Deep-Learning based Model for Prediction of Crowding in Public Transit System

Arpit Shrivastava, Devesh Pratap Singh, Itisha Jain, Amit Agarwal*

***Abstract*—Crowding in public transport (PT) is one of the reasons which nudges the road users to shift from PT to private modes of transport. To provide the passengers a facility to plan their trips as per the dynamic crowding levels, this work proposes a framework for a Passenger Information System (PIS), in which the transit choices are differentiated with respect to crowding levels on the transit routes at different times of the day. For the crowding prediction, firstly, the Transit Segment Relations (TSR) are constituted and used to make clusters based on the ridership index. Further, a time-series model is trained for each cluster using boarding TSR. A case study of Bhubaneswar, India is presented, and three months of ticketing data are used to demonstrate the performance of the proposed prediction model. The prediction model is integrated in the PIS to expedite various route choices.**

***Index Terms*—Public Transport, Occupancy, Crowding, Affinity Propogation, LSTM**

I. Introduction

Public Transport (PT) is a vital part of the urban transport system in most urban agglomerations. However, many PT systems are experiencing increasing crowding levels at different times of the day. In-vehicle and at stop crowding have negative impact on the comfort and satisfaction of travelers as well as increase the dwell time that may provoke the transition from PT to private transport or keeping private users away from PT [1]. In addition, the COVID pandemic has stuck the public transit ridership severely by shifting the passengers to private motorized vehicles due to chances of getting infection in the crowded places [2]–[4]. Addition of more supply is unlikely to satisfy the travel demand and physical distance norms [5]. The exigency to recover from the pandemic has paved the immediate need for a system that can provide alternatives based on crowding levels. In other words, a passenger may adapt to different departure time and PT route to experience lesser crowding levels. However, it will be possible only if a Passenger Information System (PIS) incorporates the real-time crowding/ riding index while proposing the alternative routes. This work attempts to develop a PIS framework, which recommends the alternative choices based on the dynamic occupancy levels in the transit vehicles.

An increasing usage of automated fare collection (AFC) using smart card data or electronic ticketing facilitate the travelers (comfort, convenience, faster boarding/ alighting, etc.) as well as operators (lesser operational costs, lesser cycle time, etc.) and planners (insights from the data). The electronic ticketing machine (ETM) or smart-card systems produce large amount of high quality data. It can be used for strategic, tactical and operation analysis [6]. Chen and Fan [7] reviews literature, which are based on smart-card data (SCD). However, this work does not include many other prediction based studies, which are also exploiting the smart-card data. Table I exhibits various studies in the last decade or so, that use smart-card or ETM data for various analyses, estimation and prediction. The applications of SCD/ ETM include spatio-temporal variation in the of public transit network [8], analysis of travel patterns of passengers [9], identification of the trip purposes of transit users [10], [11], extraction of locations (boarding stops [7], destination stops [12], home locations [10]), inference about the employment status [13], anomaly detection in rail ticketing data [14], prediction of bus-bunching on various transit routes [15], [16], estimation/ prediction of public transit OD flows [17], [18], prediction of passenger flow [19]–[24], prediction of ridership [25], prediction for passenger trips [26], etc. Clearly, the SCD/ ETM data has huge potential for various use cases. The present study also attempt to exploit the ETM data to determine the occupancy levels of the transit vehicles and use them for short-term prediction of crowding in the transit vehicles.

The occupancy in a transit vehicle can be estimated if details about number of boarding and alighting passengers at each stop is available. Equation (1) shows that the passenger volume (P_n) after departure from stop n is difference of sum of boarded passengers (b_i) from first stop to n^{th} stop and sum of alighted passengers (a_i) from second stop to n^{th} stop.

$$P_n = \sum_{i=1}^{n} b_i - \sum_{i=2}^{n} a_i \tag{1}$$

Typically, the capacity of a transit vehicle is defined as the sum of seating and standing capacities. The latter depends on the desired person density (i.e., number of persons standing per m^2). An increasing passengers density will reduce the comfort, level of service [27], [28] and negatively affect the dwell time. To determine the standing capacity, a level of service is decided during the planning stage. If the passenger volume between two stop exceeds the seating capacity of the transit vehicle, an increase in the number of standing passenger leads to rising levels of crowding. Consequently, higher crowding for longer duration is likely to reduce the overall modal share of public transit.

Manuscript submitted: May 26, 2022. (Corresponding author: Amit Agarwal). The authors are with the Department of Civil Engineering, Indian Institute of Technology Roorkee, Roorkee, Haridwar, India-247667. Email addresses: ashrivastava@ce.iitr.ac.in, dsingh4@ce.iitr.ac.in, ijain@ce.iitr.ac.in, amitfce@iitr.ac.in

The authors wish to thank National Institute of Urban Affairs (NIUA), Deutsche Gesellschaft für Internationale Zusammenarbeit (GiZ) India for facilitating the data and suggestions to make the model realistic.

Table I. Studies using data from smart card/ electronic ticketing machine data

Application	study	method/ model	location
measure variability of PT	[8]	Transportation object-oriented modeling and clustering	Gatineau, Canada
travel patterns of passengers	[9]	DBSCAN, K-Means, rough-set based algorithm	Beijing, China
bus bunching prediction	[15]	Least Squares SVM	Beijing, China
	[16]	Supply-demand seq2seq model (CNN, LSTM)	Sydney, Australia
(A) infer home location and (B) trip purpose	[10]	(A) → center point based algorithm and (B) → rule based approach	Beijing, China
infer trip purpose	[11]	rule-based modeling	Queensland, Australia
forecasting destinations	[12]	deep learning	Seoul, Korea
boarding location extraction	[7]	clustering	4 bus routes in Guangzhou, China
infer employment status of passengers	[13]	CNN	London, U.K.
OD flow prediction/ estimation	[18]	LSTM	MTR lines Hong Kong
	[17]	trip chaining	Brisbane, Queensland, Australia
ridership prediction	[25]	simple what-if analysis	The Hague, Netherland
passenger flow prediction	[19]	Kalman filtering and KNN	LRT Line Changchun, China
	[20]	Cluster-Based LSTM	Beijing, China
	[21]	ARIMA and GARCH models (for special events only)	two stations (close to Olympic sports center), Nanjing, China
	[22]	spatio-temporal LSTM	Beijing Metro Airport Line
	[23]	LSTM	2 bus lines in Guangzhou, China
	[24]	LSTM	Metro line Kochi, India
anomaly detection in railway ticketing	[14]	pre-wavelet LSTM	China
passenger trips prediction	[26]	NLP based Back propogation	Queensland, Australia

Information about crowding/ occupancy at different time of the day may assist the passengers in planning their trips. For this, prediction of occupancy in the transit vehicles is required. There are plenty of studies in the literature, which uses various machine/ deep learning models in public transit planning, operation, scheduling, etc., (c.f., Table I). For instance, 1) a hybrid stacked autoencoders (SAE) deep neural network (DNN) model was proposed to predict the passenger flow on four bus lines [29] 2) a natural language processing (NLP) based back propogation neural network model is developed for prediction of passengers in PT [26] 3) least square Support Vector Machine (SVM) is used for prediction of bus bunching [15], 4) CNN is used to predict the employment status of passengers [13], 5) random forest is used to predict the crowding level using automated passenger count data in Pittsburgh [30], etc.

Occupancy level in PT depends on many external factors (e.g., time of the day, day of the week, recurring events, weather, etc.) but these show high temporal dependency. Thus, the prediction model must have time dependence to explain the periodicity and variance in the past data of the occupancy of transit vehicles. For such purposes, use of Long Short Term Memory (LSTM) is quite common due to its performance [31]. LSTM models are very popular in different domains of transportation, e.g., (a) to forecast traffic in short-term [32], (b) to predict the bicycle sharing usages [33], (c) to predict particulate matter [34], (d) to forecast the short-term train loads [35], etc. Table I exhibits various prediction models which use LSTM directly or integrates LSTM with other models for PT. For instance, (a) to predict the bus bunching [16], (b) to predict the short-term OD flows in urban rail system [18], (c) to detect anomaly in Chinese railway ticketing data [14], (d) to predict the bus arrival time [36], (e) to predict the passenger flow [20], [22]–[24], (f) to forecast the passenger travel demand using automated passenger count data [37], etc.

From the aforementioned studies, it can be inferred that LSTM based prediction models can be be employed for the data with temporal variability. It has abstraction capabilities of neural networks to exploit the structural regularity of the time series data. On the other hand, the prediction of occupancy in the transit vehicles will not only help the existing passengers in planning their trips but also may attract the users of private modes of transportation. As per the authors best knowledge, literature lacks in such prediction model as well as passenger information system which provides alternatives while incorporating the crowding levels in the routing. For the prediction model, the following research gaps are identified. (i) The scope of many studies is limited to a few stations or a few lines rather than focusing on the whole transit system in the region. (ii) The temporal variation is included in various LSTM models; however, very few studies included the spatio-temporal variability in the prediction function. (iii) Typically, the prediction models in PT domain are trained at aggregate levels to have a correlation in the consecutive time-bins, which is likely to have a better performing LSTM model. However, the granularity of the model is compromised in such cases.

Therefore, the present study proposes a cluster-based, long-short term memory (CB-LSTM) model for prediction of occupancy in transit vehicles and crowding levels for each transit choice, which is applicable for all transit systems in a region. Further, the present study maintains the granularity to a transit stop for each trip. The proposed prediction model is integrated in the form of a crowding toolbox to a passenger information system which embeds the crowding levels in the transit choices.

II. Crowding toolbox

This section explains the crowding toolbox. The various steps are depicted in Figure 1.

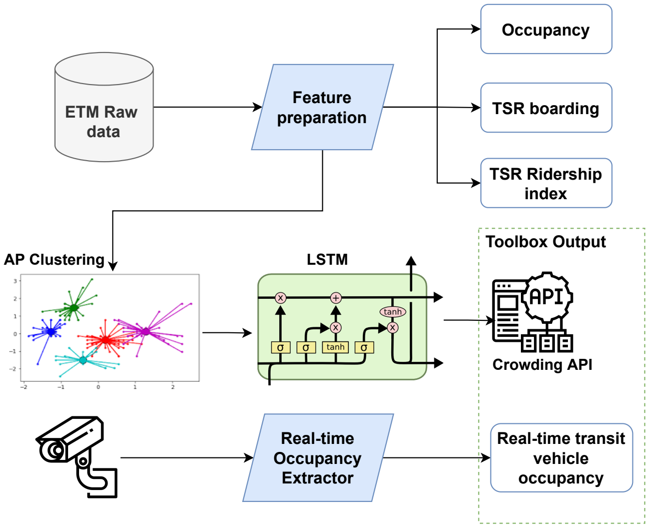


Fig. 1. Schematic of the prediction model and crowding toolbox

In the present work, transit path represents the route between an OD pair, which may consists of trips of different transit routes, i.e., including transfers. A transit route (or transit line) offers various trips throughout the day. A segment is portion of a trip between two consecutive stops in one direction.

A. ETM data

Typically, the Electronic Ticketing Machine (ETM) data comprise the following fields, “from station, to station, trip start time, total fare collected, vehicle number, transit mode, route number, timestamp”, etc., corresponding to every ticket raised. It may have other fields as per the use case in different cities. However, these are the fields which are required and used in the present work. This data is called as raw data. The raw data is cleaned for possible errors, null values, etc., and prepared/ transformed for some missing, illogical entries. For this purpose, other data sources (e.g., operator/ agency’s website) is referred.

B. Features preparation

This section describes the various features which are used in the prediction model.

1) Calculation of occupancy: The occupancy for each segment for a given route and time is determined as follows.

(A) The number of passengers boarding at each stop is required. It is estimated by dividing total fare collected (in ETM data) by the minimum fare for the given from and to stops.

(B) The raw data is sorted for the timestamp (increasing order).

(C) For each trip of a transit route, the sequence of the transit stops is verified using General Transit Feed Specification (GTFS). Further, the boarding and alighting passenger values of a transit trip are aggregated to get the occupancy at each segment of the given transit trip.

(D) The segmental occupancies ($P_{r,n}$; between stops n and $n+1$ of a transit route) are calculated for whole duration of the raw data.

The passenger occupancy ($P_{r,n}$) at n^{th} stop on one of the trips for a given transit route (r) can be written as:

$$P_{r,n} = \sum_{j=1}^{n} b_{r,j} - \sum_{j=2}^{n} a_{r,j} \tag{2}$$

where, $b_{r,j}$ and $a_{r,j}$ are number of persons boarded and alighted at stop j of transit route r.

2) Transit Segment Relation (TSR): To maintain highest level of granularity, a ‘Transit Segment Relation (TSR)’ is defined for two consecutive transit stops (i and $i+1$) in one hour time bin, which starts from s_i (e.g., start of clock hour, the departure time of the first trip, etc.) for the stop i, i.e.,

$$\text{TSR}_i = f(i, s_i)$$

For instance, if a transit stop is served between 05:40 and 23:40, 19/ 18 TSRs will be considered for the stop and the s_i for first three TSR could be 05:00, 06:00, 07:00 (Section II-B3) or 05:40, 06:40 and 07:40 (see Section II-B4), respectively. TSR is created and used in clustering (see Section II-C) and LSTM (see Section II-D).

3) Calculation of TSR ridership index: In the clustering (see Section II-C), TSR ridership index is used as a feature. In contrast to TSR boarding for time-series model (Section II-B4), the TSR ridership index is determined for each route (e.g., blue and black lines separately in Figure 2) and as a ratio (i.e., between 0 and 1). TSR ridership Index is defined as the ratio of the hourly ridership on a route over the daily ridership for the route (i.e., during operational hours). This is used to capture passenger flow variations throughout a day for the given route while ignoring the ridership volume to some extent. The TSR ridership index is a route-specific number to assist in the clusters.

$$\text{RI}_{\text{TSR}_{i,r}} = \frac{\sum_{t=c_i}^{t=c_i+1} b_{i,r,t}}{\sum_{t=1}^{t=H} b_{i,r,t}} \tag{3}$$

Equation (3) represents the ratio of passenger boarding at a stop i in an hour for a transit route r (i.e., sum over all trips in that hour) to the total passenger boarding from the stop i

for the route r in the whole duration of operation (H, i.e., all trips). c_i is clock hour.

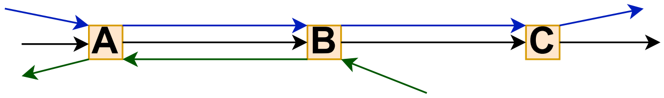


Fig. 2. Schematic of bus lines to show TSR

4) Calculation of TSR boarding: In the time-series model (see Section II-D), passenger boarding for each segment is required as a feature, which vary with respect to time of the day. For the prediction of the occupancy, the temporal trend of passenger boarding for each segment is a very important feature. Therefore, TSR boarding (b_{TSR_i}) are defined and estimated as shown in Equation (4).

$$b_{\text{TSR}_i} = \sum_{t=d_i-1}^{t=d_i} b_{i,t} \tag{4}$$

where, $b_{i,t}$ denotes the boarding at stop i for the trip departing at t from stop i. Thus, b_{TSR} represents the sum of all boarding at stop i (towards stop $i+1$), which happens between time $d_i - 1$ to d_i irrespective of transit route (i.e., sum of boarding at stop A for black and blue lines in Figure 2). Here d_i is the departure time of the first bus in the time bin (see Section II-B2 for an example of TSR and d_i). This is determined by aggregating the boarding passengers for each TSR.

C. Affinity Propagation Clustering

The clustering is the first step in the prediction modeling due to the fact that a prediction model for each TSR is not an efficient way for a larger urban area and will be computationally expensive and practically not feasible to outline the model architecture of each TSR. Therefore, similar to the past study [20], based on the assumption that some of the stops would have similar boarding patterns and some other will have another patterns, a clustering is performed. In this way, not only the requirement of number of models and computational resources will reduce but will maintain a TSR patterns too.

At first, most common K-means/ K-centered clustering was tried, however, there was a lack of clarity on number of clusters using elbow-method. Thus, to avoid the need of determining the number of clusters in advance, affinity propagation clustering algorithm is used, which performs better than K-centered [38]. Affinity Propagation Clustering (APC) algorithm simultaneously considers all data points as potential exemplars, and it recursively transmits real-valued messages along edges of the network until a good set of centers and corresponding clusters are generated [39]. Affinity propagation has many advantages such as quick convergence, good precision, etc. The algorithm takes as input a collection of real-valued similarities between data points, where the similarity $s(\text{TSR}_{r,i}, \text{TSR}_{r,k})$ indicates how well the TSR_r with index k is suited to be the exemplar for TSR_r with index i.

This provides multiple clusters have similar boarding (loading) patterns and thus, a time-series model is developed for each cluster without losing the granularity. To identify a cluster, TSR and route details will be required.

D. Time Series Model

As discussed previously, the occupancy levels in PT have high temporal dependency. In this study, ETM data is used for crowding prediction. A gated Recurrent Neural Network (RNN) suits best for this kind of data. A commonly used example of gated RNN is Long Short Term Memory (LSTM). The architecture of the model with LSTM and time distributed fully connected layers can serve the purpose of short term prediction of the occupancy of future trips [35]. The combination of convolutional layers with LSTM layers allows taking spatial, temporal and exogenous dependencies into account as explained in next sections.

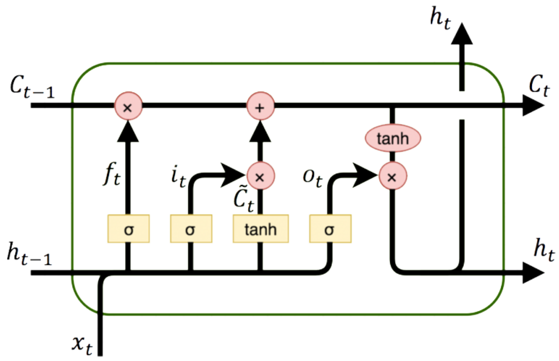


Fig. 3. LSTM unit architecture

1) LSTM unit architecture:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{5}$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{6}$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{7}$$
$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \tag{8}$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{9}$$
$$h_t = O_t \times \tanh(C_t) \tag{10}$$

The LSTM unit architecture is shown in Figure 3. From Equations (5) to (10), two activation functions are used; they are σ (sigmoid) and $\tanh$ (hyperbolic tangent) and are given by $\frac{1}{1+e^{-x}}$ and $\frac{e^x - e^{-x}}{e^x + e^{-x}}$ respectively. Equations (5) to (10) represent the forget gate value, input gate value, candidate value, direct output value, output gate value and new candidate value respectively. Ws and bs represent the weight matrices and bias vectors respectively.

2) LSTM model architecture:

$$x_t = [b_{\text{TSR}_i}, P_{r,n}] \tag{11}$$

Firstly, let X be the input matrix, which has slices as x_t; in other words, $X = \{x_1, x_2, x_3, ...x_T\}$. The slice of input matrix (x_t) is defined as shown in Equation (11), where extraction of features b_{TSR_i} and $P_{r,n}$ are explained in Section II-B. The size of the input matrix is $[N - \ell + 1, \ell, 2]$, where N is number of trips for the region including all transit routes, ℓ is lookback

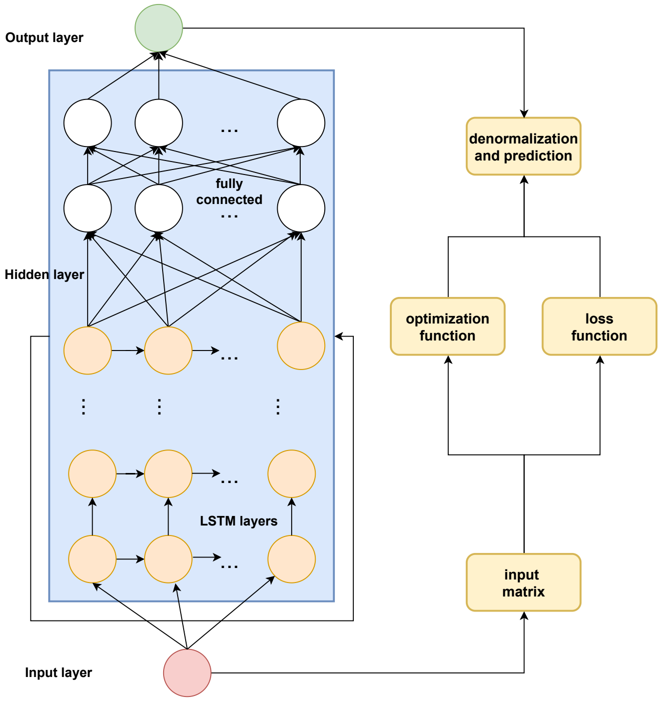


Fig. 4. LSTM model architecture

factor and last number denotes the number of features (i.e., 2). The input matrix is normalized (using min-max scaler) to bring the magnitudes of features (representing different entities) in the similar ranges.

Figure 4 shows the LSTM model architecture used in the present study. Combination of LSTM layers with some dropout layers in-between (if required) and fully connected layers are used for prediction purposes. Typically, there are many optimization functions available, such as, Stochastic Gradient Descent (SGD), Adagrad, AdaDelta, etc. However, in the present work, SGD is used due to the likelihood to reach to global minima compared to other optimization functions [40]. For the loss function, root mean squared error is used. In the final prediction layer, Rectified linear unit (ReLU) activation function is used to get output as predicted passenger counts.

E. Crowding API

At first, affinity propagation based clustering is performed (see Section II-C) to get the fewer cluster from a large number of possible TSRs in a real-world scenario. In other words, based on the similar temporal patterns, the need to have a unique model architecture for each transit stop is eliminated which is, in-general, not practical. For each cluster, a time-series model architecture is finalized (see Section II-D) which predicts the short-term crowding levels for each transit segment (i.e., number of passengers traveling between stop n and $n+1$). This combination enables to maintain the granularity at a transit stop level.

To facilitate the crowding levels for each TSR, a crowding API is developed. The trained model is picked based on the clusters of TSRs. The inputs for the crowding API are a series of transit stops (i.e., transit path coming from transit router, see Section III-B), connecting transit routes and stop-to-stop travel times. The stops are not necessarily belong to one transit route. From the transit stops, transit route and stop-to-stop travel time, TSRs are determined, which are used to identify the cluster. Since a transit stop can be part of various transit routes, transit route detail is also required to identify the cluster. Thus, corresponding to identified cluster, the trained time-series model is used to estimate the predicted number of persons in the transit vehicle. The stop-to-stop travel time and predicted number of persons are further used to estimate the crowding index.

A crowding index is defined to include the effect of seating and standing passengers between two consecutive stops for each transit path. As identified in the literature that traveling while standing is more burdensome than traveling in the sitting condition [27], [41], it is included in the travel cost function (see Equation (12)) of a transit path. The travel cost for a transit path is estimated at upstream transit stop (i.e., initiation of a segment); thus the travel cost is estimated from the first stop to $(n-1)^{\text{th}}$ stop.

$$C = \sum_{i=1}^{i=n-1} \left(A_i \cdot t_i + B_i \cdot t_i^2\right) \tag{12}$$

where, A and B are number of persons sitting and standing in a transit vehicle respectively. t is travel time from stop i to $i+1$. To compare the different transit choices, the crowding index for each transit choice (transit path) is defined in Equation (13).

$$\text{CI} = \frac{C}{C_{max}} \tag{13}$$

$$C_{max} = A_c \cdot \sum_{i=1}^{i=n-1} t_i + B_c \cdot \sum_{i=1}^{i=n-1} t_i^2 \tag{14}$$

where A_c and B_c are the sitting and standing capacities, which may vary depending on the bus type. The latter is estimated by taking the ratio of standing area and minimum area for one standing person (m^2/ person) thus, may also vary from region-to-region [27]. Eventually, crowding API returns the occupancy for each TSR and crowding index for the transit path, which may include transfers.

F. Realtime Occupancy extractor

In addition to the prediction model for the trip planning purposes, the real-time occupancy is also included in the passenger information system (see Section III). To add the functionality of real-time occupancy in each bus using the CCTV cameras installed in the transit vehicles, the present study integrates the algorithms from literature, which is briefly described in this section.

The detection of a person is done using the MobileNet SSD (Single-shot detector) Caffe model of OpenCV [42], [43][1]. For tracking a person inside the video file, the Centroid tracking algorithm [44] of OpenCV has been used and based on it, an object id is assigned to a person as long as the person is in

[1] see https://github.com/mailrocketsystems/AIComputerVision

the frame. If the person goes out of the frame, the object is de-registered. Then, the count of these ids are used to get the Live person count (LPC). In the final output, three parameters are available including FPS (frames per second), LPC (Live person count) and OPC (Overall person count). LPC is for the current number of persons present in the frame and OPC is for all the people who have been detected till now. Thus, to show the real-time occupancy in the transit vehicles, LPC is integrated in the passenger information system.

III. Passenger Information System

This work proposes an end-to-end solution in terms of a passenger information system, which facilitates trip planning while including the crowding levels at transit stops of each transit route. The overall schematic is shown in Figure 5 and explained in the following sections.

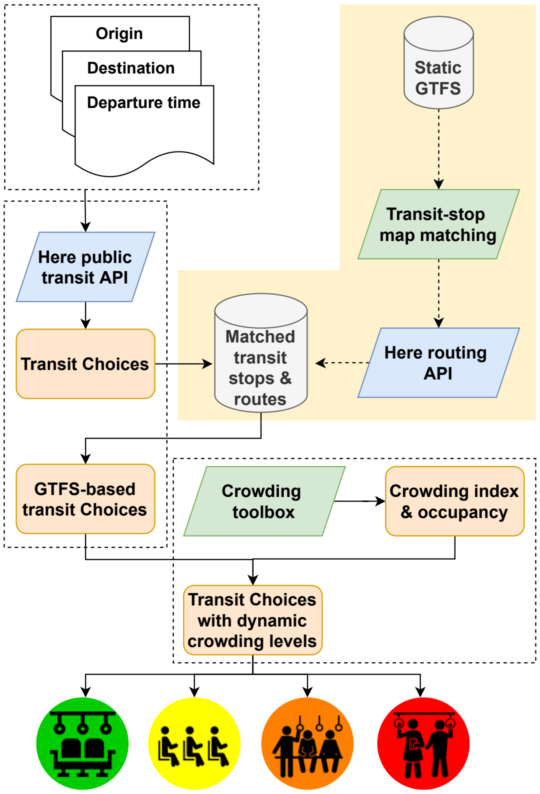

Origin
Destination
Departure time
Static GTFS
Transit-stop map matching
Here public transit API
Transit Choices
Matched transit stops & routes
Here routing API
GTFS-based transit Choices
Crowding toolbox
Crowding index & occupancy
Transit Choices with dynamic crowding levels

Fig. 5. Schematic of the proposed work

A. Transit stop map matching

At first, a transit-stop map-matching approach is developed. This will be executed only once in the passenger information system. As explained in Section II, the crowding index for each transit choice and occupancy for each transit segment are predicted using the CB-LSTM model (see Sections II-C and II-D). For the prediction model, ETM data is used which is based on GTFS. On the other hand, to get the transit choices from origin and destination, Here public transit API is used and the resulting choices does not match with the GTFS and are incomplete (absence of intermediate transit stops).[2] Clearly, a query to crowding toolbox will be successful only if transit route from the router has same route name/ number, stop name/ number as that of the GTFS. Therefore, there is a need to match the transit routes from HERE public transit API with GTFS.

The steps for transit-stop map-matching approach are shown in Figure 5 and are described below.

(a) Firstly, for each transit route in GTFS, a unique, non-overlapping section is identified (manually, by looking on the map)[3].
(b) The unique section is then used as input to Here public transit API (i.e., at most one transit route).
(c) The resulting route from Here public transit API is mapped to the transit route in GTFS.
(d) In absence of the departure times at stops in GTFS, stop-to-stop travel times are determined using Here routing API[4] version 7.

Thus, the product of this step is a database which matches the transit routes of two different sources and include stop-to-stop travel times. Thus, the database is named as “ matched transit stop & routes database”. This database is used for every query on passenger information system as shown in Figure 5.

B. Transit routing

In order to provide various choices to the passengers, a transit router is required. Traditional transit router provides transit routes based on the schedule and may also include the real-time congestion pattern. Since development of a transit router is beyond the scope of this work, an existing transit router is used as a placeholder and modified as per the need to match with the available GTFS. For a better and efficient implementation, a GTFS-based configurable routers (e.g., RAPTOR [46]) can be used.

In the present work, HERE public transit API[5], version 8 is used to get the transit choices for a given origin-destination pair and departure time. These choices are then passed from the matched transit stop & routes database, which provides transit choices as per GTFS.

C. Integration of crowding levels in the transit choices

The crowding index and occupancy for each transit choice are the output of crowding API (see Section II-E). To get these, a request is sent to crowding API; the request consists of the series of transit stops, connecting routes and stop-to-stop travel times. The API returns the crowding index for each GTFS-based transit choice and predicted occupancy for each segment of the transit choice.

After getting the crowding indices for all transit choices, the choices are sorted based on the crowding indices and showed to the users in increasing order of crowding index. For users’ convenience, the passenger information system

[2] Use of Here public transit API is not essential, see Section III-B.

[3] In contrast to the manual work, for a larger scenario this can be scripted, see [45].

[4] See https://developer.here.com/documentation/routing/dev_guide/topics/introduction.html

[5] see https://developer.here.com/documentation/public-transit/dev_guide/routing/index.html

shows only four distinct crowding levels. They are “many seats available”, “few seats available”, “many standing spaces available”, “few standing spaces available” (see Figure 5). Though, it is possible to configure the numbers to classify the crowding levels for a transit choice. The present work simply split the crowding index in four equal ranges to show the four crowding levels for a transit choice. Higher is the crowding index, fewer sitting/ standing spaces are available for most of the segments of the transit choice.

If the user disable the option to show the choices based on crowding index, the earliest departure choice is shown first. The occupancy levels for all transit stops are shown if a user opt to see the details by selecting one of the transit choice.

IV. Case study: Bhubaneswar, India

A. *Overview of study area*

In the present study, Bhubaneswar is selected as a case study due to the availability of the historical ticketing data. Bhubaneswar is the capital of Odisha, named as ‘Temple city’ of India. Currently, more than 1.1. million persons lives in Bhubaneswar.

In Bhubaneswar, only bus service (called as ‘MO’ bus) is available as a public transit system and operated by Capital Region Urban Transport (CRUT) on 40 routes.

B. *ETM and GTFS data*

For this work, three months (Dec. 2019 to Mar. 2020) of ETM data is used. The ETM data is provided by CRUT Bhubaneswar and static GTFS data is obtained from [47]. The transit vehicles include Air-conditioned (AC) and non-AC buses with varying capacities (e.g., 43 seats in regular buses and 23 seats in midi buses). However, only part of the data (i.e., vehicle type) is available for Feb. 2020 and therefore, a uniform seating and standing capacities are used in the present study; they are 40 and 22 respectively. In case of different seating capacities, it can be taken as input to the crowding API (see Section II-E), which will use it to determine the crowding indices. It will not affect the prediction model.

The ETM data has details like ticket date, depot name, bus number, route number, trip number, mode of payment, origin stop id, destination stop id, passenger type, passenger count, etc. There are 375 stations in the GTFS data. Clearly, many of those stations offer the transfer functionality (serve multiple routes) at different time of the day. This highlights the need of transit stop relations (see Section II-B2).

Figure 6 shows the average hourly passenger volume for all buses putting together at different days of the week. It exhibits the typical two-peak patterns, and has minor variation for different days of the weekdays. The morning peak is smaller than evening peak for weekends, which is also a commonly known pattern.

Figures 7 and 8 exhibit the variation in the ridership index for different stops of transit route 11 (starting at Nandan Vihar) and for different days of the week at stop Aacharya Vihar of the transit route 11, respectively. The former highlights the spatial and temporal variations over the transit route. It can also be observed that the ridership indices of multiple columns are similar (i.e., common color patterns of multiple columns), which highlights that a common model can be developed for such transit stops. Thus, a clustering model is developed. Further, the latter figure demonstrates the variation for times of the days and day of the week for a transit stop. Clearly, this emphasizes the need of spatio-temporal prediction model and therefore, the clustering model is followed by a time-series mode.

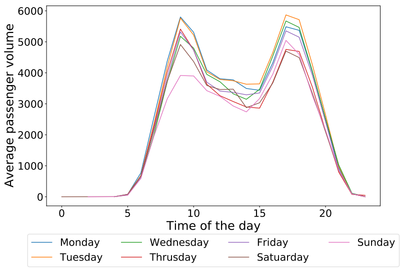


Fig. 6. Average passenger volume during days of the week at different times of the day

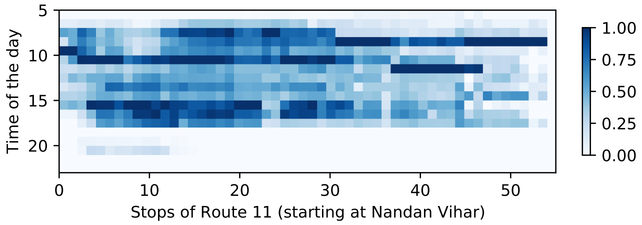


Fig. 7. Variation of ridership index for transit route 11 on 6th Jan 2021

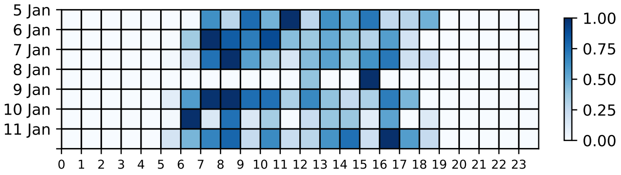


Fig. 8. Variation of ridership index at different time of the day and day of the week for Aacharya Vihar transit stop

C. *Clustering output*

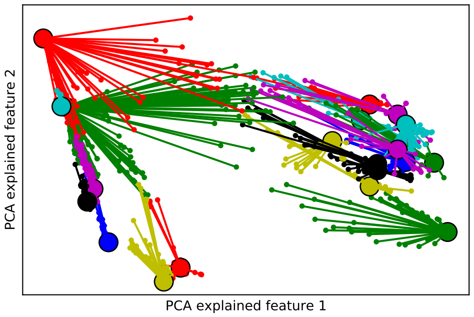


Fig. 9. Simplified representation of clusters (affinity propagation)

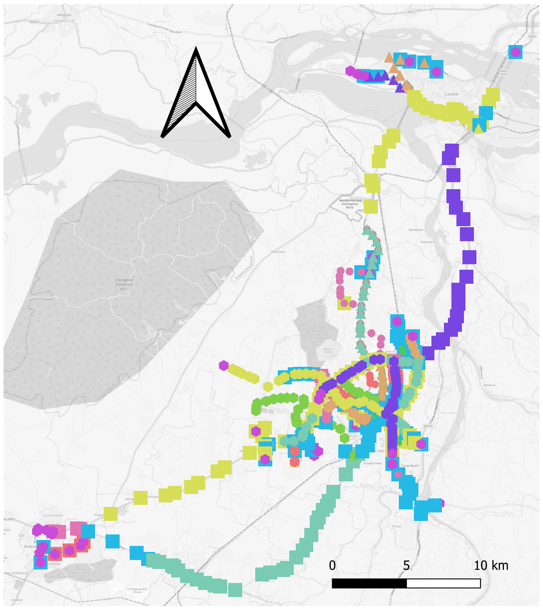
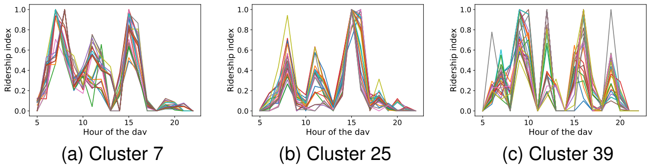


Fig. 10. Bus stops in Bhubaneswar, different color shows the different cluster.

From 375 transit stops, 1644 unique relations for each transit stop and transit route are prepared (if a stop is served by 5 transit routes, it will be counted 5 times). For the given data, total number of timebins are 2208; removing non-operation hours leaves 1656 timebins. Thus, in total $\mathrm{RI}_{\mathrm{TSR},r}$ are about 2.7 million. On the application of affinity propagation (AP) clustering, 41 clusters are formed. The affinity between the transit tops to form a cluster can be visualized from Figure 9; it shows the simplified representation of 21 clusters. For this, the Principal Component Analysis (PCA) is employed, which explains the clusters using two features only.

Figure 10 shows the result of AP clustering. To show the spatial distribution of 41 clusters from 1644 unique relations for each transit stop and transit route, different shapes (e.g., circle, triangle, square, etc.) and colors are used. Different sizes are used to show overlapping transit stops. It can be observed that due to affinity, the transit stops from one of the transit route may belong to different clusters and same stop from different transit routes may belong to different clusters (i.e., overlapping points in Figure 10).



(a) Cluster 7

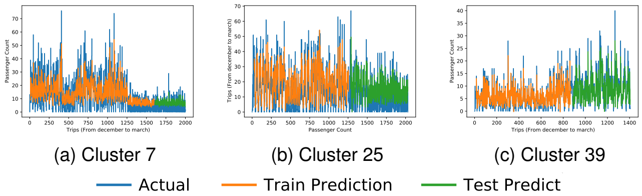
(b) Cluster 25

(c) Cluster 39

Fig. 11. A temporal distribution of ridership indices for 6th Jan.

Finally, a temporal distribution of ridership indices is shown in Figure 11. Clearly, to form the clusters, whole data is used but for representation purpose, three clusters are showing three different temporal distributions for a day. In each cluster, the ridership patterns for different TSRs look similar, which confirms the validation of results from the affinity propagation clustering.

D. LSTM output



(a) Cluster 7

(b) Cluster 25

(c) Cluster 39

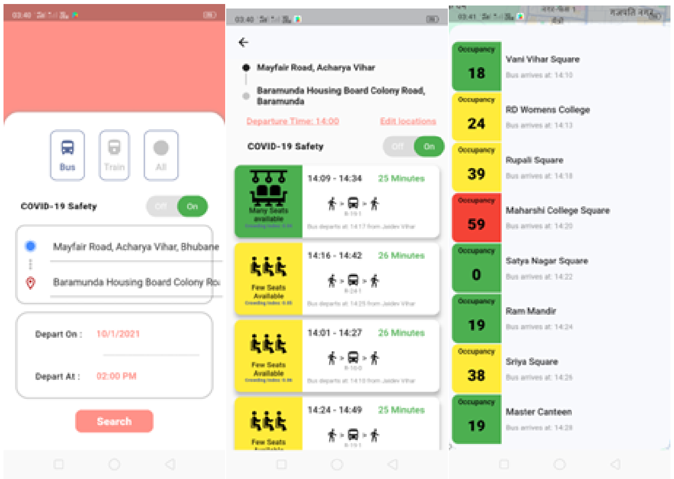
Fig. 12. Prediction results on train and test data for one of the randomly selected TSR in each cluster (7, 25 and 39)

An LSTM model is trained for each cluster. Model architecture of each cluster in the time-series model is decided so as to get the minimum root mean squared error (RMSE) for each TSR lying in the cluster. For the training, various hyper tuning parameters are used. For instance, cluster 39 has 29 TSRs, which performs good with one hidden layer with 32 nodes. In this, the lookback factor (ℓ) is taken as 15. Finally, a fully connected layer has two nodes corresponding to passenger count and boarding TSR. The average RMSE values for cluster 7, 25 and 39 on training data are 4.061, 5.039 and 4.305 respectively. The same on test data are 2.776, 3.837 and 4.425 respectively. Clearly, the lower RMSE values highlights the good performance of these models, which has simple architecture. With an increasing level of complexity in the model, the accuracy may be even higher.

The LSTM results for one of the randomly selected TSR in each cluster are illustrated in Figure 12. For instance, in cluster 7, TSR_{30} of transit route 20 in downstream direction is selected. The RMSE values on train and test data for this TSR are 3.44 & 2.19 respectively. Similarly, the RMSE values for the selected TSR in cluster 25 and 39 are 3.87 & 3.34 and 2.12 & 2.46 respectively. The prediction results for clusters 25 and 39 are in good sync with the actual data. However, the prediction on part of the train and test data in cluster 7 is unable to accurately predict the very low passenger counts. It is happening due to the sudden change in the pattern (refer to blue lines after 2 months of the ETM data in Figure 12a), and plausibly, can be eliminated with longer duration data.

E. Transit choices with dynamic crowding levels

As indicated in the Section III and demonstrated in Figure 5, the final product is a passenger information system (see Figure 13). In this, PIS, a passenger can enter origin, destination and departure time (see Figure 13a). The resulting interface shows the various transit choices (see Figure 13b), which are differentiated by the crowding levels by default. To represent the crowding levels intuitively, different colors and images are used. Further, Figure 13c shows the interface after selecting the one of the choices. In this, the occupancy for all transit segments are shown, which gives more clarity for the crowding levels on the transit routes. If the passenger would like to take



COVID-19 Safety

Mayfair Road, Acharya Vihar, Bhubane

Baramunda Housing Board Colony Ro:

Depart On : 10/1/2021

Depart At : 02:00 PM

Mayfair Road, Acharya Vihar

Baramunda Housing Board Colony Road, Baramunda

Departure Time: 14:00

COVID-19 Safety

14:09 - 14:34 25 Minutes — Many Seats available — Bus departs at 14:17 from Jaidev Vihar

14:16 - 14:42 26 Minutes — Few Seats Available — Bus departs at 14:25 from Jaidev Vihar

14:01 - 14:27 26 Minutes — Few Seats Available — Bus departs at 14:10 from Jaidev Vihar

14:24 - 14:49 25 Minutes — Few Seats

Occupancy	Stop
18	Vani Vihar Square Bus arrives at: 14:10
24	RD Womens College Bus arrives at: 14:13
39	Rupali Square Bus arrives at: 14:18
59	Maharshi College Square Bus arrives at: 14:20
0	Satya Nagar Square Bus arrives at: 14:22
19	Ram Mandir Bus arrives at: 14:24
38	Sriya Square Bus arrives at: 14:26
19	Master Canteen Bus arrives at: 14:28

(a) Input interface (b) Transit choices (c) Occupancy details

Fig. 13. Inputs and results summary interfaces of PIS

the earliest available bus, the ‘covid-19-safety’ switch can be turned off.

V. Conclusion

To facilitate the passenger in planning their trips as per the dynamic crowding levels in the public transit system, this study proposed a real-time passenger information system (PIS) and integrated a crowding prediction model in it. The proposed prediction model used affinity propagation for clustering of transit segment relations (TSR) using ridership indices. Afterward, an LSTM model is developed for each cluster which returns the dynamic passenger counts on each transit route.

Three months of ETM data for public bus system of Bhubaneswar, India was used to demonstrate the PIS framework. The lower RMSE values of the models showed good performance of the proposed model and emphasized the potential of the proposed framework for other cities.

From the perspective of transit operators, the proposed crowding prediction model can be used for optimal allocation of resources. From the passengers’ point of view, the use of this PIS can help them in reducing the waiting times at the transit stop as well as in opting for the right departure time and transit route to experience the lesser crowding. The PIS framework is structured in a way that the real-time crowding levels can also be integrated, which will be useful for the last minute trips. In future, the authors wish to (a) use a dedicated GTFS-transit router (b) integrate a dynamic bus scheduling tool for the operators and (c) incorporate real-time crowding levels in the PIS.

References

[1] A. Tirachini, D. A. Hensher, and J. M. Rose, “Crowding in public transport systems: effects on users, operation and implications for the estimation of demand,” *Transportation Research Part A*, vol. 53, pp. 36–52, 2013.

[2] K. Aghabayk, J. Esmailpour, and N. Shiwakoti, “Effects of COVID-19 on rail passengers’ crowding perceptions,” *Transportation Research Part A*, vol. 154, pp. 186–202, 2021.

[3] TERI, “Impact of COVID-19 on urban mobility in India: evidence from a perception study. New Delhi,” The Energy and Resources Institute (TERI), Tech. Rep., 2020.

[4] A. Thombre and A. Agarwal, “A paradigm shift in urban mobility: policy insights from travel before and after COVID-19 to seize the opportunity,” *Transport Policy*, vol. 110, pp. 335–353, 2021.

[5] H. K. Suman, A. Agarwal, and N. B. Bolia, “Public transport operations after lockdown: how to make it happen?” *Transactions of the Indian National Academy of Engineering*, vol. 5, no. 2, pp. 149–156, 2020.

[6] J. D. Schmöcker, F. Kurauchi, and H. Shimamoto, “An overview on opportunities and challenges of smart card data analysis,” in *Public transport planning with smart card data*, F. Kurauchi and J.-D. Schmöcker, Eds., 2017.

[7] Z. Chen and W. Fan, “Extracting bus transit boarding stop information using smart card transaction data,” *Journal of Modern Transportation*, vol. 26, no. 3, pp. 209–219, 2018.

[8] C. Morency, M. Trépanier, and B. Agard, “Measuring transit use variability with smart-card data,” *Transport Policy*, vol. 14, no. 3, pp. 193–203, 2007.

[9] X. Ma, Y. J. Wu, Y. Wang, F. Chen, and J. Liu, “Mining smart card data for transit riders’ travel patterns,” *Transportation Research Part C*, vol. 36, pp. 1–12, 2013.

[10] Q. Zou, X. Yao, P. Zhao, H. Wei, and H. Ren, “Detecting home location and trip purposes for cardholders by mining smart card transaction data in Beijing subway,” *Transportation*, vol. 45, no. 3, pp. 919–944, 2016.

[11] A. Alsger, A. Tavassoli, M. Mesbah, L. Ferreira, and M. Hickman, “Public transport trip purpose inference using smart card fare data,” *Transportation Research Part C*, vol. 87, pp. 123–137, 2018.

[12] J. Jung and K. Sohn, “Deep-learning architecture to forecast destinations of bus passengers from entry-only smart-card data,” *IET Intelligent Transport Systems*, vol. 11, no. 6, pp. 334–339, 2017.

[13] Y. Zhang and T. Cheng, “A deep learning approach to infer employment status of passengers by using smart card data,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 2, pp. 617–629, 2020.

[14] Z. Xie, J. Zhu, F. Wang, W. Li, and T. Wang, “Long short-term memory based anomaly detection: A case study of China railway passenger ticketing system,” *IET Intelligent Transport Systems*, vol. 15, no. 1, pp. 98–106, 2020.

[15] H. Yu, D. Chen, Z. Wu, X. Ma, and Y. Wang, “Headway-based bus bunching prediction using transit smart card data,” *Transportation Research Part C*, vol. 72, pp. 45–59, 2016.

[16] Z. Gong, B. Du, Z. Liu, W. Zeng, P. Perez, and K. Wu, “SD-seq2seq : a deep learning model for bus bunching prediction based on smart card data,” in 29th *International Conference on Computer Communications and Networks (ICCCN)*, 2020, pp. 1–9.

[17] A. A. Alsger, M. Mesbah, L. Ferreira, and H. Safi, “Use of smart card fare data to estimate public transport origin–destination matrix,” *Transportation Research Record*, vol. 2535, no. 1, pp. 88–96, 2015.

[18] W. Jiang, Z. Ma, and H. N. Koutsopoulos, “Deep learning for short-term origin–destination passenger flow prediction under partial observability in urban railway systems,” *Neural Computing and Applications*, vol. 34, no. 6, pp. 4813–4830, 2022.

[19] S. Liang, M. Ma, S. He, and H. Zhang, “Short-term passenger flow prediction in urban public transport: Kalman filtering combined K-nearest neighbor approach,” *IEEE Access*, 2019.

[20] J. Zhang, F. Chen, and Q. Shen, “Cluster-based LSTM network for short-term passenger flow forecasting in urban rail transit,” *IEEE Access*, vol. 7, pp. 147 653–147 671, 2019.

[21] E. Chen, Z. Ye, C. Wang, and M. Xu, “Subway passenger flow prediction for special events using smart card data,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1109–1120, 2020.

[22] X. Yang, Q. Xue, M. Ding, J. Wu, and Z. Gao, “Short-term prediction of passenger volume for urban rail systems: A deep learning approach based on smart-card data,” *International Journal of Production Economics*, vol. 231, p. 107920, 2021.

[23] Y. Xu and K. Jin, “An LSTM approach for predicting the short-time passenger flow of urban bus,” in 2nd *International Conference on Artificial Intelligence in Electronics Engineering*, 2021.

[24] T. D. Sajanraj, J. Mulerikkal, S. Raghavendra, R. Vinith, and V. Fábera, “Passenger flow prediction from AFC data using station memorizing LSTM for metro rail systems,” *Neural Network World*, vol. 31, no. 3, pp. 173–189, 2021.

[25] N. van Oort, T. Brands, and E. de Romph, “Short-term prediction of ridership on public transport with smart card data,” *Transportation Research Record*, no. 2535, pp. 105–111, 2019.

[26] M. Dou, T. He, H. Yin, X. Zhou, Z. Chen, and B. Luo, "Predicting passengers in public transportation using smart card data," in *Lecture Notes in Computer Science*, 2015, pp. 28–40.

[27] Z. Li and D. Hensher, "Crowding in public transport: a review of objective and subjective measures," *Journal of Public Transportation*, vol. 16, no. 2, pp. 107–134, 2013.

[28] Z. Zuo, W. Yin, G. Yang, Y. Zhang, J. Yin, and H. Ge, "Determination of bus crowding coefficient based on passenger flow forecasting," *Journal of Advanced Transportation*, vol. 2019, pp. 1–12, 2019.

[29] L. Liu and R. C. Chen, "A novel passenger flow prediction model using deep learning methods," *Transportation Research Part C*, vol. 84, pp. 74–91, 2017.

[30] T. Arabghalizi and A. Labrinidis, "Data-driven bus crowding prediction models using context-specific features," *ACM/ IMS Transactions on Data Science*, 2020.

[31] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[32] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, "LSTM network: a deep learning approach for short-term traffic forecast," *IET Intelligent Transport Systems*, vol. 11, no. 2, pp. 68–75, 2017.

[33] C. Zhang, L. Zhang, Y. Liu, and X. Yang, "Short-term prediction of bike-sharing usage considering public transport: a LSTM approach," in *21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 1564–1571.

[34] V. Mittal, S. Sasetty, R. Choudhary, and A. Agarwal, "Deep-learning spatio-temporal prediction framework for PM under dynamic monitoring," *Transportation Research Record*, 2022.

[35] K. Pasini, M. Khouadjia, A. Samé, F. Ganansia, and L. Oukhellou, "LSTM encoder-predictor for short-term train load forecasting," in *Machine Learning and Knowledge Discovery in Databases*, U. Brefeld, E. Fromont, A. Hotho, A. Knobbe, M. Maathuis, and C. Robardet, Eds., 2020, pp. 535–551.

[36] Q. Han, K. Liu, L. Zeng, G. He, L. Ye, and F. Li, "A bus arrival time prediction method based on position calibration and lstm," *IEEE Access*, vol. 8, pp. 42 372–42 383, 2020.

[37] S. Halyal, R. H. Mulangi, and M. M. Harsha, "Forecasting public transit passenger demand: with neural networks using APC data," *Case Studies on Transport Policy*, 2022.

[38] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *SCIENCE*, vol. 315, no. 5814, pp. 972–976, 2007. [Online]. Available: http://utstat.toronto.edu/reid/sta414/frey-affinity.pdf

[39] P. Thavikulwat, "Affinity propagation: A clustering algorithm for computer-assisted business simulations and experiential exercises," in *Developments in Business Simulation and Experiential Learning: Proceedings of the Annual ABSEL conference*, vol. 35, 2008.

[40] S. Ruder, "An overview of gradient descent optimization algorithms," 2016.

[41] A. Tirachini, R. Hurtubia, T. Dekker, and R. A. Daziano, "Estimation of crowding discomfort in public transport: Results from santiago de chile," *Transportation Research Part A*, vol. 103, pp. 311–326, 2017.

[42] Y. C. Chiu, C. Y. Tsai, M. D. Ruan, G. Y. Shen, and T. T. Lee, "Mobilenet-ssdv2: an improved object detection model for embedded systems," in *International conference on system science and engineering (ICSSE)*, 2020, pp. 1–5.

[43] A. Younis, L. Shixin, S. Jn, and Z. Hai, "Real-time object detection using pre-trained deep learning models MobileNet-SSD," in *6th International Conference on Computing and Data Engineering*, 2020.

[44] J. C. Nascimento, A. J. Abrantes, and J. S. Marques, "An algorithm for centroid-based tracking of moving objects," in *IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*. IEEE, 1999.

[45] R. Choudhary and A. Agarwal, "Bus route optimization for dynamic monitoring network to maximize spatio-temporal coverage," Indian Institute of Technology Roorkee, Tech. Rep. WP #46, 2022, uRL http://faculty.iitr.ac.in/~amitfce/publications.html.

[46] D. Delling, T. Pajor, and R. F. Werneck, "Round-based public transit routing," *Transportation Science*, vol. 49, no. 3, pp. 591–604, 2015.

[47] Dataspace, "City bus Bhubaneswar - GTFS static," accessed 2021. [Online]. Available: https://dataspace.mobi/dataset/city-bus-bhubaneswar-gtfs-static

Arpit Shrivastava got his B. Tech. in the Department of Civil Engineering, IIT Roorkee, India in 2022.

Devesh Pratap Singh got his B. Tech. in the Department of Civil Engineering, IIT Roorkee, India in 2022.

Itisha Jain got her B. Tech. in the Department of Civil Engineering, IIT Roorkee, India in 2022.

Amit Agarwal received his B.Tech. in Civil Engineering from MNIT Jaipur in 2009, M.Tech. in Transportation Engineering from IIT Delhi in 2012 and Ph.D. from TU Berlin in 2017. He is currently Assistant Professor at the Department of Civil Engineering and Joint Faculty at Mehta Family School of Data Science and Artificial Intelligence, IIT Roorkee, India. He has (co-)authored several papers in reputed leading journals. His research interests include air pollution exposure, crowdsourced data, shared mobility, public transport, MATSim, etc.