Journal of Network and Computer Applications 164 (2020) 102692

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

MARIO: A spatio-temporal data mining framework on Google Cloud to explore mobility dynamics from taxi trajectories

Shreya Ghosh [a,*], Soumya K. Ghosh [a], Rajkumar Buyya [b]

[a] Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India
[b] Cloud Computing and Distributed Systems (CLOUDS) Laboratory, School of Computing and Information Systems, The University of Melbourne, Australia

ARTICLE INFO

Keywords:
Trajectory
Spatio-temporal data analysis
Association rule mining
Deep learning
Travel demand prediction

ABSTRACT

With the major advances in location acquisition techniques, deployment of GPS enabled devices and increasing number of mobile users, substantial amount of location traces are generated from different geographical regions. It provides unprecedented opportunities to analyze and derive valuable insights of urban dynamics, specifically, *time-dependent mobility patterns* and *region-specific travel demands*. This work proposes an end-to-end mobility association rule mining framework called *MARIO*, conducive to extract urban mobility dynamics through analysing large taxi trip traces of a city. The MARIO framework consists of (i) generating mobility-dynamics network by spatio-temporal analysis of taxi-trips, (ii) finding travel demand variations in different functional regions of the urban area, (iii) extracting mobility association rules and (iv) predicting travel demands and traffic dynamics using extracted associative rules. The proposed MARIO framework is implemented in *Google Cloud Platform* and an extensive set of experiments using real GPS trace dataset of *NYC Green and Yellow Taxi trace*, *Roma Taxi Dataset* and *San Francisco Taxi Dataset* have been carried out to demonstrate the effectiveness of the framework. The performance of the proposed approach is significantly better than the baseline methods in predicting travel demands (with the reduction of average MAPE value and execution time by 50%).

1. Introduction

The ever-increasing popularity of wireless communications, sensor technologies and GPS embedded devices have motivated extensive research on analysing the enormous amount of GPS traces and various location-aware applications. The location traces of moving agents (GPS equipped private vehicles, mobile sensors of individuals, public transportation data etc.) are accumulated easily and effectively leading to generation of huge amount of location traces. This increasing availability of human location traces (also may be defined as 'human trail') from various sources has opened up interesting research directions like human behaviour/activity learning, studying city dynamics, improved route planning, resource allocation and traffic analysis (Zheng, 2015).

This paper aims to discover overall mobility patterns of a city analyzing taxi-traces to build a mobility-knowledge base (associative patterns) which captures the inherent urban-dynamics and utilize the knowledge to another region for better urban planning. Notably, the public transportation vehicles provide predictable patterns as they follow fixed routes and regular time schedules. Further, private vehicles offer fairly expected routes as the traces depict the pattern of the owner (say, commuting from home to work). On the other side, GPS enabled taxi-cabs serve diverse transportation needs of a large population and contributes improving urban transportation planning and thus acts as an integral part of intelligent transportation system. However, there exists a well-known dilemma that trajectory data are rich in terms of volume but activity information poor (Gong et al., 2016). This gap can be mitigated if contextual information such as functional regions or landmarks of the taken route, activities at stay points are considered instead of analysing raw GPS log (timestamped latitude, longitude data). As the pattern is likely to be dependent on start and destination functional regions of the cities, more specifically on travel purposes, this paper incorporates the 'POI' information (ex., industrial, residential etc.) of the trips along with the spatio-temporal attributes. In summary, our framework generates mobility knowledge base by extracting prevalent mobility association rules from city taxi-trips and predicts possible travel demand spikes.

Motivating example: Fig. 1 shows a motivating example for the proposed *MARIO* framework. Each POI (point-of-interest) or functional

* Corresponding author.
E-mail addresses: shreya.cst@gmail.com (S. Ghosh), shreya2015@iitkgp.ac.in (S. Ghosh), skg@iitkgp.ac.in (S.K. Ghosh), rbuyya@unimelb.edu.au (R. Buyya).

https://doi.org/10.1016/j.jnca.2020.102692
Received 1 October 2019; Received in revised form 14 March 2020; Accepted 29 April 2020
Available online 11 May 2020

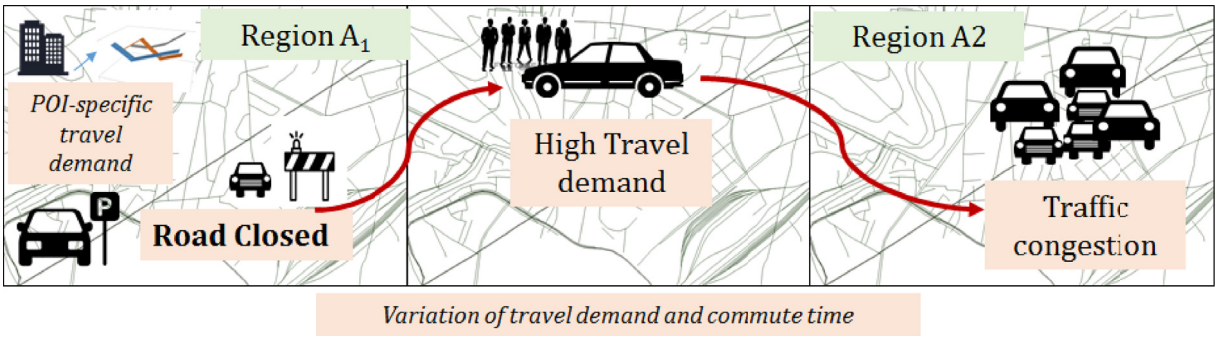

Fig. 1. *Example scenario:* Different mobility events, like road-closure, high travel demand effect other regions.

region depicts travel demand pattern in different temporal scales. Various mobility events effect other neighboring regions in a temporal sequence and subsequently it helps to predict travel demand and travel time efficiently.

Our framework generates mobility knowledge base by extracting prevalent mobility association rules from city taxi-trips and predicts possible travel demand spikes. For a given city, the proposed *MARIO* framework may explore the rules like (R1): *When the region A_1 of a city experiences traffic congestion, then with 80% probability the traffic density of region A_2 will be higher after δ time-period and consequently high travel-demands in the spatial-neighborhood region of A_2.* R1 reflects both the spatial and temporal neighborhood traffic states and the associative flow-patterns among different regions of the city. The mobility association rule mining problem becomes quite challenging for several spatio-temporal resolutions, i.e, *an infrequent item-set for the entire region or large time-interval may be a frequent item within a part of the region or a relatively small time-quantam.* Furthermore, mobility state of one region effects the neighboring regions in a temporal sequence. Therefore, there is a strong need to learn the traffic-flow inter-dependencies of the regions in different temporal scales and partition the city-network into smaller regions.

Contributions: To address the above mentioned challenges and issues, we propose mobility rule miner *MARIO* framework which involves

1. generation of mobility dynamics network by segmenting the large city road network and sub-graph decomposition to discover traffic-flow in different regions of the city
2. proposing a variant of *Apriori* algorithm utilizing frequency tree and deep learning architecture to extract prevalent mobility rules by analysing spatio-temporal neighborhood effects
3. extracting mobility rules, namely, region-specific mobility association rules, i.e. mobility dynamics (crowd behaviour, traffic flow and travel-demand) of different functional regions of a city, taxi-trip specific or individualized mobility patterns to capture the overall urban dynamics
4. predicting travel demands effectively and timely manner in different regions of the city

In summary, we have implemented an end-to-end trajectory analysis service over Google Cloud Platform (GCP) by creating interfaces with varied cloud services and vCPUs to explore the mobility dynamics of a city. The remainder of this article is organized as follows. Section 2 presents a review of recent studies in this direction. A few preliminary concepts of the work and different modules of the framework are presented in section 3 and 4 respectively. Section 5 depicts the experimental set-up and detailed discussion and evaluations of the proposed method. Finally, section 6 summarizes the study and discusses some avenues of future research direction.

2. Related work

Understanding and discovering mobility patterns facilitate better urban planning and traffic resource management. In this section, we brief the existing works on spatio-temporal association rule mining and applications of mobility trace analysis.

In short, spatial data mining aims to extract patterns which are previously unknown but potentially useful. Spatial association rule mining (Koperski and Han, 1995) discovers the frequent co-occurrences of spatial predicates, such as *adjacent_to*, *nearby* and objects, namely, *highway* and *house*. The existing works follow two strategies to extract spatio-temporal patterns: transaction-based strategy and transaction-free strategy. In transaction-based strategy, spatio-temporal data is transformed into a format of *transactions* such that traditional association rule mining technique can be deployed. These transactions are created by partitioning the study regions into different sub-regions, where the sub-regions are transactions and features in these sub-regions correspond to items (Lee, 2004; Appice et al., 2003). In the transaction-free strategy, spatial points are considered as *Boolean spatial features* (Huang et al., 2004) i.e, the presence or absence of spatial phenomenons. Another strand of research in this domain focuses on *spatio-temporal events*, such as state-based events, location-based events and change-based events (Liu et al., 2016). In location-based events (Mohan et al., 2011), spatial locations are analysed where events, such as vehicle collision, typhoon, flood occurred. Spatio-temporal association pattern reveals the connection between two such events. *However, the temporal dynamics (temporal relation between two such correlated events) of the events can not be extracted from the existing approaches. We have not found any consensus regarding the definition of spatio-temporal events and further, no such existing works delineate events related to mobility behaviours of people or vehicles.* There are very few works which explore associative patterns in human movement traces. Yang et al. (Ye et al., 2009) present a framework to extract association rules from individual mobility traces. For instance, they extract rules like “In 70% of the days, person X visits POI Y; or visits shopping mall once in a week”. In our previous works (Ghosh and Ghosh, 2016; Ghosh and Ghosh, 2017), we present the association rule mining technique from individuals’ trajectory traces, however, these approaches are not suitable for extracting aggregate movement patterns city-wide. In our present work, we study the mobility patterns thoroughly and detail the *mobility event* definition to deploy the mobility rule miner (MARIO) framework. All of the existing works mainly focus on relationships among diversity of spatial data-types (Dao and Thill, 2016; Mohan et al., 2011) and finding spatial autocorrelation (Barua and Sander, 2013). A variant of Apriori algorithm named *T-Apriori* is proposed to discover association rules in interval based data in Chen and Wu (2006). There is existing work in protecting the privacy of users’ data (Shen et al., 2017). The work considers location semantic diversity and randomness of query behaviour to protect the trajectory

privacy. Cao et al. (2018) present a classification model for location prediction based on user check-in patterns. Rosen Ivanov (2012) presents a novel algorithm for real-time GPS track simplification, which in turn helps outdoor navigation of visually impaired people. There are a broad range of applications on GPS data - a systematic survey (Pirozmand et al., 2014) on human mobility discusses the mobility features and prediction techniques. There are also works on effective forecasting of traffic flows (Zhang et al., 2017; Zhang et al., 2019). The authors (Zhang et al., 2017) propose a deep learning approach to forecast crowd flows in different regions of a city using several factors such as *weather* and *intra-region traffic*. A multi-task deep learning framework is proposed in Zhang et al. (2019) where both the node flow and edge flow are predicted. The authors consider the fact that flow at a node and transitions between nodes (edge flow) are dependent to each other. However, this work does not explore the spatial and temporal neighborhood effects of such crowd flows. Zhang et al. (2015) formulate a new problem of mining spatial co-evolving patterns from several geo-sensory data. The authors explore different sensor-records such as air-quality, bike/vehicle data; and finds out co-evolving patterns by assembling the individual sensors' patterns into a single pattern. Our problem set-up is quite different from this work, where instead of finding co-evolving patterns from several sensory information, the patterns are extracted and analysed only from GPS trajectories. Discovering the spatio-temporal dynamics is one of the most challenging issue in our mobility analysis work. It is useful to predict the traffic condition apriori to build an effective traffic recommendation system. Most of the studies (Akbari et al., 2015; Verhein and Chawla, 2006) model the geographical phenomenon (air-pollution or typhoons) as simple point-events. However, in our framework, dynamic characteristics of the moving agents and their interrelationships need to be extracted efficiently.

Table 1 summarizes the features of MARIO and other most relevant existing works. To the best of our knowledge, there are few works on spatio-temporal association rule mining but none have reported for extracting mobility association rules from taxi-trips. Further, the indexing scheme of MARIO, mobility-dynamics network based on traffic flow dynamics and analysing sptio-temporal neighborhood by deep LSTM architecture are novel propositions. In summary, MARIO provides an end-to-end framework to model mobility dynamics, extract mobility association rules and predict travel demand efficiently.

Table 1
Comparisons of existing works and MARIO framework.

Feature	Related Works					MARIO (Proposed Framework)
	Akbari et al. (2015), Verhein and Chawla (2006), Dao and Thill (2016), Koperski and Han (1995) and Appice et al. (2003)	Cudre-Mauroux et al. (2010), Zhou et al. (2013) and Zheng et al. (2008)	Gong et al. (2016), Zhang et al. (2016a) and Yao et al. (2018)	Kong et al. (2017)	Ghosh and Ghosh (2017)	
Mining Spatial/Spatio-temporal Association rule	✓	×	×	×	✓	✓
Mining Mobility Association rule (individual)	×	×	×	×	✓	✓
Mining Mobility Association rule (city-wide)	×	×	×	×	×	✓
Travel Demand Prediction	×	×	✓	✓	×	✓
Mobility event graph construction	×	×	×	×	×	✓
Indexing Scheme	×	✓	×	×	×	✓
Modelling Travel demand variation (functional region)	×	×	×	✓	×	✓
Mining Spatio-temporal Neighborhood Effects	×	×	×	×	×	✓

3. MARIO framework: workflow

Fig. 2 depicts the workflow of the *MARIO* framework which discovers varied spatio-temporal association rules of taxi-trips. The problem definition of this paper can be summarized as:

- *Given a spatial region r and the taxi-trips (Ty) of r, extract mobility association rules (MARs).* The mobility association rules (MARs) are extracted from the mobility database by deploying proposed mobility-rule miner framework.
- *Given r and Ty of r, generate the mobility dynamics network to represent the overall urban dynamics.* This network captures the overall mobility dynamics of the geographical region by analysing varied sources of movement data.
- *Predict the travel demands in different regions of the city effectively.* Finally, based on the historical movement log and spatio-temporal analysis, the travel demand in different regions of a city is computed.

The framework has three main modules: (i) *trajectory trace preprocessing module:* takes the mobility database consisting taxi-trips of a region (*r*) and road-network of *r* as input and generates the item-set or transactions from the mobility data by following application-specific mobility rule templates, (ii) *Spatio-temporal analysis module:* analyses taxi-trip database to discover correlation among different entities, such as time, place and moving objects (vehicles, people etc.) and (iii) *Mobility association rule mining module:* shows how mobility association rules

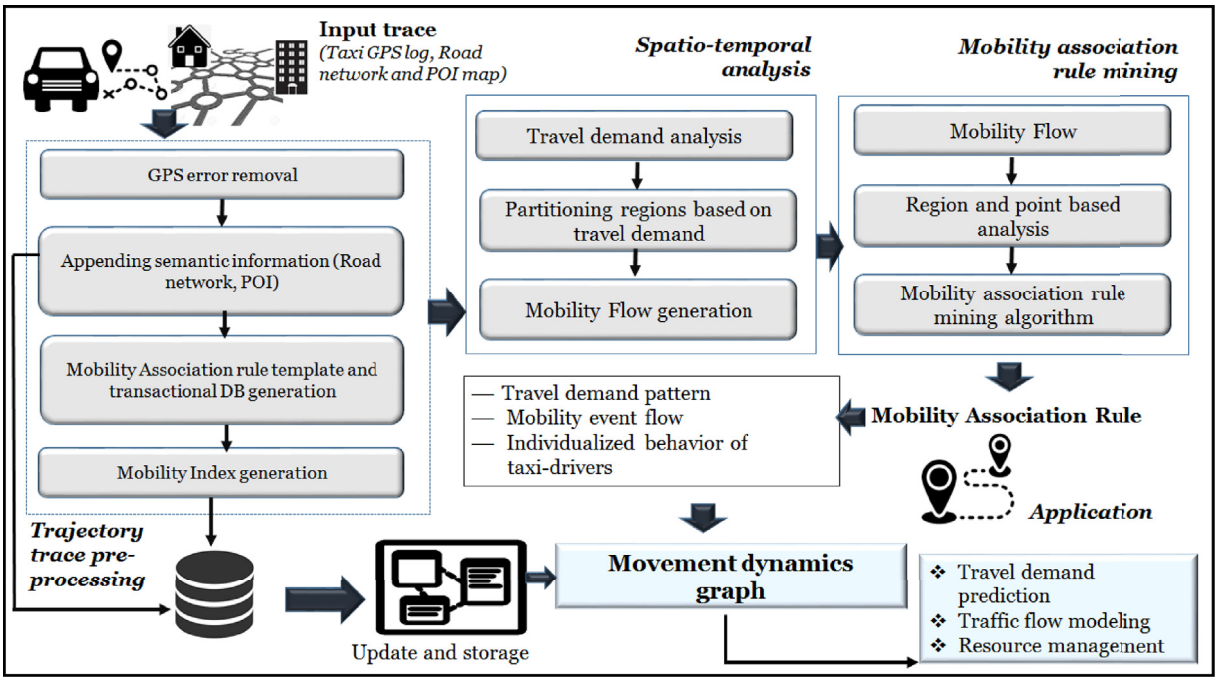

Fig. 2. Workflow of *MARIO* framework.

can be extracted to understand the urban dynamics for better traffic resource management.

We consider a set of moving agents M, say, taxis, set of POIs in a 2D space or region r. Each moving agent $m \in M$ is associated with sequences of location information (x, y). In this work, we study taxi-trajectory dataset to explore varied movement patterns in a city region. As shown in Table 2, the acquired taxi trips cover divergent types of trip-information, such as Type I stores the pick-up and drop-off locations along with passenger count, trip distances etc whereas GPS sequences of complete trip are stored in Type II trace database. These taxi-trip database facilitate different interesting patterns, namely, correlations of pick-up and drop-off locations in different time-instances, frequent-path followed by the trajectories, and even, taxi-specific (or taxi-driver specific) movement behaviours. To start with, few preliminaries are defined as follows.

1. **GPS Log or Trace or Trajectory:** GPS log is a collection of time stamped GPS points $P = \{p_1, p_2, \ldots., p_n\}$. Each GPS point $p_i \in P$ contains latitude (p_i, Lat), longitude $(p_i, Lngt)$ and timestamp (p_i, t_i) of the moving agent (Zheng, 2015). In this paper GPS log, trace or trajectory are used interchangeably.

Geotagged Point: Each GPS point is associated with most appropriate land use information. Here, each GPS point p_i contains $(p_i, place)$ along with latitude, longitude and timestamp information (Ghosh and Ghosh, 2019). For example, p_i is associated with *residential building* or the nearest landmark of p_j is *supermarket*.

2. **Road Network:** Road network of a region is represented by a directed graph $R = (V_R, E_R)$ where V_R represents all intersecting points of the road-segments, i.e., either starting or ending points of each such road-fragment and E_R is the set of edges or roads in the map. It may be noted that in real-life scenario, direction of the roads $E_R = \{v_{R_i} \xrightarrow{t} v_{R_j} | (v_{R_i}, v_{R_j}) \in V_R\}$ may depend on particular time-slot of a day and the variation of direction based on temporal slot has been considered in the road network graph construction.
3. **Trajectory Slider:** It contains the route followed by the taxi within a specific distance (D) and time threshold (T). Trajectory Slider contains a set of GPS points or traces within a specific distance and time-window boundary. $Traj\_Slider = \{(x_1, y_1, t_1), (x_2, y_2, t_2), \ldots, (x_n, y_n, t_n)\}$ iff $max(dist((x, y), (j, k))) <= D \; \forall$ (x,y) and (j,k) pair $\in$ P and $diff(t_n, t_1) <= T$
4. **Taxi-Trip Trajectory:** We define two types of taxi trajectories, $Ty_1 = (< p_{pi}, t_{pi} >, < p_{dr}, t_{dr} >, |C|, D_T, F_T)$ and $Ty_2 = (< p_{pi}, t_{pi} >, < p_i, t_i >, < p_{i+1}, t_{i+1} >, \ldots, < p_{dr}, t_{dr} >)$, where T_1 consists of start and end point of the taxi-trip. T_2 consists of all GPS points of the trajectory segments between the start and end point [Table 2].

Mobility traces offer huge opportunities to discover multi-states information, such as past, present and future states (travel demand, traffic congestion etc.) of a city road-network. For example, knowing the recurrent effect of traffic congestion of neighborhood region allows one to avoid the probable congested regions apriori. To be more specific, it is important to capture the evolution of events in neighboring regions over time and space.

Association rule mining techniques (Agrawal and Srikant, 1994) are utilized to discover the unknown dependencies among the data-items to reflect the correlations among varied item-sets. In this paper, we aim to discover spatio-temporal association rules, or to be more specific, mobility association rules from taxi traces of a geographical region. Typically, it depicts that if a spatio-temporal event takes place, a resultant event is likely to occur within a defined *spatio-temporal neighborhood*. The challenges in finding mobility association rules are following:

- Unlike traditional relational databases, mobility information are not explicitly encoded as *transactions* but are rather embedded within the spatial framework of the geo-referenced data (Shekhar and Chawla, 2003). Therefore, it is required to generate transactions from the spatio-temporal dataset such that association rule mining techniques can be deployed to discover mobility association rules.
- Conventional association rule mining works with categorical data and not with numeric data such as metric distance.
- To extract spatio-temporal association rules, huge amount of data needs to be analysed in varied spatial and temporal scales. Therefore a computationally efficient approach is required to generate the effective association rules in timely manner.

Table 2
Taxi-trip description.

Taxi-trace type	Attributes
Type I:GPS points of pickup and drop-off locations * Pattern: Travel demand	**GPS**: <pickup, drop off location, timestamp>**Others**: Passenger count, Trip distance, Fare amount
Type II:GPS log of the complete trip * Pattern: Frequent path	**GPS**: $< lat_1, lon_1, time_1 > \ \ldots < lat_n, lon_n, time_n >$**Others**: Running status (Vacant, Occupy)

In this direction, we propose a novel framework which is capable to discover the spatio-temporal interrelationships from mobility traces and utilize the association rules to predict travel demand.

3.1. Mobility-flow: spatio-temporal event

Spatio-temporal database captures *events* which represent the chronologically ordered instances of timestamp-geometry pair. Typically, geometries are used to represent space and can be depicted by set of grid cells or spatial region, such as *polygon*. The instances are application-specific and illustrate the type of the spatio-temporal event. For example, "Forest fire always occurs at region R_1 prior to the occurrence of haze in nearby region R_2" (Wang et al., 2004) - where *Forest fire* and *Haze* are two instances of spatio-temporal events occurring at two different spatial locations (R_1 and R_2) and temporal sequence is represented by *prior*. The definitions used in this paper are defined as follows:

1. *Definition 1.* **Spatio-temporal event:** A *spatio-temporal event* or simply *event*, denoted as, $e(r, t, v)$, represents a spatial feature e (namely, rain, haze) in the spatial region r at time t with an intensity value v.
   - r represents co-ordinates (latitude, longitude) or a polygon geometry (denoted by bounding-box) or nodes of a spatial network
   - v represents measurement of e, which may be estimated by specific unit (10 mm *of precipitation*) or linguistic variables (*high, medium*).
   - t represents either a particular time-stamp (1529496160 epoch) or a time-interval pair ($[t_i, t_{i+1}]$)

According to our problem set-up, we consider e as a mobility event which helps to model traffic-flow and travel demand of a region r in a given time-interval t.

2. *Definition 2.* **Mobility event:** A mobility event is a specific type of spatio-temporal event, denoted as, $M(r, t, v)$, illustrates movement dynamics of vehicles in a region r. In this work, we consider three types of mobility events:
   - Pick-up event: The number of pick-ups or starting of trips is v_{pi} from the source region r in the time-interval t
   - Drop-off event: The number of drop-offs or completion of trips is v_{dr} in the destination region r in the time-interval t
   - Moving: The count of trajectory-segments of vehicles passing region r in t time is v_m
3. *Definition 3.* **Spatio-temporal Association Rule (SAR):** Spatio-temporal association rule is defined as $SAR = (r_i, [t_a, t_b]) \Rightarrow (r_j, [t_x, t_y])$, with s support and c confidence, where objects appearing in region r_i during time-interval $[t_a, t_b]$ either
   (i) appear in region r_j for the first time or by (at or before time)
   (ii) be in region r_j during the $[t_x, t_y]$ time interval (Verhein and Chawla, 2006).

Our paper primarily considers the movement behaviours of taxis and aims to analyze the patterns by mining mobility events. We define mobility association rule as follows:

4. *Definition 4.* **Mobility Association Rule (MAR):** Mobility association rule is expressed as: $MAR = M_1(r_i, \Gamma = [t_a, t_b], v_i) \Rightarrow M_2(r_j, \Gamma + \gamma, v_j)$, which depicts that objects (such as taxis) contributing in a mobility event, M_1 also partake another mobility event M_2 after γ timestamp of the former events' time-interval. There are nine possible correlations or direct MARs from three mobility events (pick-up, drop-off and moving). Few such MARs along with their notations are given below.
   - Notation: $M_{pick-up}(r_i, \Gamma, v_i) \Rightarrow M_{drop-off}(r_j, \Gamma + \gamma, v_j)$: Objects participating in a *pick-up mobility event* in a region r_i in the time-interval Γ subsequently completes the trip in region r_j within the time-window $[t_a + \gamma, t_b + \gamma]$. This type of mobility association rules help to correlate two spatial-regions, namely, source (r_i) and destination (r_j) within a specific time-window. It may be noted that γ is essentially different for different trips, however, the goal of our work is to extract such γ value so that the *support* and *confidence* of the rules maximize.
   - Notation: $M_{moving}(r_i, \Gamma, v_i) \Rightarrow M_{moving}(r_j, \Gamma + \gamma, v_j)$: Objects passing through region r_i in the time-interval Γ also pass through region r_j within the time-window $[t_a + \gamma, t_b + \gamma]$. This *MAR* discovers frequent paths in different time-stamp. The entities, namely, r_i, r_j of the rule vary with different time-windows and given a specific time-interval and time-window our approach is capable to extract varied correlated regions having grater support and confidence than the minimum threshold.
   - Notation: $M_{drop-off}(r_i, \Gamma, v_i) \Rightarrow M_{pick-up}(r_j, \Gamma + \gamma, v_j)$: Objects completing trips in region r_i in the time-interval Γ start trips from region r_j within the time-window $[t_a + \gamma, t_b + \gamma]$. This rule assists the drivers to find out next probable pick-up regions and subsequently helps to reduce the waiting time.

The above mentioned *MARs* can be directly extracted by mining taxi-trip database. However, to predict the travel-demands efficiently and understand the overall mobility dynamics of a city-region, it is required to discover derived mobility association rule. The derived mobility association rule's templates will be discussed in the later section of the paper.

5. *Definition 5.* **Travel demand:** Travel demand (d) of a region r_i at time interval Γ is the aggregated count of *pick-up* mobility event in r_i at Γ.

$$f(r_i, \Gamma[t_a, t_b]) = \sum_{t=t_a}^{t_b} v \ where \ M_{pick-up}(r_i, t, v = v_i) \tag{1}$$

Travel demand may also include the number of trip requests from the citizens in a region. Since only taxi-trips are available as experimental purpose, we consider the travel demand of a region as the number of pick-ups in that region at a given temporal scale. However, the proposed approach is scalable enough to incorporate this value as a parameter along with the count of pick-up events in case it is available.

4. MARIO framework: algorithms and implementation

MARIO framework exploits varied type of taxi trip related information and utilizes the attributes to come up with the mobility rule template. It analyzes the spatial neighborhood effects of travel demand variation using Long Short Term Memory (LSTM) architecture where the temporal quantitative relationships of passenger count at different places are utilized as input feed. Moreover, we propose variant of temporal-apriori algorithm to discover the mobility association rules.

Table 3
Predicates of mobility rule template.

Predicate	Description
$timeVal(t)$	Timestamp (time-interval or time-slot of a day)
$POI(p)$	point-of-interest of 'p' type
$locTraversal(dis, S, D)$	Traveled dis distance from S source to D destination
$timeTraversal(t, S, D)$	Time spent t to travel from S source to D destination
$TravelD(c, r, t)$	Travel demand (number of pick-ups) in a region r at time-interval t is c
$footPrint(c, r, t)$	Number of GPS footprints in a region r at time-interval t is c

4.1. Trajectory trace pre-processing

This work uses GPS trajectory data of taxis in cities. Two different types of taxi-trace (ref. Table 2) have been utilized in this purpose. This GPS trace only includes $< latitude, longitude >; < time >$ of pick-up and drop-off locations (trace type I) or $< latitude, longitude >$ after a particular time-interval (say, 30 s) of the complete trip (trace type II). Prior to analyzing the GPS log, appending other semantic information, such as underlying *road-network* structure or nearby *landmark* is important for discovering intrinsic patterns of movement behaviour of a geographical region.

4.1.1. Enriching taxi-trip log with semantic information

Primarily in this step, GPS error is removed followed by appending semantic information, namely, geo-tagged points, road-network structure. The GPS traces of the taxis of the mobility database are segmented where each such fragments are denoted as *trajectory-segment* or *T_Seg*. Initially, the pre-processing guarantees that all GPS points are strictly ordered on timestamps in the T_Seg. For example, if there are more than one GPS point with same timestamp, then only the first occurrence of the GPS point is kept and rest points are removed.

$$p_1(x_1, y_1, t_1) > \ldots > p_n(x_n, y_n, t_n) \text{ and } T\_Seg = \{p_1, \ldots, p_n\} \quad where, t_1 > t_2 > \ldots > t_n, \forall (p_1, \ldots, p_n) \in P \tag{2}$$

Here, GPS log is fragmented such that each *T_Seg* represents a single taxi-trip, i.e., p_1 and p_n are *pick-up* and *drop-off* points of a taxi-trip. Then, GPS errors, which are relatively large, in scale of 100 m (due to GPS capturing device failure or poor satellite signals) are removed deploying *Kalman filtering* (Krakiwsky et al., 1988) technique. To address the issue of different sampling rates of various GPS-trackers of the taxis the analysis is carried out by segmenting the trips into varied *trajectory slider*s which store path followed by the taxis bounded by a specific distance and time thresholds. In the next step, geo-tagged information for each *pick-up* and *drop-off* points are extracted and appended to enrich the semantic information of the raw traces. To augment the nearest landuse information, *iterative reverse geocoding*(IRG) (Ghosh and Ghosh, 2019) is used. Additionally, we maintain a POI-database where $< latitude, longitude, POI >$ are stored for any subsequent computation.

Finally, road network structure of the region is extracted from OSM[1] to semantically enrich the mobility traces. This information contributes in the semantic enrichment process by appending geometric information of the road network (such as length and width of the road-segments), the connectivity and continuity (like, intersection of roads) of the road network. Since we aim to utilize the *MARIO* framework in both urban (dense road-segments) and non-urban regions, an efficient map-matching process with high-accuracy is required. It may be noted that our framework works in two phases:

Phase-I: Firstly, it generates the knowledge base of movement dynamics from historical mobility records by discovering the inherent patterns,

Phase-II: It applies the extracted knowledge from the first phase to predict travel demand in different places. For the first phase, map-matching is a one-time phenomenon and could be performed offline. However extracting the position of the vehicle in the road network in phase two requires online-strategy to reduce pre-processing time. To map the raw taxi-trips onto the digital map we have used *AntMapper* matching (Gong et al., 2017) which utilizes both topological information and global similarity measurement and provides result within a relatively short running time. The map-matching for phase-II of the framework exploits the advantages of *mobility-index* and map-matched trajectory data of phase-I.

To complement the missing values, two-step procedure is followed in the MARIO framework. In the first step, map-matching process is deployed. The map-matching process maps the GPS points into the underlying road-structure. The features of the road-segments: length, width, connectivity and continuity (like, intersection of roads) are extracted in this step. The map-matching algorithm named AntMapper assigns each GPS point to the appropriate road-segment. In the next step, for more refinement of the GPS points, all the GPS points are smoothened using Kalman filtering technique to reduce the error of the measurement/data collection. Using kalman filtering, in case the outlier/missing value is detected, it is replaced by other intermediate points as returned by the filtering technique. This step takes the output of the previous step (map-matching process), and finds out intermediate points to complement the missing values. The road-network and POI information are stored in Cloud Big Query Storage of Google Cloud platform. The map-matching and semantic enrichment algorithms are implemented using the compute engine, which calls Google Place API service and stores the data in a database [*Oracle Spatial and Graph*] with spatial extension.

4.1.2. Mobility association rule (MAR) template and GPS transactional database construction

The definitions of spatio-temporal event, association rule and mobility association rule are discussed earlier in section 3.1. In this section, we explore varied types of *derived* MAR templates and the construction procedure of GPS transactional database. The key-idea of exploring such MAR template is to facilitate a framework capable to effectively model mobility dynamics and discover interesting movement patterns. Given a database of mobility traces, we aim to define a language L conducive to express spatio-temporal properties of the entities in terms of mobility events. The intuitive meaning of any association rule $X \rightarrow Y$ is that transactions containing set of X items tend to contain set Y of items (Agrawal and Srikant, 1994). Similarly finding GPS records containing set of spatio-temporal entities X and Y as well, is the main idea of discovering mobility association rules. The language L is complex and there may be huge number of possible sentences of this language. In this work, we are interested to mine travel-demand specific patterns. Therefore, a set of specific rules (or sentences), which are a sub-set of L are considered.

To represent mobility association rule template, we primarily introduce few predicates [refer Table 3] with parameters which help to construct the basic structure of the rule-templates. It may be noted that all three mobility-events (*pick-up*, *drop-off* and *moving*) are also considered

[1] OpenStreetMap: https://www.openstreetmap.org/.

Table 4
Mobility Rule template (MAR).

Rule Id	Rule template
MAR_1	$M_1(r_1, t_1, v_1) \Rightarrow M_2(r_2, t_2, v_2)$: Mobility event M_1 is followed by mobility event M_2
MAR_2	$timeVal(t) \wedge POI(p) \Rightarrow travelD(c, p, t)$: The travel demand of a region largely depends on the timestamp value and the place type information
MAR_3	$travelD(c_i, r_i, t_i) \Rightarrow footPrint(c_j, r_j, t_j) \wedge travelD(c_k, r_k, t_k)$: Travel demand in a particular region impacts footprint density and generate travel demand in other regions
MAR_4	$locTraversal(dis, S, D) \Rightarrow context \wedge timeVal(t)$: Location specific information of a taxi trip effects the context information, such as fare amount and trip time

as *predicates* which take region, time and count of the event-instances as parameters. Based on the set of predicates (*Pre*) we form derived mobility association rules with spatio-temporal variables (region: r, time: t) and possible annotations. The basic syntax of such rule is:

$$\phi_1, \phi_2 \ldots \phi_n := Pre(r, t) \;\; s.t. \;\; \phi_1 \implies \phi_2 | \phi_1 \wedge \ldots \phi_i \implies \phi_2 \wedge \ldots \phi_j | \phi_1 \vee \ldots \phi_i \implies \phi_2 \vee \ldots \phi_j \tag{3}$$

Given such rules, a selection function finds out how many transactions or records in the mobility database satisfy the *implies* condition. This selection function measures the *support* and *confidence* of the rule and if the count is above the threshold values of support and confidence then that particular rule is included for mapping movement dynamics of the region.

The city dynamics can be modelled by (i) exploring travel demands in different regions and in different time-slots, (ii) how the travel demand of one region impacts the GPS footprints of other regions, and (iii) movement behaviour of taxi-drivers based on region and time. In other words, we aim to reflect peoples' movement dynamics through the above-mentioned factors. Four generic rule-templates have been designed, and we argue that the rule template is complete considering our application to model city-dynamics. Table 4 represents varied types of mobility association rules that help to model the overall mobility dynamics, such as travel demand, traffic flow, individual drivers' movement patterns in a region. MAR_1 is the template of direct *mobility association rule* which can be extracted from three mobility events directly. There are nine possible *MARs* under rule template MAR_1. We have extracted and illustrated such *MARs* with real-life data in the experimental section. Travel demand variation at different POIs can be extracted from MAR_2. It may be noted that rules are fully dependent on space and temporal values, i.e., variation in region and time change the count (or in other word, support and confidence measure) of the records which satisfy that rule. Therefore the proposed framework works in two ways, (i) extracts frequent patterns and (ii) given a rule template, region and timeslot finds out the correlated mobility event, effected region and timestamp value. The impact of the change of travel demand on other regions are represented by MAR_3 and finally, movement behaviour of taxi-drivers are depicted in MAR_4.

Given such rule template and mobility database, the next step is to construct the transactional mobility database such that *MARs* can be extracted. It is obvious that transactional database must be created before mobility association rule mining algorithm can be deployed. From the conventional definitions of *item-set* and *transaction*, we know that, $I = \{I_1, I_2 \ldots I_n\}$ set of items and transactional database contains $T = \{T_1, T_2, \ldots, T_m\}$ set of transactions, where each transaction $T \subseteq I$. In this work, from the raw GPS log, first we construct transactional database where each transaction (T) consists of $< tId, spatialIn, event, temporalSpan, context >$ tuple. It may be noted that we have considered spatial database (Oracle Spatial and Graph) with spatial data-types, such as *point, polygon, polyline* etc. Therefore *spatialIn* field stores the spatial information, namely, *polygon* for a region and *polyline* for movement trace. Table 5 shows an example of such transactional database, where the last column represents additional information, such as $|C| - 2$: count of passengers, $F_t - 50$: fare amount or O: occupy status. All the GPS records are converted to such spatial database transactions and further, mobility index is generated on such records.

Table 5
Transactional Spatial database to store mobility traces.

tId	spatialIn	event	temporalSpan	context
Ty_{1_a}	*point_geom*	*pick-up*	08.00–08.05	$\|C\| - 2$
Ty_{1_b}	*point_geom*	*drop-off*	08.40–08.50	$F_t - 50$
Ty_{2_a}	*polyline_geom*	*moving*	10.00–10.25	O

In this regard, the measurements of such *MARs* need to be quantified. It is obvious that support and *confidence* calculations of traditional association rules do not hold for this case. These measurements need to be extended in spatio-temporal domain. The spatial coverage of a rule defines the sum of the area referenced in the predicates of the rule. Temporal coverage of a rule represents the time window for which rules must be valid. The temporal coverage of different mobility rules are different. We deploy a scaling for both temporal and spatial coverage of the rules which is represented by the ratio of spatial or temporal validity of the rule and spatial and temporal information of the complete study period. Huang et al. (2004) define *participation index* to measure the strength or frequency of a co-location pattern. MARIO utilizes similar measure to define the spatio-temporal support and confidence. Given a mobility association rule $MAR(\phi_i \Rightarrow \phi_j)$ containing spatio-temporal features $f_1, \ldots, f_k$, the following measures are described.

- Spatio-temporal Support ($\phi_i(r_i, t_i) \Rightarrow \phi_j(r_j, t_j)$): It is the scaled spatial coverage and the total length of the time-intervals in the rules.

$$STsupp = \frac{|I(\phi_i, [f_1, \ldots, f_k])|}{|I([f_1, \ldots, f_k])|} \times \sum_{t=1}^{N} \frac{[\phi_i]^t}{N \times t} \tag{4}$$

  where $|I[f_1, \ldots, f_n]|$ represents the number of transactions in the database containing the spatio-temporal features and the next term denotes the temporal scaling parameter where N is the total time-scale of the study period.
- Spatio-temporal Confidence ($\phi_i(r_i, t_i) \Rightarrow \phi_j(r_j, t_j)$): It is measured as the conditional probability of the predicate ϕ_j is true given that ϕ_i is already true.

$$STconf = \frac{STsupp(\phi_i(r_i, t_i) \wedge \phi_j(r_j, t_j))}{STsupp(\phi_i(r_i, t_i))} \tag{5}$$

Our framework attempts to discover mobility rules that have spatio-temporal support above a threshold, *minSTSupp*, and confidence above *minConf*.

4.1.3. Mobility index construction

To achieve the easy and quick access of time-series data, an efficient storage scheme is required for the computation speed up. To store the trajectory data, we propose a hashing based mobility indexing scheme which is beneficial in terms of space allocation and easy accessing

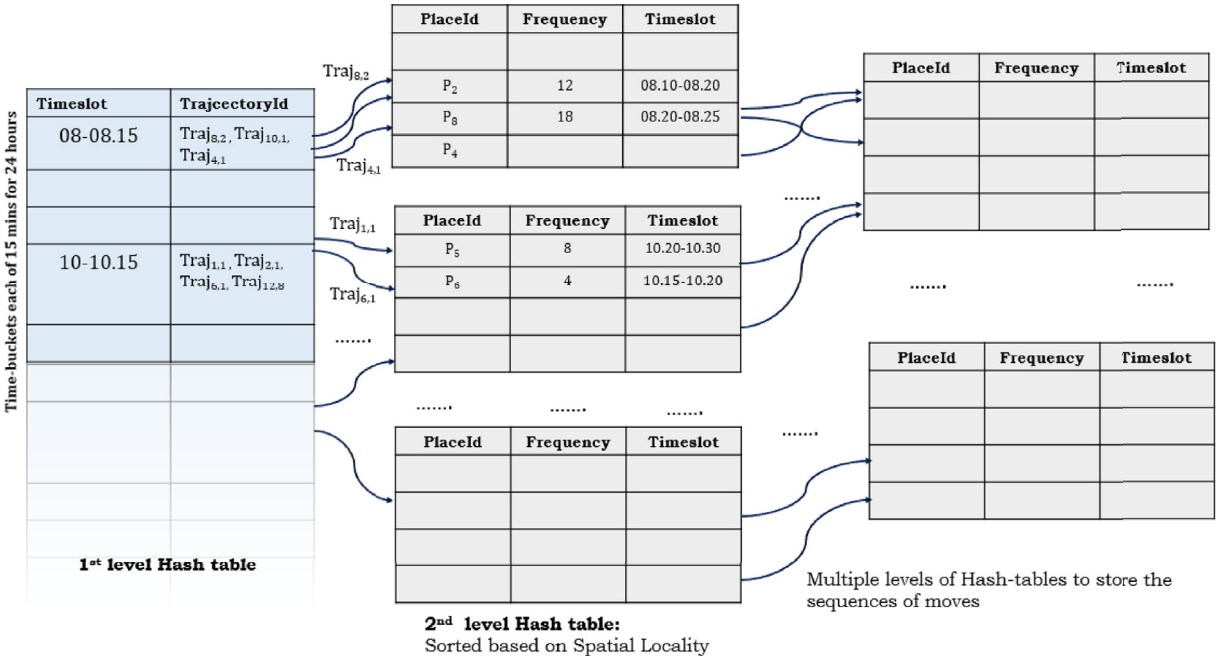

Fig. 3. K-Level temporal hashing for the storage of significant trajectory patterns of a region.

of significant trajectory-patterns in different time-slots. Fig. 3 depicts the structure of the K-level temporal hashing table where k temporal sequences of the trajectory traces are stored in different temporal buckets.

- The 1st level hash table contains trajectory-ids (*Traj*) of different taxi-trips starting at varied time-intervals. The keys of the table are time-slots of 15 min each for a day.

$$H(traj_{id}) = B, where\ [B.t_1 < traj_{id}.t_{pickup} < B.t_2) \tag{6}$$

where $B.t_1$, $B.t_2$ represent the time-intervals of the keys in the hash-table. Trajectory-id (say, $Traj_{a,b}$) maintains the taxi-id (a) and the trip-id (b).

- From the next level, sequences of the trajectory-segments are maintained in different levels along with the place-id (type and location), frequency of visit and time-slots. The hashing technique considers the spatial-locality information, i.e., the nearby places are stored in consecutive buckets. The following pairing function is used as the hash-function:

$$H(p_x, p_y) = (p_x + p_y)(p_x + p_y + 1)/2 + p_y \tag{7}$$

- Frequency plays an important role in the hashing scheme. When a new entry comes in the storage, a search function is deployed and if similar entry is already present, then frequency is incremented and the new entry is inserted in the list. This *frequency* attribute helps to minimize the redundant entries and also can be used as a feature to find out most followed trajectory sequence.

MARIO deploys the *k-level temporal hashing scheme* to store trajectory-sequences of a region into k layers. This hashing scheme is beneficial for extracting movement information efficiently and in timely manner. For example, the most followed route of a region can be discovered by analyzing the k-sequences and *frequency*. On the other side, traffic states of a place can be explored by interpreting the GPS footprints of the taxi-ids in different time-slots. We have implemented this schema using *BigTable* and *BigQuery* of *Google Cloud Platform (GCP)* as storage platform. Cloud DataProc of GCP is used to maintain the index and insertion of records in the storage. Cloud Spanner of GCP is used to store these information which supports horizontal scalability.

4.2. Spatio-temporal analysis

In this section, we discuss about movement dynamics in the study region and how to model the variation of mobility events in varied space and temporal dimension. We have presented how mobility events of one region effect other regions after a particular time-interval. We have introduced *trajectory graph* to model and store varied mobility features in different temporal scale.

4.2.1. Trajectory graph

Trajectory graph is a labeled directed graph where the properties of the graph change as a function of time. The graph evolves by appending or deleting nodes and links over time. This graph helps to characterize information flow in a network, namely, travel demand variation or mobility events on the underlying road-network structure of the region. Formally, trajectory graph is defined as $G = (V, E, \Upsilon, \tau)$ where V represents the set of vertices or nodes and $E \subseteq V \times V$ is the set of edges. The other two parameters Υ and τ depict *node labeling function* and *edge labeling function* over time respectively. The time-evolving nature of the graph is conceptually expressed by a series of directed graphs $G_{t_1}(V_{t_1}, E_{t_1}, \Upsilon, \tau), \ldots, G_T(V_T, E_T, \Upsilon, \tau)$ at different time-instances $t_1, \ldots, T$. Based on the labels assigned by Υ and τ there are three possible types of trajectory-graphs in our analysis.

The *node labeling* function assigns $V \rightarrow \Sigma$ a label to each node in the graph from an alphabet Σ. The labels illustrate the properties of the nodes in the graph. The node properties may represent static information (such as G_1: intersection of roads or G_2: POI-placements do not change with time) or may vary in different temporal scale (such as G_3: mobility-events). The trajectory graph with mobility events as nodes is constructed as follows:

- Sources and destinations of the taxi trips are converted to nodes of the graph and labeled as $< Pick - up, t_i >$ and $< Drop - off, t_j >$ respectively
- If the traces are available from source to destination of the trip, then after ζ time-interval one GPS point (p) is selected from the trace and p is converted to a node of the graph with label $< Moving, (t_j + \zeta) >$

Table 6
Trajectory graph information.

ID	Node(V)	Edge (E)	Υ	τ
G_1	Intersection of roads	Road segments	Location information (*latitude*, *longitude*)	Direction
G_2	POI	Road segments	POI-type information	Direction
G_3	< Mobility event, location, timeInterval >	Sequence (After/Before relation)	Types of mobility events	Time taken from one event to another

- The edges of the graph are generated from taxi trip's nodes and ordered on increasing timestamp. The label of the edges are the time-differences between two nodes $< t_{i+1} - t_i >$.

The detailed procedures to construct trajectory graph, namely, *mobility event graph* are noted in algorithm 1. The tabular representation of trajectory graph is shown in Table 6. For *road graph* construction, we deploy a nested adjacency list structure to store the direction of the roads in different time-intervals. It may be noted that two different road graph structures have been maintained for weekdays and weekends. It is observed that in most cities traffic directions vary typically in weekdays and weekends. The algorithm 1 constructs the mobility-event graph from the historical movement traces of a region. Let us assume, the number of taxis in the dataset is M. The average number of trips by each taxi is n. Therefore, it takes $O(M \times n)$ to find out the pick-up and drop-off points in the complete dataset. The time taken to construct the vertices of $G3$ is $O(c_1 \times M \times n)$, where c_1 is a constant [line no. 3–10]. Next, the algorithm finds out the moving objects (GPS points between pick-up and drop-off points) of the complete path. Let us assume, the average length of the taxi-trips is: L. The average time-duration of the trip: dur. The time-offset (or sampling rate) parameter is: ζ. Therefore, the number of intermediate (moving) points = (dur/ζ). The time complexity to generate the moving edges and nodes (line no 11–26) is $= O(n * L * dur/\zeta) = O(n \times L \times c_2)$. Therefore, the overall time-complexity of the algorithm 1 is $= O\sum_{k=0}^{M} n \times L \times c_2 + n \times c_1$, which is a polynomial time algorithm.

Algorithm 1 Trajectory graph (Mobility event graph) construction.

Input: Taxi-trip trajectory (Ty_1, Ty_2) log
Output: Mobility event graph $G_3(V, E, \Upsilon, \tau)$
1: **function** GENERATE$G_3(Ty)$ ▷ Where Ty_1 - taxi trajectory log with pick-up and drop-off points, Ty_2 - taxi trajectory log with all intermediate points
2: $G_3(V[], E[]) \leftarrow NULL$ ▷ 2-D matrix where each row, column is of < id, location, label > data-type
3: **for** $i = 1$ to $length(T)$ **do**
4: $Create\_node(G_3.V_i \leftarrow (id, location, label))$
5: $\Upsilon : G_3.V_i[id] \leftarrow p$ ▷ p - pick-up
6: $G_3.V_i[p][label] \leftarrow T[i].t_{pick-up}$
7: $Create\_node(G_3.V_i \leftarrow (id, label))$
8: $\Upsilon : G_3.V_i[id] \leftarrow d$ ▷ d - drop-off
9: $G_3.V_i[d][label] \leftarrow T[i].t_{drop-off}$
10: **end for**
11: **for** $j = 1$ to i **do**
12: $t \leftarrow V_j.label$
13: **for** $k = 1$ to $length(T[j])$ **do**
14: **if** $length(T[k]) > 2$ **then** ▷ Ty_2 trajectory trace type
15: $m \leftarrow 1; l \leftarrow length(T[k]);$

(continued on next page)

Algorithm 1 *(continued)*

16: **while** $m! = l$ **do**
17: $p \leftarrow$ *Select GPS point after*$(t + \zeta)time - interval\ from(T[k])$
18: $Create\_node(G_3.V_j \leftarrow (id, label))$
19: $\Upsilon : G_3.V_j[id] \leftarrow m$ ▷ m - moving
20: $G_3.V_j[p][label] \leftarrow p.timestamp$
21: $t \leftarrow p.timestamp;$
22: $\tau : Create\_edge(e_m \leftarrow (V_k[p], V_k[m]))$
23: **end while**
24: **end if**
25: **end for**
26: **end for**
27: **end function**

4.2.2. Modelling travel demand variation

Discovering the *effects* of a mobility phenomenon, such as traffic blockage and road closure, on a different region after specific time-interval is the most challenging task in travel demand analysis. For example, "If there is a road closure in *region 1*, then *region 2* experiences higher travel demand in a particular time-interval". Here, *region 1* is effecting the mobility patterns of *region 2* and prior prediction of such *effects* may help in effective resource management. Therefore, the main objective is to find out such relations among spatial regions from the historical movement log. Next, the framework aims to find out the temporal relations among the events' effect, i.e., "traffic congestion of *region 1* is propagated to *region 2* after Δ time period" - where the range of Δ value needs to be discovered. In this work, the regions which are effected by a mobility event at other regions are termed as *neighbor*.

Typically, spatial neighbors are formed with points of a grid or nodes of a graph those are *close* to one another. The *neighborhood* definition largely depends on the denotation of *close*, where close generally represent adjacent or within some threshold range of distance. However, in our analysis, distance measure is not well-suited for analysing the underlying correlations among varied regions. Here, we define *neighborhood function* and few related terms.

Region: A region $R = (V', E')$ is a sub-graph of the trajectory graph (G), where $V' \in V$ represents the nodes (intersection point or POI) and $E' \in E$ depicts the roads between two such nodes. For the ease of visualization any region is denoted by a rectangular bounding box.[2]

Source: A region is denoted *Source* (S) if the count of moving objects entering (n_e) in the region is much lesser than the objects leaving (n_l) the region in a particular time-slot.

$$|n_l - n_e| \geq min_{thresh} \tag{8}$$

Sink: A region is denoted *Sink* (S') if the count of moving objects entering (n_e) in the region is greater than the objects leaving (n_l) the region in a particular time-slot.

$$|n_l - n_e| \leq min_{thresh} \tag{9}$$

The pick-up and drop-off events are considered while counting n_e and n_l.

Flow distance: Flow distance (F) between two regions (r and r') is

[2] To avoid the ambiguity of the definition of region, the paper uses R as the underlying network and r as the bounding box (polygon geometry) of any geographical space.

defined as a time-series arraylist of (n_l, n_e) pairs:

$$F(r, r')[t_i, t_j] = \{|n_e - n_l|_{t_i,\ldots,t_j}, |n'_e - n'_l|_{t_i,\ldots,t_j}\} \tag{10}$$

In the next step, MARIO explores the travel demand variation of different regions based on functional area and time-slots. In section 4.1.1, we describe the method of augmenting geo-tagged information with the GPS log. The functional area or geo-tagged information is crucial to model the travel demand, since citizens mostly follow similar temporal rules for their social activities. The complete study region is divided into 12 social functional areas, *residential*, *commercial*, *government organization*, *industrial region*, *school or university*, *entertainment*, *area of historical interests*, *religious place*, *tourist spots*, *railway*, *airport region* and *highway region*. Each day is divided into 15 min slot where slot 0 starts from 0800 in the morning. The study-area is divided into $n = 12$ functional regions where each region is considered as the bounding box of a subgraph of the trajectory graph. The number of *pick − up*, *drop − off* and *moving* mobility events in region R_a during different time-slots $(t_0, \ldots, t_m)$ of a day (say, L_1) are represented as:

$$\begin{aligned} P^{R_a}_{L_1} &= (P^{1,a}_{t_0}, P^{1,a}_{t_1}, \ldots, P^{1,a}_{t_m}) \\ D^{R_a}_{L_1} &= (D^{1,a}_{t_0}, D^{1,a}_{t_1}, \ldots, D^{1,a}_{t_m}) \\ M^{R_a}_{L_1} &= (M^{1,a}_{t_0}, M^{1,a}_{t_1}, \ldots, M^{1,a}_{t_m}) \end{aligned} \tag{11}$$

Next, for each region 3 matrices are formed for all GPS logs $(L_1, \ldots, L_d)$ of d days.

$$M = (M_a, M_b, \ldots, M_n)^t = \begin{pmatrix} M^{R_a}_{L_1} & M^{R_b}_{L_1} & \ldots & M^{R_n}_{L_1} \\ M^{R_a}_{L_2} & M^{R_b}_{L_2} & \ldots & M^{R_n}_{L_2} \\ \ldots & \ldots & \ldots & \ldots \\ M^{R_a}_{L_d} & M^{R_b}_{L_n} & \ldots & M^{R_n}_{L_d} \end{pmatrix}^t \tag{12}$$

where M shows the distributions of moving taxis in n regions for a particular time-slot (t) of d days. Similarly, other two matrices namely pick-up distribution (P) and drop-off distribution (D) are formed. Next, we deploy *autoregressive integrated moving average* (Lee and Tong, 2011) to find out the values of pick-up, drop-off and moving events in different time-slots from the historical movement traces. Although the method can effectively model non-linear time-series, however, the *impact* or *effect* of other regions are not considered. Therefore, in the next step, MARIO aims to find out *effect* of one region to other from spatio-temporal context.

Two regions (r and r') are neighbors of each other in a time-slot $T = [t_i, t_{i+1}]$ if the flow distance between two regions are within a threshold and the regions are edge-reachable in T. Edge-reachable defines that no other neighboring region in the same time-slot T is present in the connected path of r and r'.

$$f_{neighbor}(r, r', t_i, t_j) = \begin{cases} 1 & if\ (t_j - t_i) < t_{th}\ \wedge\ edge - reachable(r, r') = 1 \\ & \wedge\ F[t_i, t_j](r, r') < d_{th} \\ 0 & otherwise \end{cases} \tag{13}$$

$f_{neighbor}$ returns whether two regions are neighbor of each other in a given time-slot. Typically, it checks whether the flow-distance and time-differences are within the threshold range.

4.2.3. Modelling the impacts of mobility events using LSTM

The mobility events occurring in one region impact mobility phenomenon of other regions. These patterns do exhibit a high degree of spatio-temporal correlation. It happens due to the road-structure, social functional areas (POI) and people's movement regularity. To this end, MARIO deploys *Recurrent Neural Network* to capture the effects of one regions' mobility variation to other regions. To be more specific, *LSTM (Long Short Term Memory)* (Sak et al., 2014) - a type of RNN is suitable to learn the long term dependency of the time-series data and determine the correlated regions.

Problem Formulation: Let us assume, the set of GPS trajectories of all vehicles ($v \in V$) is Tr, and the set of all locations in the study area is Reg, where $Tr_v \in \mathbb{R}^d$ and $Reg_l \in \mathbb{R}^d$ represent the latent vectors of trajectory-segments of vehicle v and each location l respectively. Here, each $l \in Reg$ refers to a particular grid, and represented by a grid-id[3] $(gr[i \in I, j \in J])$.[4] The grids are associated with GPS locations. The trajectory-history of the vehicles are presented as $Reg^v = \{l^v_{t_1}, l^v_{t_2}, \ldots\}$, when the v vehicle is present at $l^v_{t_i}$ at time t_i. The trajectory-segments of all vehicles can be aggregated as $Reg^V = \{Reg^{v_1}, Reg^{v_2}, \ldots\}$. In this work, the travel demand (integer value) is represented by the number of pick-up (n_{pickup}) and number of moving vehicles (n_{move}) in different parts of the city. From the trajectory segments of all vehicles a different locations, we can find out the n_{pickup} and n_{move} of all locations at different time-stamp. Therefore, a matrix $TD_{I\times J}$ can be formed where two such values (n_{pickup}, n_{moving}) are stored for each location over time. $TD \in \mathbb{R}^{2\times I\times J}$ represent the travel demand at any time. Given the historical observations $\{TD_t | t = 0, 1, \ldots, n-1\}$, the objective of this model is to predict TD_n. Learning Objective: In the training phase, MARIO learns how to estimate the travel demand of the locations based on the historical spatio-temporal patterns, and other mobility events (sudden traffic blockage, accident etc.). The framework typically works in two phases. Given the real-time mobility event information (location and timestamp of accident/blockage), it finds out the *neighbor*[5] locations of the event. Next, it predicts the travel demand of the location incorporating any effects due to the mobility events.

Typically, LSTM overcomes the vanishing gradient and exploding gradient problems of conventional RNN and conducive to learn the mobility events' patterns with long time spans and automatically predict the mobility events of other regions as an effect of the present. Fig. 4 shows the snapshot of LSTM network, which maps an input sequence to an output sequence by computing the network activations in different time-instances. The basic equations of a typical LSTM architecture are as follows:

$$\begin{aligned} i_t = \sigma(w_i[h_{t-1}, x_t] + b_i);\ \ f_t = \sigma(w_f[h_{t-1}, x_t] + b_f); \\ o_t = \sigma(w_o[h_{t-1}, x_t] + b_o) \\ c_t = f_t \cdot c_{t-1} + i_t \cdot \tanh(w_c[h_{t-1}, x_t] + b_c); \\ m_t = o_t \cdot h(c_t) \end{aligned} \tag{14}$$

where input gate, output gate and forget gates are represented by i, f and o. The scalar product between two vectors are represented by $\cdot$ whereas $\sigma(\cdot)$ denotes the logistic sigmoid function. The activation vectors for each cell is c and m are memory blocks. w_x and b_x are the weight matrices for neurons and bias vectors for respective blocks. MARIO uses tanh for cell input, output activation functions and *softmax* as network activation function. The two-layer deep LSTM network has a linear recurrent projection layer in each LSTM layer stack. Based on the features such as, velocity, change of position, timestamp etc., the moving behaviour sequences are generated. Next, this sequences are fed into the LSTM blocks, which converts the inputs into fixed-length representations. The mobility event information (type of mobility event, location, timestamp) is fed as another input into the LSTM stack. There are 6 types of mobility event: Accident, Congestion, DisabledVehicle, PlannedEvent, RoadHazard, Construction with 4 severity level (Low-Impact, Minor, Moderate and Serious). Since these are categorical values, we have used an embedding method (Gal and Ghahramani, 2016)

[3] The process of partitioning of the whole network into grids is presented in section 4.2.4.

[4] The set of grids are represented by a $I \times J$ matrix.

[5] see equation (13) for the definition of neighbor.

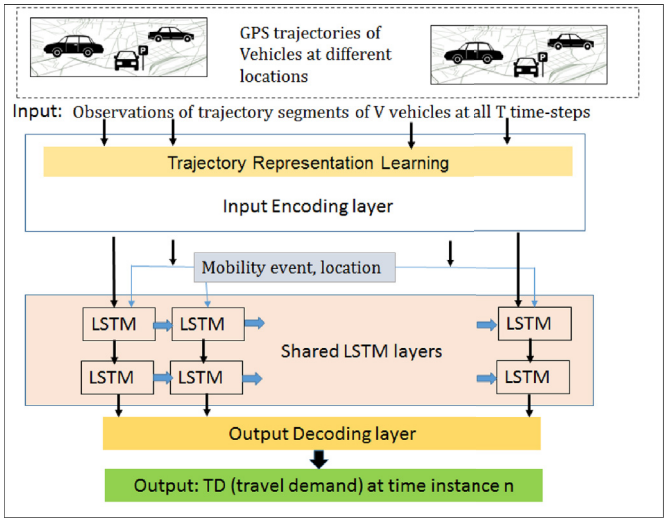

Fig. 4. MARIO: Deep LSTM architecture.

to transform these values to a low-dimensional vector. The input of the LSTM architecture is region along with mobility states and road-network structure. MARIO aims to model the mobility states jointly with road network structure and POI-placements since the correlations with other regions inherently depend on these aspects. Sak et al. (2014) demonstrates that two-layer deep LSTM architecture outperforms other existing methods where each layer has a linear recurrent projection layer. The model is trained using 100 epochs. In the input observation, each location (grid) has an average of 10 road-segments. In the output, the predicted travel demand is the number of pick-up and moving events at a particular region. The number of regions depend on the scale of the grid and area of the road-network. In the experiment, we have tested up to 10,000 regions (or grids of size 10). Firstly, it learns the effect of road-network connectivity among different regions and then extracts the regions showing variation of mobility events as a consequence of mobility event in the input region. The deep learning architecture of MARIO consists of an *encoding* and *decoding* layer and two hidden layers. The loss function is the mean squared error between predicted travel demand and actual travel demand.

The road-network is extracted from OpenStreetMap where incomplete information is present making the graph sparse in nature. To tackle this issue, we have used network embedding method to extract the structural information of the underlying road network. The embeddings of the road network has been learned from the traffic interaction of the vehicles. The process requires a set of vehicle trajectories which cover the road segments in the network to learn the missing values of the sparse network. The deep learning architecture is implemented using Google Tensorflow.[6] The implementation is done on the top of the Google App Engine, including DataStore and Task Queues. Two Cloud SQL instances are created, where one is executed from Google App engine. The other instance has the database access permission. To add storage capacity, automatic storage increase is enabled.

4.2.4. Movement dynamics network

Movement dynamics network is used to illustrate the interrelationship among spatio-temporal features of mobility data. Intuitively, movement dynamics of a region is correlated to city-infrastructure (road-network and POI placements) and the movement behaviour of mobile objects, such as people and vehicles. The modelling and analysis of movement behaviour is quite challenging since it varies dynamically with temporal value.

The movement dynamics network is constructed by hierarchically assembling trajectory-graphs of a region. Fig. 5 shows a snapshot of movement dynamics network in three levels. Firstly, the grids are generated on the road-graph of the region such that the sub-graph within each grid has nearly similar cardinality (number of edges). The grid information help to correlate the location and other contextual information, such as POI placement, mobility events etc. Next, road graph is placed on the created grids. In level 2, mobility event graph is created based on the spatial transactional database containing taxi-trips. Finally, level 3 consists of the unusual mobility phenomenon, such as *traffic congestion* (when velocity of the vehicles are less than a threshold value), traffic blockage (velocity is nearly zero) or road-closure and travel-demand changes. This level specifically helps to understand the real-time movement dynamics of a region while level 2 provides frequent movement patterns and travel demands in various time-slot. The major steps are as follows:

- Grid construction: The study area is divided into n grids such that differences of $|E|^* \sum_{e \in |E|} length(e)$ in grids are within a threshold value. We start the partitioning task in a top-down fashion, where the complete road graph is divided into n connected sub-graphs.
- Aggregated mobility event graph construction: This process takes the mobility event graph as input and deploying a label propagation clustering generate region-clusters having same mobility events.
- Mobility phenomenon: We have considered mobility phenomenons: *traffic congestion, traffic blockage* and *travel demand variation.* First, from the level 2 the matrix is created with the features, GPS footprints in each edge of the grids, travel-demand in the regions and average velocity of the vehicles in the road-segments.

Algorithm 2 provides the procedures of grid construction by graph partitioning. Firstly it computes the *geo-hash codes* of the study-region and divides the complete region into several uniform grids. Then it analyses the cardinality of road-network graph within each such grids and aims to minimize the differences between cardinalities of the sub-graphs. Finally, the algorithm outputs the geo-hash codes of generated grids having more or less similar cardinality of graph information. Let us assume, the number of edges and nodes of the road-graph (R) is $|E|$ and $|V|$ respectively. Time taken to partition the region into n uniform grid: $O(n^*E)$. Time to compute the geo-hash code of the region: $O(c_1 \times n)$, where c_1 is a constant. Time to compute the cardinality (size of edge-set) of all such grid: $O(c_2 \times (|V| + |E|))$, where c_2 is a constant (since the number of edges of all n grids will be equal to the number of edges of the road-graph). Let us assume the optimal number of partitioned graph is: a and $a <= n$. The number of edge set in each such partitioned graph is: b and $a \times b <= |E|$ [line no 5–13]. Therefore, the overall time-complexity of the algorithm 2 is $= O(n \times E + c_1 \times n + c_2 \times (|V| + |E|) + a \times b) <= O(n \times E + c_1 \times n + c_2 \times (|V| + |E|) + |E|) = O(c_3 \times n \times E + c_4 \times (V + 2E))$, since, $a <= n$ and $a \times b <= |E|$. Again, $n << V$, the worst-case time complexity is $O(c_3 \times V \times E + c_4 \times (V + 2E)) <= O(V \times E)$.

Algorithm 2 Grid construction by graph partitioning.

Input: Road graph ($R(V,E)$), min, max
Output: List of Grids ($Grid[geo - hash]$)
1: **function** GRIDCONSTRUCT($R(V,E), min, max$) ▷ Where R is the road-graph of the study region,

(continued on next page)

[6] Tensorflow: https://www.tensorflow.org/.

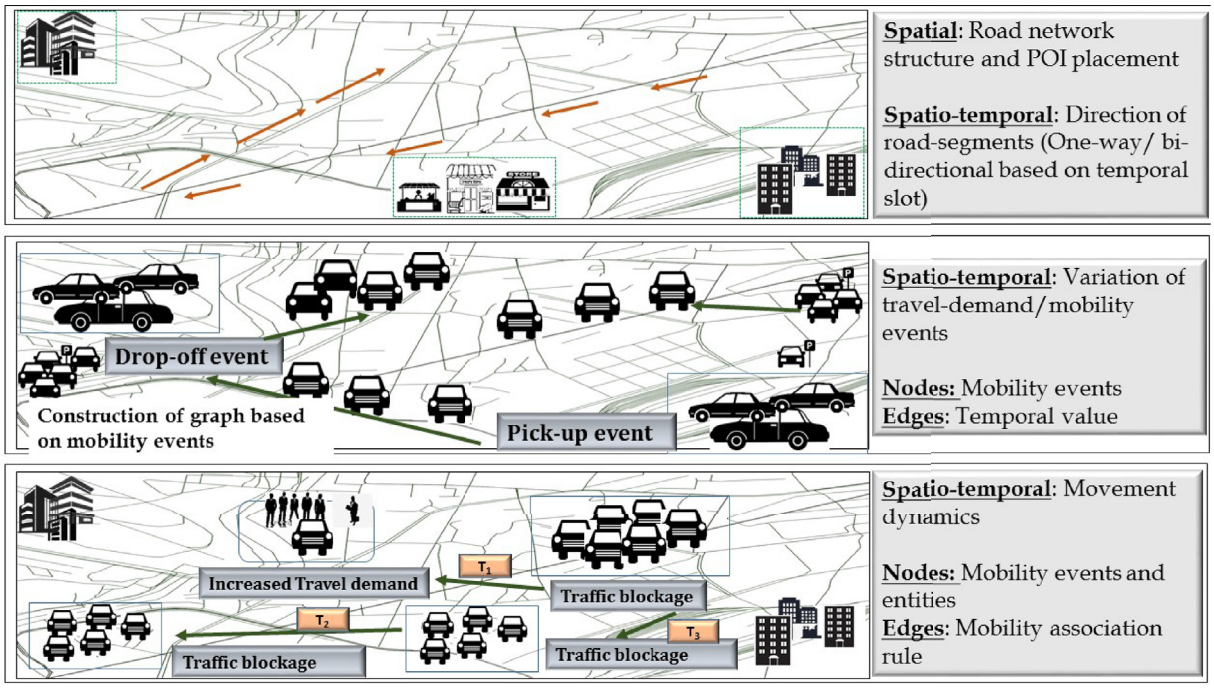

Fig. 5. Snapshot of *Movement dynamics network*.

Algorithm 2 *(continued)*

```
min and max represent the minimum and
 maximum number of grids
geo − hash_R ← generate geo − hash code(R)
subG[] ← Uniform − grid(geo − hash_R, min); n ← min
diff[ ] ← ∑ connectedComp(subG[]) × ∑_{subG_{i,j}} ||E|_{subG_i} − |E|_{subG_j}|
while n <= max do
  m ← computemindiff_n        ▷ Solution using simplex method
  for i = 1to m do
    G_1 ← mergeSubgraphs(subG, i)
    diffL[ ] ← ∑ connectedComp(G_1[]) × ∑_{G_{i,j}} ||E|_{G_i} − |E|_{subG_j}|
      diff_n ← diffL; subG ← G_1
  end for
end while
Grid[geo − hash] ← compute geo − hash(subG)
end function
```

4.3. Mobility rule mining

In this section, we demonstrate the process of extracting mobility association rules from the taxi-trips by deploying a variant of *temporal apriori* (Chen and Wu, 2006). Furthermore, we aim to predict travel demand in different places based on the extracted mobility rules. The mobility miner method consists of two major steps, namely, capturing the prevalent candidate items of the rules in different spatio-temporal resolution and extracting the rules having support and confidence greater than the threshold values. A *M-flow* or mobility-flow represent any sub-graph of level 3 of mobility dynamics network, i.e., the effects of mobility phenomenon to other neighboring region. With the help of mobility rules, we aim to analyze the spatio-temporal neighborhood effects and predict the mobility characteristics of the regions along with predicted travel demand. Here *frequency tree* is utilized and the GPS trace of different time-intervals have been used as the input feed.

Spatio-temporal property 1: *M-flow follows Apriori Property: Any sub-set of infrequent spatio-temporal event-set (sequential pattern) is also infrequent*

To this end, we can formulate the mobility-rule extraction problem as: Given a database (Ty) of spatio-temporal events, a specific location and time window, a mobility rule template *MAR*, and a minimum support threshold, the problem of extraction of mobility rules is equivalent to discovering set of frequent item-sets among a set of items in a transactional database. Extracting spatio-temporal mobility rules is not a trivial process because space and time can not be analysed in same dimension, time is an increasing sequence of small temporal-quantam while such sequence is not present in 2D space.

We consider another situation: **Spatio-temporal property 2:** *An infrequent item-set of M-flow for a time-range T and spatial region R, may be frequent item-set for a smaller region R −Δ and a time-interval T − δ*. It depicts that numerical attributes (trip-length, trip-duration etc.) depend on the spatial and temporal coverage of the rules.

Formally, *mobility association rule* is defined as $e_1 \wedge l_1 \wedge c_1 \rightarrow d_p$, where a conjunction of literals e_1, l_1, c_1, i.e., edge information, location information and other contextual information are associated with a specific travel demand. A tuple t satisfy a rule if and only if it satisfies every possible literals (space, time) in the rule. If t satisfies the rule, travel demand of the region at the particular time can be represented by the value of d_p. This leads to the former research question *"Using taxi-trajectories, how mobility dynamics of a region can be captured and represented as mobility-association rules and help to predict the travel demand?"* The extracted mobility association rules of a region will help to understand the travel demands accurately and predict the traffic flow. The next process involves extracting *mobility flow* of a region by analysing taxi-trips. The first step of the proposed method is to discover all frequent spatio-temporal event-sets followed by finding co-related events analysing spatio-temporal neighborhood of the events.

The key observation is any temporal relationship between the antecedent and consequent of the rules must follow higher order sequences. Hence, any infrequent item-set within the same temporal resolution but in a reverse order is discarded apriori. An *Iterative Deepening* based enumeration method has been proposed where different item-sets are inserted based on the temporal occurrences. In the next step, a *frequency tree* has been generated, where,

- Each node at height k, stores sequence of event-sets, and possible travel-demand at different time-resolution
- The travel demand is initialized as 1 in the beginning phase
- After each iteration, the frequency of the events are modified and different granularity of spatio-temporal information are maintained [to satisfy spatio-temporal property 2]

The rule mining algorithm is a variant of T-Apriori method which consists of a number of passes depending on the time-slots. The algorithm finds out i-itemsets (i.e., itemsets with i items having at least the minimum support) at pass *i*. It generates the set of candidate of i-itemsets and computes the count by scanning the database. Finally it finds out the MARs by inspecting the spatio-temporal support of all the candidate itemsets. The algorithm is terminated when no large itemset is discovered after a pass. Furthermore, the MAR template is used as an input feed. The algorithm only finds out the item-sets which are present in the rule-template. It significantly reduces the search time. The algorithm works in a bottom-up fashion where small temporal scale is used in the first phase followed by grouping larger time-slots given that spatio-temporal support and confidence are larger than the minimum threshold. Given the four rule-templates, MARIO extracts MAR_1 type rules by analyzing the spatio-temporal neighborhood method proposed in section 4.2.3 and deploying temporal apriori method. MAR_2 and MAR_3 type rules are extracted by computing M, P and D matrices [see section 4.2.2] followed by executing temporal apriori algorithm. Finally, the individualized movement patterns MAR_4 can be extracted by grouping unique taxi-ids and then performing rule mining process.

Our proposed rule mining algorithm differs from *T-Apriori* (Chen and Wu, 2006) which is used to discover temporal pattern for interval-based events. However, the impacts of one regions' mobility events can not be incorporated using only *T-Apriori*. MARIO exploits the extracted spatial-neighborhood effects of mobility events while extracting the mobility rules. The patterns of *effecting* other regions' mobility states are extracted by deep LSTM architecture and associated with the rule mining algorithm.

5. Performance evaluation

All experimental evaluations are carried out on VM instance of *Google Cloud Platform*[7] having 4 vCPUs, 15 GB memory and Ubuntu 16.04, Linux as the OS. The algorithms are implemented in Python, R and all the experiments are performed on three real datasets of taxi trajectories.

5.1. Dataset

The dataset[8] (Type I) is collected by the NYC Taxi and Limousine Commission (TLC). It contains trip records from all trips completed in Yellow and Green taxis in NYC from 2009 to present, and all trips in for-hire vehicles (FHV) from 2015 to present. It contains over 750 millions taxi-trips and the storage of over 190 GB. We have also evaluated MARIO with Roma taxi traces[9] and San Francisco (SF) taxi data[10] (Type II). The Roma dataset contains mobility traces of taxi cabs in Rome, Italy consisting GPS coordinates of approximately 320 taxis collected over 30 days. The SF dataset contains mobility traces of taxi cabs in San Francisco, USA consisting GPS coordinates of approximately 500 taxis collected over 30 days in the San Francisco Bay Area. Both the Rome and SF dataset contain GPS points of the taxi-trips in a particular time-interval (10–15 s). As depicted earlier, two types of taxi-trip data have been analysed (see Table 2), where type I dataset has only pick-up and drop-off points, and GPS traces of complete trip have been logged in the type II dataset. The missing values of logging occur in type II dataset. It is observed that among 320 taxis, there are 6 taxis with less than 5 days trip-history. These 6 taxis are eliminated from the data-set. Amongst other 314 taxis, the locations are logged in 12 s offset. Total number of trajectory segments are 44,286. The number of trajectory segments consisting missing values is 2795, which is around 6% of the complete trajectory dataset of Rome. On the other-side, the total number of trajectory-segments in SF dataset is 56,920 and the missing values are found in 5467 segments, which is around 10% of the SF dataset. These missing values are replaced by intermediate points by map-matching and Kalman filtering techniques.

To depict the effectiveness of the proposed framework, we have carried out the experimental evaluations in two aspects: evaluating the system performance of MARIO framework and extracting varied mobility rules, predicting travel demand in different part of the city network.

5.2. Performance evaluation of MARIO: indexing scheme

One of the major contributions of MARIO is facilitating an end-to-end framework which is conducive to resolve user travel related queries. We propose *k-level temporal hash based schema* to store the huge amount of data and reduce the information extracting time. To depict the efficacy of the storage along with indexing scheme of MARIO, we compare it with five baseline methods namely *R-tree* (Guttman, 1984), extended *historical R-tree* (HR + tree) (Deng et al., 2011), *Scalable and Efficient Trajectory Index (SETI)* (Chakka et al., 2003), *TrajStore* (Cudre–Mauroux et al., 2010) and *R2-D2* (Zhou et al., 2013). A multiversion structure is proposed in *HR + tree* where entries of different timestamps are placed in the same node leading to reduction of space. All of these methods are implemented and tested with *NYC* taxi-trip dataset. Chakka et al. (2003) propose a two-level index structure to decouple the index of spatial and temporal information. Trajstore (Cudre-Mauroux et al., 2010) maintains an optimal index and dynamically co-locates and compresses spatially and temporally adjacent segments on disk. A grid based index is proposed in Zhou et al. (2013) where the area of interest is divided into a set of rectangular cells with fixed size and trajectories are indexed in the cells they pass.

The comparisons have been carried out for index-size, and query-time. We have considered *R-query* and *T-query* (Deng et al., 2011) and report the average query-response time for all methods. Figs. 6 and 7 demonstrate the evaluation results of MARIO with other baseline methods. It has been shown that MARIO has less query execution time compared to other methods. Here, the proposed k-level temporal hashing scheme has outperformed other baselines in a huge margin (almost 50% less execution time in average). The key reason is that the trip-sequences of a region are stored into k-temporal levels and in consecutive buckets following the hash-function based on latitude and longitude information. It helps to extract the range and T-query in an efficient manner compared to other methods. Furthermore, with the increasing data size, MARIO maintains a reasonable rate of index size, since it avoids to maintain any duplicate entry using the frequency attribute of the hashing-scheme. In summary, although there are several indexing methods for spatio-temporal data-set, however they fall short in maintaining large mobility database and providing timely access of trajectory information.

5.3. Extracted mobility rules, measurements

Table 7 shows a subset of extracted mobility rules along with average scaled spatio-temporal support and confidence. The minimum threshold of spatio-temporal support and confidence are set to 0.10 and 0.75 respectively from [0,1] range. The number of extracted mobility rules in this range are 120, 356 and 288 for NYC, ROME and SF respectively. The average spatio-temporal confidence of these rules lie in the

[7] https://cloud.google.com/.
[8] NYC Taxi Trace: http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml.
[9] ROME Taxi Trace: https://crawdad.org/roma/taxi/20140717/.
[10] SF Taxi Trace: https://crawdad.org/epfl/mobility/20090224/.

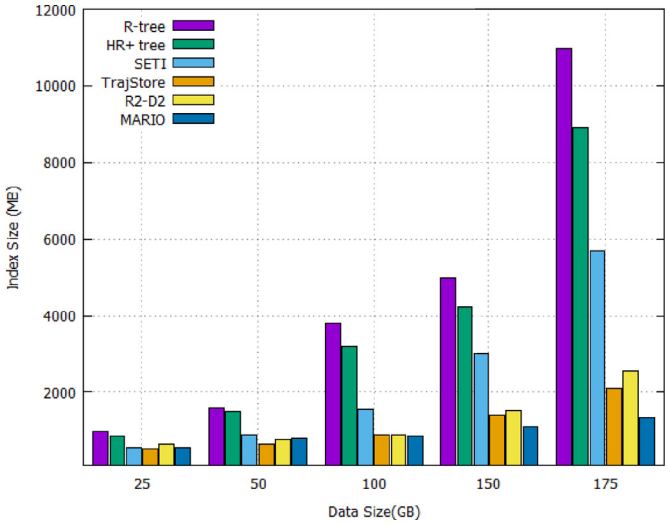

Fig. 6. Performance evaluation of MARIO: Index Size.

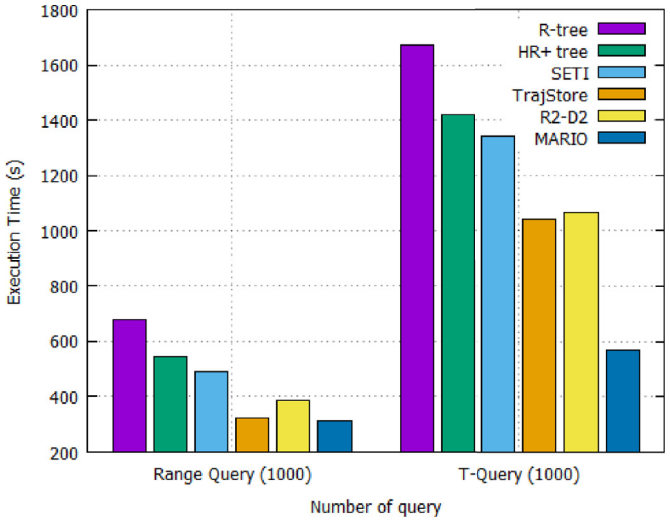

Fig. 7. Query execution time.

range of (0.78–0.84), (0.82–0.88) and (0.85–0.89) respectively for the three study-areas.

It may be observed two types of mobility rules have been extracted. The former rules ($MR_1 - MR_5$) are based on travel demand in different POIs at varied timestamps while other rules are generated depending on mobility behaviours of taxi-drivers. For example, it has been noted that few taxi-drivers start their trips at early morning and typically travel in the residential and commercial area, while another set of taxi-trips are mainly from airport region to residential region and of large time-duration. Variation of trip-distance, taxi-fare along with varied timestamps have been reflected by the extracted mobility rules. These rules then are utilized to predict travel demand in different regions. Apart from the rules shown in Table 7 the other types of rules considered in MARIO (rule template MAR_1) are neighborhood effects. These type of rules are the highest numbers of rules that have been extracted. Although this can not be shown in the tabular format however these rules significantly help to predict travel demand and GPS footprints in different regions of the city.

Based on the trip's pattern, taxi-drivers may be grouped into three different categories, (i) G_1: Starts trip early in the morning slot and typically moves around residential area, academic area and railway or bus stops. The number of trips are fairly large in a day. (ii) G_2: Trips are initiated in late afternoon from commercial and entertainment places. There are also fair amount of trips from areas of historic interests or tourist places. (iii) G_3: These taxi-trips mainly cover movements from airport area and railway stations and bus-stop. The trip-duration is large and subsequently less number of rides are covered by the taxi-drivers.

5.4. Prediction accuracy: travel demand

Based on the MARs, we compare the performance of MARIO with nine baseline methods to predict travel demand in different regions of a city. The baseline methods are historical average (HA), ARIMA, SVM, Linear regression, GBDT, VAR, ST-ANN, DeepST (Zhang et al., 2016b) and ST-ResNet (Zhang et al., 2017). The HA model predicts the demand using average values of historical events whereas ARIMA combines moving average and autoregressive components for modelling time series. We have compared MARIO with ordinary least squares regression. The kernel function is used in SVM model for prediction and a gradient boosting method is used in the GBDT model to improve the prediction accuracy. These all are broad range of classical machine learning algorithm. We have also compared it with deep-learning models, VAR, ST-ANN, DeepST and ST-ResNet, to illustrate the efficacy of the proposed deep LSTM model of MARIO. VAR or Vector Auto-Regressive explores the pair-wise relations among all flows, however, the model has high computational cost. ST-ANN model finds out the spatial and temporal features of nearby 8 regions, and fed into an ANN network. Here, we have considered nearby 8 regions' (nearest 8 grids) data and 8 previous time-intervals. We have used the DeepST-CPT variant of DeepST model where periodic, temporal and seasonal sequences are considered for crowd-flow forecasting.

The evaluation is carried out by using two metrics: *Mean Average Percentage Error (MAPE)* and *Rooted Mean Square Error (RMSE)*. We have carried out the comparisons for number of *pick-up events* and *moving events* and report the average *RMSE* and *MAPE* values. Table 8 and Table 9 demonstrate the RMSE values for travel demand and moving mobility events for MARIO and five baselines. Fig. 8 and Fig. 9 show the MAPE values for *moving* and *pick-up* events. It has been observed that MARIO achieves a lowest 7.016 and 0.1251 RMSE and MAPE respectively. Amongst all the baselines based on deep-learning, DeepST and ST-ResNet show better performances. Next, we manually select 10 days from all three datasets when specific events (accident, road-blockage, crowd due to social event) occurred and evaluate the travel demand in different places of the cities. The results of the deep learning models are shown in Fig. 10 and Fig. 11. It is observed that MARIO performs significantly better than other deep learning models in these 10 days when any event occurs. The key reason of this performance is that MARIO not only models the spatial and temporal travel demand patterns, it is also capable to model the variation of travel demand occurring due to some events.

MARIO provides significant reduction in RMSE and MAPE over the baseline methods. It may be noted that the existing methods fall short especially in predicting *moving* events, however MARIO provides significantly better results both in pick-up and mobility event predictions.

5.5. Discussions

1. This work aims to extract mobility association rules from taxi-trips of a city. These rules consist of underlying mobility patterns of a city road network, namely, travel demand of a region (say, residential or commercial) in different time-slots of a day, taxi-drivers' movement behaviours and finally how the variation of travel demand effects other regions. These rules are effective for early prediction of travel demand spikes such that several ridesharing companies, individual taxi-drivers can schedule the trips and as well as to provide a better transportation to citizens.

Table 7
Mobility Rules and Evaluation Metrics. S:Spatio-temporal Support, C: Spatio-temporal Confidence, T_1: 0800–1000, T_2: 1000–1600, T_3: 1600–2100, T_4: 2100 - 0800, R_1: Residential area, R_2: Commercial and entertainment region, R_3: Academic area, R_4: Areas of historic interest, R_5: Railway station and Bus-stops, R_6: Airport region.

M-Rule	S	C
$\boldsymbol{MR_1}$: TimeStamp$(T_1 \wedge T_3) \wedge Weekday \Rightarrow travelDemand(High, R_1)$	0.28	0.867
$\boldsymbol{MR_2}$: TimeStamp$(T_1 \wedge T_2) \wedge Weekday \Rightarrow travelDemand(High, R_3)$	0.21	0.785
$\boldsymbol{MR_3}$: TimeStamp$(T_3 \wedge T_4) \wedge Weekday \Rightarrow travelDemand(High, R_2)$	0.182	0.843
$\boldsymbol{MR_4}$: TimeStamp$(T_1 \wedge T_2 \wedge T_3) \wedge Weekday \Rightarrow travelDemand(High, R_5)$	0.145	0.874
$\boldsymbol{MR_5}$: TimeStamp$(T_1 \wedge T_2 \wedge T_3) \wedge Weekend \Rightarrow travelDemand(High, R_4)$	0.128	0.902
$\boldsymbol{MR_6}$: $Trip - duration(small) \wedge TimeStamp(T_1 to T_3) \Rightarrow noOfTrips(high) \wedge Region(R_1, R_3, R_5)$	0.23	0.821
$\boldsymbol{MR_7}$: $Trip - duration(small) \wedge TimeStamp(T_4 \wedge T_3) \Rightarrow noOfTrips(high) \wedge Region(R_2)$	0.24	0.876
$\boldsymbol{MR_8}$: $Trip - duration(large) \wedge TimeStamp(T_1 \wedge T_4) \Rightarrow noOfTrips(small) \wedge Region(R_5, R_6)$	0.206	0.743
$\boldsymbol{MR_9}$: $Trip - duration(large) \wedge TimeStamp(T_1 \wedge T_3) \Rightarrow noOfTrips(small) \wedge Amount(High) \wedge Region(R_6 \wedge R_4)$	0.32	0.870
$\boldsymbol{MR_{10}}$: $Trip - duration(small) \wedge TimeStamp(T_1 to T_3) \Rightarrow noOfTrips(large) \wedge Amount(Medium) \wedge Region(R_1 \wedge R_2 \wedge R_3)$	0.217	0.817

Table 8
RMSE Comparison (travel demand) with baselines.

Dataset	HA	GBDT	SVM	Linear Regression	ARIMA	VAR	ST-ANN	DeepST	ST-ResNet	MARIO
NYC dataset	14.182	10.091	13.96	11.12	12.89	12.52	11.86	11.02	9.58	**8.02**
Rome dataset	13.215	10.03	11.910	10.08	11.561	11.03	10.42	9.93	8.07	**7.408**
SF dataset	12.803	9.608	12.07	10.012	11.867	11.65	10.26	9.42	7.96	**7.016**

Table 9
RMSE Comparison of *moving* mobility events with baseline methods.

Dataset	HA	GBDT	SVM	Linear Regression	ARIMA	VAR	ST-ANN	DeepST	ST-ResNet	MARIO
Rome dataset	18.801	14.706	17.913	15.702	16.810	16.09	15.21	14.64	13.05	**11.862**
SF dataset	17.103	13.02	15.526	13.87	14.531	14.07	13.68	12.04	10.85	**9.098**

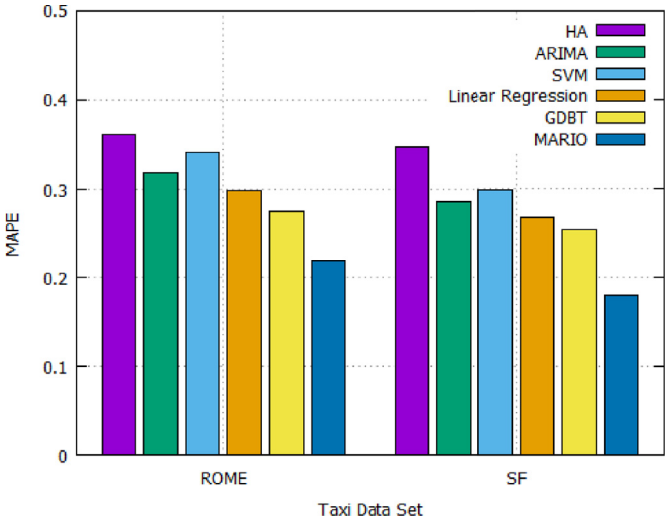

Fig. 8. Average MAPE value of moving mobility events in ROME and SF cities.

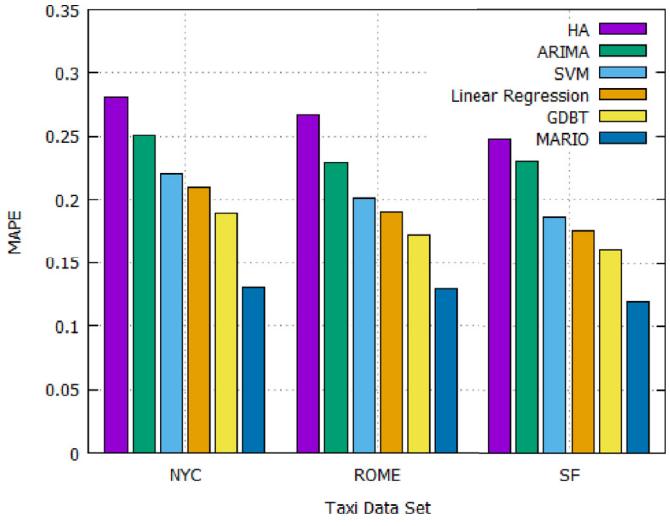

Fig. 9. Average MAPE value of travel demand in 3 cities.

2. The key challenges to extract mobility association rules are two-folds, (i) Firstly the statistical analysis to predict travel demand in near future is time and computation extensive, since it requires a huge amount of taxi trips to be analysed. (ii) Further, this analysis may fall short to extract underlying dynamics of the mobility features. Mobility association rules are extracted from historical log and utilized for demand prediction without the need to analyze enormous amount of taxi-trips. Moreover, the rules are capable to reflect mobility features of a typical city network. Further, the indexing scheme of MARIO has outperformed other baseline methods in terms of quick access of time-series data [refer Figs. 6 and 7].
3. Categorization of taxi-drivers has been done based on their mobility behaviours. For example, few taxi drivers cover only small (trip-length) trips, while others take long trips from airport to residential area and vice versa. These varied mobility behaviours also effect the total amount of payment in a day. These analysis may help to show which types of trips in a given region and timestamps are useful to reduce vacant time (no passenger) of a taxi.
4. The spatio-temporal neighboring effect is crucial to predict any variation of travel demand and re-route the trips. Our proposed method finds out the interrelationships of neighboring areas using a deep LSTM architecture [refer Fig. 4] and thus outperforms other existing

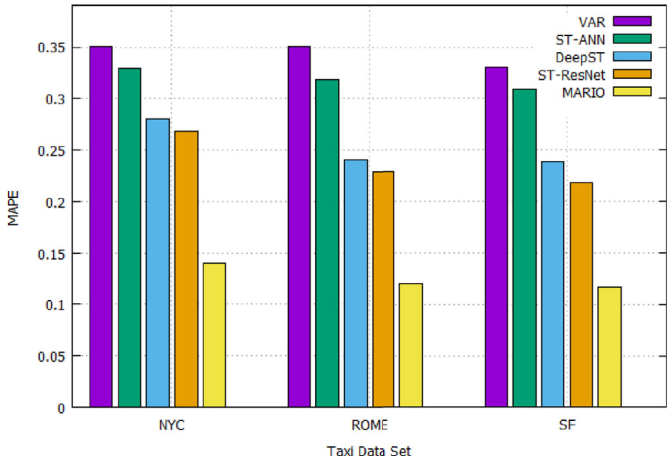

Fig. 10. Average MAPE value (10 days) of travel demand in 3 cities.

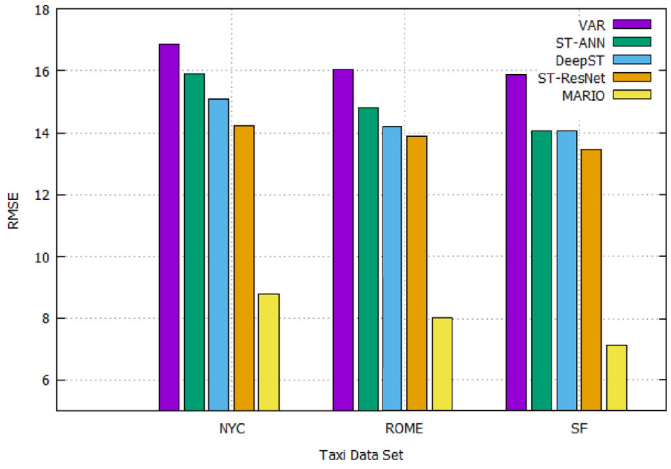

Fig. 11. Average RMSE value (10 days) of travel demand in 3 cities.

methods to predict travel demand and variation of travel demands. The proposed framework has achieved better RMSE value, MAPE values [refer Figs. 8 and 9] for predicting total passenger count, number of originating trips in a region and how travel demand variation of one region impacts other neighboring regions.

6. Conclusions and future work

Exploring city dynamics from mobility traces facilitates several location based services, such as traffic resource management and improved route planning. The mobility association rules play an important role to summarize such travel patterns, variation of travel demands and finally modelling the overall movement dynamics. However, extracting mobility association rules are challenging for time and extensive computation complexity. In this work, we present a mobility-rule miner framework named MARIO which is capable to extract spatio-temporal mobility association rules from taxi-trip dataset. Furthermore, with the help of the proposed methods, we discover mobility patterns and finally predict the spatiotemporal distribution of travel-demands for different functional regions of a city. The mobility association rules predict the travel demand of different regions of a city which may improve taxi drivers' profits and passengers' travel experience. MARIO is not only limited to find interesting patterns from trajectory database, but incorporates how the effect of any mobility event evolve over longer periods and on different spatial scale. The experimental evaluations on three real-life taxi traces demonstrate the efficacy of MARIO. We strongly believe that our framework can be utilized as an end-to-end system to model mobility dynamics of a city and subsequently predicting travel demands. MARIO has outperformed other existing methods for predicting travel demands in terms of *RMSE* and *MAPE* measures significantly.

In future, we will extend MARIO by incorporating other contextual information such as weather information to model the effects of these features on overall mobility dynamics. Furthermore, we will aim to develop an intelligent trip planner using the MARIO framework for resolving users' queries. It has been shown that the proposed approach may facilitate the development of a context-aware trip recommendation system.

CRediT authorship contribution statement

Shreya Ghosh: Conceptualization, Methodology, Software, Investigation, Formal analysis, Writing - original draft, Visualization, Validation. **Soumya K. Ghosh:** Formal analysis, Investigation, Resources, Data curation, Project administration, Writing - review & editing, Supervision, Project administration. **Rajkumar Buyya:** Conceptualization, Formal analysis, Investigation, Writing - review & editing, Supervision, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The work is partially supported by TCS PhD research fellowship.

References

Agrawal, R., Srikant, R., 1994. Fast algorithms for mining association rules. In: Proceedings of the 20th International Conference Very Large Databases, VLDB, vol. 1215, pp. 487–499.
Akbari, M., Samadzadegan, F., Weibel, R., 2015. A generic regional spatio-temporal co-occurrence pattern mining model: a case study for air pollution. J. Geogr. Syst. 17 (3), 249–274.
Appice, A., Ceci, M., Lanza, A., Lisi, F.A., Malerba, D., 2003. Discovery of spatial association rules in geo-referenced census data: a relational mining approach. Intell. Data Anal. 7 (6), 541–566.
Barua, S., Sander, J., 2013. Mining statistically significant co-location and segregation patterns. IEEE Trans. Knowl. Data Eng. 26 (5), 1185–1199.
Cao, J., Xu, S., Zhu, X., Lv, R., Liu, B., 2018. Effective fine-grained location prediction based on user check-in pattern in lbsns. J. Netw. Comput. Appl. 108, 64–75.
Chakka, V.P., Everspaugh, A., Patel, J.M., 2003. Indexing large trajectory data sets with seti. In: Proceedings of the CIDR, vol. 75. Citeseer, p. 76.
Chen, Y.-L., Wu, S.-Y., 2006. Mining temporal patterns from sequence database of interval-based events. In: Proceedings of the International Conference on Fuzzy Systems and Knowledge Discovery. Springer, pp. 586–595.
Cudre-Mauroux, P., Wu, E., Madden, S., 2010. Trajstore: an adaptive storage system for very large trajectory data sets. In: Proceedings of the 26th International Conference on Data Engineering (ICDE 2010). IEEE, pp. 109–120.
Dao, T.H.D., Thill, J.-C., 2016. The spatialarmed framework: handling complex spatial components in spatial association rule mining. Geogr. Anal. 48 (3), 248–274.
Deng, K., Xie, K., Zheng, K., Zhou, X., 2011. Trajectory indexing and retrieval. In: Computing with Spatial Trajectories. Springer, pp. 35–60.
Gal, Y., Ghahramani, Z., 2016. A theoretically grounded application of dropout in recurrent neural networks. In: Advances in Neural Information Processing Systems, pp. 1019–1027.
Ghosh, S., Ghosh, S.K., 2016. Thump: semantic analysis on trajectory traces to explore human movement pattern. In: Proceedings of the 25th International Conference Companion on World Wide Web. ACM, pp. 35–36.
Ghosh, S., Ghosh, S.K., 2017. Exploring human movement behaviour based on mobility association rule mining of trajectory traces. In: Proceedings of the 17th International Conference on Intelligent Systems Design and Applications. Springer, pp. 451–463.
Ghosh, S., Ghosh, S.K., 2019. Traj-cloud: a trajectory cloud for enabling efficient mobility services. In: Proceedings of the 11th International Conference on Communication Systems and Networks (COMSNETS). IEEE, pp. 765–770.
Gong, L., Liu, X., Wu, L., Liu, Y., 2016. Inferring trip purposes and uncovering travel patterns from taxi trajectory data. Cartogr. Geogr. Inf. Sci. 43 (2), 103–114.
Gong, Y.-J., Chen, E., Zhang, X., Ni, L.M., Zhang, J., 2017. Antmapper: an ant colony-based map matching approach for trajectory-based applications. IEEE Trans. Intell. Transport. Syst. 19 (2), 390–401.

Guttman, A., 1984. R-Trees: A Dynamic Index Structure for Spatial Searching, vol. 14. ACM.
Huang, Y., Shekhar, S., Xiong, H., 2004. Discovering colocation patterns from spatial data sets: a general approach. IEEE Trans. Knowl. Data Eng. 12, 1472–1485.
Ivanov, R., 2012. Real-time gps track simplification algorithm for outdoor navigation of visually impaired. J. Netw. Comput. Appl. 35 (5), 1559–1567.
Kong, X., Xia, F., Wang, J., Rahim, A., Das, S.K., 2017. Time-location-relationship combined service recommendation based on taxi trajectory data. IEEE Trans. Ind. Inf. 13 (3), 1202–1212.
Koperski, K., Han, J., 1995. Discovery of spatial association rules in geographic information databases. In: International Symposium on Spatial Databases. Springer, pp. 47–66.
Krakiwsky, E.J., Harris, C.B., Wong, R.V., 1988. A kalman filter for integrating dead reckoning, map matching and gps positioning. In: IEEE PLANS'88., Position Location and Navigation Symposium, Record. Navigation into the 21st Century. IEEE, pp. 39–46.
Lee, I., 2004. Mining multivariate associations within gis environments. In: Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Springer, pp. 1062–1071.
Lee, Y.-S., Tong, L.-I., 2011. Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming. Knowl. Base Syst. 24 (1), 66–72.
Liu, W., Li, X., Rahn, D.A., 2016. Storm event representation and analysis based on a directed spatiotemporal graph model. Int. J. Geogr. Inf. Sci. 30 (5), 948–969.
Mohan, P., Shekhar, S., Shine, J.A., Rogers, J.P., 2011. Cascading spatio-temporal pattern discovery. IEEE Trans. Knowl. Data Eng. 24 (11), 1977–1992.
Pirozmand, P., Wu, G., Jedari, B., Xia, F., 2014. Human mobility in opportunistic networks: characteristics, models and prediction methods. J. Netw. Comput. Appl. 42, 45–58.
Sak, H., Senior, A., Beaufays, F., 2014. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In: Proceedings of the 15th Annual Conference of the International Speech Communication Association.
Shekhar, S., Chawla, S., 2003. Spatial Databases: a Tour. Pearson.
Shen, H., Bai, G., Yang, M., Wang, Z., 2017. Protecting trajectory privacy: a user-centric analysis. J. Netw. Comput. Appl. 82, 128–139.
Verhein, F., Chawla, S., 2006. Mining spatio-temporal association rules, sources, sinks, stationary regions and thoroughfares in object mobility databases. In: Proceedings of the International Conference on Database Systems for Advanced Applications. Springer, pp. 187–201.
Wang, J., Hsu, W., Lee, M.-L., Flowminer, 2004. Finding flow patterns in spatio-temporal databases. In: Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence. IEEE, pp. 14–21.
Yao, H., Wu, F., Ke, J., Tang, X., Jia, Y., Lu, S., Gong, P., Ye, J., Li, Z., 2018. Deep multi-view spatial-temporal network for taxi demand prediction. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence.
Ye, Y., Zheng, Y., Chen, Y., Feng, J., Xie, X., 2009. Mining individual life pattern based on location history. In: Proceedings of the 10th International Conference on Mobile Data Management: Systems, Services and Middleware. IEEE, pp. 1–10.
Zhang, C., Zheng, Y., Ma, X., Han, J., Assembler, 2015. Efficient discovery of spatial co-evolving patterns in massive geo-sensory data. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 1415–1424.
Zhang, K., Feng, Z., Chen, S., Huang, K., Wang, G., 2016a. A framework for passengers demand prediction and recommendation. In: Proceedings of the IEEE International Conference on Services Computing (SCC). IEEE, pp. 340–347.
Zhang, J., Zheng, Y., Qi, D., Li, R., Yi, X., 2016b. Dnn-based prediction model for spatio-temporal data. In: Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 1–4.
Zhang, J., Zheng, Y., Qi, D., 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In: Thirty-First AAAI Conference on Artificial Intelligence.
Zhang, J., Zheng, Y., Sun, J., Qi, D., 2019. Flow prediction in spatio-temporal networks based on multitask deep learning. IEEE Trans. Knowl. Data Eng.
Zheng, Y., 2015. Trajectory data mining: an overview. ACM Trans. Intell. Syst. Technol. (TIST) 6 (3), 1–41.
Zheng, Y., Li, Q., Chen, Y., Xie, X., Ma, W.-Y., 2008. Understanding mobility based on gps data. In: Proceedings of the 10th International Conference on Ubiquitous Computing. ACM, pp. 312–321.
Zhou, J., Tung, A.K., Wu, W., Ng, W.S., 2013. R2-d2: a system to support probabilistic path prediction in dynamic environments via semi-lazy learning. Proc. VLDB Endow. 6 (12), 1366–1369.

Shreya Ghosh received the B.Tech. degree from the Department of Computer Science and Technology at Indian Institute of Engineering Science And Technology, Shibpur (IIEST Shibpur), India, in 2015. She is currently a Research Scholar with the Department of Computer Science and Engineering, IIT Kharagpur, India working towards her PhD. Her current research interests include spatial informatics, trajectory data mining and cloud computing. Shreya is the recipient of the prestigious TCS fellowship.

Soumya K Ghosh received the M.Tech. and Ph.D. degrees in computer science and engineering from the Indian Institute of Technology (IIT) Kharagpur, India. He is currently a Professor with the Department of Computer Science and Engineering, IIT Kharagpur. He was with the Indian Space Research Organization, Bengaluru, India. He has authored or coauthored more than 300 research papers in reputed journals and conference proceedings. His current research interests include spatial data science, spatial web services, and cloud computing. He is the recipient of National Geospatial Chair Professor award from Department of Science and Technology, Govt. of India.

Rajkumar Buyya is a Redmond Barry Distinguished Professor and Director of the Cloud Computing and Distributed Systems (CLOUDS) Laboratory at the University of Melbourne, Australia. He is also serving as the founding CEO of Manjrasoft, a spin-off company of the University, commercializing its innovations in Cloud Computing. He served as a Future Fellow of the Australian Research Council during 2012–2016. He has authored over 625 publications and seven text books including "Mastering Cloud Computing" published by McGraw Hill, China Machine Press, and Morgan Kaufmann for Indian, Chinese and international markets respectively. He also edited several books including "Cloud Computing: Principles and Paradigms" (Wiley Press, USA, Feb 2011). He is one of the highly cited authors in computer science and software engineering worldwide (h-index = 135, 96,000+ citations). Microsoft Academic Search Index ranked Dr. Buyya as #1 author in the world (2005–2016) for both field rating and citations evaluations in the area of Distributed and Parallel Computing. Recently, Dr. Buyya is recognized as a "Web of Science Highly Cited Researcher" in 2016, 2017, and 2018 by Thomson Reuters, a Fellow of IEEE, and Scopus Researcher of the Year 2017 with Excellence in Innovative Research Award by Elsevier for his outstanding contributions to Cloud computing. For further information on Dr.Buyya, please visit his cyberhome: www.buyya.com.