# Capacity-Aware Fair POI Recommendation Combining Transformer Neural Networks and Resource Allocation Policy

Sajal Halder[a], Kwan Hui Lim[b], Jeffrey Chan[a], Xiuzhen Zhang[a]

[a]*School of Computing Technologies, RMIT University, Australia*
[b]*Singapore University of Technology and Design, Singapore*

**Abstract**

Point of Interest (POI) recommendations have primarily focused on maximizing user satisfaction, while neglecting the needs of POIs and their operators. One such need is recommendation exposure, which can lead to envy among the POIs. Some POIs may be under-recommended, while others may be over-recommended, resulting in dissatisfaction for both staff and users due to long queues or overcrowding. Existing work has not addressed the trade-off between satisfying user preferences and being fair to POIs, which typically aim to operate at capacity. Therefore, we introduce the POI fair allocation problem to model this issue, taking into account both user satisfaction and POI exposure fairness. To address this problem, we propose a fair POI allocation technique that balances user satisfaction and POI capacity-based exposure simultaneously. Our proposed model utilizes existing (transformer neural networks and attention LSTM model) personalized POI recommendation models that capture users' spatio-temporal influences and interests in POI visits. We then propose POI capacity-based allocation using the over-demand cut policy and under-demand add policy, which ensures POI exposure ratio and envy-freeness up to certain thresholds. We evaluate the performance of our proposed model on five datasets containing real-life POI visits. Experimental evaluations show that our proposed model outperforms baselines in terms of user and POI-based evaluation metrics. To ensure reproducibility, we have publicly shared our source code at Codeocean.

*Email addresses:* `sajal.halder@student.rmit.edu.au` (Sajal Halder), `kwanhui_lim@sutd.edu.sg` (Kwan Hui Lim), `jeffrey.chan@rmit.edu.au` (Jeffrey Chan), `xiuzhen.zhang@rmit.edu.au` (Xiuzhen Zhang)

*Keywords:* Point of Interest, Fair Recommendation, User Satisfaction, POI Allocation, Over and Under Capacity.

## 1. Introduction

Point of Interest (POI) recommendation is of great importance due to its numerous applications in resource allocation, tourism, trip recommendation and event management [1]. Diversity and item exposure in recommender systems attract research attention due to several motivations. Firstly, a good recommendation system should not only provide personalized recommendations but also offer a diverse set of items that cater to users' interests and preferences. This ensures that users are presented with a range of enjoyable options, sweetening their overall experience. Secondly, it is essential for the recommendation system to maintain the long-term sustainability of providers. This involves mitigating the "rich get richer" phenomenon, where a few popular items dominate the recommendations, potentially excluding lesser-known but equally valuable options. Balancing the exposure of users' distribution is essential for fairness and promoting a vibrant ecosystem of POIs. Moreover, POI recommendation faces additional challenges related to allocation diversity and capacity limitations. Unlike traditional item recommendation scenarios, POIs are not infinitely available. There are restrictions on the number of people that can visit a particular attraction or participate in an event. Thus, recommendations must consider the capacity limitations of POIs to avoid suggesting options that are already at full capacity, ensuring a seamless user experience. To the best of our knowledge, this paper is the first to study these issues in the context of POI recommendations. By exploring capacity-aware fair POI recommendation, we aim to develop novel techniques that tackle the challenges of diversity, item exposure, and capacity constraints, thereby advancing the field and contributing to more effective and fair recommendation systems.

POI recommendation is challenging due to the need to satisfy two competing aspects of the problem, which are to maximise user personalised interests and viable POI visiter numbers concurrently. However, most studies are focused on maximising user personalised interest [2, 3, 4] and neglect POIs exposure. These prior studies focus on the personalised recommendation aspect and only consider user satisfaction to recommend top-k POIs that achieve high customer utility. The POI exposure aspect is neglected, which may result in a large number of users being recommended the same POI. In turn, that may create a long queue due to capacity limits and users may become frustrated. Our observation of real–world dataset shows a huge disparity in the exposure of the POIs, which is unfair to the POIs. Because it increases the risk of presenting a biased or homogeneous set of recommendations, which can lead to a narrow view of the available options to the users. Furthermore, low user demands will cause POIs to face exposure problems and be unable to operate for the long term, causing users to lose the opportunity to enjoy diverse POIs. These challenges motivate us to build a fair POI recommendation that users get satisfaction and POIs get a reasonable number of visitors in a real-world environment.

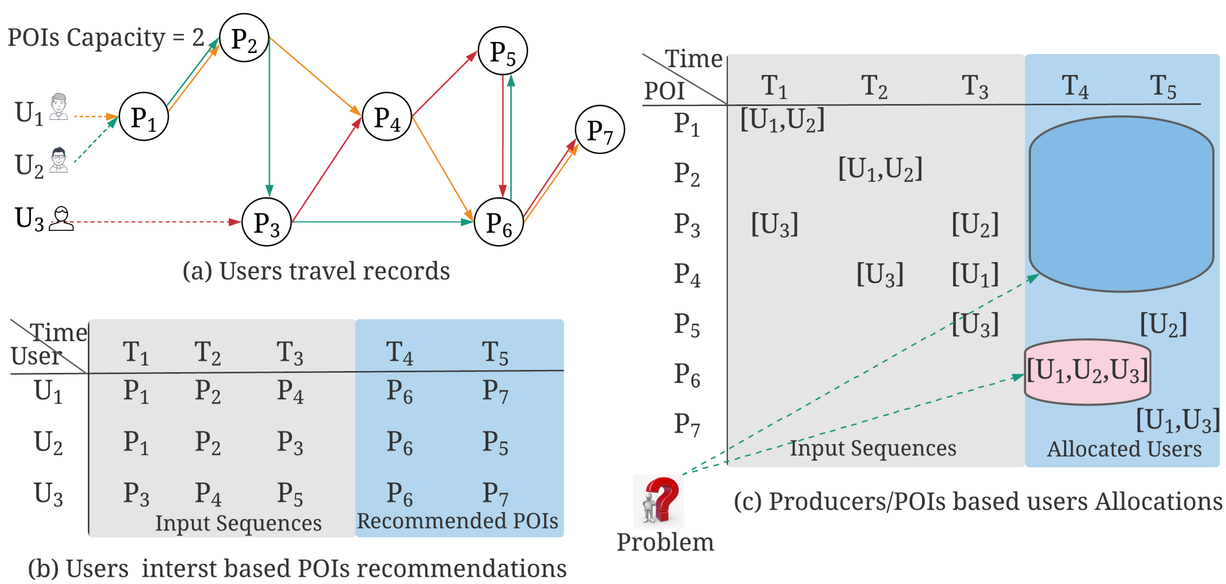


Figure 1: Necessity of fair POI recommendation.

Figure 1 shows the necessity of fair POIs recommendation. Users $U_1$, $U_2$ and $U_3$ visits POIs $P_1$ to $P_7$ at different time stamps $T_1$ to $T_5$ in Figure 1(a). The first three timestamps $T_1$ to $T_3$ are used as input sequence and the next two time stamps POIs are recommended, illustrated in Figure 1(b). Existing models focus on user personalised interest and recommend POI $P_6$ to all users at time stamps $T_4$. This recommendation creates two problems: (i) POI

$P_6$ has a capacity limit of 2, which does not allow all users to visit at a time, and (ii) some POIs struggle to find users in Figure 1(c). Thus, one model is necessary which can consider user interest and simultaneously consider POIs visit levels. Only user preference-based POI recommendations may create biased POI allocation.

To recommend POIs, different deep neural models have been proposed that consider spatio-temporal information and user preferences [2, 3, 4, 5]. Recently, *FairRec* [6] and FairRecPlus [7] focused on the two-sided satisfaction of both users and e-commerce item recommendation in which the number of recommended items is envy-free up to one item (EF1). EF1 goal is to divide a set of resources among multiple individuals in a way that no individual envies another's share, but only up to one item. We can not apply existing EF1 allocation in POI recommendations because POIs have different capacity limits. Thus, our definition of envy-free depends on POI demand ratio with capacity, in which higher capacity-based providers should get more visitors than the lower capacity-based providers. Our proposed capacity-based envy-free allocation has a few advantages. First, providers get visit levels based on invested assets that are more realistic than all providers getting the same visit level. If all providers get the same number of visitors, the providers of higher amount investing could not get enough visitors, whereas providers of small amount investing will get many visitors considering their capital. In that case, the high amount of investing providers group will suffer because they could not receive enough consumers based on their expectations. Second, the capacity ratio-based envy-free model reduces higher and lower amounts of investing group envy because each group will get a number of users based on the number of their investing resources. Third, capacity ratio–based allocation prevents over–demand problems in which a few popular items may concentrate all users' interest.

Capacity-based envy-free allocation is a resource allocation method that takes into account the capacities or capacities of individuals to receive or utilize resources. This method differs from traditional envy-free allocation, where resources are simply divided equally among individuals, regardless of their capacities. In capacity-based envy-free allocation, individuals with higher capacities receive more resources, while those with lower capacities receive fewer resources, but still in proportion to their capacities. General room allocation problems [8, 9] solve over–demand problems based on capacity diversity and budget constraints. Therefore, we can not apply the existing model because it is an NP-hard problem and solve a maximum four beds capacity-based room sharing problem. The existing room allocation model's computational complexity increases exponentially with the room capacity. Besides this, POIs visit levels are more diverse than the room-sharing problems. *FairRec* [6] and *FairRecPlus* [7] proposed fair items recommendation that each item gets envy-free allocation up to one item. *FairRec* model allocates items equally, which does not perform well in the POI recommendation because POIs have different capacities. Therefore, in POI recommendation, if we use constant capacity proportion–based allocation, we will lose user preferences. Moreover, if we apply greedy approach to distribute the users among the POIs, POIs will lose their popularity because that case POIs will get same portion of users based on their capacity.

With the above motivations, this research first introduces a fair POI recommendation to leverage user dynamic interests and POIs satisfaction in filling their capacity to a sufficient level. Here, we do not apply constant capacity–aware envy–free allocation to solve the POIs popularity reduction problem. We consider the existing models [2, 4, 10] that can capture user dynamic preferences. To capture fair POIs satisfaction, we consider capacity-based POI over–demand (allocations greater than the capacity) and under–demand (allocations tiny than the capacity) and solve the recommendation imbalance problem among the POIs. Finally, we focus on maximising user preferences and fair allocation among POIs. Therefore, this research's main objective is to select POIs that users get appropriate satisfaction and POIs get sufficient user allocation.

The contributions of this study can be summarised as follows:

- In this research work, we propose capacity-aware fair POI recommendation to solve users allocation in POI recommendation using deep learning model and allocation strategy in which users get maximum satisfaction and POIs get appropriate allocations. We study that users' equal distribution is not a good approach for POI recommendation due to its resources difference and constraints.

- Our proposed model capture users' spatiotemproal influence to calculate users' interest and capacity-based user allocation ensures that all POI providers get sufficient users to operate services in the long run.
- The model simultaneously learns user satisfaction and balances POIs exposure to solve recommendation and fairness problems in one framework.
- We conduct experimental analysis using five real datasets and show our proposed model superiority over the baselines regarding user and POI side evaluation metrics.

The remaining sections of this paper are organized as follows. The review of related works is described in Section 2. Problem statement with some preliminaries is introduced in Section 3. In Section 4, we present our capacity–aware fair POI recommendation model. After that, we analyze experiment results using five datasets in Section 5. Finally, we conclude the proposed model with future research direction in Section 6.

## 2. Related Work

Personalised and fair recommendations are two areas that are highly relevant to our proposed problem and model. We discuss key literature from these two areas in the following sections.

### *2.1. Personalised Next Top-K POI Recommendations*

The effectiveness of POI recommendation is associated with POI geographical locations, users visiting time, and user's interest [3, 11, 12, 13, 14]. These models consider users who have visited similar set of locations as having similar preferences. POIs and users' latent features are used to predict user preferences for unvisited locations, which improves recommendation performance [15, 16]. Previous studies also considered spatial and temporal correlations [17, 18]. Besides these, in the attention-based spatiotemporal influence ATST-LSTM [2] model, an attention memory network [19] and self-attention [5] were proposed for time-aware POI recommendation. People's behavior is influenced by the geography of the location which means people are more likely to visit nearby locations than distant ones. Guo et al. [20] proposed location perspective-based neighborhood-aware POI recommendation instead of users perspective POI recommendation. Check-in sequence and contents of POIs are used in *CAPE* [21] model. Feng et al. [22] proposed a non-Euclidean embedding model capturing user preference, region and category for the next-POI recommendation. Wu et al. [23] introduced personalized next POI recommendation using long-and short-term preferences. Li et al. [24] proposed a deep neural network for crossing-town POI recommendation. Different category–based POI recommendation has been applied in [25]. Moreover, CNN and multi-layer preceptors to POI recommendation have been employed in [26, 27]. Nowadays, attention-based transformers have shown significant improvement in capturing all dependencies at once using a non-recurrent encoder-decoder model for POI recommendation [4, 10]. Pang et al. [28] introduced a hierarchical attention mechanism for POI recommendation. Shi et al. [19] applied memory network and correlation-based embedding for time-aware POI recommendation. A sustainable tourist trip recommendation for groups of users has been addressed using a multi-objective approach in [29]. Qi et al. [30] introduced a group-based POI category recommendation model for LBSNs by addressing the challenges of sparse user preferences. It incorporates group influence, LSTM with attention mechanism, and utilizes POI categories to efficiently capture long-term dependencies and user interests. Liu et al. [31] proposed an interaction-enhanced and time-aware graph convolution network, for successive point-of-interest (POI) recommendations in location-based social networks. However, these models do not focus on both sides of the recommendation, i.e., the user's perspective and the location perspective.

### *2.2. Fair Recommendations*

Fair recommendation involves a multi-sided consideration such that all platforms are fair and unbiased. For example, Hannak et al. [32] investigated the biases of race and gender in freelance marketplaces. Hort et al. [33] introduced semantically correct word embedding for reducing gender bias. It has been shown that popular POIs often acquire most of the follower's visibility where new, but good ones are often starved for follower visibility [34]. In our daily life, recommendation platforms often involve multiple stakeholders [35]. Serbos et al. [36] proposed customers individual fairness-based group tour recommendations on travel booking sites. Recent research shows that multiple issues of unfairness and biases have happened on different platforms [37, 38, 39]. Suhr et al. [40] and Chakraborty et al. [41] introduced models for two-sided fairness combination problem. Patro et al. [6] proposed *FairRec* model that mapped product fair recommendation problem to a fair allocation problem to the POIs and customers. *FairRecPlus* [7] improves the *FairRec* model user satisfaction by using envy-free cycles. Chen et al. [9] proposed a room allocation model in which precisely two persons are assigned in each room. Li et al. [8] introduced a room allocation model considering various room capacity and budget constraints, which maximise social welfare. Therefore, our model differs from the existing models due to users' spatio-temporal dependencies, POIs capacity limits and personalisation.

### *2.3. Differences from existing studies*

Our proposed capacity-aware fair POI recommendation model differs from these earlier works. Prior studies show the top-k POI recommendations only consider user preferences where POI fairness was ignored. This research first introduces fair user allocation among the POIs based on capacity level, unlike E-commerce item recommendation and room allocation. Second, we simultaneously apply personalised user interest and POIs fairness to maximise user satisfaction and minimize POIs bias problems, whereas existing models are either user-centric or POI-centric. Last but not least, our proposed model generates a fair top-k POIs list that addresses the recommendation and allocation problems in one framework, whereas earlier works consider only one among them. Table 1 describes the main differences between the proposed model and earlier works.

## 3. Preliminaries and Problem Statement

This section describes the necessary definitions and problem formulation of capacity-aware recommendations.

Table 1: Comparison among proposed model and baselines.

| Models | Users Interest | Producer/POI Exposure | Technique | Recommendation | Allocation |
|---|---|---|---|---|---|
| ATST-LSTM [2] | ✓ | | Attention + LSTM | ✓ | |
| TLR-M [4] | ✓ | | Transformer | ✓ | |
| TLR-M_UI [10] | ✓ | | Transformer + POI Description | ✓ | |
| TOPK | ✓ | | TOPK Interest | ✓ | |
| LOWK | ✓ | | LOW-K Interest | ✓ | |
| Random | ✓ | | Random Allocations | ✓ | |
| FairRec [6] | ✓ | ✓ | Greedy Algorithm | ✓ | ✓ |
| FairRecPlus [7] | ✓ | ✓ | Greedy + Cycle Free | ✓ | ✓ |
| **CAFPR** (Proposed) | ✓ | ✓ | Demand Policy | ✓ | ✓ |

### *3.1. Preliminaries and Necessary Definitions*

**Definition 1.** Point of Interest (POI): *A POI is a place, typically in a city or theme park, where people find it interesting to visit, e.g., a roller coaster ride, a church or a museum. A set of POI is defined by* $P = \{p_1, p_2, \cdots, p_n\}$*, where each POI* $p_i \in P$ *has position coordinates, category, and openning times.*

**Definition 2.** Capacity of POI: *Let* $p_i \in P$ *be a POI. The capacity of* $p_i$ *is the maximum number of users who can visit* $p_i$ *at any timestep* $t \in T$*, where* $T$ *is a total ordinal set of timestamps.*

*The capacity may not be available and can be estimated as follows:*

$$Cap(p_i) = max\big(\sum_{u \in U} \delta(u, p_i, t_k)\big) \tag{1}$$

*where* $\delta(u, p_i, t_k) = 1$ *if user* $u$ *stays (not just arriving/entering) at POI* $p_i$ *at time* $t_k$*, otherwise* $\delta(u, p_i, t_k) = 0$.

**Definition 3.** Demand of POI*: Let* $p_i \in P$ *be a POI, the demand of* $p_i$ *is the number of users who want to visit* $p_i$ *at time* $t \in T$ *and it is defined as:*

$$Dem(p_i, t) = \sum_{u \in U} \gamma(u, p_i, t) \tag{2}$$

*where,* $\gamma(u, p_i, t) = 1$ *if the user* $u$ *want to visit* $p_i$ *at times* $t$*, otherwise* $\gamma(u, p_i, t) = 0$*. Here* $U$ *is the set of all users.*

We define the demands of POIs as three types: over-demand, under-demand and exact demand. They are defined in terms of visitor demand and POI capacity as follows:

**Definition 4.** Over, Under and Exact Demand*: The demand of a POI relative to the POI capacity is defined as follows:*

$$M(p_i) = Dem(p_i, t) - Cap(p_i) \tag{3}$$

*where,* $M(p_i)$ *is net demand of* $p_i$*. If* $M(p_i) > 0$*, then it is over demand,* $M(p_i) < 0$ *is under demand and* $M(p_i) = 0$ *is exact demand.*

**Definition 5.** Visit Level*: It represents the number of users who visit the particular POI* $p_i$ *at a particular time* $t$*.*

User interest in the POI depends on the POI category [16, 10], user–to–POI distance and time [2]. Preferences may also change based on time [4, 42]. Our model can utilise any interest function that computes a relevance score for a user–POI–time triple. Therefore, the user satisfaction of a POI depends on multiple complex factors, which can be defined as follows:

**Definition 6.** User Satisfaction to POI*: The satisfaction of user* $u_j$ *at time* $t$ *for POI* $p_i$ *is denoted by* $S_{u_j}(p_i, t) \in [0, 1]$*, the preference score of POI.*

Allocation refers to the distribution of resources, such as goods or tasks, among a group of individuals. In the context of resource allocation, the concept of envy-freeness is often discussed. Envy-freeness means that no one in the group envies the resources of others. Envy-freeness is considered a desirable property in resource allocation because it ensures fairness and equitable distribution of resources. Achieving envy-freeness can be a challenging task, especially when resources have different values for different individuals and when the number of resources is limited. However, finding an envy-free allocation is possible in some cases, such as when resources can be divided into equal portions.

In POI recommendation, users visit a POI for a period of time, depending on their interest in the POI and other factors (e.g., time budget). This POI allocation should be a fair allocation; otherwise, some POI will experience over-demand, while others will face the under-demand problem. The POI allocation should be envy–free as much as possible. An envy-free user allocation is an allocation of users among the POIs in which no POI provider will envy the other POI providers. Existing room allocation [9, 8] or item allocation [6] processes are not suitable for envy-free POI allocations, as the room capacity is limited (maximum of four occupants), whereas POI capacity is up to the hundreds. Furthermore, room and item allocation preference does not change based on time and spatial distance, whereas POI allocation demand depends on these variables. Finally, POI capacities vary from one POI to another; we cannot consider fixed user number differences based on envy-free allocation, e.g., EF1 (envy free up to one item) or EFN (envy–free up to $N$ items). To address these limitations, we need a capacity–based envy–free allocation approach in which POI exposure utilisation depends on user demand and POI capacity ratio.

Capacity-based envy-free POI allocation is an allocation method that considers POIs capacities to distribute users. This method varies from other traditional envy-free allocations, where resources are simply divided equally among individuals, regardless of their capacities. If we use constant capacity proportion-based allocation, we will miss input from users' preferences and some POI will lose their popularity. To make a balance between user preferences and capacity–based free allocation we defined our capacity–based envy-free allocation as follows:

**Definition 7.** Capacity-based Envy–free Allocation: *A pair of POIs allocated among the users will be envy-free if each POI gets a minimum portion ($min_{dem}$) of users based on their capacity and the differences of demand portion for each POI in the pair will be less than the user–defined threshold value ($Envy_{cap}$).*

The demand ratio of POIs based on capacity is greater than or equal to the threshold value $min_{dem}$, and the allocation difference between a pair of POIs is smaller than the $Envy_{cap}$ value. Two POIs $p_i$ and $p_h$ allocations are capacity–aware envy–free allocations if it follows both constraints at time $t$ as follows.

- $\frac{Dem(p_i,t)}{Cap(p_i)} \geq min_{dem}$ and $\frac{Dem(p_h,t)}{Cap(p_h)} \geq min_{dem}$,

- $|\frac{Dem(p_i,t)}{Cap(p_i)}$ - $\frac{Dem(p_h,t)}{Cap(p_h)}| < Envy_{cap}$.

**Definition 8.** Envy-free: *Two POIs $p_i$ and $p_h$ will be envy-free at time $t$ if these two POIs get sufficient POI exposure (based on $min_{dem}$) and their capacity-based exposure ratio is smaller than the $Envy_{cap}$ threshold.*

Figure 2 presents an example involving three POIs, where the capacity of $p_1$ is 10 and the demand represented by $Dem(p_1, t)$ is 8. Similarly, the capacity of POIs $p_2$ and $p_3$ are 6 and 8 and the demands are 2 and 4, respectively. We assume $Envy_{cap} = 0.4$ and $min_{dem} = 0.5$. Thus, according to Definition 3.1, $p_1$ and $p_2$ are not envy-free; for the other pairs, the $(p_1, p_3)$ ratio difference is 0.3. The $(p_2, p_3)$ ratio difference is 0.17, and both are envy-free.

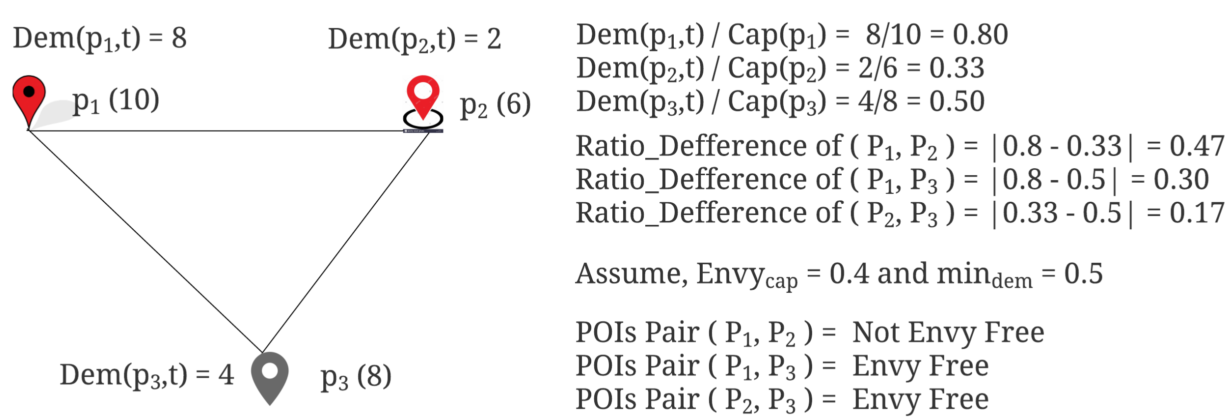


Figure 2: Capacity–based envy–free allocation.

*3.2. Problem Statement*

**Capacity–aware Fair POI Recommendation:** Consider a user set $U = \{ u_1, u_2, \cdots, u_m \}$, POI set $P = \{ p_1, p_2, \cdots, p_n \}$, timestamp set $T = \{t_1, t_2, \cdots, t_l \}$ and POI capacity set $C = \{\ Cap(p_1), Cap(p_2), \cdots, Cap(p_n)\ \}$. The goal of fair POI recommendation is to recommend $k$ POIs such that users will receive maximum satisfaction, while the POIs will be envy–free and receive capacity–aware fair POI recommendation. This means that our proposed model maximises user satisfaction based on user preferences and also maximises envy–free POI pair allocations based on fair allocation policy.

Figure 3 illustrates the two objectives: maximising user satisfaction and maximising envy–free POI pairs. At each time $t$, attempts are made to allocate many users to a POI, resulting in many-to-one mapping. Let $X_{i,j,t}$ indicate that user $u_i$ is allocated to POI $p_j$ at time $t$, where $X_{i,j,t} \in \{0, 1\}$. Thus, the user objective is as follows:

$$\max_{X_{i,j,t}} \sum_t \sum_{p_j} \sum_{u_i} \left( \underset{\substack{\downarrow \\ \text{User Satisfaction}}}{X_{i,j,t} S_{u_i}(p_j, t)} \right) \tag{4}$$

where $S_{u_i}(p_j, t)$ represents the user $u_i$ satisfaction score with POI $p_j$ at time $t$. A user can only visit / be allocated to one location at a time; hence, we have the constraint $\sum_j x_{i,j,t} = 1, \forall i, t$. Alternatively, if users do not have to be allocated at a POI across all times, we have $\sum_j x_{i,j,t} \leq 1, \forall i, t$.

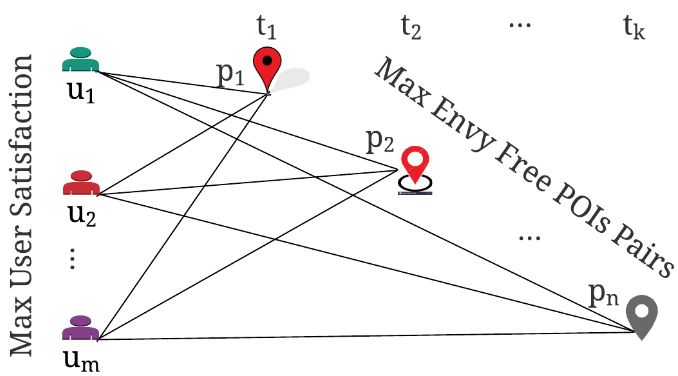


Figure 3: Objectives of fair POI recommendation model.

Here, we assume that users can transit from one POI to another POI based on their user preferences. However, our proposed model is also able to incorporate travel costs between POIs and user interests. Moreover, a user will not change their POI allocation unless their satisfaction function changes with time. The POI objective is as follows:

$$\max \sum_{t} \sum_{p_j, p_k \in P, p_l \neq p_k} \Big( \underset{\text{POIs Envy}}{\underset{\downarrow}{Envy\_Free(p_j, p_k, t)}} \Big) \tag{5}$$

where, $Envy\_Free(p_j, p_k, t) = 1$ if POI pair $p_j$ and $p_k$ are envy–free at time $t$; otherwise, $Envy\_Free\ (p_j, p_k, t) = 0$. Here, $Envy\_Free\ (p_j, p_k, t)$ depends on the user allocation between the POIs $p_j$ and $p_k$. These two objectives ensure user satisfaction and minimum envy among the POIs, respectively. Notably, optimising these objectives is challenging because the two aims may conflict with each other.

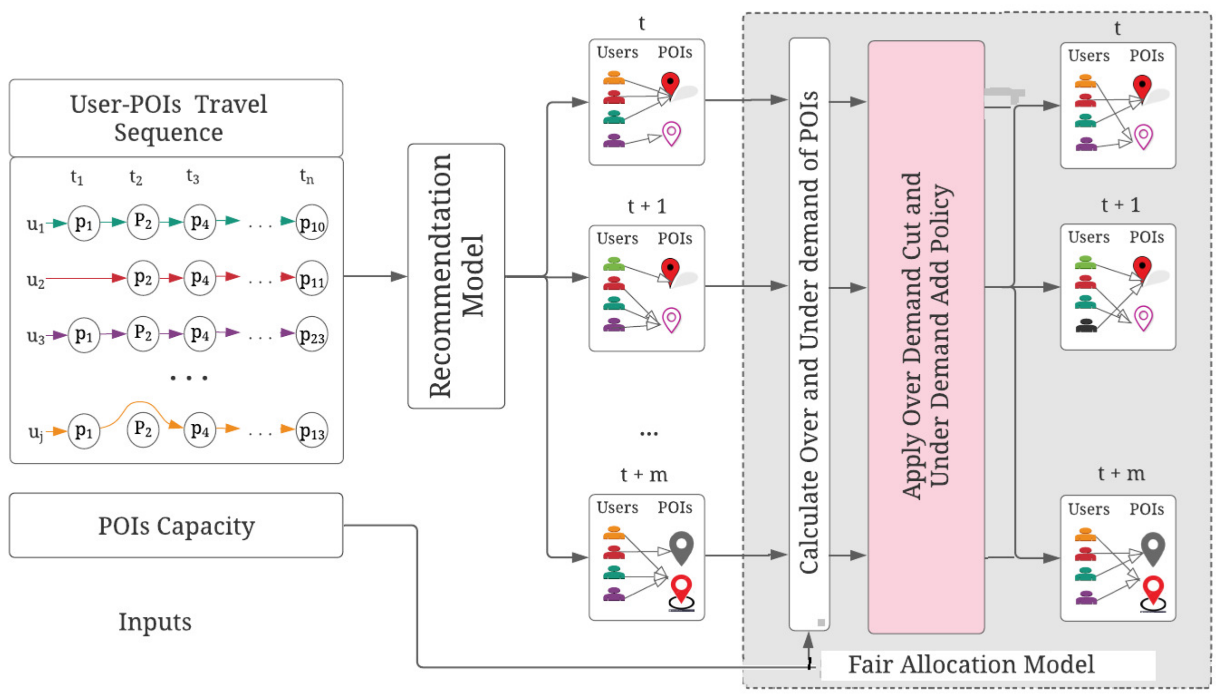


Figure 4: Proposed capacity-aware fair POI recommendation (CAFPR) model.

## 4. Proposed Model

Figure 4 illustrates our proposed model that trades off between user satisfaction and fair POI recommendations. The recommendation model recommends a set of POIs to the users, whereas the allocation policy selects the best POIs in which POIs get sufficient visit levels.

The recommendation part of the proposed model captures user dynamic behaviours and spatiotemporal periodic dependencies. The recommendation part recommends top-k POIs that users get the highest satisfaction. After that, the fair allocation balances user satisfaction and POIs visit demands. In the recommendation part, we need to capture user spatio-temporal dependencies. Thus, we seek the model that can capture user–personalised preferences and spatio-temporal influences. The existing models [4, 10] can capture user spatio-temporal dependencies better than the state-of-the-art models. To model user interests-based POI recommendations, we apply multi-attention-based transformer architecture [4] to capture heterogeneous factors' influence appropriately and attention-based LSTM model [2] to capture spatiotemporal information in POI recommendation. The allocation part uses over-demand cut and under-demand add policies to allocate POIs among the users to make the POI capacity-based envy-free. We describe each part of the proposed model in the next subsections.

### *4.1. Recommendation Model*

Existing POI recommendation models [4, 2, 3, 14, 17] employed user interest–based POI recommendation that considers spatiotemporal influences and personalised user preferences that ensure user satisfaction. Our main motive of this research paper is to propose a fair POI recommendation method that ensures that both users and POIs are satisfied with the recommendations. Thus, we avoid a detailed description of the POI recommendation model. We can apply any recommendation model in our proposed fair $k$ POI recommendation. Each model recommends candidate sets to the users. The candidate set focuses on user satisfaction but not user allocation among POIs, as the latter is addressed by the allocation model. Here, selection probability distributions consider a user satisfaction score that depends on spatiotemporal dependencies and personalised preferences. To make a fair POI recommendation, however, we need to focus on POI utilisation. The POI utilisation is based on users' visit level, which depends on the user allocation.

### *4.2. Allocation Model*

User interest–based recommendation may create biased POI allocation; as a result, POIs are assigned over their capacity, leading to a crowding problem, or under-allocated, leading to user scarcity problem. Appropriate POI allocation is necessary to solve these problems. However, this is challenging because of variability in POI demand. Due to the variations in the number of visitors at different times, exact demand is hard to determine. Hence, we introduce some margin / flexibility with the thresholds. This allocation model solves over–demand and minimises under–demand, seeking to achieve exact demand for the POIs to the greatest extent possible. Figure 5(a) depicts the over-demand scenario, in which three users are interested in being allocated to POI $p_1$, with a capacity of two. However, $p_1$ cannot fulfill demands from all of these users at one time because its capacity is two. Figure 5(b) illustrates POI under–demand, in which the capacity of each POI is two, but only one user is allocated to each POI. However, solving POI over and under-demand is challenging

General room sharing problems [8, 9] solve the over–demand problem based on capacity diversity and budget constraints. However, we cannot apply an existing model because the problem in our case is NP-hard, meaning that we cannot use existing models designed to solve a room-sharing problem with a maximum capacity of four beds. Moreover, POI visitation levels are more significant than can be accounted for in room-sharing problems. Thus, we are unable to apply the existing room allocation models, in which computational complexity increases exponentially with the POI capacity. The *FairRec* model [6] proposed a fair item recommendation that allows each POI to receive envy-free allocation up to one item. However, the *FairRec* model allocates items equally; this approach does not perform well in POI recommendation because POIs have different capacities. We can not apply a greedy approach to distribute users based on capacity because in that case, users will not have their individual preferences satisfied. Furthermore, some POIs will lose their popularity because all POI will get the same proportion of user numbers.

To solve these challenges, we propose a new POI allocation model to solve the over and under–demand problems. The model first identifies over–demanded and under-demanded

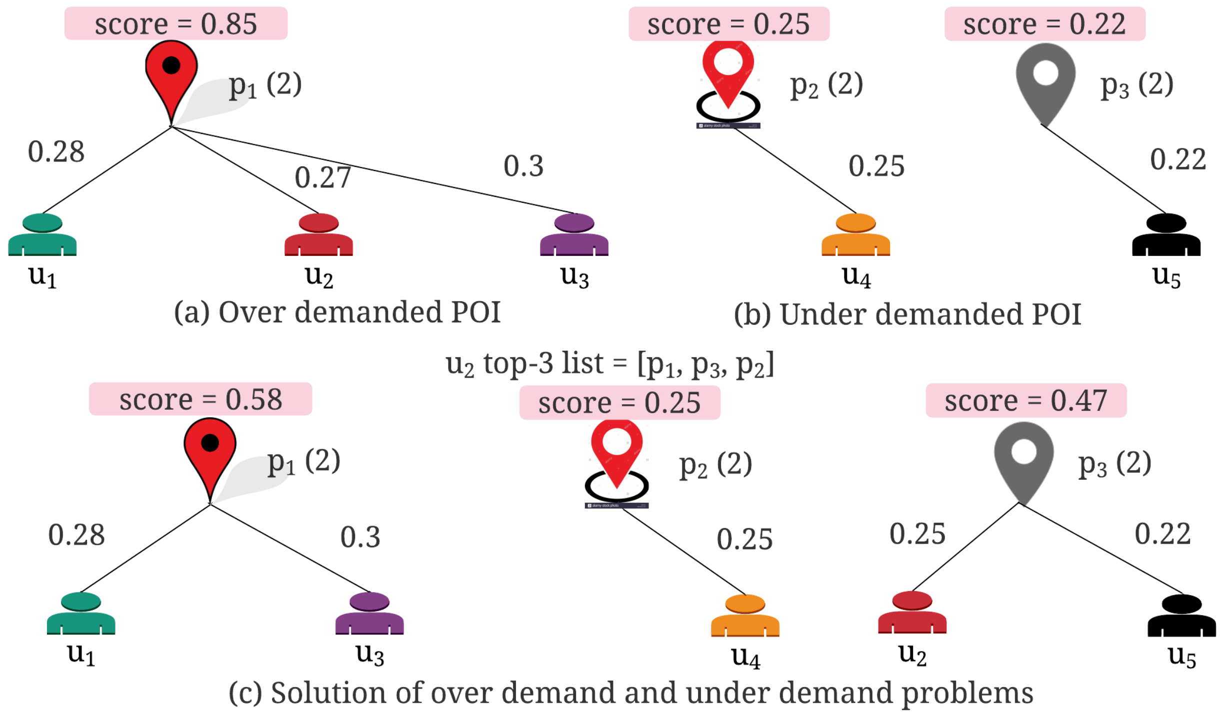


Figure 5: Example of POI over and under–demand problems. (a) The over-demand scenario, where three users are interested in visiting $p_1$ at a time when capacity is two. (b) The under–demand scenarios where demand is one and capacity is two. (c) Illustration of an over–demand cut and under–demand add policy, where the score indicates the user interest in the POI. Here, the weight between user and POI indicates the users' personalised interest in the POI.

POIs. Subsequently, among the users in the over–demanded POI, we select a sub-group of these users. Below, we will describe how to select a capacity-based user group for POI allocation. After that, we assign these users from over-demanded POIs to under-demanded POIs, to balance out the demands among POIs.

#### *4.2.1. Over–demand Cut Policy*

In allocation problems, over-demand is a common occurrence. In a POI network, over-demand may occur due to POI popularity. Users generally prefer to visit popular POIs in a network. In Figure 5(a), three users, $u_1$, $u_2$ and $u_3$, are interested in visiting POI $p_1$ at the same time. Therefore, we have to select two of the three users to get maximum satisfaction. This work applies a cut policy that removes users until the number of users matches the POI capacity. In this example, we have to cut one user from the POI's assigned list. While we can cut extra users randomly from this list, this may reduce user satisfaction. User satisfaction will vary depending on how much users like the POI. Here, users $u_1$, $u_2$ and $u_3$ like POI $p_1$ with a value of 0.28, 0.27 and 0.3, respectively. These values come from the POI recommendation transition probability. We therefore remove user $u_2$ from the over-demand POI $p_1$, as that maximises the utilisation of the POI while maximising the total satisfaction among the remaining users.

Figure 5(a) shows that the POI allocation satisfaction is 0.85 (0.28 + 0.27 + 0.3). If we cut $u_1$ users from the network POI, the assigned weight of POI $p_1$ will equal the capacity, and the new score will be 0.57 (0.27 + 0.3). Similarly, we get a satisfaction score of 0.58 (0.28 + 0.3) when we remove $u_2$ and 0.55 (0.28 + 0.27) when we remove $u_3$. Our main goal is to maximise satisfaction; thus, we cut user $u_2$ from the POI assign list, which maximises the overall satisfaction associated with the POI and assigned users. This cut process is continued until the number of users in the allocated POI is less than the capacity of the POI.

#### *4.2.2. Under–demand Add Policy*

After removing users based on the over-demand cut policy, we need to assign these users to the under–demanded POIs. Figure 5(b) shows that $p_2$ and $p_3$ are under-demanded POIs that are assigned only one user, although their capacity is two. The question that now arises is how to select an appropriate POI for the cut users that will maximise satisfaction. Our solution is to add users to the POI that can generate the highest satisfaction possible. For example, in Figure 5, $p_3$'s score is higher than $p_2$'s score because the recommended top-3 list for user $u_2$ is $[p_1, p_3, p_2]$. Thus, we assign $u_2$ to the $p_3$ demand list. If the assigned POI demand is equal to the capacity, we remove it from the under-demanded POI group and set it as a fair POI allocation.

#### *4.2.3. Fair Allocation Algorithm*

Algorithm 1 describes the POI fair allocation process. The algorithm takes user set, POI set, POI demand, POI capacity as input and returns fair allocated POIs to the system. Initially, the set of un-allocated users, under-allocated POIs and fair-allocated POIs are empty in line 1. For each POI in POIs set, calculate demand and if the demand is higher

**Algorithm 1:** $FairPOIAllocation(U, P, P_{dem}, P_{cap})$

**Data:** $U$ : Users set; $P$ : POI set; $P_{dem}$ : Users POI demand; $P_{cap}$: POI capacity set.

**Result:** $POI_{allo}$: Fair Allocated POIs

```
U_un_allo, POI_und_allo, POI_allo ← ∅, ∅, ∅;
for p^i ∈ P do
    while p^i_dem > p^i_cap do
        u_cut ← ∅; max_score = 0.0;
        for u ∈ p^i_dem do
            score = Score after removing u from p^i_dem
            if max_score < score then
                max_score = score
                u_cut = u                                   /* Least significant User*/
            end
        end
        U_un_allo = U_un_allo ∪ u_cut                       /*Unassigned users*/
        p^i_dem = p^i_dem − 1                               /* Reduce POI demand */
    end
    if p^i_dem < p^i_cap then
        Applying POI_und_allo = POI_und_allo ∪ p^i
    else
        POI_allo = POI_allo ∪ p^i_dem
    end
end
for u_j ∈ U_un_allo do
    p_sel = ∅;
    for p^l ∈ Rec_List_{u_j} do
        if p^l ∈ POI_und_allo then
            p_sel = p^l                                     /* Next significant POI */
            break
        end
    end
    Add u_j to p^sel and update p^sel_dem.
    if p^sel_dem == p^sel_cap then
        POI_allo = POI_allo ∪ p^sel_dem
        Remove p^sel_dem from POI_und_allo
    end
end
POI_allo = POI_allo ∪ POI_und_allo
Return POI_allo;                                            /* Return Fair POI allocation */
```

than the capacity, it finds unallocated users using the over-demanded user cut policy in lines 4-11. The removed user is added to the under-allocated users' list in line 12. This process is continued until the POI demand reaches POI capacity using the loop in lines 3-14. After removing over demand, we add POI demand list as fair POI allocation if its demand is equal to the capacity in line 18. Otherwise, add to the under-allocation list in line 16. This process is continued for each POI using the loop in lines 2-20. Then, we need to add unallocated users to the under-allocated POIs. For each un-allocated user in $U_{un\_allo}$ list, identify the appropriate POI among the under-demanded POIs list in lines 22-28. The cutting user $u_j$ is added with $p^{sel}$ and update $p^{set}_{dem}$ in line 29. If the update demand is equal to capacity, remove the POI from the under-demanded list and add it to the fair allocation list in lines 30-33. After allocating all un-allocating users in lines 21-34, the under-demanded POIs are added with fair allocated POIs in line 35. Finally, the algorithm returns fair POI allocation in line 36.

### *4.3. The CAFPR Algorithm*

Our proposed **C**apacity **A**ware **F**air **P**OI **R**ecommendation (**CAFPR**) model is shown in Algorithm 2. The main challenges of this model are to incorporate recommendation and allocation using multiple features. The testing and training sets are first partitioned in line 1, after which the sample input based on batch size is fed into the recommendation model in lines 2–3. Next, the recommendation model generates user satisfaction scores in line 4. We apply the POI allocation model that ensures fair user distributions based on their capacity in line 5. The top-k POI recommendations learned and generated by the model are distributed to the users in line 6. Finally, the algorithm returns to the *CAFPR* model in line 8. Upon the completion of this algorithm, users get appropriate satisfaction, and POIs get a fair user allocation.

Here, POI allocation is an NP-hard because POI will get visitation level considering any subset of users. Hence we consider both user satisfaction and POI visitation level together, we can not solve the allocation problem using existing mathematical programming, e.g., cake-cutting, picking sequence, EFN. Thus, we apply over-demand cuts and under-demand

**Algorithm 2:** CAFPR Model (U, P, C, SH)

**Data:** U=Users, P=POIs, C=Capacity, SH=Sequence History

**Result:** CAFPR Model

$Seq\_train, Seq\_test = Partition\_Data(SH, test\_ratio)$

**for** $X_b \leftarrow sample(Seq\_train)$ **do**

  Pass these $x_b$ into in recommendation model.

  Find User satisfaction $US_{u_j}(p_i)$ from the output of recommendation model.

  Get $PS_{p_i}$ using Algorithm 1.

  Build the Fair Top-K POI Recommendation CAFPR model.

**end**

*Return* CAFPR Model

add policy-based heuristics to ensure the POI visitation level at a threshold value.

## 5. Experiments

This section outlines our experiments, evaluates and compares our proposed *CAFPR* model with the selected baselines, and discusses the results. We first describe the datasets and then compare a set of baselines against our proposed model.

### *5.1. Datasets*

We conduct extensive experimental analyses based on the five datasets [1] that were used in [16, 4]. We applied the same data pre-processing steps outlined in [4]. Figure 6 shows the POI capacities in the California Adventure dataset. We can see that the maximum capacity of POIs is 13 and that the minimum is 6. This shows that 13 POIs out of the 25 POIs have the same capacity level. This is a theme park dataset, meaning that the number of users is limited. We can set a unique time-based number of users who visit those POIs for 15 min, 30 min, or one hour; in this capacity calculation, we have applied a number of users based on a 15-min visit duration.

[1] https://codeocean.com/capsule/5378181/tree/v1

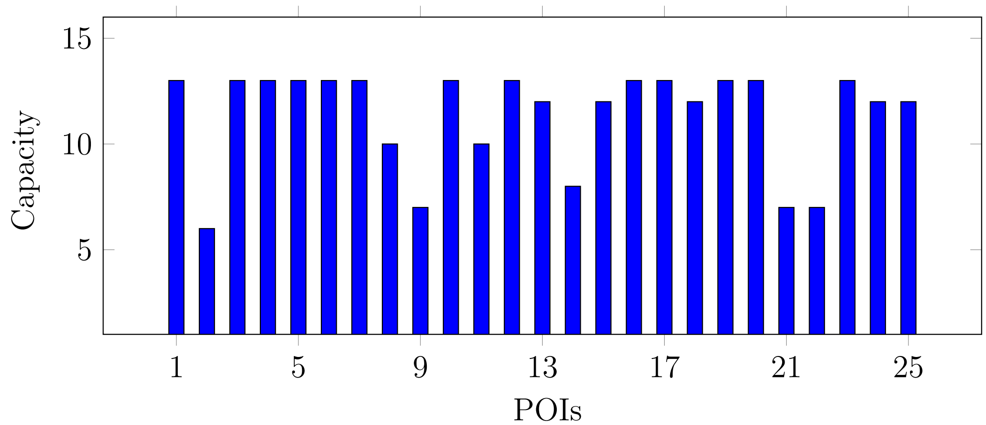


Figure 6: The POI capacities in the California Adventure dataset.

We split the data into testing and training sets; specifically, 70% is used as the training set, and the remaining 30% is designated as the testing set. The details of the datasets are summarised in Table 2.

Table 2: Description of various datasets (K = 1000)

| Dataset | # Photos / # Check-ins | # POI Visits | # Users | # POIs |
|---|---|---|---|---|
| California Adventure | 193.0K | 57.2K | 2.6K | 25 |
| Magic Kingdom | 133.2K | 74.0K | 3.3K | 27 |
| Budapest | 36.0K | 18.5K | 0.9K | 39 |
| Toronto | 157.5K | 39.4K | 1.4K | 30 |
| Melbourne | 17.1K | 5.8K | 0.9K | 242 |

*5.2. Baseline Algorithms*

In this research work, our proposed *CAFPR* model recommends $k$ POIs to the users and ensures POI's capacity–aware allocation fairness. Thus, to evaluate the performance of the proposed model, we consider two recommended algorithms. The first one is **TLR-M** [**4**], which is a multi-head attention-based transformer model that incorporates user interests, distance and temporal information to simultaneously recommend top-k POIs to the users and predict their queuing time. And, the second one is **ATST-LSTM** [**2**], which is an attention-based spatiotemporal LSTM–based model that uses check-in contextual information to recommend the top-k POIs. The model uses spatial and temporal information to predict users' future check-in patterns accurately.

Subsequently, we run the proposed *CAFPR* model on all datasets considering two values of the recommended size k, which are 5 and 10. To facilitate fair comparison, we use the following baselines:

- **FairRec [6]:** The model uses a two-sided definition of fairness, which encompasses both social or judicial precepts and long-term sustainability. The model maps the fair recommendation problem to a fair allocation problem.
- **FairRecPlus** [**7**]: It is a modification of FairRec that improves the recommendation performance for the customers ensuring the same level of fairness.
- **TOPK:** The TOPK model recommends top-k POIs based on the recommendation output.
- **LOWK:** The LOWK model recommends low-k POIs based on the recommendation output.
- **Random:** It randomly selects k POIs for user recommendation based on the recommendation output.

### *5.3. Performance Evaluation*

To measure the performance of our *CAFPR* model, we divide the evaluations into two sides: user–side metrics and POI–side metrics. Here, the metrics for both sides are those used in [6]. Moreover, we also employ the additional user-side evaluation metrics used in [4]. Therefore, to evaluate the *CAFPR* and the baselines, we use the following evaluation metrics.

#### *5.3.1. User–side Metrics*

The following metrics have been selected for use in evaluating the proposed model's fairness to the users:

- **Precision@k:** Assume that $P_r$ denotes the next POIs in the actual visit sequence and that $P_k$ represents the top–k recommended POIs. The precision represents the ratio of the next top-k POI that is present in the original next POIs as follows:

$$Precision@k = \frac{|P_r \bigcap P_k|}{|P_k|} \tag{6}$$

- **Recall@k:** We use the same $P_r$ and $P_k$ as above. Here, Recall@k represents the proportion of real next POIs that are also present in the top-k recommended POIs that is defined as follows:

$$Recall@k = \frac{|P_r \bigcap P_k|}{|P_r|}. \tag{7}$$

- **F1-Score@k:** This is the harmonic mean of both recall and precision of user $u$, defined as follows:

$$F1 - Score@k = \frac{2 \times Recall@k \times Precision@k}{Recall@k + Precision@k} \tag{8}$$

- **NDCG@k:** This evaluates the performance of next POI recommendation based on the position of the next POI in the result list. It is defined as follows:

$$NDCG@k = \frac{1}{U} \sum_{u \in U} \frac{DCG@k(u)}{IDCG@k(u)} \tag{9}$$

$$DCG@k(u) = \sum_{i=1}^{k} \frac{2^{Rel_u} - 1}{log_2(Ind_u + 2)} \tag{10}$$

  where $Rel_u$ is 1 if $hit@N = 1$ and 0 otherwise. $Ind_u$ is the hit position index and takes values ranging from 0 to N-1. Finally, IDCG@k(u) is the ideal DCG@k(u), meaning that the index values range from 0 to k-1.

### *5.3.2. POI-side Metrics*

Fair recommendation depends not only on user-side metrics but also on POI-side metrics. We modify [6, 7] evaluation metrics based on the capacity-aware evaluation and define the following evaluation to demonstrate POI–side fairness and efficiency.

- **Capacity-based Fairness of Satisfied POIs (CFSP):** Capacity–based POI fairness depends on the number of user allocations considering the capacity limit. If the user allocation of a POI is greater than the minimum percentage of capacity exposure, then the POI will be satisfied. The fraction of the satisfaction score can be defined as follows:

$$CFSP = \frac{1}{|P|}\sum_{p\in P} Dem(p) \geq Cap(p) * min_{dem} \tag{11}$$

  where $Dem(p)$ and $Cap(p)$ represent the allocated users and capacity of $p$, respectively. Here, $min_{dem}$ is the minimum under–demand threshold parameter.

- **Capacity based Envy Free Allocation (CEFA):** User recommendation depends on the users' personalised interests. Therefore, if the ratios of POI exposures to POI capacities are diverse, low-exposure POIs will lose market value and will be considered to envy high-exposure POIs. Thus, we calculate the envy-free allocation score as follows.

$$CEFA = 1.0 - \frac{\sum_{p_i=1}^{P-1}\sum_{p_j=p_i+1}^{P} Envy\_Score(p_i, p_j)}{|P| \times (|P|-1)/2} \tag{12}$$

  where, $Envy\_Score(p_i, p_j) = 1$ if two POIs $p_i$ and $p_j$ are not envy free according to the figure 2. Otherwise, $Envy\_Score\ (p_i, p_j) = 0$.

- **Gini Index (Gini)**: This index measures item frequency distribution inequality [43], e.g. the number of users (exposure) in the POI recommendation context. In short, it measures POI individual level–based exposure. Given a set of POIs $P = \{p_1, p_2, \cdots, p_n\}$ and their exposure numbers $\{e_1, e_2, \cdots, e_n\}$, the Gini is calculated as follows:

$$Gini(P) = \frac{1}{2|P|^2\bar{e}}\sum_{i=1}^{|P|}\sum_{j=1}^{|P|}|e_i = e_j| \tag{13}$$

  where $\bar{e}$ is the average exposure number of all POIs.

### *5.3.3. Balancing Metrics between User and POI Sides*

To provide a balanced consideration of two-sided metrics, we use rank metrics incorporating NDCG and CEFA metrics as follows. Here, the first three user–side evaluations represent user satisfaction scores, while NDCG expresses the selection of POIs' order ranking. Moreover, for the provider-side metrics, Gini represents capacity-free user exposure, and CFSP expresses POI utilisation value, although this cannot describe the level of fairness. Thus, we select the CEFA–based ranking, which can account for the capacity–based envy–free allocations among the POIs.

- **Rank@k**: Rank metrics indicate the average rank of NDCG (user side) and CEFA (POI side), which is as follows:

$$Rank@k = \frac{Rank(NDCG@k) + Rank(CEFA@k)}{2} \tag{14}$$

## *5.4. Results Analysis*

The fairness and recommendation efficiency of our proposed *CAFPR* model and the existing baselines are evaluated on two sides: the user side and the POI side. In the following sections, we describe the fairness results for each side in detail.

Table 3: Performance analyses between the proposed *CAFPR* and baselines using *TLR-M* recommendation baselines in top-5 and top-10 POI allocation. Here, ↑ and ↓ indicate that high and low values, respectively, are desirable, while numbers in bold indicate the best results and underlined scores the second-best results. Numbers within brackets show the rank based on F1-score and CEFA scores, where 1 = best and 5 = worst.

| | | | topk = 5 | | | | | | | | topk = 10 | | | | | | | | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | User side evaluations | | | | POI side evaluations | | | Rank↓ | User side evaluations | | | | POI side evaluations | | | Rank↓ | Rank↓ |
| | | | Pre ↑ | Re ↑ | F1 ↑ | NDCG ↑ | CFSP ↑ | CEFA ↑ | Gini ↓ | | Pre ↑ | Re ↑ | F1 ↑ | NDCG ↑ | CFSP ↑ | CEFA ↑ | Gini ↓ | | |
| *TLR-M* | California | Random | 0.037 | 0.186 | 0.062 | 0.039 (5) | 0.7 | 0.752 (3) | 0.259 | 4.0 | 0.034 | 0.339 | 0.062 | 0.040 (4) | 0.52 | **1.0** (1) | 0.262 | 2.5 | 3.25 |
| | | LOWK | 0.033 | 0.165 | 0.055 | 0.035 (6) | 0.44 | 0.483 (5) | **0.189** | 5.5 | 0.033 | 0.330 | 0.063 | 0.035 (5) | 0.60 | 0.677 (4) | **0.207** | 4.5 | 5.0 |
| | | TOPK | 0.037 | 0.191 | 0.064 | 0.042 (4) | 0.560 | 0.497 (4) | 0.254 | 4.0 | 0.037 | 0.372 | 0.068 | 0.044 (2) | 0.48 | 0.767 (3) | 0.265 | 2.5 | 3.25 |
| | | FairRec | 0.038 | 0.192 | 0.064 | 0.043 (3) | 0.52 | 0.98 (2) | 0.263 | 2.5 | 0.038 | 0.375 | 0.068 | 0.043 (3) | 0.64 | 0.940 (2) | 0.264 | 2.5 | 2.5 |
| | | FairRecPlus | 0.038 | 0.192 | 0.064 | 0.044 (2) | 0.52 | **1.0** (1) | 0.263 | 1.5 | 0.038 | 0.376 | 0.068 | 0.043 (3) | 0.64 | 0.940 (2) | 0.264 | 2.5 | 2.0 |
| | | ***CAFPR*** | **0.041** | **0.206** | **0.069** | **0.045(1)** | **0.84** | **1.0** (1) | 0.246 | **1.0** | **0.040** | **0.401** | **0.073** | **0.046** (1) | **0.88** | **1.0** (1) | 0.253 | **1.0** | **1.0*** |
| | Magic K | Random | 0.030 | 0.145 | 0.048 | 0.037 (4) | 0.407 | 0.874 (2) | 0.262 | 3.0 | 0.031 | 0.315 | 0.057 | 0.037 (4) | 0.444 | 0.902 (2) | 0.265 | 3.0 | 3.0 |
| | | LOWK | 0.025 | 0.128 | 0.042 | 0.033 (5) | 0.407 | 0.652 (3) | 0.258 | 4.0 | 0.035 | 0.346 | 0.063 | 0.035 (5) | 0.593 | 0.761 (4) | 0.262 | 4.5 | 4.25 |
| | | TOPK | 0.031 | 0.153 | 0.051 | 0.038 (3) | 0.407 | 0.607 (4) | 0.265 | 3.5 | 0.034 | 0.344 | 0.062 | 0.041 (3) | 0.481 | 0.644 (5) | 0.271 | 4.0 | 3.75 |
| | | FairRec | 0.037 | 0.185 | 0.062 | 0.040 (2) | 0.444 | **1.0** (1) | 0.258 | 1.5 | 0.035 | 0.354 | 0.064 | 0.042 (2) | 0.407 | 0.846 (3) | 0.264 | 2.5 | 2.0 |
| | | FairRecPlus | 0.037 | 0.185 | 0.062 | 0.040 (2) | 0.444 | **1.0** (1) | 0.258 | 1.5 | 0.035 | 0.354 | 0.064 | 0.042 (2) | 0.407 | 0.846 (3) | 0. 264 | 2.5 | 2.0 |
| | | ***CAFPR*** | **0.038** | **0.188** | **0.063** | **0.043** (1) | **0.926** | **1.0** (1) | **0.254** | **1.0** | **0.037** | **0.372** | **0.068** | **0.044** (1) | **0.852** | **1.0** (1) | **0.253** | **1.0** | **1.0*** |
| | Budapest | Random | 0.020 | 0.102 | 0.037 | 0.026 (3) | 0.538 | 0.690 (2) | 0.280 | 2.5 | 0.022 | 0.221 | 0.040 | 0.026 (3) | 0.462 | 0.702 (3) | 0.28 | 3.0 | 3.0 |
| | | LOWK | 0.021 | 0.108 | 0.036 | 0.014 (5) | 0.461 | 0.419 (4) | 0.324 | 4.5 | 0.017 | 0.166 | 0.037 | 0.018 (4) | 0.564 | 0.490 (5) | **0.248** | 4.5 | 4.5 |
| | | TOPK | 0.024 | 0.108 | 0.039 | 0.027 (2) | 0.410 | 0.416 (5) | 0.289 | 3.5 | 0.022 | 0.221 | 0.040 | 0.028 (2) | 0.564 | 0.503 (4) | 0.279 | 3.0 | 3.25 |
| | | FairRec | 0.022 | 0.103 | 0.038 | 0.025 (4) | 0.436 | 0.654 (3) | 0.280 | 3.5 | 0.025 | 0.250 | 0.045 | 0.026 (3) | 0.436 | 0.733 (2) | 0.284 | 2.5 | 2.75 |
| | | FairRecPlus | 0.022 | 0.103 | 0.038 | 0.025 (4) | 0.436 | 0.654 (3) | 0.280 | 3.5 | 0.025 | 0.251 | 0.046 | 0.026 (3) | 0.436 | 0.733 (2) | 0.284 | 2.5 | 3.0 |
| | | ***CAFPR*** | **0.025** | **0.127** | **0.042** | **0.033** (1) | **0.897** | **0.994** (1) | **0.272** | **1.0** | **0.026** | **0.259** | **0.047** | **0.032** (1) | **0.897** | **0.853** (1) | 0.256 | **1.0** | **1.0*** |
| | Toronto | Random | 0.025 | 0.131 | 0.047 | 0.032 (4) | 0.433 | 0.924 (2) | 0.348 | 3.0 | 0.031 | 0.31 | 0.054 | 0.032 (4) | 0.367 | 0.924 (2) | 0.269 | 3.0 | 3.0 |
| | | LOWK | 0.022 | 0.119 | 0.037 | 0.029 (5) | 0.366 | 0.409 (5) | **0.214** | 5.0 | 0.025 | 0.249 | 0.052 | 0.032 (4) | 0.433 | 0.683 (5) | 0.269 | 4.5 | 4.75 |
| | | TOPK | 0.028 | 0.144 | 0.048 | 0.035 (2) | 0.367 | 0.720 (4) | 0.281 | 3.0 | 0.031 | 0.316 | 0.058 | 0.035 (3) | 0.503 | 0.823 (4) | 0.302 | 3.5 | 3.25 |
| | | FairRec | 0.029 | 0.144 | 0.049 | 0.034 (3) | 0.367 | 0.903 (3) | 0.263 | 3.0 | **0.033** | **0.334** | **0.061** | 0.036 (2) | 0.333 | 0.910 (3) | 0.267 | 2.5 | 2.75 |
| | | FairRecPlus | 0.029 | 0.145 | 0.049 | 0.032 (4) | 0.367 | 0.903 (3) | 0.2363 | 3.5 | **0.033** | **0.334** | **0.061** | 0.036 (2) | 0.333 | 0.910 (3) | 0.265 | 2.5 | 3.0 |
| | | ***CAFPR*** | **0.030** | **0.148** | **0.05** | **0.04** (1) | **0.867** | **1.0** (1) | 0.246 | **1.0** | 0.032 | 0.322 | 0.059 | **0.046** (1) | **0.867** | **1.0** (1) | **0.265** | **1.0** | **1.0*** |
| | Melbourne | Random | 0.004 | 0.019 | 0.006 | 0.005 (3) | 0.442 | 0.506 (3) | 0.189 | 3.0 | 0.006 | 0.057 | 0.010 | 0.004 (4) | 0.554 | 0.904 (3) | 0.323 | 3.5 | 3.25 |
| | | LOWK | 0.005 | 0.026 | 0.008 | 0.005 (3) | 0.453 | 0.426 (5) | 0.167 | 4.0 | 0.005 | 0.051 | 0.010 | 0.005 (3) | 0.285 | 0.746 (5) | **0.286** | 4.0 | 4.0 |
| | | TOPK | 0.005 | 0.023 | 0.008 | 0.007 (2) | 0.465 | 0.50 (4) | 0.166 | 3.0 | 0.005 | 0.047 | 0.009 | 0.007 (2) | 0.289 | 0.762 (4) | 0.305 | 3.0 | 3.0 |
| | | FairRec | 0.005 | 0.027 | 0.009 | 0.002 (5) | 0.50 | 0.838 (2) | 0.190 | 3.5 | **0.007** | **0.066** | **0.012** | 0.003 (5) | 0.612 | 0.917 (2) | 0.320 | 3.5 | 3.5 |
| | | FairRecPlus | 0.005 | 0.028 | 0.009 | 0.003 (4) | 0.50 | 0.838 (2) | 0.190 | 3.0 | **0.007** | **0.066** | **0.012** | 0.003 (5) | 0.612 | 0.917 (2) | 0.320 | 3.5 | 3.25 |
| | | ***CAFPR*** | **0.007** | **0.036** | **0.012** | **0.011** (1) | **0.607** | **0.932** (1) | **0.144** | **1.0** | 0.005 | 0.054 | 0.010 | **0.008** (1) | **0.702** | **1.0** (1) | 0.291 | **1.0** | **1.0*** |

*5.4.1. User-side Fairness Results Analysis*

Table 3 shows the user–side performance analysis of our proposed *CAFPR* and the baselines using the *TLR-M* [4] recommendation model. It shows that using *TLR-M* recommendation, our model outperforms the baselines in 20 evaluation scores among 20 scores on user side in top-5 recommendations. Our model outperforms *TOPK* method because *TOPK* method only focuses on user interest without considering POIs capacity. Due to POIs capacity, users may sacrifice their preferences in a real-life scenario. That is why our model performs better in precision@k, recall@k and F1-score@k. Here, *FairRecPlus* [7] is the best baseline and our model outperforms *FairRecPlus* [7] and *FairRec* [6] as well as other baselines. *FairRecPlus* and *FairRec* distribute POI among the users and allow free allocation of up to one product envy. Due to POI's various capacities, envy-free up to one POI does not satisfy all producers. In that case, our proposed capacity-based recommendation gets a real scenario and outperforms user evaluation metrics.

Table 4 shows the user–side performance analysis of our proposed *CAFPR* and the baselines using the*ATST-LSTM* [2] recommendation model. We also see that the proposed model outperforms 14 evaluation metrics among 20 metrics on user side in top-10 recommendations. On the other hand, in *ATST-LSTM* recommendation, our proposed model outperforms only 3 evaluation metrics among 20 metrics on user side in top-5 recommendations and 7 evaluation metrics among 20 metrics in top-10 recommendations. These two recommendation model performances are different because *TLR-M* model focused on user personalised preferences along with spatio-temporal impacts, whereas *ATST-LSTM* focused on only spatio-temporal dependencies. It is common that existing *TOPK* performs well on user-side evaluation metrics. Therefore, our model *CAFPR* outperforms *TOPK*, *FairRec* and *FairRecPlus* when personalised preferences are more significant.

Table 4: Performance analyses between the proposed *CAFPR* and baselines using *ATST-LSTM* recommendation baselines in top-5 and top-10 POI allocation. Here, ↑ and ↓ indicate that high and low values, respectively, are desirable, while numbers in bold indicate the best results and underlined scores the second-best results. Numbers within brackets show the rank based on F1-score and CEFA scores, where 1 = best and 5 = worst.

| | | | topk = 5 | | | | | | | | topk = 10 | | | | | | | | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | User side evaluations | | | | POI side evaluations | | | Rank ↓ | User side evaluations | | | | POI side evaluations | | | Rank ↓ | Rank ↓ |
| | | | Pre ↑ | Re ↑ | F1 ↑ | NDCG ↑ | CFSP ↑ | CEFA ↑ | Gini ↓ | | Pre ↑ | Re ↑ | F1 ↑ | NDCG ↑ | CFSP ↑ | CEFA ↑ | Gini ↓ | | |
| *ATST-LSTM* | California | Random | 0.033 | 0.162 | 0.056 | 0.030 (3) | <u>0.780</u> | <u>0.760</u> (2) | 0.344 | 2.5 | 0.033 | 0.329 | 0.060 | 0.026 (5) | <u>0.80</u> | <u>0.720</u> (2) | 0.352 | 3.5 | 3.0 |
| | | LOWK | 0.035 | 0.175 | 0.058 | 0.028 (4) | 0.200 | 0.667 (3) | <u>0.276</u> | 3.5 | 0.032 | 0.322 | 0.058 | 0.030 (4) | 0.480 | 0.393 (5) | **0.228** | 4.5 | 4.0 |
| | | TOPK | **0.040** | **0.201** | **0.067** | **0.035** (1) | 0.480 | 0.480 (4) | 0.297 | 2.5 | <u>0.038</u> | 0.379 | <u>0.069</u> | <u>0.034</u> (2) | 0.520 | 0.443 (4) | 0.360 | <u>3.0</u> | 2.75 |
| | | FairRec | 0.034 | 0.171 | 0.057 | <u>0.032</u> (2) | **0.880** | **0.780** (1) | 0.346 | <u>1.5</u> | **0.039** | <u>0.391</u> | **0.071** | **0.035** (1) | 0.640 | 0.547 (3) | 0.392 | **2.0** | <u>1.75</u> |
| | | FairRecPlus | 0.034 | 0.173 | 0.058 | <u>0.032</u> (2) | **0.880** | **0.780** (1) | 0.346 | <u>1.5</u> | **0.039** | **0.393** | **0.071** | **0.035** (1) | 0.640 | 0.547 (3) | 0.392 | **2.0** | <u>1.75</u> |
| | | ***CAFPR*** | <u>0.036</u> | <u>0.182</u> | <u>0.061</u> | **0.035** (1) | **0.880** | **0.780** (1) | **0.254** | **1.0** | 0.033 | 0.332 | 0.060 | 0.033 (3) | **0.840** | **0.780** (1) | <u>0.241</u> | **2.0** | **1.5*** |
| | Magic K | Random | 0.029 | 0.147 | 0.049 | <u>0.036</u> (2) | <u>0.752</u> | <u>0.715</u> (2) | 0.339 | <u>2.0</u> | 0.029 | 0.295 | 0.054 | 0.033 (3) | 0.839 | 0.725 (3) | 0.350 | 3.0 | 2.5 |
| | | LOWK | 0.029 | 0.139 | 0.049 | 0.032 (5) | 0.444 | 0.453 (5) | 0.290 | 5.0 | 0.032 | 0.319 | 0.057 | 0.033 (3) | 0.630 | 0.587 (4) | <u>0.300</u> | 3.5 | 4.25 |
| | | TOPK | 0.031 | 0.153 | 0.051 | 0.033 (4) | 0.630 | 0.519 (4) | <u>0.283</u> | 4.0 | 0.032 | 0.320 | 0.058 | <u>0.034</u> (2) | 0.556 | 0.547 (5) | 0.356 | 3.5 | 3.75 |
| | | FairRec | **0.036** | **0.179** | **0.060** | 0.035 (3) | 0.719 | 0.713 (3) | 0.345 | 3.0 | **0.037** | **0.366** | **0.067** | **0.035**(1) | <u>0.869</u> | <u>0.745</u> (2) | 0.345 | <u>1.5</u> | <u>2.25</u> |
| | | FairRecPlus | **0.036** | **0.179** | **0.060** | 0.035 (3) | 0.719 | 0.713 (3) | 0.345 | 3.0 | **0.037** | **0.366** | **0.067** | **0.035**(1) | <u>0.869</u> | <u>0.745</u> (2) | 0.345 | <u>1.5</u> | <u>2.25</u> |
| | | ***CAFPR*** | <u>0.033</u> | <u>0.159</u> | <u>0.053</u> | **0.037** (1) | **0.878** | **0.887** (1) | **0.210** | **1.0** | <u>0.034</u> | <u>0.342</u> | <u>0.062</u> | **0.035** (1) | **0.889** | **0.795** (1) | **0.250** | **1.0** | **1.0*** |
| | Budapest | Random | 0.020 | 0.106 | 0.032 | 0.024 (3) | 0.410 | <u>0.478</u> (2) | 0.333 | <u>2.5</u> | 0.014 | 0.137 | 0.025 | <u>0.024</u> (2) | 0.410 | 0.676 (3) | 0.347 | <u>2.5</u> | <u>2.5</u> |
| | | LOWK | 0.021 | 0.105 | 0.035 | 0.021 (5) | 0.385 | 0.408 (5) | 0.318 | 5.0 | <u>0.025</u> | <u>0.247</u> | <u>0.045</u> | 0.023 (3) | 0.436 | 0.447 (5) | <u>0.323</u> | 4.0 | 4.5 |
| | | TOPK | **0.026** | **0.132** | **0.044** | <u>0.026</u> (2) | <u>0.462</u> | 0.440 (3) | <u>0.298</u> | <u>2.5</u> | 0.024 | 0.242 | 0.044 | <u>0.024</u> (2) | <u>0.487</u> | 0.545 (4) | 0.326 | 3.0 | 2.75 |
| | | FairRec | <u>0.024</u> | <u>0.121</u> | <u>0.040</u> | 0.022 (4) | 0.410 | 0.422 (4) | 0.340 | 4.0 | 0.023 | 0.232 | 0.042 | 0.023 (3) | 0.410 | <u>0.693</u> (2) | 0.341 | <u>2.5</u> | 3.25 |
| | | FairRecPlus | <u>0.024</u> | <u>0.121</u> | <u>0.040</u> | 0.023 (4) | 0.410 | 0.422 (4) | 0.340 | 4.0 | 0.023 | 0.232 | 0.042 | 0.023 (3) | 0.410 | <u>0.693</u> (2) | 0.341 | <u>2.5</u> | 3.25 |
| | | ***CAFPR*** | 0.021 | 0.105 | 0.035 | **0.027** (1) | **0.513** | **0.493** (1) | **0.290** | **1.0** | **0.026** | **0.258** | **0.047** | **0.027** (1) | **0.846** | **0.727** (1) | **0.277** | **1.0** | **1.0*** |
| | Toronto | Random | 0.030 | 0.150 | 0.050 | 0.031 (5) | 0.600 | <u>0.772</u> (2) | 0.353 | 3.5 | 0.028 | 0.279 | 0.051 | 0.028 (4) | 0.600 | **0.814** (1) | 0.334 | 2.5 | 3.0 |
| | | LOWK | 0.032 | 0.159 | 0.053 | 0.034 (4) | 0.200 | 0.439 (4) | 0.564 | 4.0 | 0.030 | 0.296 | 0.054 | 0.032 (3) | 0.433 | 0.506 (4) | 0.427 | 3.5 | 3.75 |
| | | TOPK | **0.037** | **0.185** | **0.062** | <u>0.036</u> (2) | <u>0.633</u> | 0.623 (3) | <u>0.275</u> | 2.5 | <u>0.033</u> | 0.326 | 0.059 | <u>0.033</u> (2) | <u>0.667</u> | <u>0.759</u> (3) | <u>0.282</u> | 2.5 | 2.5 |
| | | FairRec | **0.037** | **0.185** | **0.062** | **0.038** (1) | 0.700 | **0.814** (1) | 0.343 | **1.0** | <u>0.033</u> | <u>0.335</u> | <u>0.061</u> | **0.034** (1) | 0.600 | **0.781** (2) | 0.344 | **1.5** | **1.25*** |
| | | FairRecPlus | **0.037** | **0.185** | **0.062** | **0.038** (1) | 0.700 | **0.814** (1) | 0.343 | **1.0** | <u>0.033</u> | <u>0.335</u> | <u>0.061</u> | **0.034** (1) | 0.600 | **0.781** (2) | 0.344 | **1.5** | **1.25*** |
| | | ***CAFPR*** | <u>0.034</u> | <u>0.172</u> | <u>0.057</u> | 0.035 (3) | **0.900** | **0.814** (1) | **0.265** | <u>2</u> | **0.034** | **0.339** | **0.062** | <u>0.033</u> (2) | **0.900** | **0.814** (1) | **0.266** | **1.5** | <u>1.75</u> |
| | Melbourne | Random | 0.002 | 0.010 | 0.003 | 0.005 (4) | 0.433 | 0.478 (3) | 0.339 | 3.5 | 0.004 | 0.040 | 0.007 | 0.004 (3) | <u>0.612</u> | <u>0.552</u> (2) | 0.334 | 2.5 | 3.0 |
| | | LOWK | 0.004 | 0.020 | 0.007 | 0.004 (5) | 0.471 | 0.466 (4) | <u>0.323</u> | 4.5 | 0.004 | 0.045 | 0.008 | 0.004 (3) | 0.545 | 0.518 (4) | <u>0.327</u> | 3.5 | 4.0 |
| | | TOPK | **0.011** | **0.054** | **0.018** | **0.010** (1) | 0.248 | 0.437 (5) | 0.612 | <u>3.0</u> | **0.008** | **0.084** | **0.015** | <u>0.005</u> (2) | 0.339 | 0.439 (5) | 0.534 | 3.5 | 3.25 |
| | | FairRec | <u>0.010</u> | 0.050 | <u>0.017</u> | <u>0.008</u> (2) | <u>0.517</u> | <u>0.613</u> (2) | 0.351 | **2.0** | <u>0.005</u> | <u>0.054</u> | <u>0.010</u> | **0.008** (1) | 0.574 | 0.536 (3) | 0.355 | <u>2.0</u> | <u>2.0</u> |
| | | FairRecPlus | <u>0.010</u> | <u>0.051</u> | <u>0.017</u> | <u>0.008</u> (2) | <u>0.517</u> | <u>0.613</u> (2) | 0.351 | **2.0** | <u>0.005</u> | <u>0.054</u> | <u>0.010</u> | **0.008** (1) | 0.574 | 0.536 (3) | 0.355 | <u>2.0</u> | <u>2.0</u> |
| | | ***CAFPR*** | 0.005 | 0.025 | 0.008 | 0.007 (3) | **0.529** | **0.687** (1) | **0.335** | **2.0** | 0.004 | 0.041 | 0.007 | <u>0.005</u> (2) | **0.620** | **0.576** (1) | **0.319** | **1.5** | **1.75*** |

#### 5.4.2. POI–side Fairness Results Analysis

Table 3 shows that our proposed model outperforms all baselines on POI–side fairness evaluation metrics in terms of CFSP, CEFA and Gini. The CFSP and CEFA are capacity-based evaluations, while Gini relates to POI exposures without capacity. In *TLR-M* recommendation model, our proposed *CAFPR* model outperforms 13 evaluation metrics among the 15 evaluations in top-5 values and 12 metrics among 15 metrics in top-10 values in POI evaluation metrics. Besides this, using *ATST-LSTM* recommendation, our allocation model outperforms 15 evaluation metrics and 14 evaluation metrics, respectively, in top-5 and top-10 values among 15 evaluation metrics in provider evaluation metrics. In *TLR-M*, the CFSP evaluation score is maximum 0.926 and minimum 0.670 in top-5 values and maximum 0.897 and minimum 0.702 in top-10 values. The proposed *CAFPR* outperforms the best in five datasets. The main reason for this performance is capacity diversity. If the dataset's POIs capacity is not same, our model performs well. If all POIs capacity is the same, our model and *FairRecPlus* performance will be the same. However, in the real-world scenario, POI capacity is different. Thus our model performs better than the *FairRecPlus* model. It has been seen that *CAFPR* outperforms all other baselines in CEFA evaluation metrics. The main reason is that our model solves the overcrowding and user scarcity problems.

Although *FairRecPlus* solves the overcrowding problem, it cannot solve capacity-based user scarcity problems. In that case, high capacity-based POIs get similar user distribution to low capacity-based POIs. In that case, POIs with high capacity receive a similar user distribution to low-capacity POIs, which causes high-capacity POIs to face scarcity problems and encounter difficulties with running their business. Furthermore, on the Gini, *LOWK* sometimes performs well because it depends only on POI exposure while ignoring user interest. Our model performs well on this metric for two reasons: the capacity-based allocation model distributes users based on the POI capacity ratio, meaning that POIs receive sufficient users to meet their minimum under-demand and envy capacity threshold values. Thus, POIs get sufficient users to fulfill their minimum under-demand and envy capacity threshold value. The *TOPK* model produces the worst Gini evaluation results because it only focuses on user interests. Finally, we conclude that our proposed model outperforms all baselines.

### *5.4.3. Trade–off between User Utility and POI Fairness*

We propose rank evaluation based on F1-score rank from the user side and CEFA rank from the POI side to show the impacts of two sides' fairness in one evaluation metric. We then take the average rank score, which reveals that our model rank is best than the baselines. It has been shown that in the maximum case, our proposed model outperforms the baselines in terms of the Rank evaluation metric. Except for the Toronto and Melbourne datasets' top-10 Rank values, all cases attain rank 1 values. The last column of Table 3 indicates the average rank of the top 5 and top 10 ranks. Here, values in bold that are marked with a star (*) indicate that our model has outperformed the baselines. We have also verified our model using the *ATST-LSTM* recommendation model; the results show that our *CAFPR* model outperforms the baselines on most of the evaluation metrics. We present the *ATST-LSTM* recommendation–based results in Table 4 based on $min_{dem} = 0.7$ and $Envy_{cap} = 0.3$.

## *5.5. Parameter $min_{dem}$ Value Impact Analysis*

The two charts in Figure 7 show the impacts of $min_{dem}$ where $Envy_{cap}$ is fixed (0.3). The user-side evaluation F1-score does not change substantially as this parameter changes. The parameter $min_{dem}$ indicates the number of envy pairs of POIs in the model. The results show that if $min_{dem}$ is large, the envy score is higher because POIs are not happy if they do not receive their capacity-based user distributions. However, if $min_{dem}$ is small, this means that POIs agree to compromise their utilisation; in this case, most of the models get high values. However, top-k allocation obtains a maximum of 85% envy-free allocation when POIs are flexible and require only 10% capacity to solve the under-demand problem. This shows that our model's F1–score is always better than that of the baselines.

## *5.6. Parameter $Envy_{cap}$ Value Impact Analysis*

Figure 7 shows $Envy_{cap}$ impacts where $min_{dem}$ is fixed and it is 0.7. Our proposed model gets the better results for all value changes and *FairRec* and *Random* also obtain the highest POI exposure at high values. However, *TOPK* allocation obtains a maximum of

80% envy-free allocation when POIs are flexible and require only 10% capacity to solve the under-demand problem. It shows that our model F1-Score is always good than the baselines.

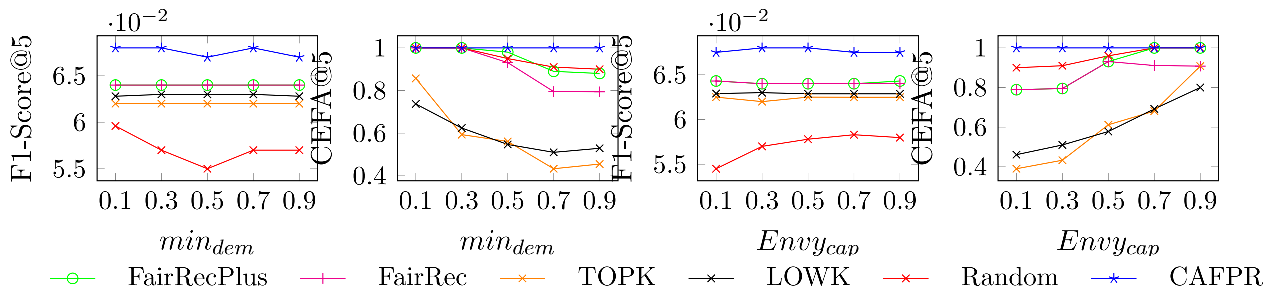


Figure 7: Impacts of parameters $min_{dem}$ and $Envy_{cap}$ in Magic Kingdom dataset.

### *5.7. Execution Time Comparison*

Figure 5 shows the execution time (s) of our proposed model and baselines in top-10 allocation. Our proposed model is faster than the existing *FairRec* and *FairRecPlus* models. The main reason is that the existing model used round robin technique to distribute users among the POIs. It takes extra time compared to our proposed model. Our proposed model takes a little bit extra time than TOPK, Randam and LOWK because these models allocate users only once; there are no re-allocation strategies. However, our proposed *CAFPR* model requires user re-allocation sometimes due to POI capacity limits. Finally, our proposed model performs better than the baselines and execution time is significantly comparable with existing *FairRec* and *FairRecPlus* models.

Table 5: Execution time (s) performance analyses between the proposed *CAFPR* and baselines using *TLRM* recommendation baselines in top-10 POI allocation.

| | Dataset | Random | LOWK | TOPK | FairRec | FairRecPlus | ***CAFPR*** |
|---|---|---|---|---|---|---|---|
| *TLR-M* | California | 45.05 | 43.37 | 43.64 | 184.91 | 185.15 | 45.49 |
| | Magic K | 82.69 | 77.66 | 75.73 | 480.99 | 482.37 | 78.77 |
| | Budapest | 7.11 | 6.95 | 6.72 | 15.37 | 15.38 | 6.96 |
| | Toronto | 10.91 | 11.69 | 10.91 | 16.21 | 16.22 | 11.45 |
| | Melbourne | 7.59 | 7.27 | 7.65 | 16.46 | 16.78 | 8.00 |

## 6. Conclusion

Capacity-aware fair recommendation is a significant and interesting research direction in the field of location-based recommendation systems. This model takes into account not only user preferences but also the capacity of POIs such as restaurants, museums, or parks. In our proposed model, Capacity-Aware Fair POI Recommendation System, we aim to provide personalized recommendations that are not only relevant but also fair and sustainable. In this paper, we introduce capacity-aware fair top-k POI recommendations considering user satisfaction and POI allocation. We apply a deep learning model that simultaneously incorporates recommendation and allocation in one framework. We use two deep learning models, ATST-LSTM and TLR-M, for initial demand mining. Then, we propose an over-demand cut policy and an under-demand add policies that ensure capacity-based envy-free allocation for fair allocation. We evaluate our experiments based on user-side metrics and POI-side metrics. The experimental results show that our proposed CAFPR model outperforms all baseline models in five real datasets.

In this paper, our focus is primarily on individual personalized interests. However, we acknowledge that when visitors are in a group or family setting, their preferences and recommendations may vary. As part of our future work, we aspire to explore the concept of user group happiness and develop an approach to construct comprehensive itineraries that cater to the diverse preferences, budget constraints, and time limitations of both individuals and groups. By optimizing both individual and group satisfaction, we aim to provide a holistic recommendation system that ensures an enjoyable and fulfilling experience for all users involved.

## CRediT authorship contribution statement

**Sajal Halder:** Conceptualization, Methodology, Investigation, Software, Formal analysis, Writing - Original Draft preparation. **Kwan Hui Lim:** Methodology, Validation, Writing - Review & Editing. **Jeffrey Chan:** Methodology, Validation, Writing - Review & Editing, Supervision. **Xiuzhen Zhang:** Methodology, Validation, Writing - Review &

Editing, Supervision.

## References


[1] Kwan Hui Lim, Jeffrey Chan, Shanika Karunasekera, and Christopher Leckie. Tour recommendation and trip planning using location-based social media: a survey. *Knowledge and Information Systems*, pages 1247–1275, 2018.

[2] Liwei Huang, Yutao Ma, Shibo Wang, and Yanbo Liu. An attention-based spatiotemporal LSTM network for next POI recommendation. *IEEE Transactions on Services Computing*, pages 1585–1597, 2021.

[3] Hossein A. Rahmani, Mohammad Aliannejadi, Mitra Baratchi, and Fabio Crestani. Joint geographical and temporal modeling based on matrix factorization for point-of-interest recommendation. In *European Conference on Information Retrieval*, pages 205–219. 2020.

[4] Sajal Halder, Kwan Hui Lim, Jeffrey Chan, and Xiuzhen Zhang. Transformer-based multi-task learning for queuing time aware next POI recommendation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 510–523. 2021.

[5] Qianyu Guo and Jianzhong Qi. Sanst: A self-attentive network for next point-of-interest recommendation. *arXiv e-prints*, pages arXiv–2001, 2020.

[6] Gourab K Patro, Arpita Biswas, Niloy Ganguly, Krishna P Gummadi, and Abhijnan Chakraborty. Fairrec: Two-sided fairness for personalized recommendations in two-sided platforms. In *The World Wide Web Conference*, pages 1194–1204, 2020.

[7] Arpita Biswas, Gourab K Patro, Niloy Ganguly, Krishna P Gummadi, and Abhijnan Chakraborty. Toward fair recommendation in two-sided platforms. *ACM Transactions on the Web*, pages 1–34, 2021.

[8] Yunpeng Li, Yichuan Jiang, Weiwei Wu, Jiuchuan Jiang, and Hui Fan. Room allocation with capacity diversity and budget constraints. *IEEE Access*, pages 42968–42986, 2019.

[9] Pak Hay Chan, Xin Huang, Zhengyang Liu, Chihao Zhang, and Shengyu Zhang. Assignment and pricing in roommate market. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 446–452, 2016.

[10] Sajal Halder, Kwan Hui Lim, Jeffrey Chan, and Xiuzhen Zhang. Poi recommendation with queuing time and user interest awareness. *Data Mining and Knowledge Discovery*, pages 2379–2409, 2022.

[11] Haochao Ying, Jian Wu, Guandong Xu, Yanchi Liu, Tingting Liang, Xiao Zhang, and Hui Xiong. Time-aware metric embedding with asymmetric projection for successive poi recommendation. *World Wide Web*, pages 2209–2224, 2019.

[12] Zhiyuan Zhang, Yun Liu, Zhenjiang Zhang, and Bo Shen. Fused matrix factorization with multi-tag, social and geographical influences for poi recommendation. *World Wide Web*, pages 1135–1150, 2019.

[13] Fan Zhou, Xiaoli Yue, Goce Trajcevski, Ting Zhong, and Kunpeng Zhang. Context-aware variational trajectory encoding and human mobility inference. In *The World Wide Web Conference*, pages 3469–3475, 2019.

[14] Fan Zhou, Ruiyang Yin, Kunpeng Zhang, Goce Trajcevski, Ting Zhong, and Jin Wu. Adversarial point-of-interest recommendation. In *The World Wide Web Conference*, pages 3462–34618, 2019.

[15] Sajal Halder, Kwan Hui Lim, Jeffrey Chan, and Xiuzhen Zhang. Efficient itinerary recommendation via personalized poi selection and pruning. *Knowledge and Information Systems*, pages 963–993, 2022.

[16] Kwan Hui Lim, Jeffrey Chan, Shanika Karunasekera, and Christopher Leckie. Personalized itinerary recommendation with queuing time awareness. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 325–334, 2017.

[17] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. Predicting the next location: a recurrent model with spatial and temporal contexts. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 194–200, 2016.

[18] Yingxue Ma and Mingxin Gan. Exploring multiple spatio-temporal information for point-of-interest recommendation. *Soft Computing*, pages 18733–18747, 2020.

[19] Meihui Shi, Derong Shen, Yue Kou, Tiezheng Nie, and Ge Yu. Attentional memory network with correlation-based embedding for time-aware poi recommendation. *Knowledge-Based Systems*, pages 106747–106757, 2021.

[20] Lei Guo, Yufei Wen, and Fangai Liu. Location perspective-based neighborhood-aware poi recommendation in location-based social networks. *Soft Computing*, pages 11935–11945, 2019.

[21] Buru Chang, Yonggyu Park, Donghyeon Park, Seongsoon Kim, and Jaewoo Kang. Content-aware hierarchical point-of-interest embedding model for successive poi recommendation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3301–3307, 2018.

[22] Shanshan Feng, Lucas Vinh Tran, Gao Cong, Lisi Chen, Jing Li, and Fan Li. Hme: A hyperbolic metric embedding approach for next-poi recommendation. In *Proceedings of the 43rd International ACM SIGIR*, pages 1429–1438, 2020.

[23] Yuxia Wu, Ke Li, Guoshuai Zhao, and Xueming Qian. Personalized long-and short-term preference learning for next poi recommendation. *IEEE Transactions on Knowledge & Data Engineering*, pages 1944–1957, 2022.

[24] Dichao Li and Zhiguo Gong. A deep neural network for crossing-city poi recommendations. *IEEE Transactions on Knowledge & Data Engineering*, pages 3536–3548, 2022.

[25] Zheng Dong, Xiangwu Meng, and Yujie Zhang. Exploiting category-level multiple characteristics for poi recommendation. *IEEE Transactions on Knowledge & Data Engineering*, pages 1488–1501, 2023.

[26] Carl Yang, Lanxiao Bai, Chao Zhang, Quan Yuan, and Jiawei Han. Bridging collaborative filtering and semi-supervised learning: a neural approach for poi recommendation. In *23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1245–1254, 2017.

[27] Suhang Wang, Yilin Wang, Jiliang Tang, Kai Shu, Suhas Ranganath, and Huan Liu. What your images reveal: Exploiting visual contents for point-of-interest recommendation. In *Proceedings of the 26th International Conference on World Wide Web*, pages 391–400, 2017.

[28] Guangyao Pang, Xiaoming Wang, Fei Hao, Liang Wang, and Xinyan Wang. Efficient point-of-interest recommendation with hierarchical attention mechanism. *Applied Soft Computing*, pages 106536–106548, 2020.

[29] José Ruiz-Meza, Julio Brito, and Jairo R Montoya-Torres. A grasp-vnd algorithm to solve the multi-objective fuzzy and sustainable tourist trip design problem for groups. *Applied Soft Computing*, pages 109716–109728, 2022.

[30] Lianyong Qi, Yuwen Liu, Yulan Zhang, Xiaolong Xu, Muhammad Bilal, and Houbing Song. Privacy-aware point-of-interest category recommendation in internet of things. *IEEE Internet of Things Journal*, 9(21):21398–21408, 2022.

[31] Yuwen Liu, Huiping Wu, Khosro Rezaee, Mohammad R Khosravi, Osamah Ibrahim Khalaf, Arif Ali Khan, Dharavath Ramesh, and Lianyong Qi. Interaction-enhanced and time-aware graph convolutional network for successive point-of-interest recommendation in traveling enterprises. *IEEE Transactions on Industrial Informatics*, 19(1):635–643, 2022.

[32] Anikó Hannák, Claudia Wagner, David Garcia, Alan Mislove, Markus Strohmaier, and Christo Wilson. Bias in online freelance marketplaces: Evidence from taskrabbit and fiverr. In *Proceedings of the 2017 ACM conference on computer supported cooperative work and social computing*, pages 1914–1933, 2017.

[33] Max Hort, Rebecca Moussa, and Federica Sarro. Multi-objective search for gender-fair and semantically correct word embeddings. *Applied Soft Computing*, pages 109916–109928, 2023.

[34] Matthew J Salganik, Peter Sheridan Dodds, and Duncan J Watts. Experimental study of inequality and unpredictability in an artificial cultural market. *Science*, pages 854–856, 2006.

[35] Himan Abdollahpouri, Gediminas Adomavicius, Robin Burke, Ido Guy, Dietmar Jannach, Toshihiro Kamishima, Jan Krasnodebski, and Luiz Pizzato. Multistakeholder recommendation: Survey and research directions. *User Modeling and User-Adapted Interaction*, pages 127–158, 2020.

[36] Dimitris Serbos, Shuyao Qi, Nikos Mamoulis, Evaggelia Pitoura, and Panayiotis Tsaparas. Fairness in package-to-group recommendations. In *Proceedings of the 26th International Conference on World Wide Web*, pages 371–379, 2017.

[37] Christian Bommer, Axel Dreher, and Marcello Perez-Alvarez. Home bias in humanitarian aid: The role of regional favoritism in the allocation of international disaster relief. *Journal of Public Economics*, pages 104604–104619, 2022.

[38] Xue Xiao, Lu Liu, Liangyuan Xu, Lisha Liu, Chuansheng Chen, and Yanfang Li. Group bias in children's merit-based resource allocation. *Journal of Experimental Child Psychology*, pages 104660–104674, 2019.

[39] Mingyang Li and Zequn Wang. Active resource allocation for reliability analysis with model bias correction. *Journal of Mechanical Design*, pages 051403–051415, 2019.

[40] Tom Sühr, Asia J Biega, Meike Zehlike, Krishna P Gummadi, and Abhijnan Chakraborty. Two-sided fairness for repeated matchings in two-sided markets: A case study of a ride-hailing platform. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3082–3092, 2019.

[41] Abhijnan Chakraborty, Aniko Hannak, Asia J Biega, and Krishna Gummadi. Fair sharing for sharing economy platforms. In *Fairness, Accountability and Transparency in Recommender Systems-Workshop on Responsible Recommendation*, pages 1–4, 2017.

[42] Sajal Halder, Kwan Hui Lim, Jeffrey Chan, and Xiuzhen Zhang. Deep learning of dynamic poi generation and optimisation for itinerary recommendation. *In Submission*, 2023.

[43] Corrado Gini. *Gini Index*, pages 231–233. 2008.