ComfRide: A Smartphone based System for Comfortable Public Transport Recommendation

Rohit Verma, Surjya Ghosh, Mahankali Saketh, Niloy Ganguly, Bivas Mitra, Sandip Chakraborty
Indian Institute of Technology, Kharagpur, INDIA 721302
{rohit.verma,surjya.ghosh}@iitkgp.ac.in,m.saketh96@gmail.com,{niloy,bivas,sandipc}@cse.iitkgp.ernet.in

ABSTRACT

Passenger comfort is a major factor influencing a commuter's decision to avail public transport. Existing studies suggest that factors like overcrowding, jerkiness, traffic congestion etc. correlate well to passenger's (dis)comfort. An online survey conducted with more than 300 participants from 12 different countries reveals that different personalized and context dependent factors influence passenger comfort during a travel by public transport. Leveraging on these findings, we identify correlations between comfort level and these dynamic parameters, and implement a smartphone based application, *ComfRide*, which recommends the most comfortable route based on user's preference honoring her travel time constraint. We use a '*Dynamic Input/Output Automata*' based composition model to capture both the wide varieties of comfort choices from the commuters and the impact of environment on the comfort parameters. Evaluation of *ComfRide*, involving 50 participants over 28 routes in a state capital of India, reveals that recommended routes have on average 30% better comfort level than Google map recommended routes, when a commuter gives priority to specific comfort parameters of her choice.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Human-centered computing** → *Smartphones*;

KEYWORDS

Route recommendation; City transports; Dynamic Input/Output Automata

ACM Reference Format:
Rohit Verma, Surjya Ghosh, Mahankali Saketh, Niloy Ganguly, Bivas Mitra, Sandip Chakraborty. 2018. ComfRide: A Smartphone based System for Comfortable Public Transport Recommendation. In *Twelfth ACM Conference on Recommender Systems (RecSys '18), October 2–7, 2018, Vancouver, BC, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3240323.3240359

1 INTRODUCTION

Existing studies have shown that multiple factors, such as the nature of the road, bus condition, traffic congestion, driving pattern, overcrowding, waiting time at different bus stoppages etc. significantly influence individual commuter's preference while traveling in a public city bus [14, 31]. Findings of a commuter pain survey conducted by IBM (2011) show that deterioration of such factors increases stress and anger among commuters [1]; in a separate study on New Delhi, commuters desire that safety and individual preferences be honored in public transport [21]. The poor conditions get exacerbated in developing countries where in addition to infrastructural limitations like poor road conditions, unplanned creation of bus stoppages and absence of lanes, public buses daily witness sights of overcrowding and chaotic behavior of the bus drivers [7, 16, 20, 27, 35].

This background motivates us to explore the possibility of designing a low-cost smartphone-based personalized route recommender application, which takes into account individual commuter comfort while recommending. Majority of the existing mobile based transit recommender systems, such as [2, 3, 11, 12, 18, 19] provide recommendation based on the fare and the travel time optimization; individual preferences which impact on-route comfort is still a largely unexplored area. In this paper, we propose *ComfRide* – an end-to-end smartphone based personalized bus route recommender system, which recommends the most preferable bus route based on commuters' individual comfort preferences (§ 6.4). It captures diverse features through smartphones, which impact the commuter's comfort level while using public transport and develops a personalized route recommender employing the fuzzy set theory along with TOPSIS approach [28], which considers individual comfort level based on spatio-temporal road and route characteristics. Besides this diverse and wide suite of factors across various possible routes, single or multiple breakpoints in the journey is also considered, if that increases comfort.

In order to capture the spatio-temporal dynamics over a wide choice of features, we design the route recommendation algorithm using a specialized compositional model, called *Dynamic Input/Output Automata* (DIOA) [4]. The DIOA ensures that the system is not overwhelmed with the huge amount of data to be processed thus reducing the load on the system. Effectively, *ComfRide* utilizes DIOA based compositional model to identify the most preferable route efficiently based on the historical information and the context of the query. Moreover, the DIOA also provides a mechanism to dynamically modify the model to suit the personalized preference of a commuter on obtaining a feedback from her after a trip, thus improving the quality of recommendation after every trip. An extensive deployment of *ComfRide* involving 50 volunteers for a period of two years (at weekday, weekend; at 3 different time periods) over 28 routes in a state capital of India reveals that it can recommend routes with on average 30% better comfort level than the ones proposed by Google Maps as well as other baselines.

2 RELATED WORK

In this section, we present a brief survey of the existing literature, broadly focusing on two aspects related to our work – (a) comfort features and their measurements and (b) personalized transit route recommender systems.

Comfort features and measurement of comfort: The relative importance of comfort features is a subjective issue as it depends on personal, regional and socio-economic attributes. Several works done to understand comfort features in public transport suggest that the choice of a bus route is affected by comfort levels [20, 24, 34, 35], and comfort further depends on roughly two aspects, viz, vehicle performance related features, such as vibrations, jerk etc. [9, 30], and bus-operating environment related features, like over-crowding, travel-time and so on [20, 22, 32, 35]. In most of the transport research literature, such as [13, 32, 34] and the references therein, comfort is measured via personal interviews, which is time-consuming and labor-intensive, hence lacks scalability and timeliness. Recently, participatory sensing (crowdsensing) using smartphone apps (like CMS [25], RESen [33], CommuniSense [29], UrbanEye [38]) use a new sensing paradigm, called war-driving, where volunteers contribute their phones' sensor data that can be used to analyze comfort levels.

Personalized transit recommendation: Several public bus trip planner systems are available, like the Google Transit [17], TRipGo [3], GOTransit [2] etc., which give information on alternative routes, fares, schedules, as well as map-based visualization of the real time traffic information. Further, some of the personalized bus route recommendation systems like MetroCognition [5], PATRASH [26] etc. rely on specialized cards [18], war driving [38] or explicit user feedback [5] for route database generation. The existing personalized route recommender systems and tour planners, like PaRE [23], FAVOUR [8], Routeme [19], Feeder [40], Treads [15] etc. are mostly developed for private cabs and taxis and do not consider the wide sets of comfort parameters as experienced during public bus travels in our daily life, along with the dynamics of the environments as well as personalized choices of the commuters.

3 MOTIVATIONAL STUDY

In this section, we discuss the findings that motivate us to develop *ComfRide*. Our findings are based on the analysis of Google Transit data and a world-wide user survey conducted involving 300 public transport commuters.

3.1 Google Transit Data Analysis

First, we explore the number of possible routes between a source-destination pair and their features. For this purpose, we select all the country capitals of the world, where Google Transit information is available. We randomly select 50 locations in these cities and find number of bus routes between all the possible pairs. We use Google Directions API to get this information. Figure 1 shows the distribution of the percentage of number of routes between a source-destination pair. We observe that more than 60% of the pairs have at least 4 bus routes between them and close to 25% have more than 8 bus routes. We also calculate the results separately for developed and developing countries and it can be seen that the result is almost similar in both cases.

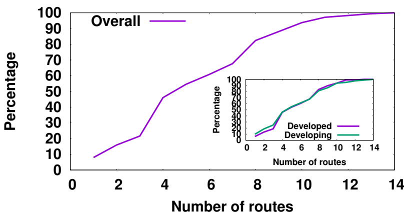


Figure 1: Cumulative Distribution of number of bus routes between random (source, destination) pairs calculated for all the capital cities of the world where Google Transit information is available. Inset: Same result shown for developed and developing countries.

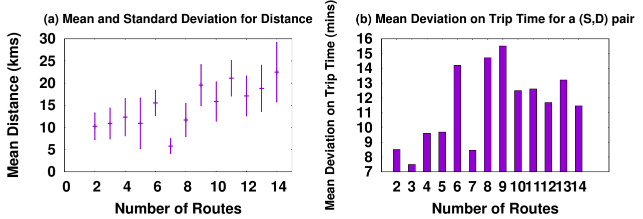


Figure 2: a) Mean and standard deviation over distance calculated over source, destination pairs having similar number of routes. (b) Mean standard deviation of time duration calculated over all the routes between a given source, destination pair for a given number of routes.

Although an alternate route may be beneficial in terms of parameters like being less crowded, but it may not be feasible if the travel time is too high compared to the shortest route. To explore whether the alternate route choices are feasible, we randomly select source-destination pairs having different number of routes between them. For each of these pairs, we measure the difference in distance traveled and travel time in Figure 2. Figure 2(a) shows that for every type of route, there is a considerable variation in the trip distance. It also demonstrates that the trend is similar for both long and short trips. Selection of an alternate route is also driven by the fact that it does not consume high travel time. For this purpose, we find the standard deviation of travel time over all the routes for a given source-destination pair. We group these by the number of routes and calculate the mean as shown in Figure 2(b). We observe that for none of the routes the mean value of standard deviation in travel time exceeds 15 minutes, which we assume is acceptable by a commuter based on the survey discussed as follows.

3.2 Commuter Survey

We have conducted a large scale online survey involving more than 300 participants across the globe, who mostly access the public transport system. Responses were obtained from different developed and developing countries including Italy, UK, Netherlands, Norway, Germany, France, USA, India, Nepal, Iran, Pakistan, Sri Lanka and Vietnam. Around 25% participants are female and are aged between 20 to 65 years. In our survey, we design the questionnaire to assess factors like *total travel time*, *traffic congestion*, *sitting probability*, *road condition*, *bus type* (AC or non-AC buses), *number*

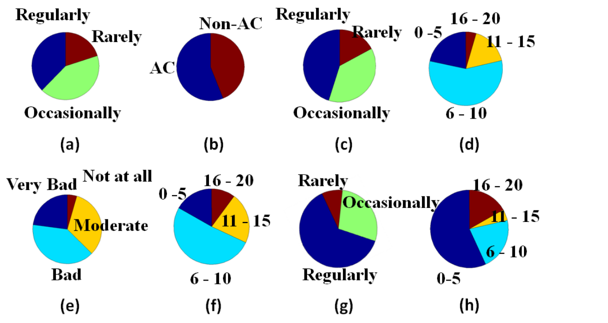


Figure 3: Commuter Survey Results: (a) Delay due in reaching, (b) Type of Bus, (c) Travel without seat, (d) Compromise Travel Time to get a seat, (e) Travel on bad roads, (f) Compromise Travel Time for better road, (g) Stuck in Traffic, (h) Compromise Travel Time for less congested route (time units are in minutes)

of stops, break journey, which reflect factors regulating commuter comfort and their interplay for a comfortable journey.

Table 1: Country-wise Responses to issues faced by commuters (All values are in percentage; Re: Regularly, O: Occasionally, R: Rarely, VB: Very Bad, B: Bad, M: Moderate, N: None)

Countries	Delay in Reaching			No seat			Bad Road				Traffic Jam		
	Re	O	R	Re	O	R	VB	B	M	N	Re	O	R
India	40.7	43.5	15.8	47.8	37.2	15	24.9	42.3	30	2.8	67.6	26.5	5.9
Nepal	80	20	0	40	60	0	40	20	40	0	60	40	0
Iran	12.5	62.5	25	62.5	25	12.5	0	12.5	75	12.5	62.5	37.5	0
Sri Lanka	66.7	16.7	16.6	83.3	16.7	0	16.7	33.3	50	0	83.3	16.7	0
USA	0	42.9	57.1	14.2	42.9	42.9	14.3	42.9	28.6	14.3	28.6	14.3	57.1
France	0	25	75	75	25	0	0	75	25	0	25	25	50

In a nutshell, this survey reveals that there exist multiple parameters (Figure 3), viz., speed of the bus, congestion, probability of getting a seat, etc, which impact commuters' comfort during the trip, and individuals differ widely in their perception of comfort primarily depending on their age and gender. Moreover, the trend is quite similar all over the world (Table 1).

3.3 Summary

We summarize the findings below from this study. (a) Among the various routes available in between a source-destination pair, traveling on an alternate route apart from the minimum distance often provides benefit in terms of comfort choices, but finding the best one manually may not be feasible. (b) There are different factors influencing commuter comfort on a public bus, which are more pronounced in developing countries. These findings call for developing a public transport recommender application which can identify the most preferable route for a commuter based on her preferences. Existing studies show that different sensors present in a smartphone, like accelerometer, gyroscope, compass, GPS, sound sensors etc., are sufficient enough to effectively determine various road and route features and to correlate them with the commuter comfort [5, 10, 33, 36, 39]. By utilizing such crowdsourced features from a smartphone, we propose *ComfRide* to recommend the most comfortable route based on the personalized preferences of a commuter.

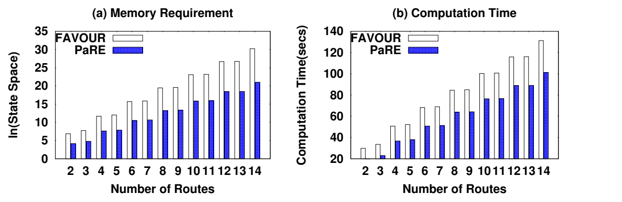


Figure 4: Impact of increase in the number of routes on the memory requirement and computation time of the two approaches (a) Natural logarithm of the average state space varying with number of routes for both the algorithms. (b) Variation of average computation time when each of the two algorithms are executed.

4 CHALLENGES INVOLVED

Development of a recommendation system considering various on-road diversities and personalized choices of commuters has multiple challenges, as discussed in this section.

4.1 Developing a Personalized System

Users may have personalized choices towards finding out the most comfortable route. While one user may prefer a route with minimum travel time even if the bus is crowded; a second user may prefer a route that is not crowded and therefore maximizes the possibility of getting a seat. The system should adopt to the personalized choices of the users while making a recommendation.

4.2 Optimizing Memory & Computation Time

The general approach of addressing a route recommendation problem would be to convert it into a graph-based problem and run queries on the graph. A graph-based approach for route recommendation has following challenges.

4.2.1 ***Generation of Complete Network Graph.*** The direct approach would be to store the complete route data as a graph, and then any query is processed on this graph. FAVOUR [8] is a route recommender system for private vehicles based on this approach. Considering the factors like (a) possibility of multiple feasible routes between a source-destination pair, (b) possibility of opting for one or more breakpoints during the journey for improving comfort choice, and (c) users' personalized choices on route parameters (like travel time, road condition, crowdedness), the state space of the graph can increase exponentially. To observe this, we have executed the FAVOUR algorithm [8] on the public bus transit routes obtained through Google Directions API. Considering the plot given in Figure 4(a), it is evident that an increase in the number of routes increases the average state space exponentially. In addition to increasing the memory requirement, this also increases the computation time as seen in Figure 4(b). The average computation time increases to more than a minute (over a system with Intel(R) Xeon(R) E5-2620 v3 @ 2.40GHz CPU, 32GB memory, Debian 9.3) when 6 routes are possible, and becomes double when the number of possible routes is 14. A high response time is not suitable for a mobile based system.

4.2.2 ***Generating a Query-based Sub-graph.*** Another approach would be to not generate the complete graph at one go; instead whenever a query comes, a graph is generated based on the query information. PaRE utilizes such approach for query execution for route recommendation based on historical trajectory data [23]. Employing such an approach reduces the state space as shown in Figure 4(a) (from similar experiments over the routes from Google Directions API as mentioned earlier). However, the computation time is not decreased considerably pertaining to the overhead of generation of the query graph for each query. Here we observe that the average computation time reaches close to 2 minutes for 14 possible routes.

5 COMFRIDE SYSTEM

Developing an end-to-end system requires tackling the previously mentioned technical challenges efficiently. In this section we first discuss how we plan to overcome the challenges and then give an overview of the *ComfRide* system.

5.1 Mitigating the Challenges

Based on previous discussion, we need an approach which optimizes both memory and graph generation time. A natural choice for this is to remember the relevant information based on the context of a query (like the time of the day when a query is fired) and prune the nodes which are irrelevant to the context. For example, a route may remain very crowded during the evening, and therefore, if a user asks for a less crowded route with a budget on travel time during the evening, the crowded routes may be pruned a priori based on the memory and the context.

To utilize such historical patterns and the context of a query, we use DIOA [4] as a specialized data structure. A DIOA is a state machine where any transition is linked to some named actions or signatures. The signatures are programs or algorithms that can be executed dynamically on the fly to change the structure of the automata. These signatures can be both to interact with the environment (external entities like some other automata) for extracting the historical patterns and the context information (called *input* or *output* signatures), or to perform local tasks (*internal* signatures). As an example, the input signature of an automata can provide historical patterns (like the statistics of a route) and the context (like time of the day), whereas the internal signatures can prune out unnecessary nodes from the graph and find out the best possible route according to commuter's choice. As shown in Figure 5, the input is received from the user as well as from the database that contains historical information, and the system triggers various *internal* signatures. These *internal* signatures apply filters based on the context of the query as defined above. Routes passing all these filters are only considered and hence the final graph generation is fast. The dynamic nature ensures that the automata can be updated anytime during the execution based on the signatures. Moreover, this ensures that a copy of the automata can be generated in parallel, when the automata is already being used by some other signature, and a separate query can be run on this copy.

The dynamic nature of DIOA also helps in designing a personalized system based on user feedback as an input signature, as shown in Figure 5. Once a commuter experiences a travel based on the recommendation of the system, she has the option to provide a feedback to the system. This feedback can be processed by the automata signatures for embedding the personalized choice of the commuter inside the internal signatures.

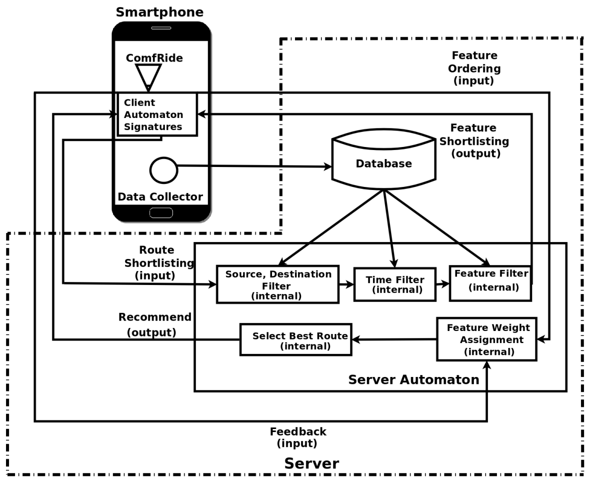

Smartphone
ComfRide
Client Automaton Signatures
Data Collector
Feature Ordering (input)
Feature Shortlisting (output)
Database
Route Shortlisting (input)
Source, Destination Filter (internal)
Time Filter (internal)
Feature Filter (internal)
Recommend (output)
Select Best Route (internal)
Feature Weight Assignment (internal)
Server Automaton
Feedback (input)
Server

Figure 5: System Architecture

5.2 System Overview

The overall system can be divided into two broad modules (Figure 5) – (a) the client module (the smartphone app) which runs the client automaton and (b) the server module (runs over a remote server) that runs the server automaton with various filters as internal signatures. Additionally, *ComfRide* uses a database to store the historical information that are used by the server module to apply various filters using the internal signatures.

5.3 Database Generation

The rich database is the core component of *ComfRide*. The database contains various road and route related features that influence the comfort level of a commuter. This database is stored at the *ComfRide* server and queried during the route recommendation. In *ComfRide*, a bus route is divided into multiple segments, and the comfort features are computed for individual segments at different times of the day to capture the spatio-temporal diversity of feature values over the route. The data collection module of the *ComfRide* app runs as a background process that collects various sensor data periodically from the smartphone sensors, like accelerometer, gyroscope, compass, periodic GPS, sound sensors etc. leveraging on the exiting crowdsensing techniques [5, 33, 37]. From these sensor data, we extract various *Point of Concerns* (PoC) that can impact travel choice, like sharp turn, speed breakers, bad road patches, congested roads and crowded route segments. Existing methodologies from the literature [29, 33, 36, 38] have been utilized for this purpose. Based on the survey conducted on general commuters (§ 3), we primarily focus on following features that can impact users' choice in selecting a bus route. These features are classified into two categories. (a) *Segment specific features:* Features like (i) *average speed of a bus from one stoppage to the next stoppage*, (ii) *crowding*, (ii) *probability of skipping a bus stop*, (iii) *average stopping time at a bus stop*, (iv) *jerkiness* and (v) *speed before a PoC* would vary between route segments; hence are calculated separately for every segment.

These features also vary widely at different times of the day; so, we capture temporal distribution of the feature values by dividing the day into multiple time zones. (b) *Route specific features:* Features like the distribution of PoCs over a bus route are evaluated over the complete route.

The database of *ComfRide*, populating the aforementioned features, consists of following two major tables, *Route Table* and *Feature Table*. The *Route Table*, indexed by the bus route information (route name & terminal stops), stores the *route specific features*. The *Feature Table* is indexed by the segment information (the delimiter bus stops & corresponding route) and stores the respective *segment specific features*. Notably, all the features are stored in the database with the respective time zones.

6 ROUTE RECOMMENDATION USING DIOA

We construct the signatures of the DIOA for *ComfRide*, that executes a query based on the information available at the database, context of the query and personalized choices of the user. During the query execution, the user provides the source-destination pair and her personalized choices on the query features through the *ComfRide* app, which is executed at the server based on the DIOA signatures.

6.1 ComfRide DIOA

The *ComfRide* DIOA consists of two components - the client automaton (runs at the Smartphone app) and the server automaton (runs at the processing server), as shown in Figure 5. Figure 6 gives the states, signatures and transition details for the client and the server automata. The broad features of *ComfRide* DIOA are as follows. At the beginning of a travel, the client sends a query to the server. Based on the source (s)-destination (d) pair in the query, the server extracts a set of candidate routes ($\mathbb{R}$) based on the filters discussed in the previous section. It then shortlists the predominant route features ($\mathbb{F}$) from the database for the given *s*-*d* pair and requests the clients to provide a feature ordering and weights based on the personalized choice. This is the first filter that removes unfeasible routes based on the query context (like time of the day, spatial characteristics of the route etc.). According to the feature ordering $\mathbb{F}$ and the feature weights $\mathcal{W}$ along with the context of the query and historical statistics about the route, the server DIOA applies several internal signatures to find out the most suitable route $\mathcal{R}$ for recommendation. Finally, the commuter provides a feedback ($\mathcal{F}$) to the client automaton at the end of the trip, which helps the DIOA to integrate personalization within the feature weights. The details of the client and the server automata are discussed next.

Client Automaton: The client has a four-state automaton, with an initial `idle` state and three other states – (i) query for feature selection (`querying`), (ii) waiting for the recommended route after a query (`waiting`) and (iii) when the commuter is en-route (`riding`). The client automaton is initialized at the `idle` state, and the initial input to the automaton is from the commuter with the *s*-*d* pair (`request`(s,d)). On receiving the input, the query is forwarded to the server, and the client state is changed to `querying`. At this state, the client can receive an input signature from the server for providing the personalized choice of feature ordering (`response`$(\mathbb{F})$), and based on the user input for feature ordering and weights, it generates an output signature `ftr_order`$(\mathbb{F},\mathcal{W})$

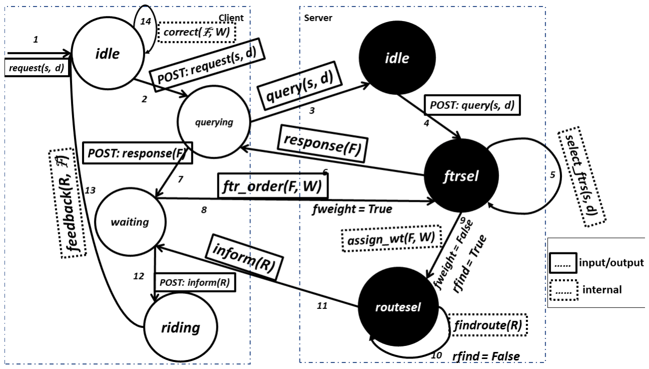


Figure 6: The complete state diagram of the DIOA based model in *ComfRide*. Numbers below the transition arrow represent the execution steps in the working of *ComfRide*. "POST: <sig>" transitions are the effect of the signature sig.

to response the same. Here, $\mathcal{W}$ are the feature weights calculated from the feedback value $\mathcal{F}$. It is to be noted that for the first trip, the weights are non-personalized. These weights are updated after each trip based on the feedback received from the user. At this stage, the client moves to the `waiting` state. Finally, once the recommended route from the server is received as an input signature `inform`$(\mathcal{R})$, the client moves to the `riding` state. At the end of the trip, the client automaton is reset to the `idle` state after providing the feedback through the signature `feedback`$(\mathcal{R},\mathcal{F})$. Once here, the client executes the internal signature `correct`$(\mathcal{F},\mathcal{W})$, to set the personalized weights based on the user feedback.

Server Automaton: The server automaton has three states – (i) feature selection (`ftrsel`), (ii) route selection (`routesel`) and (iii) idle (`idle`). Moreover, `ftrsel` and `routesel` each have two intermediate boolean state variables (`fweight` and `rfind`) indicating the status of the information available from the clients. Initially the server is in the `idle` state, while the intermediate state variables are set to false. It receives inputs as the route query (`query`(s,d)) and feature order (`ftr_order`$(\mathbb{F},\mathcal{W})$) from the client automaton. The outputs are the responses that it sends to the client corresponding to selected predominant features for a given *s*-*d* pair in the query (`response`$(\mathbb{F})$) and the final recommended route (`inform`$(\mathcal{R})$). The server automaton has three internal signatures to prune out irrelevant information (the filters) – (i) selecting predominant features based on the historical statistics and the query context (`select_ftrs`(s,d)), (ii) assigning weights to the features for user personalization (`assign_wt`$(\mathbb{F}, \mathcal{W})$), and (iii) finding the route for recommendation from the set of candidate routes (`findroute`$(\mathbb{R})$ where $\mathbb{R}$ is the set of candidate routes). We now describe the methodology used by the internal signatures.

6.2 Feature Shortlisting (`select_ftrs`(s,d))

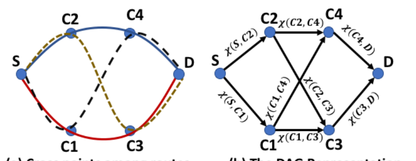
Among the set of available features, a subset of them becomes predominant for a given route based on the context of the query. For example, certain routes never remain congested at the night time; so, it may not be a prominent feature for a query over that route during night time. *ComfRide* aims to identify and flash only the prominent features to the commuters. To find the predominant features $\mathbb{F}$, the server automaton computes the occurrence density (number of occurrences per km of the route) of all the features of a route. `select_ftrs(s,d)` signature selects the features that are having the top $|\mathbb{F}|$ highest occurrence density among all possible routes between the given *s-d* pair.

6.3 Adjust Feature Weight (correct($\mathcal{F}$, $\mathcal{W}$))

In the *ComfRide* app, a commuter gives a score on a 5-point scale to each of the predominant features between the *s-d* pair of the query. Consequently, the weights (ω) of the features are assigned following the Likert Scale (1, 3, 5, 7 and 9 for five features) for the first trip by a commuter. For every subsequent trips, the feedback provided by the commuter during the previous trips is used to tune the weights of the features. Assume that for a feature f_1, the commuter gives a feedback $\epsilon_i^{f_1} \in \mathcal{F}$, and the previously assigned weight was $\omega_{i-1}^{f_1}$ after the $(i-1)^{\text{th}}$ trip. Then, we assign the weight for the i^{th} trip as $\omega_i^{f_1} = \omega_{i-1}^{f_1} + \omega_{i-1}^{f_1} * \epsilon_i^{f_1}$. It is to be noted that the relative weights of two features should follow their rankings based on the Likert scale. So, if f_1 and f_2 are two features with Likert scale values WL_{f_1} and WL_{f_2} ($WL_{f_1} \leq WL_{f_2}$), then $\omega_i^{f_1} \leq d \times \omega_i^{f_2}$, where $d = \frac{WL_{f_2}}{WL_{f_1}}$. After every i^{th} trip, if the commuter is dissatisfied with respect to a feature f_k by $\epsilon_i^{f_k}$, then after a fixed number of trips, $\epsilon_i^{f_k} \longrightarrow 0$. Eventually, the weight assigned to the feature f_k converges to the same as desired by the commuter. This is because $\epsilon_i^{f_k}$ is non-increasing as an increase in the weight for a feature would give a better or same route. Hence, the new recommendation wouldn't dissatisfy the commuter more than the previous case.

6.4 Route Recommender

Based on the user provided feature ranking and personalized feature weight adjustment by the server automaton, the internal signature `findroute(`$\mathbb{R}$`)` extracts the recommended route from the set of candidate routes $\mathbb{R}$. The set of candidate routes includes all the feasible routes between the given *s-d* pair in the query, which conform to the user's travel time budget. The allowable deviation in travel time is taken as an input during the `ftr_order(`$\mathbb{F}, \mathcal{W}$`)` signature execution. We consider the routes between the given *s-d* pair, where the median travel time as obtained from the historical data is not more than the historical median travel time for the least cost route (the route with minimum travel time) for the *s-d* pair plus the allowed deviation specified by the commuter. In *ComfRide* we employ the fuzzy set theory based recommendation technique along with TOPSIS to develop our recommender [6, 28]. TOPSIS constructs a feature matrix for various available alternatives and ranks the alternatives based on the distance similarity with the worst alternative in comparison to the best alternative. The major challenges here are (a) to obtain the feature set and a matrix of possible alternatives and (b) to optimize this matrix over different factors to get the best route based on a ranking strategy. The data collection application and database generation take care of identifying and defining the feature set. The pruning techniques employed using the DIOA further simplify the resulting matrix of feature sets for various alternate routes. Consequently, the scoring as discussed in [28] gives the value of the metric called *Route Comfort Index* (χ), defined as follows. Let ρ be the feature value as computed for a route, and ω is the feature weight obtained from the internal signature `assign_wt(`$\mathbb{F}, \mathcal{W}$`)`. The feature values are calculated from the crowdsourced data collected from the commuters using the techniques discussed in [36], and are normalized over different routes based on TOPSIS feature normalization [6]. Then, $\chi = \sum_{n \in \mathbb{F}} \rho_n * \omega_n$. Now, for every route in $\mathbb{R}$, χ is calculated from the feature matrix, and the route having the highest rank based on TOPSIS distance similarity over χ is recommended.



(a) Cross points among routes between S-D (b) The DAG Representation

Figure 7: Considering breakpoints during a travel

Opting for a break journey: Based on the commuter's choice of comfort parameters, there may be instances when taking a breakpoint during the journey and availing a different bus from the breakpoint to the destination are more fruitful than taking a direct route. If the commuter opts for a break journey, then we have to consider both break and non-break scenarios.

We solve this problem through a dynamic programming approach. The basic intuition is as follows. A breakpoint can be taken at a stoppage where two different bus routes intersect. From the set of candidate routes $\mathbb{R}$ between the given *s-d* pair, we first find out such intersecting stoppages. Let $\mathbb{C}$ be the set of such intersecting stoppages. Then we construct a weighted direct acyclic graph (DAG) $\mathcal{G}(\mathbb{V}, \mathbb{E})$ as follows. The vertex set $\mathbb{V}$ contains the elements $\mathbb{C} \cup \{s, d\}$. We construct an edge $e(v_1 \in \mathbb{V}, v_2 \in \mathbb{V}) \in \mathbb{E}$ if there is at-least one direct bus route from v_1 to v_2 without having any other intersecting stoppages in between. Note that there can be more than one bus routes from v_1 to v_2. The edge weight for $e(v_1, v_2) \in \mathbb{E}$ is the χ value for the best bus route between the two intersecting stoppages v_1 and v_2, calculated using TOPSIS as discussed earlier. Figure 7 shows an example with 4 different routes between the *s-d* pair and the corresponding DAG structure. We apply the dynamic programming over this DAG structure. Consider the route from s to d via $C1$ and $C3$, as shown in Figure 7. There are two substructures if the commuter wants to take a single break – either break at $C1$ or take a break at $C3$, and choose a different bus route. We consider that χ of an end-to-end travel is the average of all the χ values of the route segments that the commuter follows. Let $\chi_{s,d}^{n}$ denote the RCI between s and d with n number of breaks. Then the optimal substructure of this problem is represented as follows.

$$\chi_{s,d}^{n} = \max_{c \in \mathbb{C}, 1 \leq m \leq n} \frac{1}{n} \left((n-m) \times \chi_{s,c}^{n-m} + m \times \chi_{c,d}^{m} \right) \tag{1}$$

Let a commuter prefers at most β number of breaks. Then based on the dynamic programming approach, we compute the recommended travel as the sequence of bus routes with zero or more breakpoints, which provide the maximum RCI as follows.

$$\chi_{s,d}^{max} = \max_{0 \leq n \leq \beta} \chi_{s,d}^{n} \tag{2}$$

Table 2: Source-destination (S-D) Pair Details; Distance and travel time are given for all possible routes between the S-D pairs

(S, D) Pair	Source	Destination	Distance (km)					Avg. Travel Time (min)				
			1	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
P1	KM	RH	10	10	15	15	9	55	55	65	65	50
P2	SC	RH	6	6	6	6	-	30	30	30	30	-
P3	SC	GH	8	8	8	8	-	45	45	50	50	-
P4	KM	JP	15	15	20	20	14	70	65	95	90	65
P5	TG	CC	4	4	5	6	-	25	27	35	40	-
P6	NT	MS	14	13	-	-	-	55	50	-	-	-
P7	SS	RG	14	14	15	15	-	42	58	46	46	-

7 IMPLEMENTATION AND RESULTS

We have developed *ComfRide* as an Android app. The deployment and testing of the system spanned for around two years – from March 2016 to April 2018, when we conducted rigorous data collection and analysis in a state capital city (area 1887 sq.km) in India. The experiment has been conducted on 7 source-destination pairs, with 28 different bus routes. The details of the routes for these 7 source-destination pairs are given in Table 2. The ComfRide server is implemented on Debian 9.3 server, with a Intel(R) Xeon(R) E5-2620 v3 @ 2.40GHz CPU and 32GB memory. Diverse set of smartphones have been used during experiments with cost ranging from $90US\$$ to $300US\$$ and the Android version 4.4 to 6.0.

7.1 Experiment Planning and Setups

For data collection through controlled crowdsourcing, we recruited 50 volunteers, who were primarily undergraduate college students. These students were provided with suitable incentives (approx. \$50 as honorarium and a volunteer certificate). 20 volunteers were given specific travel routes (source, destination and the road to be followed) where they traveled in their leisure time (semi-controlled experiments), while the remaining volunteers collected data during their regular travels from home to college and back (general deployments). Volunteers were asked to perform experiments for two sets of routes – with and without a break journey, based on *ComfRide* recommendation. Every trip was taken by a group of volunteers on different days at various times of the day. During the semi-controlled experiments, a group of volunteers traveled through the *ComfRide* recommended route based on their choice of comfort, whereas another group of volunteers traveled using Google Maps (*G-Maps*) recommended route (the baseline mechanism) on the same day during the same time. It can be noted that the *G-Maps* navigation system recommends a series of routes, which are ordered according to the total expected travel time. For baseline comparison, the volunteers have taken the route with the least expected travel time as recommended by the Google navigation.

7.2 ComfRide vs. Google Navigation

First, we compare the performance between *ComfRide* recommendation and *G-Maps* recommendation in terms of various comfort features. However, due to space constraints, here we discuss and compare the performance in terms of crowding of a bus, which is measured in terms of the possibility of getting a seat after a commuter boards a bus. In this case, the commuters give highest preference to the less crowded bus routes followed by less travel time, jerkiness and congestion. Figure 8(a) compares recommended

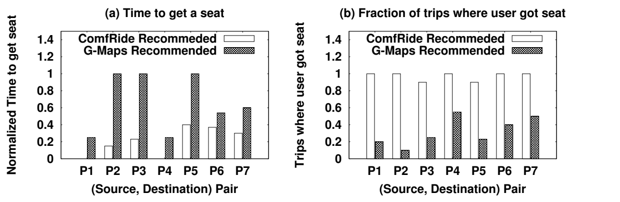


Figure 8: Comparison of *ComfRide* recommended and *G-Maps* recommended routes with respect to probability of sitting

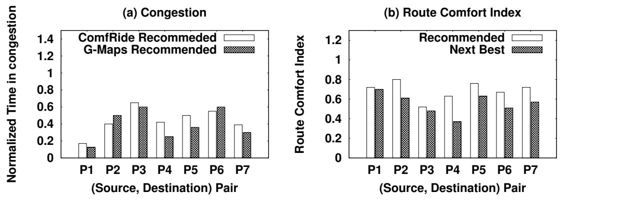


Figure 9: (a) Trade-off w.r.t. congestion when giving preference to probability of sitting, (b) Variation of RCI for different S-D pairs

and non-recommended routes w.r.t. average time after which the commuter secures a seat. We normalize this time by the total travel time. We observe that for all the trips, this time is very low for the recommended route, except for P6 where there are only two possible routes and hence less possible variations. We see zero value for P1 and P4; this is mainly because of the fact that the source is a terminal bus stop for the *ComfRide* recommended route. Figure 8(b) shows the fraction of trips when the commuter was able to get a seat. The commuter gets a seat for all the trips except in case of one trip for P3 and P5, when the recommended route is preferred. However, giving a higher preference to the sitting probability can result in a trade-off w.r.t. other parameters. We show this trade-off for congestion in Figure 9(a). In most of the cases, time in congestion is higher when on a *ComfRide* recommended route based on sitting probability. However, this is not always true as for P2 and P6.

We check the average RCI for the *ComfRide* recommended routes and *G-Maps* recommended routes, as shown in Figure 9(b). Our experiments show that the route recommended by *ComfRide* has a better average RCI than the other routes in between the same source-destination pair. We observe that on an average, the recommended routes have a 30% better comfort level. It can be noted that the recommendation is based on the commuter's choice and priority assigned for different features during the trip.

7.3 Impact of Personalization

Next, we analyze how different features impact route recommendation over different bus routes. There are certain cases when the route recommended by *ComfRide* and *G-Maps* are similar, primarily for the cases where there are less alternate routes, or the commuter gives priority to the travel time. During the semi-controlled experiments, we have considered multiple orderings of features at different times of the day, and checked for how many instances, the *ComfRide* recommended route and *G-Maps* recommended routes

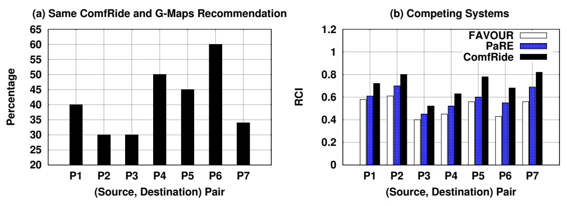


Figure 10: (a) Percentage of times *G-Maps* and *ComfRide* recommend the same route (b) Average RCI of different S-D pairs for the competing systems

Table 3: State space comparison for Graph & DIOA based approach

(S,D) Pair	Stops	Available Routes	Graph	DIOA
P1	85	5	599760	425
P2	24	4	46368	120
P3	60	4	297360	300
P4	150	5	1877400	750
P5	57	4	268128	285
P6	64	2	338688	320
P7	95	4	750120	475

are same. In Figure 10(a) we show the percentage of trips when both the systems have recommended the same route. P6 has higher percentage because it has only two different bus routes. Interestingly, we observe that the *ComfRide* recommended routes differ from *G-Maps* recommended routes for many instances, which is as high as 70% for P2 and P3, indicating that a route recommendation based on commuters' choice of comfort features is important and not a trivial extension of *G-Maps* based navigation system.

7.4 ComfRide vs Competing Systems

We compare *ComfRide* with two personalized route recommender systems proposed recently – *PaRE* [23] and *FAVOUR* [8]. *PaRE* relies on historical data collected from previous trips of the commuter to identify important landmarks and frequently used routes. It then recommends the route to maximize the familiarity while minimizing number of segments. *FAVOUR* asks a set of questions to the commuter, and then uses mass preference prior to predict best route for her. This prediction is improved using Bayesian learning techniques. We execute PaRE and FAVOUR over the collected dataset and compare the recommendation performance with *ComfRide*.

7.4.1 ***Performance in terms of RCI.*** None of these two personalized route recommender consider the features used in *ComfRide*, and several scenarios are seen when the competing systems perform poorly. The average RCI for different source-destination pairs is higher for *ComfRide* as compared to the other two, as seen in Fig 10(b). FAVOUR gives priority to the general choices of the commuters over a route, and so, fails to capture the personal choices of a commuter. On the other hand, PaRE gives priority to the personal choices, and thus ignores environmental impacts. *ComfRide* balances both the personal choice and the environmental impact, and therefore improves the RCI compared to others.

7.4.2 ***Advantage of DIOA in ComfRide.*** We also compare the advantage of using DIOA over graph-based approaches, as utilized in several route recommender such as [5, 23]. It is evident

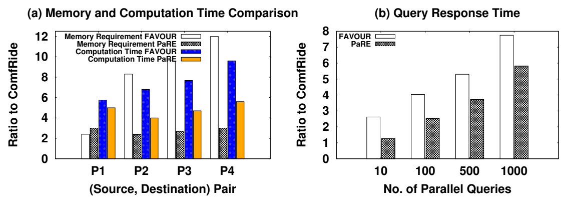


Figure 11: (a) Memory and computation time improvement (b) Query response time improvement in *ComfRide*

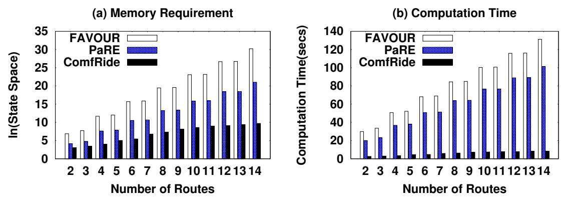


Figure 12: (a) Memory and (b) Computation Time for varying number of routes from Google Direction API

from Table 3, that the state space is significantly reduced when using DIOA compared to the graph-based approach. This provides fast recommendation with reduced computation complexity.

We have also compared various system parameters, viz. memory requirement, computation time and query response time, when the same recommendation is done by PaRE and FAVOUR. The plots in Figure 11(a) shows the ratio of the improvement in memory requirement and computation time of *ComfRide* with respect to the competing systems. As we observe from the plot, *ComfRide* optimizes these parameters compared to the baselines in all the four scenarios. Figure 11(b) shows the improvement ratio of query response time in *ComfRide* when parallel queries are fired to the system. A similar result is shown in Figure 12 for all the routes we generated through the Google Direction API (details in §3). As is evident, the pruning of the graph has helped to reduce the memory requirement and the computation time by a huge margin in all the scenarios. It is to be noted that in the competing systems, a separate query processing graph is generated for each query. Therefore, there is a significant impact on the query response time for large number of parallel queries. Nevertheless, the DIOA helps in reducing this load by pruning the graph based on contextual information.

8 CONCLUSION

Commuter comfort is a challenging problem for public transportation systems, as it is person specific and varies widely based on the environment and multiple other socio-economic factors. In this paper, we proposed *ComfRide*, a smartphone based system to recommend the most comfortable route according to commuter's preferences over various environmental and personal choices. The key concept behind *ComfRide* is to embed the *general awareness* and *intelligence* used by a regular commuter to choose the best (comfortable) bus route to reach her desired destination. *ComfRide* relies on the crowdsourced GPS and inertial sensor data, collected through a mobile app, from the city commuters. Given a source-destination pair, *ComfRide* recommends most comfortable route based on commuter's preference even by considering single or multiple breakpoints during the journey. The novelty of the system comes from utilizing a DIOA based composition model for effectively processing the queries based on historical as well as contextual information. From a field trial for 2 years over 28 different bus routes in a state capital of India, we observed that *ComfRide* recommended routes have on average 30% better comfort level than Google navigation based recommended routes, for various combinations of commuters' priorities to the comfort features. We believe that *ComfRide* can take us one step ahead attracting commuters more towards public transport, without waiting for the implementation of long term policies, especially in developing regions.

REFERENCES

[1] 2011. IBM Global Commuter Pain Survey: Traffic Congestion Down, Pain Way Up (available online, last accessed: February, 2017). (2011). http://www-03.ibm.com/press/us/en/pressrelease/35359.wss.

[2] 2017. GOTransit, (available online, last accessed: May, 2017). (2017). https://gotransitnc.org.

[3] 2017. TRipGo, (available online, last accessed: May, 2017). (2017). https://tripgo.skedgo.com.

[4] Paul C Attie and Nancy A Lynch. 2016. Dynamic input/output automata: a formal and compositional model for dynamic systems. *Information and Computation* 249 (2016), 28–75.

[5] Garvita Bajaj, Georgios Bouloukakis, Animesh Pathak, Pushpendra Singh, Nikolaos Georgantas, and Valérie Issarny. 2015. Toward enabling convenient urban transit through mobile crowdsensing. In *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*. IEEE, 290–295.

[6] Majid Behzadian, S Khanmohammadi Otaghsara, Morteza Yazdani, and Joshua Ignatius. 2012. A state-of the-art survey of TOPSIS applications. *Expert Systems with Applications* 39, 17 (2012), 13051–13069.

[7] Maria Bordagaray, Luigi dell'Olio, Angel Ibeas, and Patricia Cecín. 2014. Modelling user perception of bus transit quality considering user and service heterogeneity. *Transportmetrica A: Transport Science* 10, 8 (2014), 705–721.

[8] Paolo Campigotto, Christian Rudloff, Maximilian Leodolter, and Dietmar Bauer. 2017. Personalized and situation-aware multimodal route recommendations: the FAVOUR algorithm. *IEEE Transactions on Intelligent Transportation Systems* 18, 1 (2017), 92–102.

[9] Juan C Castellanos and Fabiano Fruett. 2014. Embedded system to evaluate the passenger comfort in public transportation based on dynamical vehicle behavior with user's feedback. *Measurement* 47 (2014), 442–451.

[10] Megha Chaudhary, Aneesh Bansal, Divya Bansal, Bhaskaran Raman, KK Ramakrishnan, and Naveen Aggarwal. 2016. Finding occupancy in buses using crowdsourced data from smartphones. In *Proceedings of the 17th International Conference on Distributed Computing and Networking*. ACM, 35.

[11] Giusy Di Lorenzo, Marco Sbodio, Francesco Calabrese, Michele Berlingerio, Fabio Pinelli, and Rahul Nair. 2016. Allaboard: visual exploration of cellphone mobility data to optimise public transport. *IEEE transactions on visualization and computer graphics* 22, 2 (2016), 1036–1050.

[12] Karoly Farkas, Gabor Feher, Andras Benczur, and Csaba Sidlo. 2015. Crowdsending based public transport information service in smart cities. *IEEE Communications Magazine* 53, 8 (2015), 158–165.

[13] Massimo Florio. 2013. *Network industries and social welfare: The experiment that reshuffled European utilities*. OUP Oxford.

[14] André Luís Policani Freitas. 2013. Assessing the quality of intercity road transportation of passengers: An exploratory study in Brazil. *Transportation Research Part A: Policy and Practice* 49 (2013), 379–392.

[15] Kaiqun Fu, Yen-Cheng Lu, and Chang-Tien Lu. 2014. Treads: A safe route recommender using social media mining and text summarization. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 557–560.

[16] Edward Glaeser and J Vernon Henderson. 2017. Urban Economics for the Developing World: An Introduction. *Journal of Urban Economics* (2017).

[17] Google. 2017. Google Transit, (available online, last accessed: May, 2017). (2017). https://www.google.com/transit.

[18] Danhuai Guo, Ziqi Zhao, Wei Xu, Jinsong Lan, Tao Zhang, Shuguang Liu, Jianhui Li, and Yuanchun Zhou. 2015. How to find a comfortable bus route – towards personalized information recommendation services. *Data Science Journal* 14 (2015).

[19] Daniel Herzog, Hesham Massoud, and Wolfgang Wörndl. 2017. Routeme: A mobile recommender system for personalized, multi-modal route planning. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. ACM, 67–75.

[20] Şükrü İmre and Dilay Çelebi. 2017. Measuring Comfort in Public Transport: A case study for Istanbul. *Transportation Research Procedia* 25 (2017), 2441–2449.

[21] Suresh Jain, Preeti Aggarwal, Prashant Kumar, Shaleen Singhal, and Prateek Sharma. 2014. Identifying public preferences using multi-criteria decision making for assessing the shift of urban commuters from private to public transport: A case study of Delhi. *Transportation Research Part F: Traffic Psychology and Behaviour* 24 (2014), 60 – 70.

[22] Jue Ji and Xiaolu Gao. 2010. Analysis of people's satisfaction with public transportation in Beijing. *Habitat International* 34 (2010), 464–470.

[23] Yaguang Li, Han Su, Ugur Demiryurek, Bolong Zheng, Tieke He, and Cyrus Shahabi. 2017. PaRE: A System for Personalized Route Guidance. In *Proceedings of the 26th International Conference on World Wide Web*. 637–646.

[24] Zheng Li and David A Hensher. 2011. Crowding and public transport: a review of willingness to pay evidence and its relevance in project appraisal. *Transport Policy* 18, 6 (2011), 880–887.

[25] Zheng Li and David A Hensher. 2013. Crowding in public transport: a review of objective and subjective measures. *Journal of Public Transportation* 16, 2 (2013), 6.

[26] Hiroyuki Nakamura, Yuan Gao, He Gao, Hongliang Zhang, Akifumi Kiyohiro, and Tsunenori Mine. 2014. Adaptive user interface agent for personalized public transportation recommendation system: PATRASH. In *International Conference on Principles and Practice of Multi-Agent Systems*. Springer, 238–245.

[27] Dorina Pojani and Dominic Stead. 2017. The urban transport crisis in emerging economies: An introduction. In *The Urban Transport Crisis in Emerging Economies*. Springer, 1–10.

[28] Faisal Rehman, Osman Khalid, and Sajjad Ahmad Madani. 2017. A comparative study of location-based recommendation systems. *The Knowledge Engineering Review* 32 (2017).

[29] Darshan Santani, Jidraph Njuguna, Tierra Bills, Aisha W Bryant, Reginald Bryant, Jonathan Ledgard, and Daniel Gatica-Perez. 2015. Communisense: Crowdsourcing road hazards in nairobi. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM, 445–456.

[30] Dragan Sekulić, Vlastimir Dedović, Srdjan Rusov, Slaviša Šalinić, and Aleksandar Obradović. 2013. Analysis of vibration effects on the comfort of intercity bus users by oscillatory model with ten degrees of freedom. *Applied Mathematical Modelling* 37, 18 (2013), 8629–8644.

[31] Khaled Shaaban and Rania F Khalil. 2013. Investigating the customer satisfaction of the bus service in Qatar. *Procedia-Social and Behavioral Sciences* 104 (2013), 865–874.

[32] Xianghao Shen, Shumin Feng, Zhenning Li, and Baoyu Hu. 2016. Analysis of bus passenger comfort perception based on passenger load factor and in-vehicle time. *SpringerPlus* 5, 1 (2016), 1–10.

[33] Chao Song, Jie Wu, Ming Liu, Haigang Gong, and Bojun Gou. 2012. Resen: Sensing and evaluating the riding experience based on crowdsourcing by smart phones. In *Mobile Ad-hoc and Sensor Networks (MSN), 2012 Eighth International Conference on*. IEEE, 147–152.

[34] Alejandro Tirachini, David A Hensher, and John M Rose. 2013. Crowding in public transport systems: Effects on users, operation and implications for the estimation of demand. *Transportation research part A: policy and practice* 53 (2013), 36–52.

[35] Dea van Lierop and Ahmed El-Geneidy. 2018. Is having a positive image of public transit associated with travel satisfaction and continued transit usage? An exploratory study of bus transit. *Public Transport* (2018), 1–16.

[36] Rohit Verma, Surjya Ghosh, Niloy Ganguly, Bivas Mitra, and Sandip Chakraborty. 2017. Smart-phone based Spatio-temporal Sensing for Annotated Transit Map Generation. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 16.

[37] Rohit Verma, Surjya Ghosh, Aviral Shrivastava, Niloy Ganguly, Bivas Mitra, and Sandip Chakraborty. 2016. Unsupervised annotated city traffic map generation. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 59.

[38] Rohit Verma, Aviral Shrivastava, Bivas Mitra, Sujoy Saha, Niloy Ganguly, Subrata Nandi, and Sandip Chakraborty. 2016. UrbanEye: An outdoor localization system for public transport. In *INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE*. IEEE, 1–9.

[39] Xiao Wang, Xinhu Zheng, Qingpeng Zhang, Tao Wang, and Dayong Shen. 2016. Crowdsourcing in ITS: The state of the work and the networking. *IEEE Transactions on Intelligent Transportation Systems* 17, 6 (2016), 1596–1605.

[40] Desheng Zhang, Juanjuan Zhao, Fan Zhang, Ruobing Jiang, Tian He, and Nikos Papanikolopoulos. 2017. Last-mile transit service with urban infrastructure data. *ACM Transactions on Cyber-Physical Systems* 1, 2 (2017), 6.