajectories may contain cycles (resulting in a harder inference problem). It is also worth mentioning that the problem discussed in this paper differs from that of learning the structure of a directed graphical model or Bayesian network (e.g., [20]). Specifically, graphical models do not model network properties (e.g., that nodes in an observed trajectory must lie along a path in the network), whereas most Bayesian network inference algorithms try to determine the most likely DAG that is consistent with a fixed ordering. Moreover, neither approach considers long paths (and their contribution to the final transition probability) between nodes over temporal networks.

Parts of the material in this article have been presented in our prior work [21]. The present article expands on [21] both in breadth and depth, as follows. First, we provide a detailed description of the process used to infer the graph of homeless services, and analyze the computational complexity of the proposed trajectory similarity estimation and probabilistic prediction model. Second, motivated by our findings in [21], we propose a tiered model that improves prediction accuracy, and therefore the chances of the proposed approach to be adopted in the real–world. Third, we extend our experimental evaluation with additional results and insights on the methods presented in [21], and comparisons to the tiered model proposed here.

3 Problem Statement

Homeless service providers offer services that are organized in *project types* (e.g., emergency shelters, transitional housing) [22]. We denote the set of project types as $P = \{p_1, p_2, \ldots, p_n\}$, and the set of individuals requesting services multiple times (i.e., reentering the homeless system more than once) as $C = \{c_1, c_2, \ldots, c_m\}$. Reentries can be viewed as temporally ordered sequences of tuples $(p_i, t_i = [s_i, e_i])$, where $p_i \in P$, and s_i and e_i are the times at which individual $c \in C$ enters to, and exits from p_i, accordingly. Such a *trajectory*, $T_c = (p_1, [s_1, e_1]), (p_2, [s_2, e_2]), \ldots, (p_N, [s_N, e_N]))$ for each individual $c \in C$, where for each two consecutive tuples $s_{i+1} \geq e_i$, records her transitions from p_i to p_{i+1}, which in turn record the trail of each individual within the unobserved homeless service network. Given a set of trajectories $\mathcal{T}$, and query trajectory $T_q = ((q_1, [s_1, e_1]), \ldots, (q_N, [s_N, e_N])$ of an individual $q \in C$, up to time e_N, we aim at predicting the project type $\hat{q}_{N+1} \in P$ that she will be assigned to immediately after q_N (or exit if she is likely to exit the system).

4 Inferring the graph of homeless services

Administrative data collected by homeless service providers record a timeline of services received by each individual, including the beginning and end dates of each service, transitions between service types, and exits and reentries (i.e., receipt of services after exiting the system). Such data offers a unique opportunity to study how individuals navigate through a complex sociotechnical system over time, with the ultimate goal of securing stable housing. Unfortunately, the data only records what services an individual receives and when, but not why is she assigned to a particular program (e.g., emergency shelter) versus another (e.g., long term housing).

Since the underlying connectivity of homeless services (i.e., the potential paths an individuals can take once she is admitted into the homeless system) is neither directly observable nor known, we set forth to uncover the aggregate dynamics of the homeless system from the observed sequences of services (and corresponding entry and exit dates) that it generates for each individual. For every edge of the network, we additionally wish to estimate the corresponding transition probability between services. This is an important step towards

Table 1 Explanation of main symbols used in this article.

Symbol	Description
C	Set of chronically homeless individuals in our dataset
P	Set of homeless service providers' offered project types
m, n	Number of individuals $c \in C$, and project types $p \in P$
$t_i = [s_i, e_i]$	i^{th} time interval at which an individual enters and exits a given project type, p_i
p_i	Project type at i^{th} step of a trajectory
$\mathcal{T}$	Set of all trajectories
$\mathcal{T}_{train}$	Set of trajectories $\in \mathcal{T}$ used for inferring the homeless service network
$\mathcal{T}_{test}$	Set of trajectories $\in \mathcal{T} \setminus \mathcal{T}_{train}$ used for testing
$\mathcal{H}$	Set of historical trajectories $\in \mathcal{T}_{test}$. Given a query Q, similar trajectories are identified from this set
Q	Set of query trajectories $\in \mathcal{T}_{test} \setminus \mathcal{H}$
T_c, T_q	Trajectories of individual c and q, accordingly
$T[t]$	Segment of trajectory T during time interval t
$\mathtt{t}$	Duration of time interval t
$t_i \cap t_j$	Temporal overlap of time intervals t_i and t_j
$\mathcal{G} = (P, E)$	Directed graph with node set P and edge set E
w_{ij}	Weight of edge $(i, j) \in E$
$f_{ij,kl}$	Frequency of $k-$step paths from p_i to p_j at offset l
α	Attenuation factor used in edge weight computation
β	Attenuation factor used in similarity computation for baseline **Sim–attenuate**
N, M	Maximum number of steps and offsets, respectively
$\sigma(T_q, T_c)$	Similarity between two trajectories, T_q and T_c
$d(q, T, t_q)$	Distance between node q and trajectory T within time interval t_q
$NB(\cdot)$	Out–neighbor of a node $\in V$
p_c, q_N	Last matching project type in trajectory T_c, project type at N^{th} time interval in trajectory T_q
$\hat{q}_{N+1}$	Service predicted to be assigned at time step $N + 1$
γ	Number of past assignments (i.e., length of historical data) in T_q used for identifying similar trajectories
$\mathcal{F}$, λ	Unique values of effective lengths of trajectories in $\mathcal{T}$, element in $\mathcal{F}$
$\mathcal{M}_{exit}, \mathcal{M}_{int}$	Predictive model for exit and interim points accordingly
$\mathcal{D}_X, \mathcal{D}_Y$	Input and output of MetaTier, respectively
$X_{\mathcal{M}}$, $Y_{\mathcal{M}}$	Input and output of Meta model, respectively

deriving computational models to predict individuals' transitions over such network, and subsequently developing methods to maximize outcomes (e.g. securing stable housing).

We begin by modelling the network as a directed graph $\mathcal{G} = (P, E)$, where P is the set of nodes representing services visited by individuals in C, and E is the set of edges between nodes, such that a directed edge appears between p_i and p_j if at least one trajectory in $\mathcal{T}$ exists, in which p_j appears after p_i. We determine the weight of each edge $(p_i, p_j) \in E$ based on the *number of steps* taken before reaching p_j from p_i, and the position where p_i appears in the trajectory (i.e., *offset*). Specifically, a path from p_i to p_j in a trajectory T involves $j - i + 1$ steps starting from offset i. Therefore:

$$w_{ij} = \sum_{k=1}^{N} \sum_{l=0}^{M-1} \alpha^{k-1+l} f_{ij,kl}, \tag{1}$$

where N is the maximum number of steps, and M the largest offset, $\alpha \in [0, 1]$ is some attenuation factor, and $f_{ij,kl}$ denotes the number of times a transition from p_i to p_j appears in any path across all trajectories in $\mathcal{T}$ with k steps at offset l. Finally, we normalize the weights of outgoing edges at each node to sum to 1. Edge weights satisfy the following properties:

- $w_{ij} > 0, \forall p_i, p_j \in P$,
- w_{ij} is undefined if $\exists$ no path from p_i to p_j,
- $w_{ij} \approx 0$ if path from p_i to p_j is long, and
- $\sum_{j \in V} w_{ij} = 1, \forall p_i \in P$.

Figure 1 shows the key steps of the graph inference process. First, for each trajectory, all possible unique paths are extracted. The value of $f_{ij,kl}$ is then computed by counting the frequency of each path across all trajectories. For example, in Figure 1(c), the path between 11 and 13 appears in the trajectory of individual X three times. For offset 0, the path $\{11, 13\}$ appears two times with 1 and 3 steps between the two project types. Similarly for offset 2, it appears once with 1 step between them. Note that the 1−step path appears twice in this toy example, albeit with different offsets. To avoid double counting, only unique paths at each number of steps are included in the computation of $f_{ij,kl}$. Next, edge weights are calculated using Eq.(1), number of steps, and offset of each path appearing in the trajectories. Consider for example the weight $w_{11,exit}$ of edge (11, exit). Three terms, corresponding to (i) α^{4-1+0} for the 4−step path at offset 0, (ii) α^{2-1+2} for 2−step path at offset 2, and (iii) α^{2-1+0} for the 2−step path at offset 0 are computed. Next, the number of trajectories these paths appear in are counted, resulting in $f_{(11,exit),(3,0)} = 1$, $f_{(11,exit),(2,2)} = 1$, and $f_{(11,exit),(2,2)} = 2$. Using Eq.(1), we get $w_{11,exit} = \alpha^{3+0} \times 1 + \alpha^{1+2} \times 1 + \alpha^{1+0} \times 2 = 1.25$. The resulting graph is shown in Fig. 1(e).

For comparison, Figs. 1(d) and (f) show two alternative representations of the unobserved network of homeless services. Specifically, Fig. 1(d) models the homeless system as an aggregate network, in which an directed edge from node i to j signifies that there is at least one *single step* transition from i to j in the dataset. Conversely, Fig. 1(f) shows a temporal graph representation of the homeless system. While often used for dynamic networks [24], a temporal graph view of the homeless system would result in extremely sparse snapshots (as illustrated in Fig. 1(f)), which would in turn have limited predictive power. In contrast, the proposed inference mechanism accounts for *paths*, as opposed to direct edges [23], while at the same time ensuring low sparsity by constructing snapshots not based on arbitrarily chosen durations, but based on *effective lengths*. Specifically, the *effective* trajectories of individuals towards exiting the system can be obtained from their *actual paths* by ignoring backward transitions (i.e., admission to a program type already admitted in the past) [23]. For instance, the effective trajectory of client X in Fig. 1 is $\{11, 13\}$, as opposed to her actual trajectory, which is $\{11, 13, 11, 13, exit\}$. Given the unique effective lengths $\mathcal{F}$ of all trajectories in $\mathcal{T}$, a transition graph $\mathcal{G}_\lambda$ is

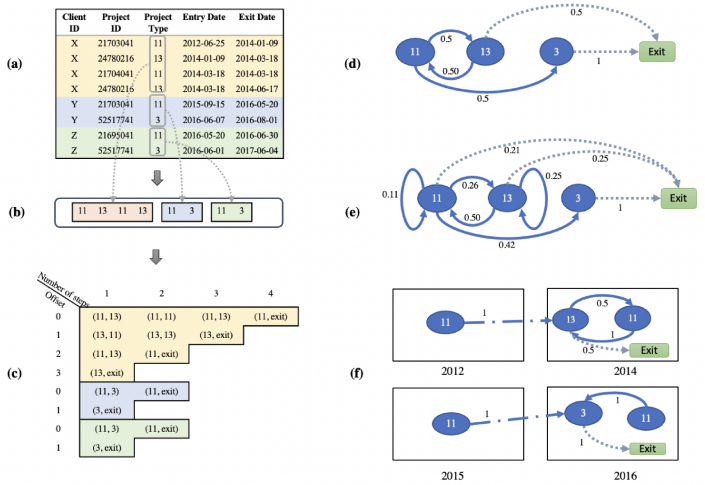


Fig. 1 Illustration of the graph inference process: (a) Sample data for three individuals, (b) Trajectories extracted from the data, (c) All possible unique paths for number of steps and offset, (d) Aggregate transition graph, as defined in [23], where a directed edge from node i to j signifies that there is at least one direct transition from i to j in $\mathcal{T}$, (e) Inferred weighted transition graph, where edge weights are computed using Eq.(1), (f) Temporal network view of the toy dataset shown in (a), where edges within a snapshot exist if corresponding transitions are recorded within a specific time interval (e.g., year), and intra–edges denote transitions that span time intervals (e.g., individual X transitioned from project type 11 to 13 in 2014).

inferred $\forall \lambda \in \mathcal{F}$. As an example, Fig.1(e) shows the inferred graph $\mathcal{G}_2$ (i.e., effective length 2) for the toy data in (a).

5 Trajectory Similarity

To compare a given individual's trajectory with historical trajectories, we need a notion of similarity. Specifically, given graph $\mathcal{G}$, a query trajectory T_q, and a historical trajectory T_c, we wish to measure similarity, σ, between T_q and T_c, while taking into account the distance between nodes in $\mathcal{G}$ appearing in T_q and T_c respectively, the temporal overlap between the trajectories, and the time intervals individuals stayed on each node in the corresponding trajectories. Figure 2 illustrates a query, T_q, and two trajectories, T_1, and T_2. Intuitively, T_1 is more similar to T_q due to higher temporal overlap and lower distance between nodes in the two trajectories. Similarly, the temporal overlap between T_q and T_2 is substantial, however, the two sequences are less aligned in the early stages. This misalignment is captured by the distance between the corresponding nodes in the transition graph.

Let the distance between two nodes $q_i, p_j \in P$ that appear in T_q and T respectively, be the weighted shortest path distance, $d(q_i, p_j)$, between them in $\mathcal{G}$. The distance between $q_i \in T_q$ and T within time interval t_q can then be

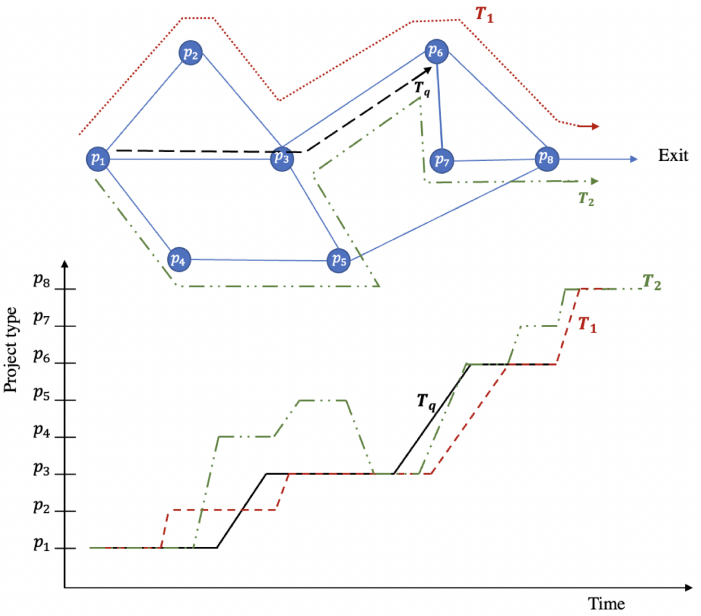


Fig. 2 Sample trajectories over a toy network of homelessness services. A query trajectory (T_q) and two similar trajectories (T_1 and T_2) are shown in black, and red and green, accordingly. Bottom plot shows a naive label encoding (i.e., project types converted into integers) of trajectories as time–series. A time–series prediction model (e.g., regression) could then be trained, but such model would miss the rich context the underlying network provides.

computed as:

$$d(q_i, T, t_q) = \frac{\min_{(p_j,t_j)\in T[t_q]} \max_{|t_j \cap t_q|} d(q_i, p_j)}{D_{\mathcal{G}}}, \tag{2}$$

where $D_{\mathcal{G}}$ is the maximum weighted distance between any two nodes in $\mathcal{G}$, and $T[t_q]$ denotes the sequence of nodes in T visited during t_q. Intuitively, this distance is minimized for $(p_j, t_j) \in T[t_q]$ that can be reached quickly from q_i in $\mathcal{G}$, and for which the interval t_j significantly overlaps with t_q. By definition, $0 \leq d(q_i, T, t_q) \leq 1$. The similarity between trajectories T_q and T for time interval t, is therefore:

$$\sigma(T_q, T, t) = \frac{\sum_{(q,t_q)\in T_q[t]} |t_q \cap t_j| \times e^{-d(q,T,t_q)}}{\sum |t_q|}, \tag{3}$$

where . denotes the length of a time interval, and t_j is the time interval in T corresponding to the node that minimizes $d(q, T, t_q)$. By definition, $0 \leq \sigma(T_q, T, t) \leq 1$, with $\sigma(T_q, T_q, t) = 1$ for any time interval t, and $\sigma(T_q, T, t) = 0$ for any two trajectories T_q and T with no temporal overlap.

6 Trajectory Similarity Estimation and Probabilistic Prediction

In this section, we describe TRACE, a novel approach for Trajectory SimilaRity EstimAtion and ProbabilistiC PrEdiction. Specifically, given the history $T_q = ((q_1, [s_1, e_1]), \ldots, (q_N, [s_N, e_N])$ of an individual, and time interval $t = [s_{N-\gamma}, e_N]$, where γ is the number of prior services received in the past, as well as the set $\mathcal{T}$ of trajectories of other individuals, TRACE begins by calculating the effective length λ of T_q. Then, TRACE identifies the most similar trajectory $T_c \in \mathcal{T}$ to T_q within t, using graphs G_λ and $G_{\lambda+1}$. The rationale for this design choice is that the next node may either be a node already visited in the past (in which case the effective length of T_q will remain unchanged) or a new node (in which case the effective length of T_q will increase by 1). The project type $\hat{q}_{N+1}$ that q is expected to be assigned to next is therefore estimated to be the one that maximizes the transition probability from p_c, the last matching project type in trajectory T_c that maximizes $\sigma(T_q, T_c, t)$ over either G_λ or $G_{\lambda+1}$. Therefore, $\hat{q}_{N+1}$ is obtained by maximizing the following objective:

$$\mathbb{1}_{(q_N, p_i) \in E} \times \mathbb{1}_{(p_c, p_i) \in E} \times P(p_i \in \mathrm{NB}(q_N) \cap \mathrm{NB}(p_c) | p_c), \tag{4}$$

where $\mathbb{1}$ is the indicator function. To ensure the predicted node is reachable from q_N, p_i is constrained to be in the out–neighborhood of both p_c and q_N. If no such node can be found within T_c, the search over trajectories continues, identifying the next maximum similarity trajectory that satisfies Eq.(4). The search terminates when a trajectory is identified that satisfies this constraint, or if no further trajectories are left to be examined. Furthermore, only those trajectories temporally overlapping with t are considered.

6.1 Computational Complexity of TRACE

The worst–case computational complexity of TRACE follows. During training (offline), the unique effective lengths $\mathcal{F}$ are identified for the set of trajectories $\mathcal{T}_{train}$. For each $\lambda \in \mathcal{F}$, paths with all possible number of steps, M, and offsets, N, in each trajectory in $\mathcal{T}_{train}$ are computed, resulting in $O(|\mathcal{F}| \times |\mathcal{T}_{train}| \times M \times N)$ complexity. In our study, we find both $|\mathcal{F}|, M$, and N to be small ($\leq 4, 55$, and 55, respectively), which means that this step is linear to the size of the training set. The calculation of edge weights involves recording the number of occurrences of a particular path with a specific number of steps and offset $\forall e \in E$, leading to an $O(\mathcal{T}_{train} \times M \times N \times |E|)$ complexity. The overall training (i.e., graph inference) complexity is therefore $\Theta(|\mathcal{T}_{train}| \times |E|)$.

Presented with a query trajectory T_q, TRACE first iterates through each trajectory in $\mathcal{H}$ to filter out trajectories with no temporal overlap with T_q. Since the length of each trajectory can be at most N, the complexity of this step is $O(|\mathcal{H}| \times N)$. The distance of all historical trajectories with the query is computed in $O(|\mathcal{H}| \times N \times \gamma)$, whereas similarity is computed in $\Theta(|\mathcal{H}| \times \gamma)$.

Algorithm 1 TRACE

Input: $\mathcal{H}, T_q, \gamma$

Output: Node $\hat{q}_{N+1}$ individual c is going to visit after q_N

1: Identify subset $\mathcal{H}_t \subseteq \mathcal{H}$ of trajectories temporally overlapping $t = (s_{N-\gamma}, e_N)$
2: Compute effective length, λ, of query trajectory T_q
3: **for each** $T_c \in \mathcal{H}_t$ **do**
4: $\quad \forall q_i \in T_q[t]$ compute distance to $T_c[t]$ using Eq.(2) on $\mathcal{G}_\lambda$ (similarly for $\mathcal{G}_{\lambda+1}$)
5: $\quad$ Compute similarity between $T_q[t]$ and T_c using Eq.(3)
6: $\quad T_{max} \leftarrow T_c$ with highest similarity score
7: **end for**
8: Using T_{max}, find $\hat{q}_{N+1}$ that maximizes Eq. (4) **return** $\hat{q}_{N+1}$

The prediction step requires constant time. Assuming that $|\mathcal{F}|, M, N$, and γ are all small compared to the size of the set of historical trajectories $\mathcal{H}$ used during testing, the overall prediction complexity is linear to $|\mathcal{H}|$.

7 Tiered Model For Better Performance

The experimental results we presented in [21] suggested that TRACE performs well in predicting both low and high frequency project types. At the same time, we noticed that some of the baselines are particularly good at predicting exit or high frequency (HF) project types. This observation prompted us to explore a tiered system, in which a prediction would be made whether an individual is more likely to exit or not, and in the latter case, TRACE would be used to predict the next service to be received by the individual.

Since, the number of combining the possible base models to construct a tiered system is exponentially large, verifying the performance of every possible combination is computationally prohibitive. We therefore propose a simple stacking ensemble, MetaTier, which combines the predictions of two candidate models: (*i*) with higher accuracy in predicting exit and HF project type, and (*ii*) with high recall in predicting project types. We denote the predictive model for exit and project types with $\mathcal{M}_{exit}$ and $\mathcal{M}_{int}$ respectively. Let $\{\mathcal{D}_X, \mathcal{D}_Y\} \in \mathcal{D}$ be the input and the output data of MetaTier. Each instance $\mathcal{D}_{X_i} \in \mathcal{D}_X$ is given by the set:

$$\{\mathcal{D}_{1,\mathcal{M}_{exit}}, \ldots, \mathcal{D}_{k,\mathcal{M}_{exit}}, \mathcal{D}_{1,\mathcal{M}_{int}}, \ldots, \mathcal{D}_{k,\mathcal{M}_{int}}\}, \tag{5}$$

where $\mathcal{D}_{a,b} \in \mathcal{D}_{X_i}$ encodes the a^{th} prediction of model b as a $|P|-$dimensional vector, $\mathcal{D}_{a,b}[j] = \{1$ if $j = p_i \in P$, and 0 otherwise$\}$. Furthermore, each instance $\mathcal{D}_{Y_i} \in \mathcal{D}_Y$ is the project type belonging to the set P. Finally, MetaTier is a logistic regression model that combines the predictions of $\mathcal{M}_{exit}$ and $\mathcal{M}_{int}$ as in Equation 5 and predicts whether an individual will exit or be assigned to a project type in the next step. Figure 3 shows an overview of MetaTier.

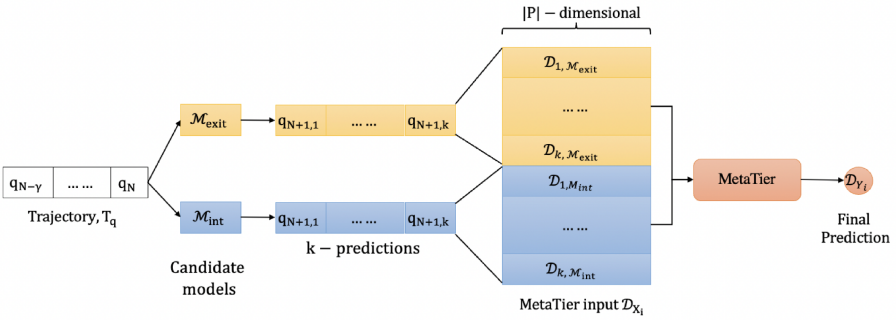


Fig. 3 Overview of MetaTier model. First, the query trajectory T_q is fed into the candidate models ($\mathcal{M}_{exit}$ and $\mathcal{M}_{int}$). Both of these candidate models output $k-$predictions in the form $[q_{N+1,1}, \ldots, q_{N+1,k}]$ which are then encoded and stacked in $\mathcal{D}_{X_i}$. MetaTier model takes $\mathcal{D}_{X_i}$ as input and outputs the final prediction $\mathcal{D}_{Y_i}$.

8 Experimental Evaluation

8.1 Data

Experiments are performed using an anonymized set of 50,469 records of all services provided by homeless services providers in the Capital Region of the state of New York to a total of 38,954 individuals over the time period of 2012 and 2018. The data was provided by CARES of NY. We focus on the 6,011 individuals that received services multiple times, out of which 3,475 exit to stable or fairly stable exits. Specifically, "stable" exits indicate individuals exiting to rental or owned housing without any housing subsidy, whereas "fairly stable exits" include permanent housing, rental or owned housing with housing subsidy, and long term facility [23].

We split the data into $\mathcal{T}_{train}$ and $\mathcal{T}_{test}$ as follows. Trajectories with entry date up to the end of 2016 are included in training set, $\mathcal{T}_{train}$, whereas trajectories with entry dates from the beginning of 2017 onward are included in test set, $\mathcal{T}_{test}$. This split ensures that predictive models are trained on data that does not contain future information, and therefore avoids data leakage. The test set $\mathcal{T}_{test}$ is further split into two disjoint sets, namely historical set, $\mathcal{H}$, and query set, Q. For each query in Q, the most similar trajectory in $\mathcal{H}$ is obtained using Eq.(3). Graphs $\mathcal{G}_\lambda$ and $\mathcal{G}_{\lambda+1}$ are computed over $\mathcal{T}_{train}$. Finally, $\mathcal{D}$ consists of the prediction for $\mathcal{T}_{test}$ which is then randomly split into training set, $\mathcal{D}_{train}$, and testing set, $\mathcal{D}_{test}$. MetaTier, is trained over $\mathcal{D}_{train}$ and tested on $\mathcal{D}_{test}$. We used 4-fold cross validation to avoid selection biases.

8.2 Metrics

We differentiate our analysis between individuals that exit the system (i.e., exit point) and those that transition to a new service (i.e., interim point) for a granular assessment of the predicting capabilities of TRACE and the baselines using the following metrics:

- **Precision@k**: measures how many times the predicted service is correct using the top k predictions.
- **Recall@k**: measures how many times each project type is identified correctly at k^{th} prediction.
- **Number of attempts**: measures the minimum number of similar trajectories that must be examined before a successful prediction.

To evaluate MetaTier, we compute the following metrics, based on all predictions (i.e., regardless of being exit or interim points):

- **Accuracy**: measures how many times the predicted service (or exit) is correct.
- **Class–specific Recall**: measures how many times each project type (or exit) is identified correctly. We additionally report mean recall for an overall comparison.

8.3 Baselines

We evaluate two variants of TRACE, where TRACE_1 uses only G_λ and TRACE_2 uses both G_λ and $G_{\lambda+1}$. We compare these TRACE models with the baselines described below. Given the most similar trajectory T_c to query T_q, $\hat{q}_{N+1}$ is predicted to be:

- $\mathbf{p_c - Next}$: p_{c+1} (i.e., the service following p_c in T_c).
- $\mathbf{p_c - NB}$: the highest transition probability out–neighbor of $p_c \in P$. Note that we found no difference in performance when requiring the out–neighbor of p_c to also be an out–neighbor of q_N ($\mathbf{p_c q_N - NB}$).
- $\mathbf{q_N - NB}$: the out–neighbor of $q_N \in V$ with the highest transition probability from q_N.
- **RN**: a random node $p \in P$. We consider two variants, namely selecting a node (**UR**) uniformly at random, and (**PA**) with a probability that is proportional to a node's in–degree (i.e., preferential attachment [25]).
- **Sim–attenuate**: the node identified using Eqs.(3) and (4), with the difference that an attenuation factor β^k, where $0 \leq k \leq K$ is the number of nodes in $T[t]$, is used in Eq.(3) to penalize intervals which are further in the past.
- **Log–Reg**: the project type predicted by a logistic regression model. Given a trajectory $T_c = (p_1, [s_1, e_1]), (p_2, [s_2, e_2]), \ldots, (p_N, [s_N, e_N]))$, we denote the input and the output of Log–Reg model as X and Y respectively. Every instance $x \in X$ is organized as a sequence of encoded project types given by $x = [x_{N-1}, ..., x_{N-1-\gamma}]$ where γ represents the window of past information fed to the model for prediction. Furthermore, each $x_i \in x$ is an $|P|-$dimensional vector defined as $x_i[j] = \{1 \text{ if } j = p_i, \text{ or } 0 \text{ otherwise}\}$. Moreover, for individuals with trajectories smaller than the window γ, x_i is padded with zeros indicating that no project type appeared in the trajectory at time instance i. On the other hand, $y \in Y$ records the scalar project type that is expected to be assigned at N. Finally, Log–Reg is trained with the train set, $\mathcal{T}_{train}$ and tested with query set $\mathcal{Q}$.

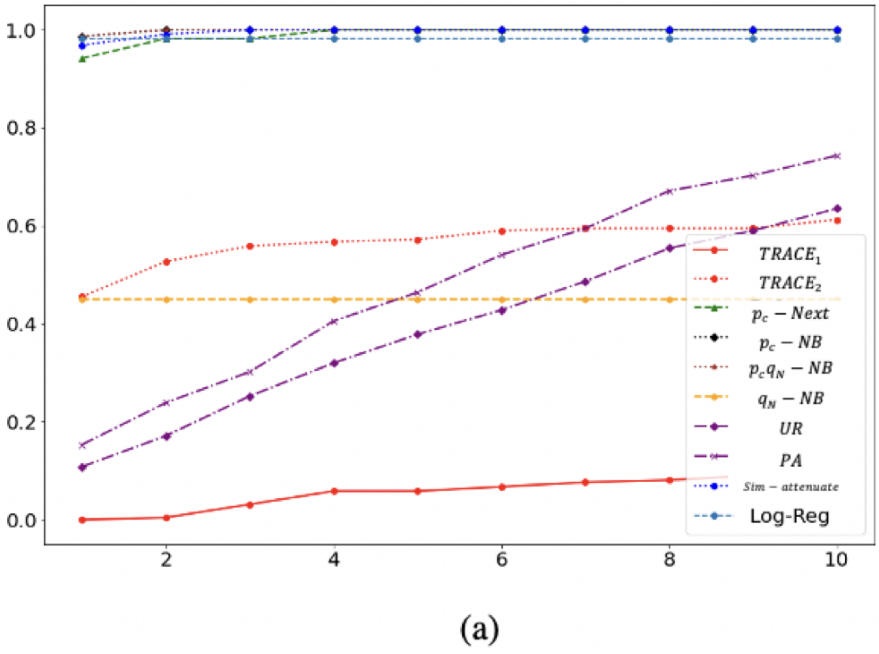
Finally, we set TRACE_2 as $\mathcal{M}_{int}$ in **MetaTier** because of its overall performance compared to TRACE_1, and consider $p_c - Next$, $p_c - NB$, $p_c q_N - NB$, $q_N - NB$, Sim–attenuate, and TRACE_1 as candidates for $\mathcal{M}_{exit}$. Note that both the baselines and TRACE are probabilistic in nature and can be used to make $k-$predictions each. Therefore, for a fair comparison, we design stacking ensemble models (**Meta model**) for each of the baselines and TRACE (i.e., base models), similar to MetaTier. The difference between Meta and MetaTier models lies in their inputs; while MetaTier model stacks the $k-$predictions from two predictive models ($\mathcal{M}_{int}$ and $\mathcal{M}_{exit}$) as input, Meta model stacks the $k-$predictions from one specific predictive model (either $\mathcal{M}_{exit}$ or $\mathcal{M}_{int}$) denoted by $\mathcal{M}$. Specifically, let $X_{\mathcal{M}}$ and $Y_{\mathcal{M}}$ be the input and output of the Meta model, where each instance $x_{i,\mathcal{M}} \in X_{\mathcal{M}}$ is given by the set $\{x_1, \ldots, x_k\}$ with $x_a \in x_{i,\mathcal{M}}$ encoding the a^{th} prediction of the model $\mathcal{M}$ as a $|P|-$dimensional vector, $x_a[j] = \{1 \text{ if } j = p_i \in P, \text{ and } 0 \text{ otherwise}\}$ and the output $y_{i,\mathcal{M}} \in Y_{\mathcal{M}}$ being the project type belonging to the set P. Finally, similar to MetaTier, the Meta model is also a logistic regression model that takes the predictions of a model $\mathcal{M}$ and predicts whether an individual will exit or be assigned to a project type in the next step. The MetaTier models and the Meta models are denoted by the prefixes **MetaTier**– and **Meta**– respectively.

9 Results and Analysis

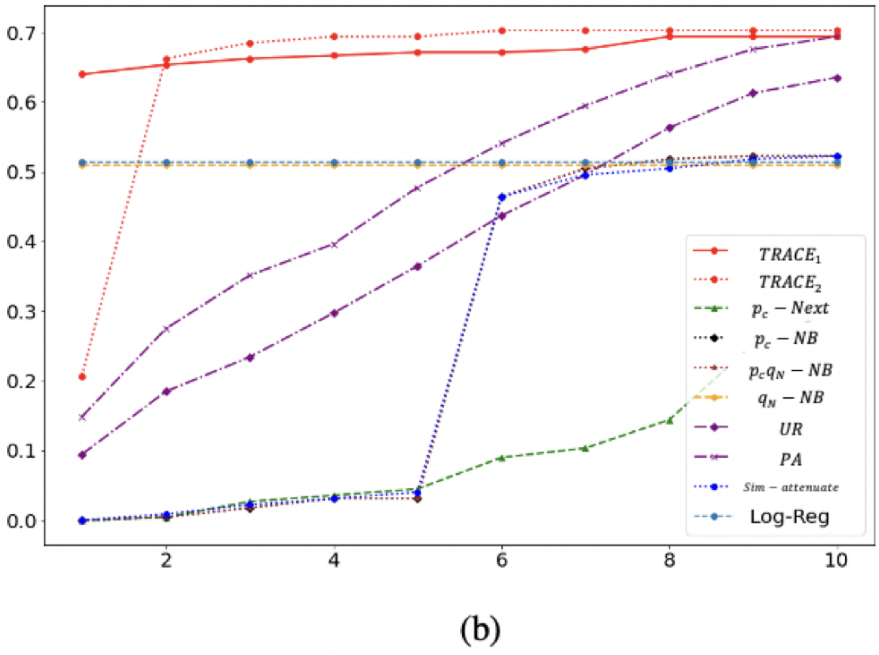
9.1 Result Analysis of TRACE

Fig. 4 shows precision@k. While, most methods perform well at predicting exit points, only TRACE_1 and TRACE_2 excel at predicting interim points. In fact, $q_N - NB$ consistently predicts the most frequent project type as the next transition (see Fig. 6) and is therefore meaningless. Similarly, both $p_c - NB$ and Sim–attenuate predict project type 1 with high probability (80%) resulting in good performance with respect to exit prediction, but meaningless prediction of interim points. Interestingly, the performance of $p_c - NB$ and Sim–attenuate improves dramatically for $k \geq 5$. However, Fig. 5 suggests that this is an artifact of their high recall for project type 1, which comprises 50% of the ground truth. $p_c - Next$ focuses on the HF project types (e.g., 1, 11, 13). Finally, TRACE_1 performs poorly for exit points mainly due to its low recall. Instead, TRACE_2 seems to predict exit at first (i.e., $k = 1$), explaining its reduced performance for interim points. However, TRACE_2's recall improves dramatically with k increasing, as shown in Fig. 5. Moreover, we compare TRACE_2 with Log–Reg, Log–Reg performs well with exit points, however the performance drops with interim points (Figure 4 and 5). Although Log–Reg is able to predict most of the project types unlike most of the other baselines, the recall for each project type is quite low in comparison to TRACE models.

Given enough opportunities, prediction models may eventually be able to "get a prediction right". We therefore additionally report the distribution of number of attempts required for a successful prediction in Fig. 7. Evidently,



(a)



(b)

Fig. 4 Precision@k plot for TRACE and baselines predicting (a) exit points and (b) interim points (Precision at x–axis and k at y–axis).

$p_c - NB$, $p_c q_N - NB$, $q_N - NB$, Sim–attenuate require few attempts to predict exit points due to their bias towards such nodes, as explained above. In contrast, TRACE is better at predicting interim points, although occasionally requiring more attempts before being able to correctly predict cases that significantly deviate from the rest of the trajectories in the historical data. Fig. 8 shows that TRACE's incorrect predictions are mostly for extremely short trajectories, which lack enough historical data, indicating the significance of the contextual information provided by the underlying and unobserved network.

Furthermore, we analyze Sim–attenuate to better understand why it converges to $p_c q_N - NB$ as β approaches 1. Fig. 9 shows that penalizing the intervals further in past diminishes the model's ability to predict exit points. Intuitively, this indicates the necessity of past information to determine the most similar trajectories to a query.

Next, we experiment with different values of α to determine its effect on TRACE's recall, as shown in Fig. 10. When $\alpha = 0.5$, the TRACE_1's exit performance improves dramatically, whereas $\alpha = 0.8$ only slightly improves performance for interim points. On the other hand, $\alpha = 0.2$ achieves the best results for TRACE_2 for both exit and interim points, but the improvement over $\alpha = 0.5$ is not significant. We therefore report results for $\alpha = 0.5$ for both TRACE_1 and TRACE_2 when comparing their performance against the baselines.

We further explore the effect of hyperparameter γ, and find that its value becomes critical for lengthy trajectories, which require more past information to improve the chances of accurately predicting the next interim point. To demonstrate this, we focus on queries of length greater than 4. Fig. 11(b) implies that the higher the value of γ, the higher the precision. In our comparisons, our results are therefore for $\gamma = 6$.

Finally, we investigate how the time–frame of $\mathcal{T}_{train}$ affects TRACE_2. We observe that the performance of TRACE_2 decreases with the size of $\mathcal{T}_{train}$, as shown in Figure 12, indicating that increased historical information better equips TRACE_2 to predict the next transition (or exit). Furthermore, Figure 13 shows that the tendency of TRACE_2 to predict exit at first (i.e., k = 1) decreases for year 2015 and 2016; this provides an explanation for the increase in precision at $k = 1$ for interim points during those years. On the other hand, a training dataset, $\mathcal{T}_{train}$, spanning a longer time period (2012 – 2016) in comparison to $\mathcal{T}_{test}$ (2017 – 2018) does not influence the performance of TRACE. This finding can be interpreted as a result of the fact that the length of trajectories lies within 1 and 20 for both $\mathcal{T}_{train}$ and $\mathcal{T}_{test}$, despite the longer time span of $\mathcal{T}_{train}$, as shown in Figure 14. Even though few longer trajectories can be observed in $\mathcal{T}_{train}$, their frequency is so low it does not have a significant affect on the performance of TRACE.

9.2 Result Analysis of MetaTier

Figure 15 captures the accuracy of MetaTier model for different choices of $\mathcal{M}_{exit}$ and their respective Meta model across different values of k. It is

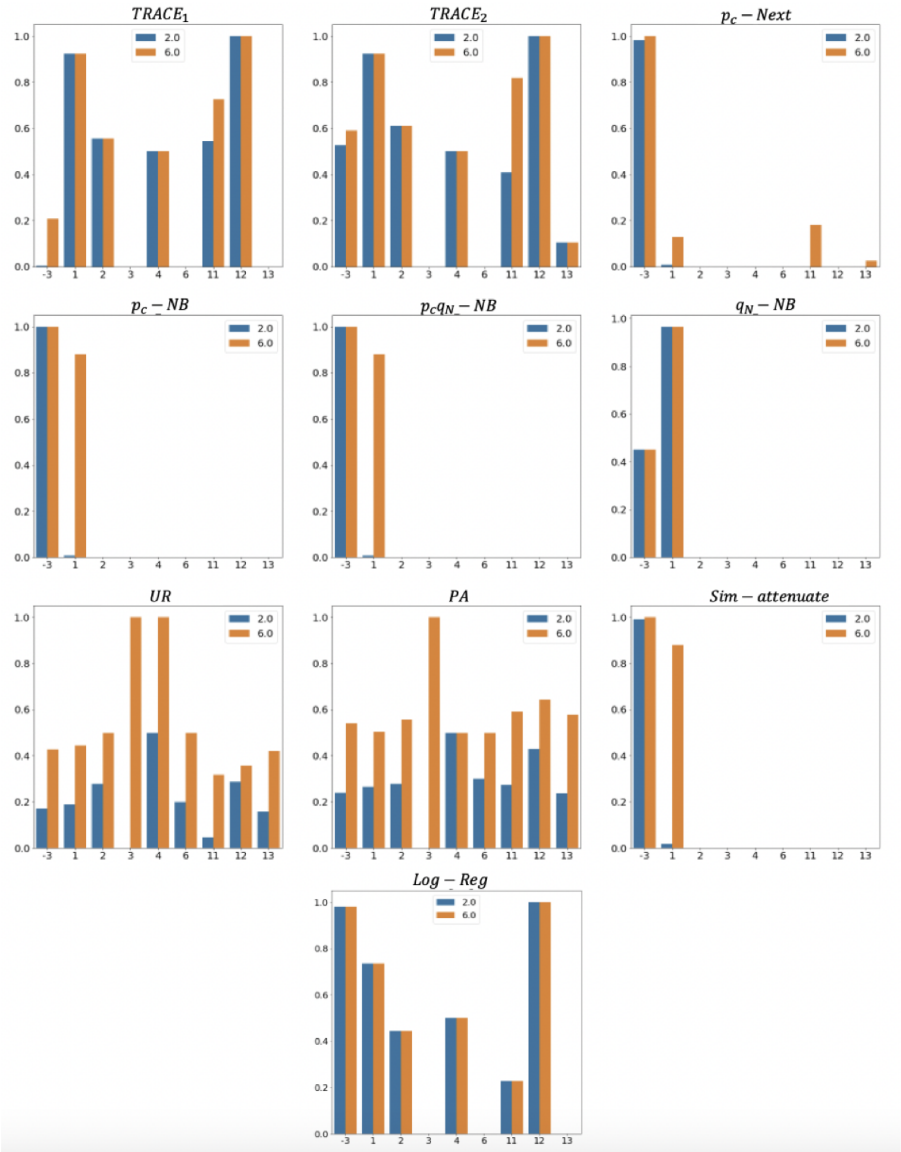


Fig. 5 Recall (y–axis) achieved by TRACE and the baselines at $k = 2$ (blue) and $k = 6$ (orange) for each project type (x–axis) separately.

clear that MetaTier models perform significantly better than their respective Meta models. Moreover, Figure 16 indicates that, unlike the Meta models, MetaTier models performs well at predicting both HF and low frequency (LF) project types. This shows that stacking the two models (i.e., meta model and $TRACE_2$) takes advantage of their strengths, thus enhancing the overall predicting capability. On the other hand, the accuracy of Meta and MetaTier

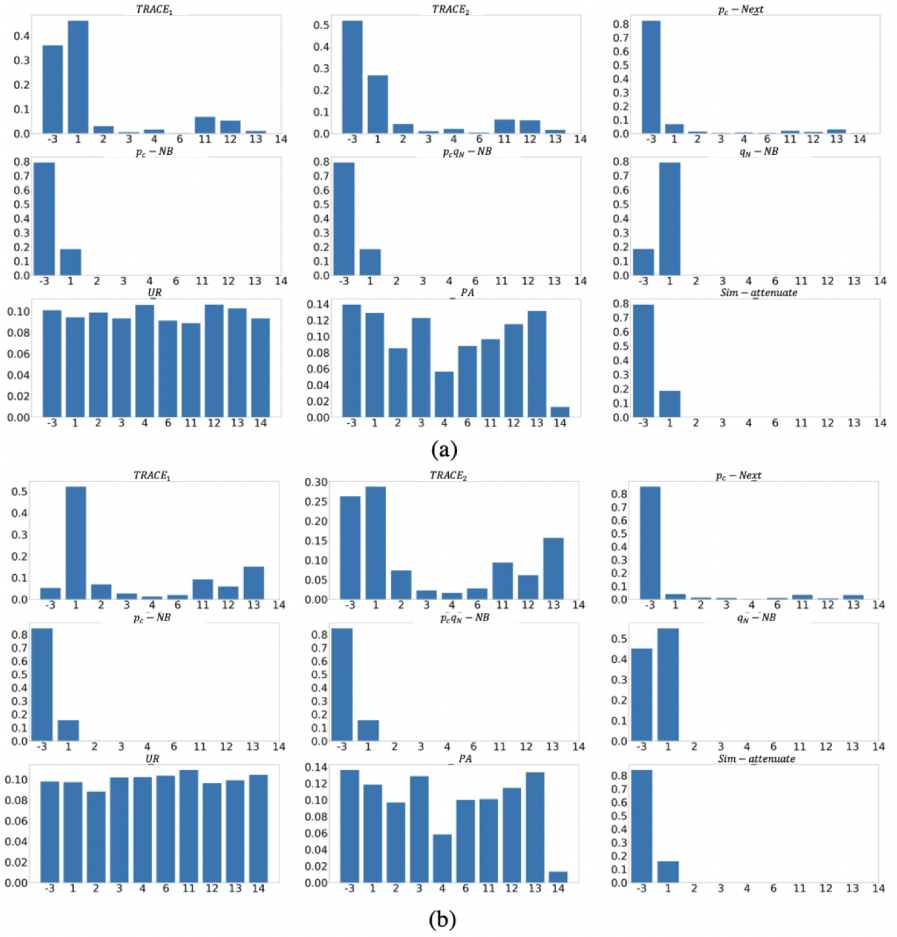


Fig. 6 Aggregate prediction distribution for (a) exit points and (b) interim points (Frequency at y–axis and project type at x–axis).

models for TRACE_1 shows quite the opposite effect in comparison to the rest of the models. This may be due to the similar capability of TRACE_1 and TRACE_2 (Figure 4). It would make sense to consider TRACE_1 as an alternative $\mathcal{M}_{int}$ for MetaTier model, however, while TRACE_1 and TRACE_2 have similar performances in predicting interim points, TRACE_2 performs notably better at predicting exit than TRACE_1. Therefore, considering the overall performance of the two variants of TRACE, we chose TRACE_2 as a better choice for $\mathcal{M}_{int}$.

For MetaTier models, k is a hyperparameter that determines the number of predictions fed to the MetaTier model, making it important to select a specific value of k. We begin by selecting the top–5 highest accuracy and mean recall values and their corresponding k values for all the MetaTier models. The idea was to find the values of k which are simultaneously present in the top–5

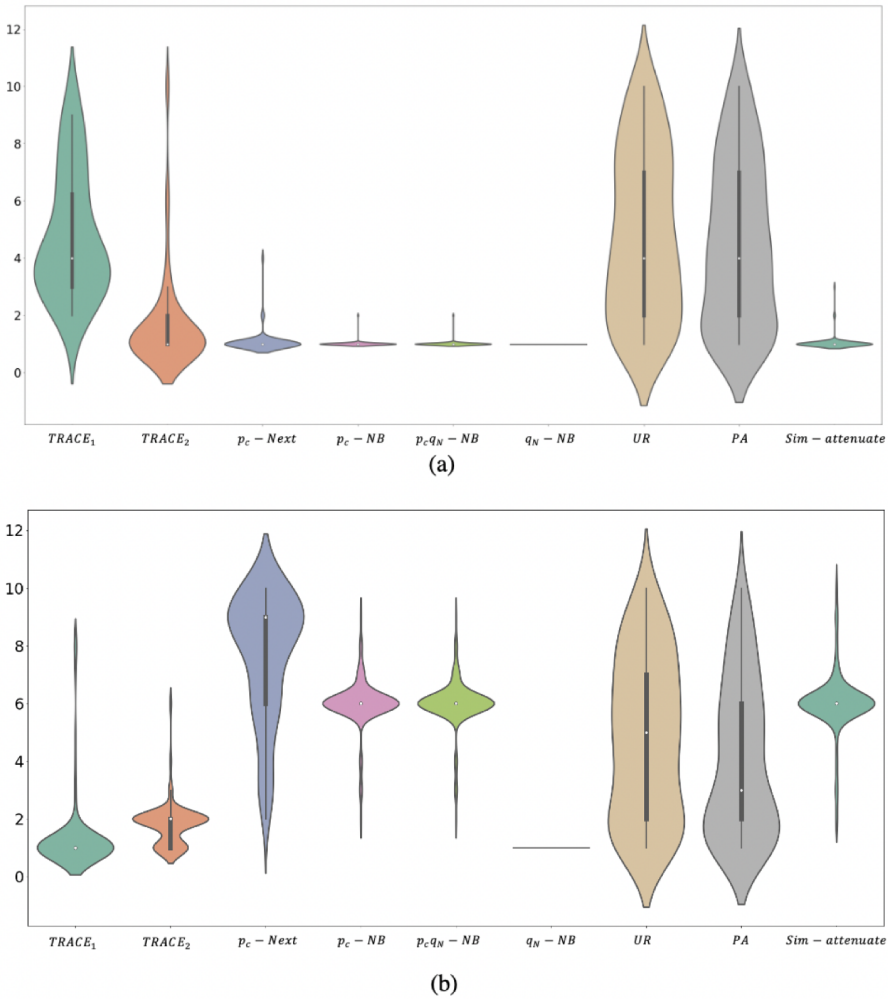


Fig. 7 Distribution of the minimum number of attempts needed to make a successful prediction of (a) exit point and (b) interim points.

MetaTier models, in this case, $k = 8$ and $k = 9$. Table 2 shows that the loss in accuracy and mean recall is higher for $k = 9$ than $k = 8$, thus we select k to be 9 for our further analyses.

Next, we analyze how MetaTier performs in comparison to TRACE_2. MetaTier stems from the intent to exploit the ability of $\mathcal{M}_{exit}$ to predict exit and HF project type, and concurrently the ability of TRACE_2 to predict interim points. Table 3 shows that MetaTier models have better performance than TRACE_2 with higher mean recall and accuracy than TRACE_2. However, the main advantage of MetaTier models over TRACE_2 lies in predicting exit and HF project types (e.g., project type 1, which stands for emergency

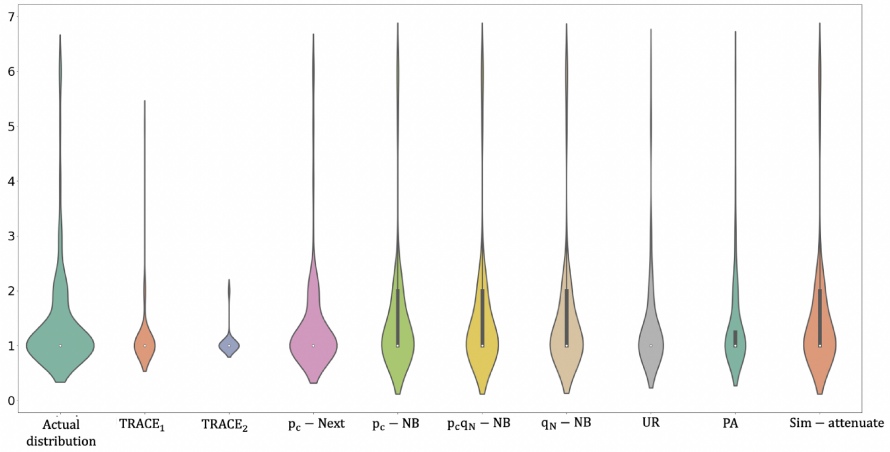


Fig. 8 Distribution of length of trajectories with incorrect predictions in comparison to the actual distribution.

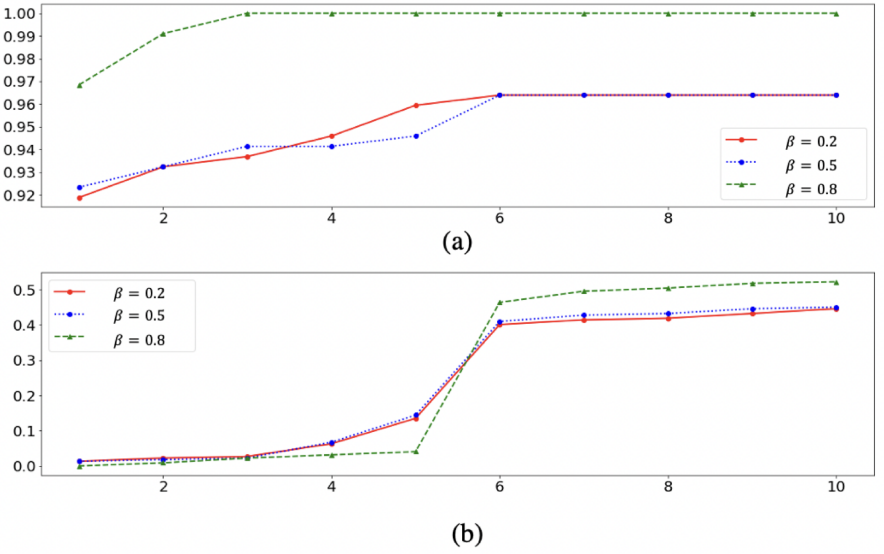


Fig. 9 Precision@k of Sim–attenuate with varying β for (a) exit points and (b) interim points.

shelters). Table 4 shows that majority of the meta models with higher mean recall than TRACE$_2$ in predicting HF project types and exit results in their corresponding MetaTier models to have better mean recall in predicting HF project types and exit than TRACE$_2$.

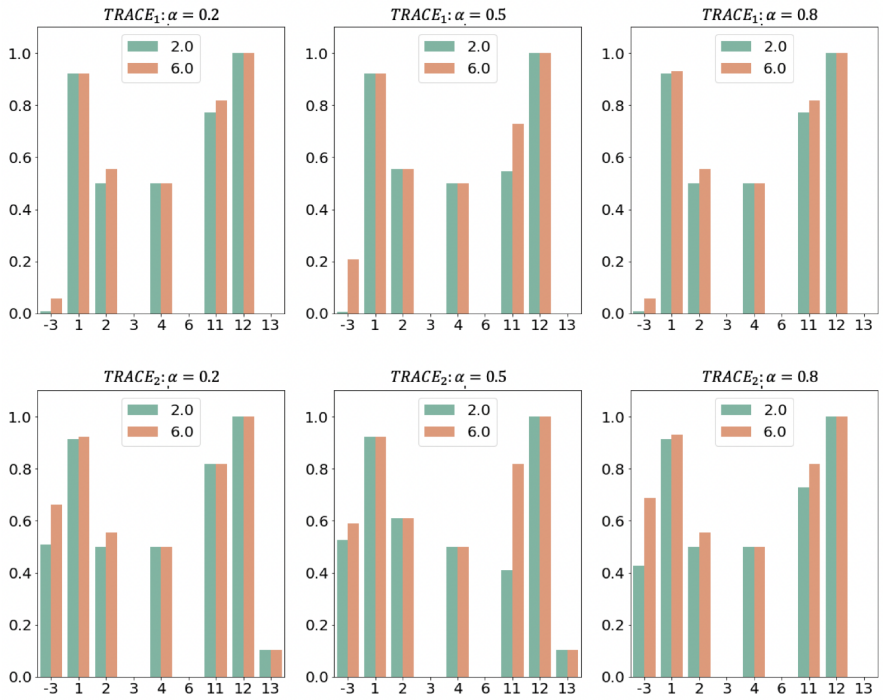


Fig. 10 Recall@k ($y-$axis) of TRACE with varying α.

Table 2 Accuracy (ACC) and Mean Recall (mR) at $k = 8$ and $k = 9$

Model	ACC@8/ACC@9	mR@8/mR@9
MetaTier–TRACE_1	0.53/0.56	0.26/0.29
MetaTier–$p_c - NB$	0.61/0.60	0.29/0.30
MetaTier–$p_c - Next$	0.53/0.52	0.25/0.27
MetaTier–$p_c - NB$	0.61/0.60	0.29/0.30
MetaTier–$q_N - NB$	0.57/0.59	0.29/0.30
MetaTier–Sim–attenuate	0.59/0.60	0.30/0.31

10 Conclusion

In this study, we proposed an approach that begins by inferring the network of services that the homeless visit as they strive to secure permanent housing. We subsequently defined a score to assess the similarity of their trajectories. Based on these two contributions we proposed a method to predict the most likely service an individual will be assigned to next given her history. Our experimental evaluation showed the ability of the proposed approach to better match the observed sequences as opposed to baselines. Our approach can be used as a building block for more complex applications, such as recommending service assignment. However, in replicating the observed data, biases in the service

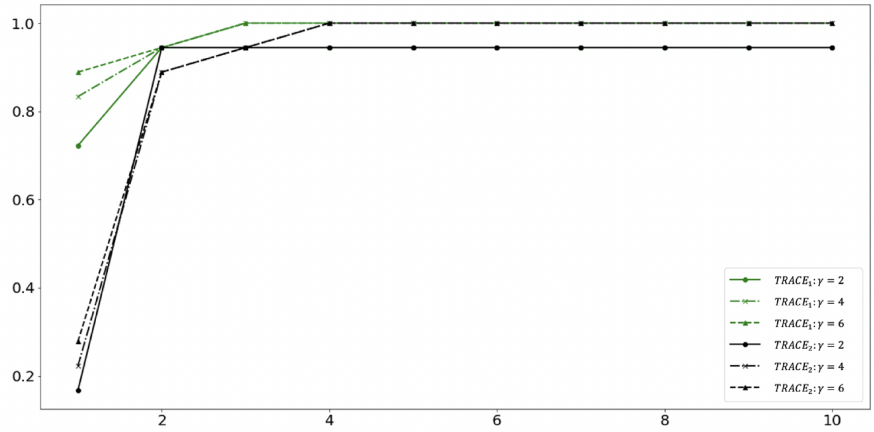


Fig. 11 Precision@k for interim points of $TRACE_1$ and $TRACE_2$ with varying $\gamma = \{2, 4, 6\}$.

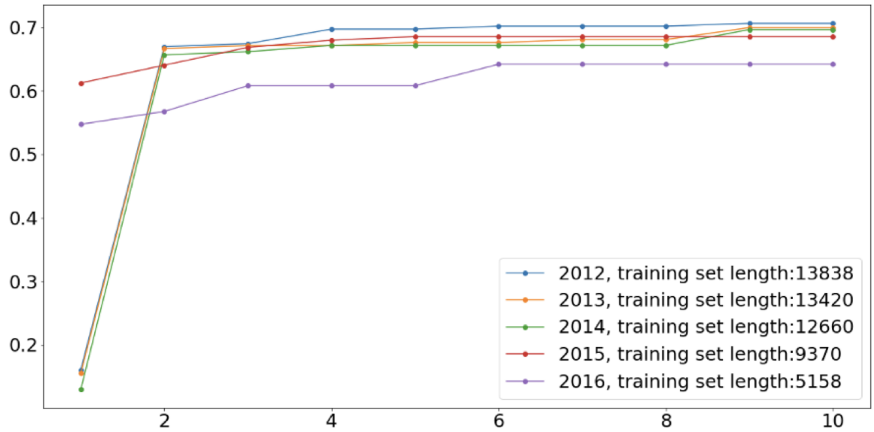


Fig. 12 Precision@k versus k for different starting year

assignment process may be replicated. In order to avoid this and to additionally evaluate the potential ability of "wrong" predictions to lead to better outcomes, we will explore counterfactual predictions in future work. We additionally plan to address limitations, such as accounting for imbalances among project types. Another potential line of research concerns the explainability and/or interpretability of our approach, given the potential human impact of its application in this high–stakes domain. Although our approach can indeed provide explanation (i.e. reason) for its prediction based on the identified next step in the most similar trajectory and in future, it was not explicitly designed to state the reasons behind its predictions. We therefore plan to rigorously

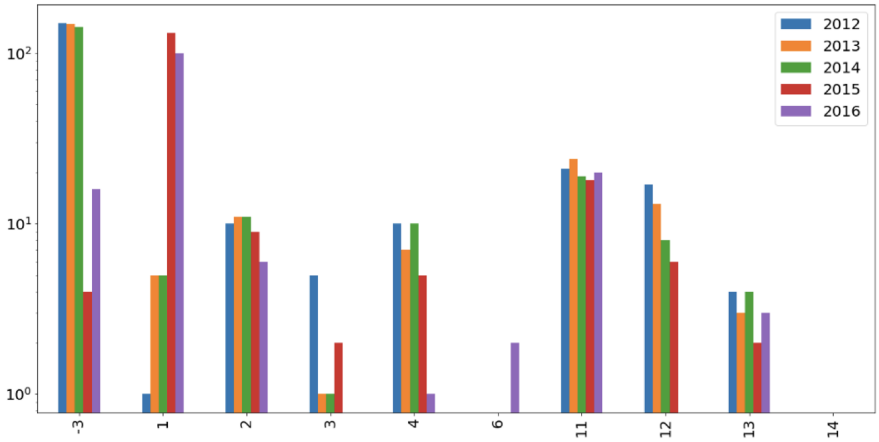


Fig. 13 Frequency of predicted project types for different starting year at $k = 1$

Table 3 Accuracy (ACC) and Mean Recall (mR) at $k = 9$

Model	mR@9	ACC@9
$TRACE_2$	0.25	0.56
MetaTier–$TRACE_1$	0.28	0.56
MetaTier–$p_c - NB$	0.28	0.60
MetaTier–$p_c - Next$	0.25	0.52
MetaTier–$p_c - NB$	0.28	0.60
MetaTier–$q_N - NB$	0.28	0.59
MetaTier–Sim–attenuate	0.29	0.60

Table 4 Mean Recall (mR) of meta and MetaTier models at $k = 9$ for exit and HF project types

Model	Meta	MetaTier
$TRACE_2$	0.66	–
MetaTier–$TRACE_1$	0.62	0.61
MetaTier–$p_c - NB$	0.72	0.70
MetaTier–$p_c - Next$	0.47	0.56
MetaTier–$p_c - NB$	0.72	0.70
MetaTier–$q_N - NB$	0.50	0.69
MetaTier–Sim–attenuate	0.73	0.70

evaluate this quality of TRACE. Finally, we plan to further improve the computation of similarity between project types by better incorporating domain semantics, such as for example that an emergency shelter is more similar to a day shelter as compared to transitional housing).

Declarations

Funding. Funding was provided by the National Science Foundation under Grant No. ECCS-1737443.

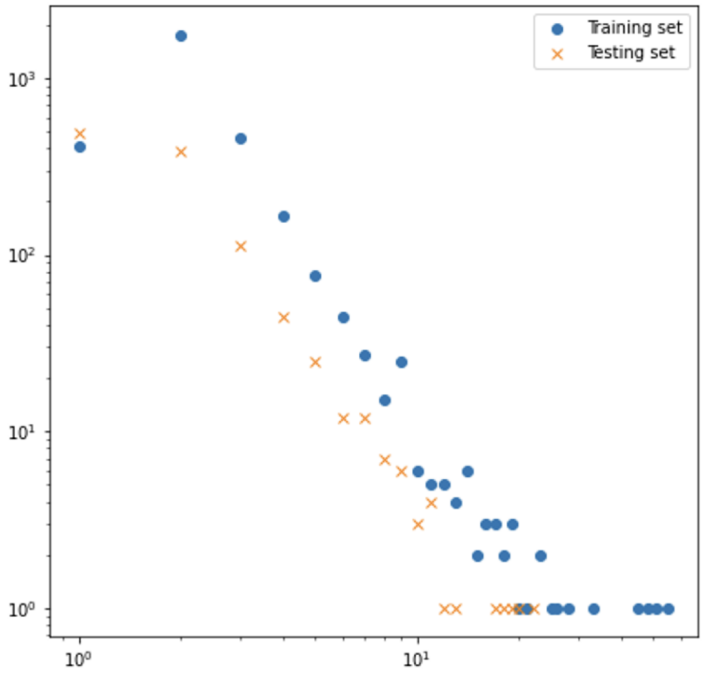


Fig. 14 Distribution of trajectory lengths for $\mathcal{T}_{train}$ and $\mathcal{T}_{test}$

Conflict of interest. The authors declare that there is no conflict of interest.

Ethics approval. This study was conducted in accordance to all relevant policies and procedures set forth by the University at Albany Institutional Review Board for the protection of human subjects.

Availability of data and materials. The dataset that support the findings of this study is not publicly available, as it has been provided to the authors under a data sharing agreement with CARES of NY, Inc. Information on how to obtain the dataset and reproduce the analysis is available from the corresponding author on request.

Code Availability. To ensure the reproducibility of the work, the source code is available at https://github.com/IDIASLab/TRACE.

Authors' contribution. Conceptualization, KSR and CC; Methodology, KSR and CC; Analysis, KSR; Writing-Editing, KSR and CC; Data collection, CC; Supervision, CC. All authors have read and agreed to this version of the manuscript.

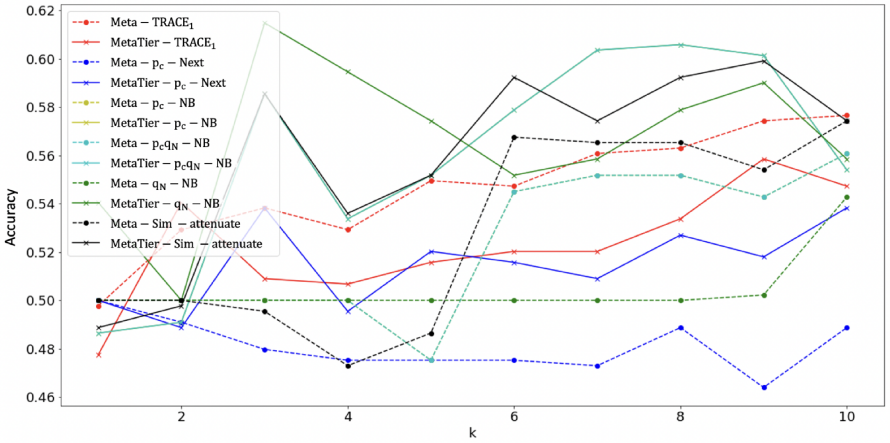


Fig. 15 Accuracy of meta and MetaTier models with varying k

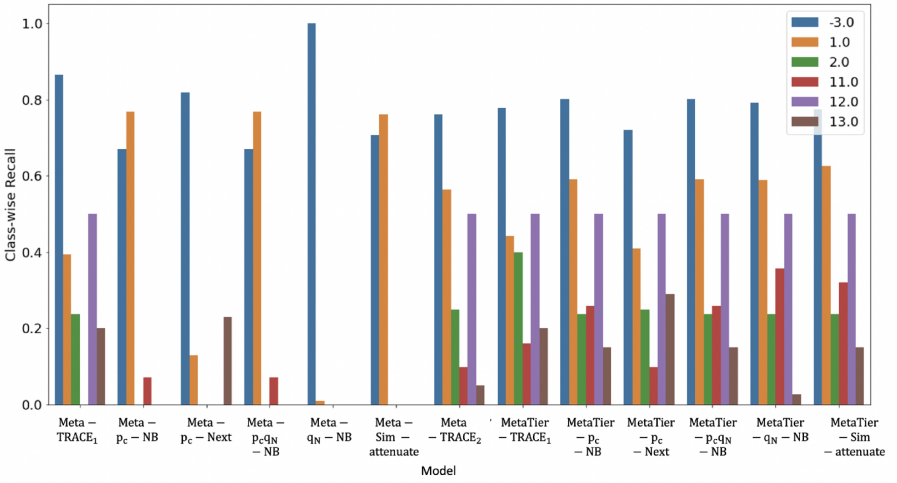


Fig. 16 Recall at $k = 9$ for meta and MetaTier models

References

[1] Sullivan, A.A.: What does it mean to be homeless? how definitions affect homelessness policy. Urban Affairs Review, 10780874221095185 (2022)

[2] Fleury, M.-J., Grenier, G., Sabetti, J., Bertrand, K., Clément, M., Brochu, S.: Met and unmet needs of homeless individuals at different stages of housing reintegration: A mixed-method investigation. PloS one **16**(1), 0245088 (2021)

[3] Henry, M., de Sousa, T., Roddey, C., Gayen, S., Joe Bednar, T., Associates, A.: The 2022 Annual Homeless Assessment Report (AHAR) to Congress. Part 1: Point-in-time estimates of homelessness. The US Department of Housing and Urban Development (2022)

[4] Gao, Y., Das, S., Fowler, P.: Homelessness service provision: a data science perspective. In: Workshops at the Thirty-First AAAI Conference on Artificial Intelligence (2017)

[5] Hong, B., Malik, A., Lundquist, J., Bellach, I., Kontokosta, C.E.: Applications of machine learning methods to predict readmission and length-of-stay for homeless families: The case of win shelters in new york city. Journal of Technology in Human Services **36**(1), 89–104 (2018)

[6] Kube, A., Das, S., Fowler, P.J.: Allocating interventions based on predicted outcomes: A case study on homelessness services. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 622–629 (2019)

[7] VanBerlo, B., Ross, M.A., Rivard, J., Booker, R.: Interpretable machine learning approaches to prediction of chronic homelessness. Engineering Applications of Artificial Intelligence **102**, 104243 (2021)

[8] Messier, G., John, C., Malik, A.: Predicting chronic homelessness: The importance of comparing algorithms using client histories. Journal of Technology in Human Services, 1–12 (2021)

[9] Chelmis, C., Qi, W., Lee, W., Duncan, S.: Smart homelessness service provision with machine learning. Procedia Computer Science **185**, 9–18 (2021)

[10] Wang, H., Su, H., Zheng, K., Sadiq, S., Zhou, X.: An effectiveness study on trajectory similarity measures. In: Proceedings of the Twenty-Fourth Australasian Database Conference-Volume 137, pp. 13–22 (2013)

[11] Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast subsequence matching in time-series databases. Acm Sigmod Record **23**(2), 419–429 (1994)

[12] Keogh, E.J., Pazzani, M.J.: Derivative dynamic time warping. In: Proceedings of the 2001 SIAM International Conference on Data Mining, pp. 1–11 (2001). SIAM

[13] Ye, L., Keogh, E.: Time series shapelets: a new primitive for data mining. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 947–956 (2009)

[14] Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. The Journal of Machine Learning Research **11**, 2837–2854 (2010)

[15] Abbott, A., Tsay, A.: Sequence analysis and optimal matching methods in sociology: Review and prospect. Sociological methods & research **29**(1), 3–33 (2000)

[16] Elzinga, C.H.: Sequence analysis: Metric representations of categorical time series. Sociological methods and research (2006)

[17] Brugere, I., Gallagher, B., Berger-Wolf, T.Y.: Network structure inference, a survey: Motivations, methods, and applications. ACM Computing Surveys (CSUR) **51**(2), 1–39 (2018)

[18] Gomez-Rodriguez, M., Leskovec, J., Krause, A.: Inferring networks of diffusion and influence. ACM Transactions on Knowledge Discovery from Data (TKDD) **5**(4), 1–37 (2012)

[19] Du, N., Song, L., Yuan, M., Smola, A.: Learning networks of heterogeneous influence. Advances in neural information processing systems **25** (2012)

[20] Heckerman, D., Geiger, D., Chickering, D.M.: Learning bayesian networks: The combination of knowledge and statistical data. Machine learning **20**(3), 197–243 (1995)

[21] Rahman, K.S., Chelmis, C.: Learning to predict transitions within the homelessness system from network trajectories. In: Proceedings of the 2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 181–189 (2022)

[22] United States Department of Housing and Urban Development: HMIS Data Standards Manual. Retrieved May 26, 2021, from https://www.hudexchange.info/resource/3824/hmis-data-dictionary/ (2020)

[23] Chelmis, C., Rahman, K.S.: Peeking through the homelessness system with a network science lens. In: Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 69–73 (2021)

[24] Holme, P., Saramäki, J.: Temporal networks. Physics reports **519**(3), 97–125 (2012)

[25] Albert, R., Barabási, A.-L.: Statistical mechanics of complex networks. Reviews of modern physics **74**(1), 47 (2002)