Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 5943567, 15 pages
https://doi.org/10.1155/2021/5943567

Research Article

A Data-Driven Scalable Method for Profiling and Dynamic Analysis of Shared Mobility Solutions

Bogdan Toader,[1] **Assaad Moawad**,[2] **Thomas Hartmann**,[2] **and Francesco Viti**[1]

[1]*Mobilab Transport Research Group, University of Luxembourg, L-4364, Esch-sur-Alzette, Luxembourg*
[2]*DataThings S.A.R.L., L-1811 Luxembourg, Luxembourg*

Correspondence should be addressed to Francesco Viti; francesco.viti@uni.lu

Received 17 January 2020; Revised 26 August 2020; Accepted 31 December 2020; Published 19 January 2021

Academic Editor: Zhuping Zhou

The advent of Internet of Things will revolutionise the sharing mobility by enabling high connectivity between passengers and means of transport. This generates enormous quantity of data which can reveal valuable knowledge and help understand complex travel behaviour. At the same time, it challenges analytics platforms to discover knowledge from data in motion (i.e., the analytics occur in real time as the event happens), extract travel habits, and provide reliable and faster sharing mobility services in dynamic contexts. In this paper, a scalable method for dynamic profiling is introduced, which allows the extraction of users' travel behaviour and valuable knowledge about visited locations, using only geolocation data collected from mobile devices. The methodology makes use of a compact representation of time-evolving graphs that can be used to analyse complex data in motion. In particular, we demonstrate that using a combination of state-of-the-art technologies from data science domain coupled with methodologies from the transportation domain, it is possible to implement, with the minimum of resources, the next generation of autonomous sharing mobility services (i.e., long-term and on-demand parking sharing and combinations of car sharing and ride sharing) and extract from raw data, without any user input and in near real time, valuable knowledge (i.e., location labelling and activity classification).

1. Introduction

The transportation industry is on the edge of unprecedented change. A survey shows that 81% of respondents believe Internet of Things (IoT) will revolutionise the transport sector [1] and the transportation industry spending is expected to surpass the $1 trillion mark in 2022 [2]. The advent of IoT in the last decade enabled the connectivity of people, goods, means of transportation, and the entire transportation infrastructure. The result is an unparalleled amount of data, delivered at revolutionary speed and in continuous expansion [3]. At the same time, the transportation industry remains the sector with the fastest-growing concerns in terms of emissions and one solution is provided by the shared mobility services [4].

The large-scale adoption of IoT caused similar issues in different industries and domains, e.g., electrical smart grid domain [5, 6], where the objective is to analyse data collected in a cyber-physical system in near real-time and to ultimately support decision-making processes based on the results of this analysis [7]. Similarly, ITS drives the implementation of data science techniques for real-time analytics of data in the transportation domain. This means that new methodologies must be able to handle not only data at rest applications (i.e., data that have been collected and are then analysed after the event occurs) but also data in motion (i.e., the analytics occur in real time as the event happens). Data in motion gathered from advanced sensing (such as built-in sensors from mobile devices) and other types of traffic information (such as traffic metering) can be combined to better analyse in near real time users' travel behaviour and derive their mobility needs.

The literature shows the necessity of shifting from classical ITS to smart social mobility services (SSMS) [8], which consists of an integrated and cooperative approach to sense users' individual needs and interactions and offers user-centred mobility services. Following the recommendations from [4], additional research must be done as the ITS

must be prepared for analysing data in motion in real time, learning users' behaviour and performing fast searches in large datasets, which could instead contribute to a more integrated, fast, and flexible method for implementing collaborative mobility services at different levels and for different needs.

In this study, we propose a method for dynamic and near real-time profiling of travel behaviour in time and space, using data in motion. In our case, profiling means the extraction of user habits for visiting a specific location. The real-time aspect is a mandatory requirement for shared collaborative systems (e.g., car sharing and parking sharing) where a large number of people and goods are moving at high speeds and solutions to combine them in an efficient way must be provided in real time (e.g., peer-to-peer ride sharing). In this sense, the proposed methodology makes use of multidimensional profiling techniques described in Section 3 to automatically build the profile as soon as the data become available and proposes efficient techniques to store the results in a temporal index for fast access.

The remainder of this paper is organised as follows. First, Section 2 presents a background of this study making the link with previous work and the proposed methodology in Section 3. The evaluation of the proposed method is presented in Section 4, along with practical usage examples in Section 5 and future work in Section 6. We conclude the paper with a discussion of future directions in Section 7.

2. Literature and Background

Travel behaviour is an interdisciplinary problem, which combines (*a*) specific methodologies for travel behaviour analysis and user's habits profiling and (*b*) data science methods for efficient computation. In the following sections, we present the general background and literature from each domain and the links with the previous work. This will help to define the base terminology, notions, and scientific background necessary for understanding the contribution of this study.

2.1. Travel Behaviour Analytics. In general, data-driven travel behavioural profiling refers to the process of constructing and applying various learning techniques, using the mobility data generated by users and other entities (e.g., sensors from different means of transportation and traffic counters). In the transportation domain, profiling is a method used with different objectives. Driver behaviour has been profiled using the advanced motion sensors from cars and smartphones to detect driving events and to classify drivers in specific categories. Profiling methods are used in fleet management, insurance policies, fuel consumption optimisation, or gas emission reduction [9–12], as well as in route choice in multimodal networks in order to consider the individual preferences in route recommendation systems [13] and in Internet-oriented user centric intelligent transportation systems [14].

More recently, attention has been focused on understanding the human mobility using the profiling of users [15]. Data generated by static and mobile sensors implemented in different transportation systems and smartphones allow to understand the patterns and citizens' habits at large scale [16]. This is used for semantic information extraction on the mobility of users but also for the study of spatiotemporal variation in travel regulations through transit data [17]. Mobility user profiles can offer valuable information for understanding the disaggregate and aggregate spatiotemporal activity patterns [18], but the proposed methods are static and do not take into consideration data in motion and the performance has not been tested with large datasets.

Several challenges have been identified in order to effectively profile user behaviour in smart mobility systems, including the learning issues for missing values, data cleansing, dimension reduction, sparse learning, and heterogeneous learning [19]. Massive amounts of raw data collected by nomadic devices (e.g., smartphones) must be cleaned, aggregated, and then processed using state-of-the-art methods and algorithms. This is the case of the shared mobility services. Previous research focused on the investigation of ride sharing opportunities. Bicocchi and Mamei [20] showed that through mobility data analysis, efficient solutions for extraction of suitable information from mobility traces can be used to identify ride sharing opportunities. The literature shows that there is a need for optimisation of these systems, which need to solve different problems related to the required features and characteristics, e.g., the dynamic character, automated matching, and cost sharing [21]. A suggested solution comes from a good understanding of users' behaviour and preferences, which is an essential feature when designing dynamic shared mobility systems.

In order to make use of collected data for large-scale mobility sharing services, users' travel behaviour and preferences must be extracted. The very first step in this process is the extraction of the duration and location of activities from raw data. A detailed review and comparison of the methodologies from literature is presented in [22]. However, all mentioned methodologies suffer from limitations when applied to dynamic and live profiling on large datasets. In practice, recommendation systems require to profile users in an environment with continuous data generated by dynamic movements of users and means of transport. They need to extract knowledge that can contribute to the mobility services to understand the human travel behaviour and automatically recommend suitable sharing services for each individual. Moreover, the analytics must be done at different levels of aggregation and resolutions, with dynamic precision and scaling, e.g., ride sharing requires a higher accuracy than the classification of secondary activities (e.g., shopping, gym, or restaurant).

In a next step, using the detected locations from the previous step, methodologies must be implemented in order to learn user mobility patterns and to perform the knowledge discovery from raw data. An example of knowledge discovery from literature is the trip purpose identification from GPS tracks [23]. They identified two main groups of trip purpose imputation routines in the literature: rule-based systems based on the position of the activity, timing, and

geographic information system (GIS) data and machine learning approaches which focus more on the activity and less on position. Montini et al. [23] used random forests [24], a machine learning algorithm that has been successfully applied in different transport-related classification problems. Data from GPS and accelerometer sensors were used as input, and the respondents were asked to correct an automatically generated travel diary that was used to extract specific features for semantic interpretation of the said data.

A similar application used in the current research is the identification/classification of each activity/visited location (e.g., home and work). The proposed profiling methodology uses only the GPS data, specific data science techniques for indexing, clustering, and querying, and as the training data, a set of known location visit patterns for each location type. The key novelty of our approach is that our methodology is able to capture detailed and complex visit patterns of users and locations through the profiling layer, which can be used in a multitude of applications simultaneously. Some usage examples are explained and evaluated in the rest of the paper (e.g., parking sharing, ride sharing, location type, and activity classification).

2.2. Data Science Methods for Efficient Computation. One of the most complex operations in big data systems is the study of the relationships between a group of entities interacting with each other in a given space. This is also the case of smart mobility systems (e.g., ride sharing), where a high number of entities represented by people, cars, and locations interact with each other. The problem becomes even more difficult when those entities interact in multidimensional spaces, represented by the properties of the above entities in motion (e.g., day, hour, geolocation, and so on). In physics, this type of problem is defined as the n-body problem [25], which is one of the most challenging topics in high-performance scientific computing, addressed by the Barnes–Hut simulation [26].

In order to reduce the costly $O(n^2)$ computation time complexity for calculating the distance between each entity, alternative optimisations have been developed in form of tree algorithms. Binary search trees are used to efficiently search and sort as by traversing the tree from root to leaf means that each comparison allows the operations to skip about half of the tree. This results in a complexity of $O(\log n)$. In a quadtree [27], each internal node has exactly four children and is most often used to partition a two-dimensional space by recursively subdividing it into four quadrants or regions. Similarly, octrees [28] are the three-dimensional analog of quadtrees. For spatial representations, each node in an octree subdivides the space it represents into eight octants.

In order to profile on more dimensions, trees with unlimited number of spaces must be used. *K*-dimensional trees (*K-d* trees) [29] are a special case of binary space partitioning trees for organising points in a *k*-dimensional space. Even if *k-d* trees are a solution for more than three dimensions, this is not suitable for efficiently finding the nearest solution in high-dimensional spaces (as if the dimensionality is *k*, the number of points in the data should be $N = 2^k$). Using the *k-d* tree with high-dimensional data, the efficiency is no better than an exhaustive search [30].

Recent studies demonstrate that ND-tree [31] data structures are efficient in high-dimensional spaces. For example, in the nearest neighbour search problem, ND-tree algorithms are effective in improving query performance in both uniform and nonuniform datasets [32] but also in the study of similarity searches in multidimensional nonordered discrete data spaces [31], a feature that is used in this study.

3. Methodology

3.1. Multidimensional Profiling in Previous Work. Multidimensional and dynamic profiling requires technologies which allow the fast processing, indexing, and querying of big datasets. In the remainder of this section, we present the data modelling framework which incorporates graphs and time series in multidimensional data models, along with the major technological implementation challenges.

The GreyCat [33] framework, formerly known as KMF [34], presented in a previous work [35], is a solution for analysing complex data in motion at scale with temporal graphs [5]. There are a number of required features (e.g., modelling with graphs, temporal aspects, and what-if analysis exploring different alternatives) presented in [35].

Another feature that is important in big data systems is the ability to lazy load nodes, meaning to load into main memory only the necessary data that need to be processed rather than to load and query each time the entire dataset. Naturally, many analytic tasks are processing only parts of the dataset. This also counts for the case study of this paper. Therefore, we suggest to load data, i.e., the nodes of our data graph only on-demand, while the graph is traversed. As an example, even if high accuracy datasets are available at an order of a few meters, if the application needs to profile to a maximum of, e.g., one kilometer accuracy, there is no need to load and process the data at a higher resolution. This will save both resources and time.

If in the previous work the proposed framework is used for the first time in the transportation domain to find possible groups of users that can use a ride sharing system using data at rest, the current work proposes a more user centric approach that can handle data in motion at scale for multiple applications (e.g., parking sharing, location classification, and nonrecurrent trips profiling).

3.2. Overview of Data-Driven Scalable Method. In this paper, we present a new way of profiling multidimensional and temporal data, specifically designed to deal with large quantities of data—in live and with different physical constraints, such as limited memory or processing power (as is the case of nomadic devices like smartphones).

Our methodology is generic in the sense that it can use any specific profiling algorithm that uses a tree-like structure to divide a parent space into two or more children subspaces. For instance: binary trees, quadtrees, octrees, and *K-d* trees

are all easily implementable and can be integrated in our methodology.

The proposed architecture shown in Figure 1 has three independent layers. The base layer is the lowest level which represents the *raw data layer*, dealing with data management (e.g., collect and store data). The *processing layer* deals with data processing to produce well-structured and fast-to-query spatiotemporal profiles. Among different processing tasks, the profiling is our main focus in this work, which is done through other different tasks (e.g., reduce, map, and apply to spatiotemporal data trees). Finally, the highest layer is the *application layer* where any specific transportation problem can be translated in high-level profile queries. The main advantage of the proposed architecture is that the profile layer is built once and then shared across several transportation applications, hence reducing the required infrastructure and resources.

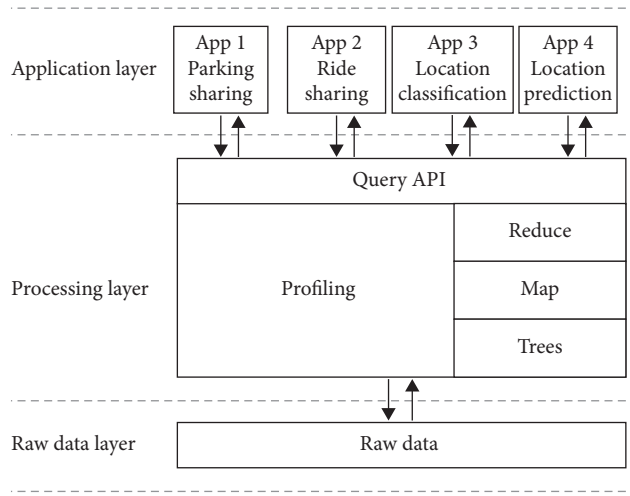

Figure 1: Architecture and abstract layers.

3.3. Terminology. As the current work includes terms from different domains, it is necessary to define the terminology that will be used throughout the entire work.

(i) *Tree*: a directed acyclic graph starting from a root node.

(ii) *Space coverage*: the N-dimensional min-max vectors that define the boundaries of the space covered by the subtree.

(iii) *The root node* is the top node in the tree. It covers the widest N-dimensional space of the tree.

(iv) *Child node*: a node directly connected to another node when moving away from the root node. Child nodes have always a smaller space coverage than their parent nodes.

(v) *Parent*: the opposite notion of a child.

(vi) *Leaf node*: a node without children.

(vii) *Siblings*: a group of nodes with the same parent.

(viii) *Degree*: the number of children of a node.

(ix) *Path*: a sequence of nodes connecting a node with a descendant.

(x) *Level*: the number of connections between the root and the node. The root node is of level 0.

(xi) *Size* of the tree is the total number of data indexed in the tree.

(xii) *Height*: the height of a tree is the maximum level reached by its nodes.

(xiii) *Resolution*: the smallest space coverage allowed for the leaf nodes. It is an N-dimensional vector representing the minimum difference allowed between the minimum and the maximum on each of the N dimensions.

(xiv) *Number of dimensions* represents the number of different features we want to profile (e.g., day of the week, time of the day, and geolocation). By default, the proposed architecture supports until 32 dimensions that are easily extensible to 64.

(xv) *Max buffer size*: the maximum size of the data stored in a node before creating a sublevel of child nodes.

(xvi) *Timeline*: a sequence of ordered timepoints.

(xvii) *Temporal resolution* represents the maximum quota in time for each profiling tree before creating another tree.

3.4. Live Profiling, Indexing, and Preprocessing. In this section, we describe specifically the preprocessing step, in which the data in motion are indexed as soon as they are received from specific sensing systems (e.g., mobile devices) or databases.

The following example deals with location data represented by points in the geographical space, represented in Figure 2. We describe how each tree structure is created based on the space partitioning and indexing of each quadrant, from the root (*Level 0*) to the leaf level (in our example Level 3). It is important to stress that the indexing and profiling methods are completely independent of the final applications that will access and use the data on the application layer.

The profiling methodology can be summarised as the following chronologically ordered steps, which are continuously performed as soon as new data are available:

(1) Start with an empty timeline.

(2) Create the first profiling tree once the data are loaded from a dataset or received through a sensing system, as can be seen in Figure 2, at *Level 0*.

(3) Once the buffer is full at the root *Level 0* (reached the max *buffer* limit of data stored at the node level, defined at design time), create child nodes of *Level 1* and redistribute the data into the corresponding subspaces. Any new data received at *Level 0* will be automatically forwarded to *Level 1* subspaces. At this step, the node at *Level 0* is transformed from a node that stores data in a *router node*, defined as a node

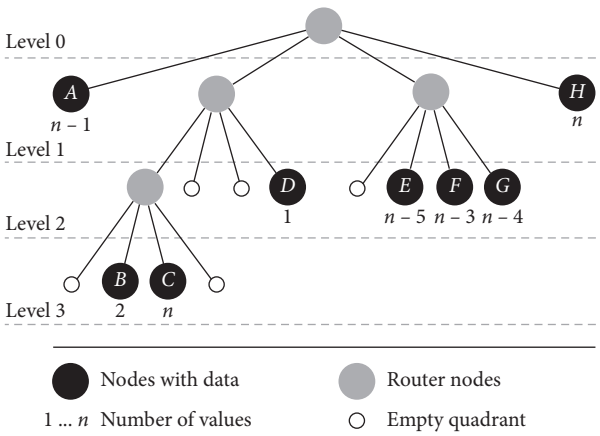

Figure 2: Profiling tree structure.

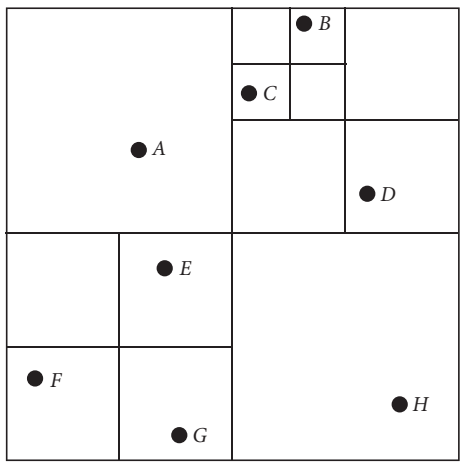

Figure 3: Profiling space partitioning for geolocation data.

that has no data but acts as a *path* that connects the node with its descendant subspaces.

(4) Each subspace has its own buffer and divides the parent dimension boundaries by two or more on each dimension. In the case of geolocation data, Figure 3 shows how the space is divided in quadrants. At *Level 1*, there are four subspaces: (1) *A*, (2) the quadrant formed by *B*, *C*, *D*, and two empty, (3) the quadrant formed by *E*, *F*, *G*, and one empty, and (4) *H*.

(5) Repeat steps 3 to 5 recursively until one of the following conditions is met:

  (a) The temporal resolution of the current profiling tree has expired. As an example, if we set the maximum temporal resolution to one hour, even if the max buffer size was not reached, a new tree will be created.

  (b) The tree reaches the maximum allowed size. This is a requirement to keep the process as fast as possible, since the use of too large trees makes the search computationally harder and more time-consuming.

  (c) We can observe in Figure 2 that nodes (2) and (3) created at Step 4 on *Level 1* are split again into four children and transformed from nodes with data in router nodes. The same process continues also at *Level 3* until one of the above conditions is met.

(6) Once a tree is complete, it is stored and the process continues with the creation of a new tree. As can be observed in Figure 4, the new tree will have a new timepoint and the entire process from 3 to 6 will be repeated.

3.5. Querying and Postprocessing. The multidimensional and temporal features of the proposed profile offer several ways to query it in order to allow a wide range of applications. A query can specify a range in time, specific days and hours of the week, and a level of precision in the multidimensional space and can ask either for all the results within a specific

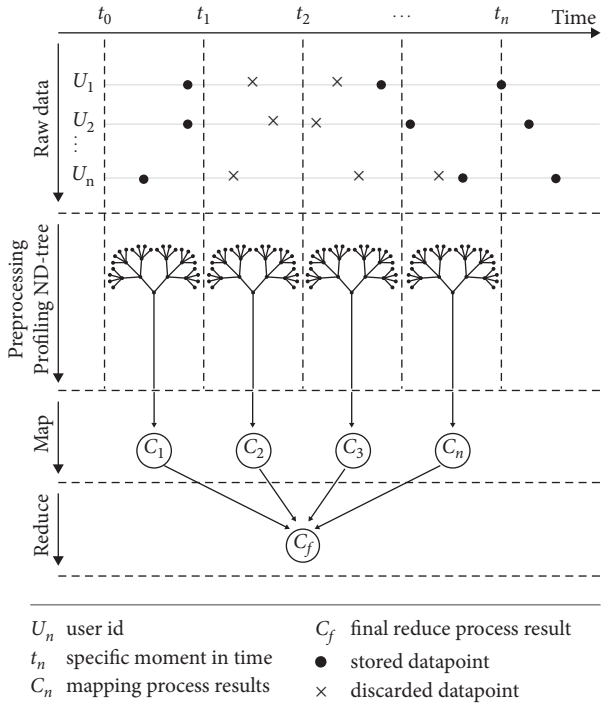

Figure 4: Data structure, query, and process flow.

range or the top *N* results from a specific complex query. The entire flow of the query process is described in Figure 4 and will be described in the remaining of this section.

In order to demonstrate the process flow, we present the example of a smart mobility shared system (e.g., car sharing, carpooling, and parking sharing) that provides the geolocation data for a number of users U_n through a smartphone application using the integrated sensing system. The system can then perform specific complex queries which must return a result in an order of milliseconds.

For example, a ride sharing system can perform a query to get all the locations that a user visited in the last two years in a specific geographical region, in specific days of the week, and specific time intervals during the day, with a specific geographic resolution. This can be useful to find possible

matches with other users that have similar profiles. The results of the queries can be stored to save the user profiles for future fast access to avoid repetitive queries or can be discarded, according to the objectives of the application. Another example can be a location classification application that extracts the user visit pattern for all the visited locations that have been visited at least s times during a specific time period and a specific geographical area. This can be useful to instantly filter trajectories points that do not represent a specific location and to classify the filtered locations based on specific location duration and time of the day.

Moreover, a query can become even more complex and can be used to show the top N locations visited, in a specific day of the week and interval of the day. This query can be used to detect the most visited locations and to quickly detect, e.g., home and work location, in order to propose a personalised itinerary end, e.g., at the user's home or to be used to match users that are compatible for parking sharing.

It is important to mention that for all of the above examples independent of any other application, the same process flow is applied. This is described below:

(1) *Data Management and Temporal Resolution*. The data are captured and processed according to the temporal resolution, in specific time intervals t_n. This temporal resolution must be set in the very beginning and represents the minimum time interval that two specific trees are indexed. For example, if from the applications domain it is known in advance that no application that uses the profiling data will need a higher time resolution than one minute or a higher geographical resolution than four meters, there is no sense to set this limit lower. A lower time interval than the minimum required will also require more resources or time to process the entire flow, providing also redundant data. There is no upper limit but just the one given by the indexing method (e.g., for the geographical space, a quadrant, no matter of the dimension in measurement units). This dynamic is important to mention as in some applications like carpooling it is possible to perform different queries with different parameters and to increase/decrease the resolution to determine, e.g., which specific routes the user uses. This information can be useful to calculate the compatibility for matching different profiles.

(2) *Efficient Storage*. There are some important aspects to mention regarding the data management and temporal resolution. First, if a user is changing the location between two consecutive t_n, t_{n+1} data points, the geolocation is stored in a node. Second, if the user is in the same location for more than two consecutive time intervals, the same information is not replicated through consecutive timepoints but is discarded, represented in Figure 4, raw data layer with x. Thus, for a visited location, only the arrival and departure timestamps are stored, which helps in cleaning the dataset of duplicate values and reduce the required storage resources. Third, if at any time t_n a query is performed and no points are found at t_n, the data from t_{n-1} will be returned, and the process begins to backtrack until a stored point is found.

(3) *Profiling Phase*. The trees are created for each time interval, following the methodology from Section 3.4.

(4) *Map Phase*. When a query is performed, the query is divided into several subqueries touching several trees and several subspaces (e.g., return the top n points from a specific time interval, specific day of the week and hour, and from a specific geospace, with a specific accuracy). The search phase can be distributed among any number of computation units and threads as needed, according to each application and specific domain requirements.

(5) *Results*. The results are then collected, and then the *reduce phase* does the synchronisation, waits for all the running threads to finish, removes the duplicates, sorts the results, and does a final postfiltering if needed.

Another important feature in the context of profile sharing with multiple applications is the ability to have a high level of parallelism. This feature brings important advantages:

(1) All queries can run in parallel: this is an important requirement when multiple applications share the same profiling layer, and multiple queries can be performed, at the same time, on the same tree indexing.

(2) Each query can be mapped to one or several profile trees according to the targeted time range of the query. The search within the targeted trees can be done as well as in parallel.

(3) Since a query can involve several subspaces within a tree, the search within these subspaces can be executed in parallel as well.

3.6. Location Visit Pattern Extraction. In addition to filtering in near real time the visited locations based on spatiotemporal complex queries, the final result of the entire process flow (i.e., starting from indexing and preprocessing, live profiling, querying, and postprocessing) is used also to extract the weekly activity pattern visit for each location visited. This is done by clustering all the visit records from the time range specified in the query parameters in a matrix with dimensions of 24 hours and the seven days of the week. Each matrix element represents the number of times a person visited a specific location, normalised by 1000, as can be seen, e.g., in Figure 5.

3.7. Location Profiling and Activity Classification. An earlier exploratory study from a previous work [36] presents the extended methodology behind the classification methodology. To summarise, the classification and labelling of each location is done by computing the Euclidean distance

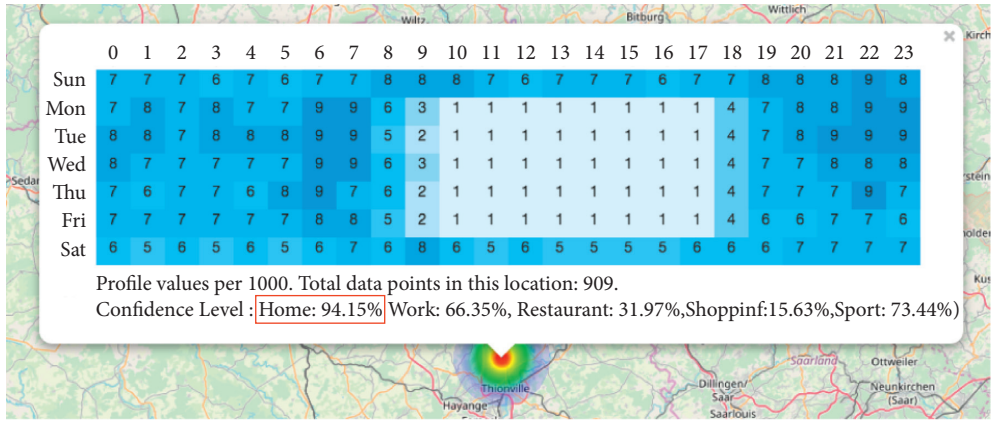

Figure 5: Example of home location classification.

(between 0 and 1) from a generated training matrix representing a known location type (e.g., home and work) and the matrix obtained for each location visit pattern. The smaller the Euclidean distance is obtained, the higher the probability that the profile of a specific location can be labelled as a type of a specific location. Figure 6 shows an example of the training matrix for a home location.

Similar matrix can be generated for other type of locations and activities (e.g., work, restaurant, shopping, and sport). Figure 5 shows an example of the classification result for a location that has 94.15% confidence level that is a home location.

Evaluation and usage examples of location profiling and classification are presented in Sections 4 and 5.

4. Evaluation

4.1. Dataset Description. In this study, two types of datasets were used in order to perform different evaluations of profiling results. Each dataset has different purposes and requirements, as follows.

First, a dataset for which we have the ground truth was used in order to evaluate the location classification and labelling accuracy based on the profile extracted for each location visited by the users. For this type of evaluation, the dataset must be accurate with less GPS errors and the results must be validated by the respondents. As the validation is done at the individual level, we did not use a huge dataset for the validation. The dataset used was based on data collected by Google Maps of 17 users from the University of Luxembourg; it is individual based and each respondent was able to easily test and validate online the results [37]. Moreover, the data come already error-filtered as the data are collected using not only the GPS sensors from the smartphones but also a fusion of data from native Android sensors like Bluetooth, Wifi, and motion sensors, which are used to validate the location even when the GPS signal is poor, e.g., inside the buildings.

Second, a larger dataset was used in order to evaluate the computational speed, performance when scaling, and accuracy. For this type of evaluation, the most important aspects are the size of the dataset, i.e., the period of time covered and the number of users. This dataset must be sufficiently large; more specifically, it should cover a relatively long time period and contain a large number of users in order to meet computational requirements within the scaling phase. The dataset used was the Geolife dataset [38]. This publicly available dataset contains around 24 million GPS points from China, collected in the Geolife project [39] from 182 users with smartphones and GPS loggers in a period of five years.

4.2. Location Classification Accuracy. The evaluation of location classification accuracy was performed by a group of 17 respondents who uploaded their GPS data exported from Google Maps to the publicly online version of the tool developed in the current research [37]. The respondents were then asked if the home and work locations were accurately detected. The results show that 100% of the respondents stated that the home and work location were correctly classified and labelled with the highest confidence level. Of course one can argue that the home and work locations are trivial to be used as an example of classification, but the scope of our evaluation is only to demonstrate that the proposed framework and methodology are able to automatically classify locations and activities performed in near real time, without any user input and only based on the GPS data. In another work [40], we extended the evaluation and methodology also for other type of locations and activities (e.g., restaurants, shopping, and sport activities) which is out of the scope of this paper.

4.3. Computational Speed. First, in order to test the computation speed of the proposed profiling method when scaling, multiple tests have been performed with different amounts of data. Figure 7 shows the results obtained when processing 12 different amounts of data, from 1 million to 24 million valid points. Moreover, as can be seen in Figure 7, the general trend line when scaling is close to logarithmic, something that confirms the complexity reduction explained in Section 2.

Second, a speed comparison has been performed between a classical linear computation in comparison with a

	Home																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Sun	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
Mon	9	9	9	9	9	9	9	9	0	0	0	0	0	0	0	0	0	0	0	9	9	9	9	9
Tue	9	9	9	9	9	9	9	9	0	0	0	0	0	0	0	0	0	0	0	9	9	9	9	9
Wed	9	9	9	9	9	9	9	9	0	0	0	0	0	0	0	0	0	0	0	9	9	9	9	9
Thu	9	9	9	9	9	9	9	9	0	0	0	0	0	0	0	0	0	0	0	9	9	9	9	9
Fri	9	9	9	9	9	9	9	9	0	0	0	0	0	0	0	0	0	0	0	9	9	9	9	9
Sat	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9

Figure 6: Generated matrix for location classification training process.

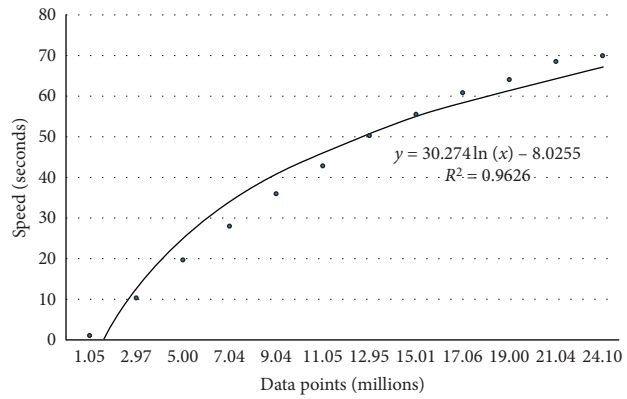

Figure 7: Computation speed when scaling.

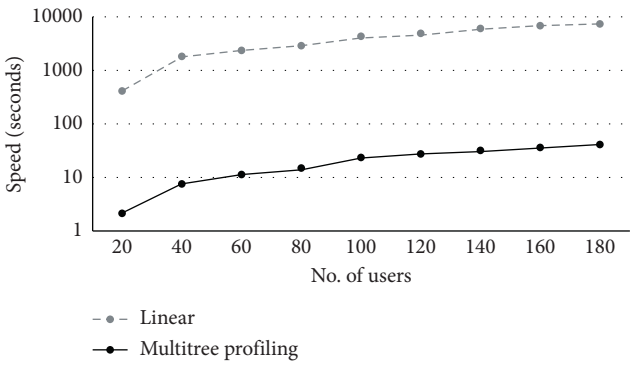

Figure 8: Linear profiling vs multitree profiling speed.

multitree profiling method, presented in Figure 8. In this experiment, 9 speed tests have been performed, with data from different number of users, from 20 to 180. The results are displayed on a log scale at y axis. The experiment clearly shows that the speed of using a multitree architecture is around 10^2 faster compared with a classic linear architecture.

Another important aspect in practical implementation of fast processing is the amount of resources needed. Most of the time, the bigger the size of the dataset is, the more the processing resources are needed in order to have the highest speed. The database research community identified graphics processing units (GPUs) as the most effective co-processors for parallel data processing [41] mainly because the dataset is processed by hundreds or thousands of small CPU nodes.

To the best of our knowledge, one of the fastest massive parallel architectures is MapD [42]. Recent experiments show that massive datasets with billions of geolocation routes can be processed and visualised in milliseconds [43]. But everything comes at a cost. Table 1 shows a comparison of speed and resources needed to process the Geolife dataset using MapD with a linear in hardware scaling method and very powerful but costly hardware, compared with a multitree profiling performed by a user PC which is a logarithmic in time or hardware method.

4.4. Accuracy. Since the profiling is done for different subspace sizes (resolution), the accuracy is highly correlated with the resolution (the smaller the resolution, the better the accuracy) and the number of levels that need to be queried to reach from the root to the leaf level (more levels means higher details, but longer time to search).

In our specific case, the accuracy depends on the size (width and height) of each resolution. Using the Geolife dataset, accuracy tests have been performed, and the results are presented in Table 2 from a minimum of 4.77×4.77 m to a maximum of 5000×5000 km.

The experiments confirm that the dataset average error is very close to the mathematical predicted error. The provided results can be used as a guideline for choosing the minimum resolution for each transportation application, based on the average and maximum error of each resolution. Thus, for any type of application, it must be assessed if the average and maximum errors are acceptable and tolerated in the domain. Different transportation applications require different precision and maximum errors. Table 3 presents an example of comparison for different applications with the precision and amount of data needed.

For some applications like ride sharing, the accuracy is important as, e.g., the meeting point of different users that can share the same car cannot have large error. A study [44] shows that on average only about 60% of the passengers will accept to walk 150 meters for transit to another bus stop and 90% of the passengers will accept to walk 50 meters. The same strict requirement has also the location classification (e.g., home, work, shops, and restaurants) or activity classification (e.g., sport and shopping) systems. In order to keep a higher quality of service and a higher rate of user retention, the maximum error must be lower than the acceptable distance that a passenger has to walk if, e.g., the suggested

Table 1: Resource and performance comparison.

Speed (ms)	Cost (EUR)	Hardware	Scaling	Method	Summary
0.5–20	40,000	8x Nvidia Tesla K80	Linear in hardware	Load all points in memory	Big data on big hardware
36–51	2000	User PC	Log in time or hardware	Multitree profiling	Big data on small hardware + smart models
6895–7579	2000	User PC	Linear on time or hardware	Linear parsing	Big data on small hardware

Table 2: Accuracy.

L_x	L_y	Unit	Worst err.	Dataset avg. err.	Math. err.
4.77	4.77	m	3.372899	1.824923	1.824993
38.2	19.1	m	21.354449	11.330335	11.330761
153	153	m	108.187337	58.537250	58.537314
1.22	0.61	km	0.682000	0.361813	0.361873
4.89	4.89	km	3.457752	1.870842	1.870902
39.1	19.5	km	21.846398	11.589896	11.590412
156	156	km	110.308657	59.691685	59.685239
1250	625	km	698.771243	370.746174	370.771200
5000	5000	km	3535.533906	1913.060314	1912.992000

Table 3: Usage examples.

	Type	Precision needed	Maximum error	Amount of data needed
1	Ride sharing	High	<50 m	Medium
2	Location/activity classification	High	<50 m	Small
3	Parking sharing	Medium	<150 m	High
4	Nonrecurrent trips	Low	<50 km	Medium

location/meeting point is not precisely in the location designated by, e.g., a recommendation system/trip planner.

For other transportation problems like parking sharing, the error can be higher as it is not unusual to park the car and walk for a decent distance until the destination, but again under the limit of the maximum user's tolerated error [45]. There are also other applications where the error can be bigger; there is no need of very detailed profile and the error can be much higher, of an order of dozens of kilometers. This is the case of nonrecurrent trip analysis, e.g., holidays or business trips where anyway the clusters and visualisation are much bigger than the above examples.

The capability to be versatile in order to handle various application requirements in the same system represents a requirement in a shared architecture. In the next section, practical usage examples of all the applications compared in Table 3 will be presented.

5. Usage Examples

Collaborative mobility services (e.g., ride sharing) represent one of the best case studies of the methodology presented in Section 3. Different types of data collected from users' smartphones represent a new dimension that adds complexity on finding efficient solutions for combining users and transportation resources in collaborative systems. Each dimension is represented by the properties of those entities, combined with the types of queries which are performed, e.g., day, hour, location, age, sex, etc. Some of them have subdimensions, e.g., the location where users perform activities can have as subdimensions the starting and ending hour of an activity, the geographic coordinates of the location, and the radius of the geographic space that represents a location. In this study, the multidimensional profiling refers to the concept of profiling all of these dimensions, where each entity has specific properties.

5.1. Parking Sharing. A group of two or more users can share the same parking place if they use it at different times of the day. In other words, the more dissimilar the users' profiles are for the same location, the higher the compatibility for parking sharing there is. The dynamic character of the proposed profiling methodology makes possible the assessment of parking sharing both for (*a*) planned long-term parking sharing and (*b*) short-term or ad hoc parking sharing.

In order to demonstrate the usefulness of a flexible, dynamic, and fast profiling framework, we present the following case study.

Figure 9 shows the profile of *User 1* who works in the proximity of the home of *User 2*.

User 2 is part of a peer-to-peer (P2P) parking sharing application. As we can observe, for *User 1*, the highest probability to be in the parking location is from *9 AM* to *7*

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Sun											1	1						1	1	1				
Mon									1	17	32	23	20	22	17	16	17	15	16	9	3			
Tue										16	24	17	26	25	18	16	15	16	17	7	2			
Wed										15	18	13	17	21	20	13	15	12	15	7	3	1	1	
Thu									1	11	14	12	18	19	17	18	17	20	17	8	2	1	1	
Fri									1	11	23	13	17	20	15	16	15	14	15	7	3	2		
Sat											1	3	2	3	3	1	1	1	1	1	1			

Profile values are per 1000. Total data points in this location : 59845

Figure 9: Profile of User 1 for long-term parking sharing.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Sun	6	7	8	7	7	7	7	7	11	10	8	8	7	7	7	7	6	5	5	6	8	8	9	9
Mon	8	8	7	7	7	9	8	9	9	2	1	1	1	1	1	1	1	1	1	5	7	7	8	9
Tue	9	9	8	7	7	8	8	10	9	3	1		1	2	2	2	2	2	3	5	7	7	9	9
Wed	8	7	7	7	7	7	7	9	9	3	1	1	1	1		1	1	1	1	4	8	10	9	7
Thu	8	8	7	7	6	7	7	9	9	4	2	2	2	2	2	2	2	2	3	5	7	7	8	9
Fri	9	10	9	7	7	6	7	8	9	3	1	1	1	1	1	1	1	2	3	6	6	5	6	6
Sat	5	5	5	6	7	6	6	7	8	10	8	5	2	2	3	5	6	7	6	6	6	5	6	6

Profile values are per 1000. Total data points in this location : 152440

Figure 10: Profile of User 2 for long-term parking sharing.

PM, from *Monday* to *Friday*. Similarly, Figure 10 shows the profile of *User 2*, who has the lowest probability to be in the same location when *User 1* is in the same location.

With this information, a system can match the two profiles with highest compatibility index (i.e., the highest Euclidean distance between profiles) to share the same parking location as they do not overlap. Moreover, this is done without asking the users any prior information but profiling their behaviour, extracting their pattern to visit the location, classifying the location, and matching profiles that are synchronised for specific sharing services.

The results denote a good example of how the presented profiling methodology can be used to assess the compatibility for long-term parking sharing of two or more users using, e.g., specific indicators for collaborative mobility between individuals [22]. More precisely, the profiling can be used to search in a specific region and users that have profiles that match other users for specific applications/sharing services. Nevertheless, there are some conditions that must be met to have an accurate long-term profiling.

First, there should be enough time data in order to have an accurate profiling. This will ensure that the location profile has a specific pattern over time and is not just a location that is visited randomly. *Home* and *work* locations are typical examples of locations that have a specific pattern over time.

Second, it is not required for the location accuracy to be extremely precise, as in the case of long parking duration, people are likely willing to walk for a decent location from the parking lot to destination [45].

In the same way, a P2P parking sharing service can be used also for ad hoc or instant parking sharing, presented in the following example. Particularly, if a user is part of the P2P parking sharing application and during a trip he notifies the application that he must stop for a specific time period in a specific place, the application can instantly search for other users in the system which has free parking slots during that specific time interval. In order to test this case study, using the Geolife dataset described in Section 4, we took a random user and a random visited location and performed a search for compatible users to simulate a match of an ad hoc P2P parking sharing request. Table 4 presents the results of different searches, at different resolutions.

As can be observed, at very small resolutions (i.e., 4.77×4.77 m and 38.2×19.1 m), no compatible users have been found, as the search is too detailed. When increasing the resolution to 153×153 m, three compatible users were found and the maximum error can be 108.187337 m which is acceptable. If we increase the resolution, more compatible users are found, but also the maximum possible error increases, to the extent that some results are not relevant, as the walking distance will be then too long and most likely not acceptable for the user to walk. In this case, we can argue that the best resolution would be in this case 153×153 m, which can give the best results regardless of the maximum possible error.

This concrete example demonstrates the capabilities that the proposed profiling methodology offers: fast processing large quantity of data, on demand and in near real time, coupled with the ability to extract user insights, behaviour, and travel pattern with minimum computation, storage resources, and user input. This is of a great importance for the next generation of AI autonomous travel planners and sharing services, as in most of the cases, the data collection

Table 4: Ad hoc P2P parking sharing matching.

L_x	L_y	Unit	Users matched	Max error
4.77	4.77	m	0	3.372899
38.2	19.1	m	0	21.354449
153	153	m	3	108.187337
1220	610	m	7	682.00000
4890	4890	m	11	3457.752000
39100	19,500	m	24	21846.398000

and processing are done through the passenger's mobile device. In a matter of seconds, the system must process years of geolocation data, extract the insights, user habits, and preferences, and provide reliable services. The fact that now it is possible to use the online tool [37] on any browser without installing any software and all the computation is done locally on the device is in line with the requirements of mobile devices which have limited autonomy and computation resources. At the same time, it also fits the mobile applications' user preferences because asking continuously user input information is no more applicable and sustainable in our days.

5.2. Ride Sharing. The profiling of users' mobility for the days and hours of the week is an important information that can be used for a recommendation system in order to analyse which users can match for ride sharing/carpooling. There are some conditions that should be met in order to organise a ride sharing between two or more users, such as the departure and arrival position to be suitable for all the participants and the departure and arrival time to synchronise matching at best their schedule. The latter condition can be assessed by analysing the probability to be in a specific location, by the days and hours of the week, and can be used in a collaborative mobility system [22].

In order to exemplify this case study, we searched in the small database presented in Section 4 for compatible respondents that can match a carpooling service. Figure 11 presents the extracted weekly heatmap of time spent in the residence for two neighbours (User 1 and User 2) that are working also in the same area, as can be seen in Figure 12. This is a typical situation where users can participate in a long-term ride sharing, as their schedule is pretty fixed in most cases.

As can be observed from the heatmap, they are at home typically outside of the working hours. Moreover, they both leave the house during the week around *9 AM* and they return at home around *7 PM*, resulting in a good synchronisation. Figure 13 shows that the schedule that they have for the work location makes it suitable for long-term carpooling as they can share the same car for commuting to work.

At the same time, we noticed also that also User 3 works close to User 1 and User 2. Moreover, User 3 can join the ride sharing when available. Analysing Figure 12, we can observe that in order to pick up User 3, the trip must be rerouted. If the first two users consider the travel time very important for them, this can be an inconvenience because the trip will be four minutes longer. However, if the users are flexible on timing and accept this trade-off, another passenger will be picked up in the same car and there will be a car less on the road and an available parking space at the destination. This approach will maximise the objectives of car sharing and parking sharing, by reducing the traffic and, respectively, the demand for parking space. Of course, users should assess different variations and accept or deny different options. In order to simplify the process, each user can state his preferences, limitations, and flexibility in an application interface so that the searching algorithm provides only suitable results for each user. Nevertheless, if this principle is applied at the level of big cities, it will result in more people in less cars and a reduced traffic congestion.

This concrete example shows how the profiling can be used both for long-term carpooling and short-term ride sharing as combined services. It is important to observe that using the proposed profiling methodology, all the necessary steps for matching people and sharing services (i.e., location visit pattern extraction, search of compatible users, and trip planner) can be done automatically and dynamically, without any user input but only the access of history GPS data, which in our days can be easily obtained via mobile devices.

5.3. Location Type and Activity Classification. Profiling the pattern for visiting specific locations gives also a possibility to automatically classify the location to a specific category. As we can observe in Figures 9–11, the location visit patterns obtained can be clearly identified as *Home* and *Work* locations and dynamically displayed as in Figure 14.

The same can be done with any type of location for which there are defined patterns which can be identified and the classification can be done automatically. In an extended exploratory study [36], we showed that using the combination of the proposed profiling method combined with observed user habits learned from surveys and extracted as activities matrix, it was possible to automatically classify the type of activity performed in a specific location.

Moreover, as the profiling can take a large time period into consideration, it is now possible to detect changes of user travel habits by detecting changes of regular visit to specific locations. In Figure 14, we can see that for the same user, two homes and workplaces are detected, as the profiling detects recursive similar patterns in different time periods. This means that the proposed profiling method can not only detect recurrent habits to secondary activities but also change of habits, something that is not easy to detect with a static method. The insights obtained based on these changes can be used to adapt and personalise transportation services to match the passenger's habits, which will result in a better service quality.

6. Future Work

Future work includes finding common subqueries across several requests and execute them once, something that can reduce the number of operations and tasks executed.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Sun	6	7	8	7	7	7	7	7	11	10	8	8	7	7	7	7	6	5	5	6	8	8	9	9
Mon	8	8	7	7	7	9	8	9	9	2	1	1	1	1	1	1	1	1	1	5	7	7	8	9
Tue	9	9	8	7	7	8	8	10	9	3	1		1	2	2	2	2	2	3	5	7	7	9	9
Wed	8	7	7	7	7	7	7	9	9	3	1	1	1	1		1	1	1	1	4	8	10	9	7
Thu	8	8	7	7	6	7	7	9	9	4	2	2	2	2	2	2	2	2	3	5	7	7	8	9
Fri	9	10	9	7	7	6	7	8	9	3	1	1	1	1	1	1	1	2	3	6	6	5	6	6
Sat	5	5	5	6	7	6	6	7	8	10	8	5	2	2	3	5	6	7	6	6	6	5	6	6

(a)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Sun	7	6	6	6	5	6	7	10	10	9	10	9	9	7	7	6	8	8	8	7	9	8	9	10
Mon	9	7	6	7	7	7	10	11	5	1										2	8	10	12	11
Tue	8	7	7	6	7	8	11	12	7	1										3	9	11	12	11
Wed	8	7	5	5	6	7	10	10	6	1	1	1					1	1	2	3	6	9	9	9
Thu	8	6	6	6	7	7	10	11	8	1									1	3	7	9	11	9
Fri	8	6	6	6	6	7	10	11	7	2	1	1	1	2	1	1	1	2	4	6	8	9	9	9
Sat	8	7	7	5	6	7	8	10	12	9	8	7	5	4	4	4	5	5	7	6	6	6	6	7

(b)

Figure 11: Home location profile for User 1 and User 2.

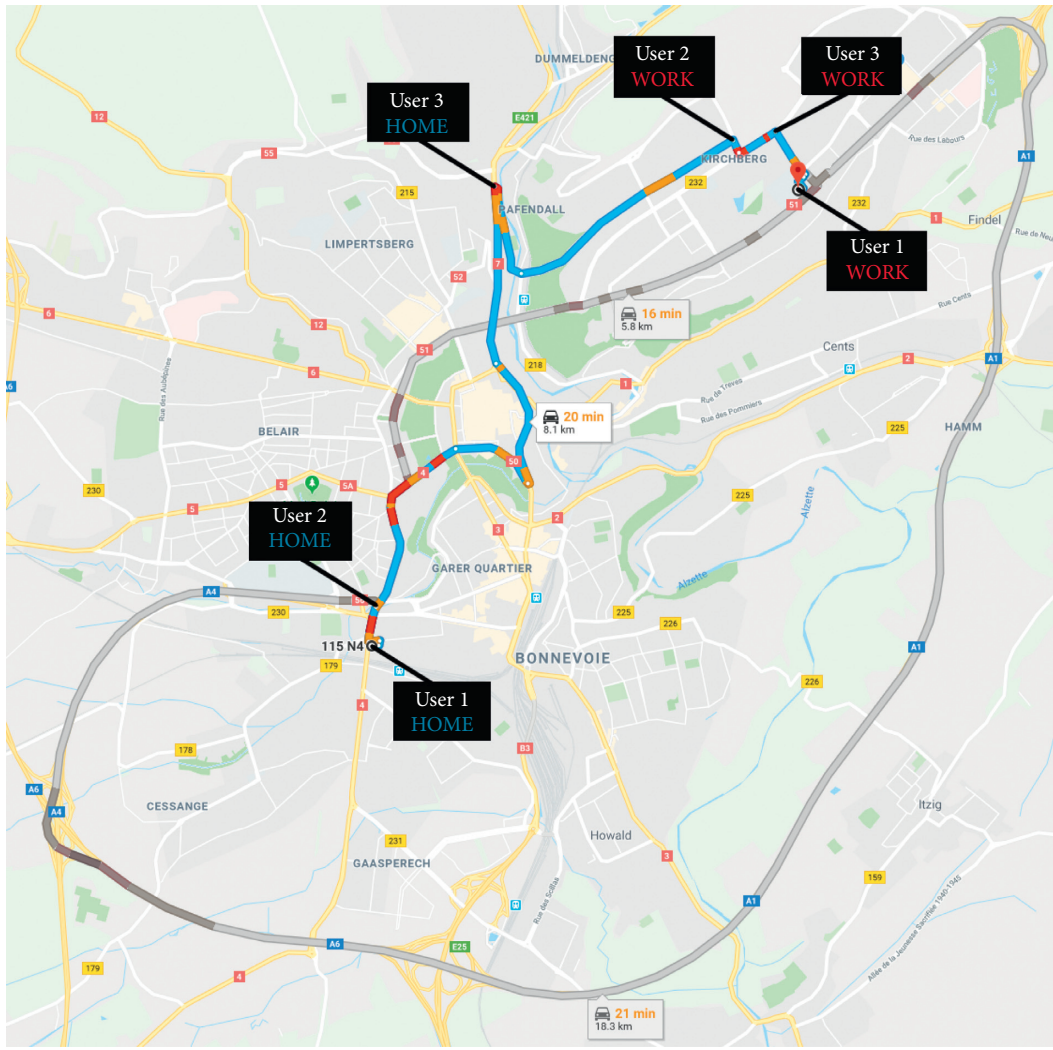

Figure 12: Ride sharing case study with three users.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Sun											1	1						1	1	1				
Mon									1	17	32	23	20	22	17	16	17	15	16	9	3			
Tue										16	24	17	26	25	18	16	15	16	17	7	2			
Wed										15	18	13	17	21	20	13	15	12	15	7	3	1	1	
Thu									1	11	14	12	18	19	17	18	17	20	17	8	2	1	1	
Fri									1	11	23	13	17	20	15	16	15	14	15	7	3	2		
Sat											1	3	2	3	3	1	1	1	1	1	1			

(a)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Sun	1														2	3	3	3	3	2	1	1		
Mon									6	16	17	15	10	15	19	22	17	10	8	8	6	3	1	
Tue									5	17	20	20	11	21	21	19	17	14	11	6	5	1		
Wed									6	15	19	18	16	20	18	15	16	15	12	12	8	6	2	
Thu									3	15	18	17	9	17	19	17	16	13	8	9	5	2		
Fri									5	17	17	17	9	15	15	15	16	14	9	8	3	1	1	
Sat												4	2		1	5	6	3	5	2	2		1	1

(b)

Figure 13: Profile of work location for User 1 and User 2.

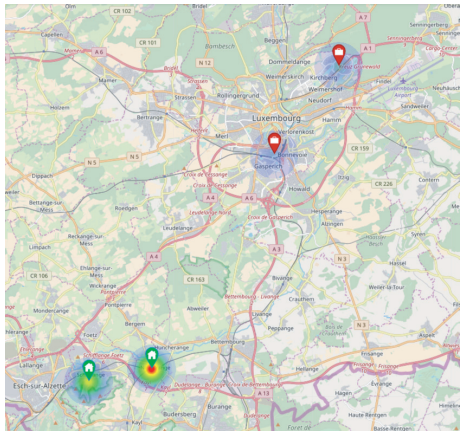

Figure 14: Location type classification: Home and Work classification. Residence and workplace change detection.

At the same time, caching techniques can be useful in the future to store temporally the results of the most asked queries on the most recent trees. Another optimisation would be the implementation of a subscription system which will perform automatically live updates of latest queries and trees.

The presented usage examples show that only using the geolocation data, it is already possible to support some sharing services (e.g., parking sharing and car sharing—as for those services only presence/absence of users/resources in time and space is needed), detect travel habits, and label/classify location and activities. In the future, an implementation of a route planner can offer the possibility to have a complete ride sharing service which can match people and vehicles. On the other hand, the usage of semantic external data of visited locations (e.g., type of facilities from existing maps) can better infer secondary activity types and reduce the identification and classification errors.

7. Conclusion

The contribution of this paper is twofold: on the one hand, we present a novel methodology that provides a dynamic profiling of users' mobility and locations' visit pattern. The proposed profiling method can be used in many applications and even in a simultaneous manner. The usage examples explained and evaluated throughout the current paper (i.e., parking sharing, ride sharing, location type, and activity classification) provide the first directions on how the profiling can be used for a dynamic analysis of sharing mobility users and solutions.

On the other hand, using state-of-the art technologies from data science and computer science, we provide a complete implementation of the proposed methodology which can be tested through an online demonstrative prototype. The demo application demonstrates how it is possible to load the data and extract complex profiles from geolocation data (i.e., location data from Google Maps), with different accuracy levels and spatiotemporal scales, in an order of magnitude of milliseconds. Moreover, for any visited location, a classification is dynamically performed, which demonstrates that different actions and computations can be performed in motion, at scale and in near real time. Different evaluations were performed in order to assess the speed, scalability, and to evaluate the required resources for implementation, which demonstrates that the proposed profiling can be implemented in a distributed way at the

smallest hardware level (e.g., microcomputers or mobile devices).

Data Availability

This study employed Google Visited Places API for the small dataset example, whereas it used the Microsoft GeoLife dataset (https://www.microsoft.com/en-us/download/details.aspx?id=52367) for the large scale example.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by the Luxemburgish FNR (Fonds National de la Recherche) through an AFR grant for the PLAYMOBeL project (9220491) and by the EU Marie-Curie-funded project InCoMMune (618234).

References

[1] Inmarsat research programme report: the future of IoT in enterprise, 2017, https://goo.gl/rvumkN.

[2] IDC, "Worldwide internet of things spending guide report," https://www.idc.com/getdoc.jsp?containerId=prUS44596319.

[3] J. Manyika, M. Chui, B. Brown et al., *Big Data: The Next Frontier for Innovation, Competition, and Productivity*, Technical Report, McKinsey Global Institute, Washington, DC, USA, 2011.

[4] ITF Forum, *ITF Transport Outlook 2017*, OECD Publishing, Paris, France, 2017.

[5] T. Hartmann, F. Fouquet, M. Jimenez, R. Rouvoy, and Y. Le Traon, "Analyzing complex data in motion at scale with temporal graphs," in *Proceedings of the 29th International Conference on Software Engineering & Knowledge Engineering (SEKE'17)*, p. 6, KSI Research, Pittsburgh, PA, USA, July 2017.

[6] T. Hartmann, F. Fouquet, J. Klein et al., "Generating realistic smart grid communication topologies based on real-data," in *Proceedings of the 2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 428–433, IEEE, Venice, Italy, November 2014.

[7] T. Hartmann, *Enabling model-driven live analytics for cyber-physical systems: The case of smart grids*, Ph.D. thesis, University of Luxembourg, Luxembourg, Luxembourg, 2016.

[8] A. Sassi and F. Zambonelli, "Coordination infrastructures for future smart social mobility services," *IEEE Intelligent Systems*, vol. 29, no. 5, pp. 78–82, 2014.

[9] G. Castignani, T. Derrmann, R. Frank, and T. Engel, "Driver behavior profiling using smartphones: a low-cost platform for driver monitoring," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 1, pp. 91–102, 2015.

[10] S.-T. Cheng, G.-J. Horng, and C.-L. Chou, "Using cellular automata to form car society in vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1374–1384, 2011.

[11] J. F. Júnior, E. Carvalho, B. V. Ferreira et al., "Driver behavior profiling: an investigation with different smartphone sensors and machine learning," *PLoS One*, vol. 12, no. 4, p. 959, Article ID e0174, 2017.

[12] E. Ozatay, S. Onori, J. Wollaeger et al., "Cloud-based velocity profile optimization for everyday driving: a dynamic-programming-based solution," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 6, pp. 2491–2505, 2014.

[13] P. Campigotto, C. Rudloff, M. Leodolter, and D. Bauer, "Personalized and situation-aware multimodal route recommendations: the favour algorithm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 92–102, 2017.

[14] S. Canale, A. Di Giorgio, F. Lisi et al., "A future internet oriented user centric extended intelligent transportation system," in *Proceedings of the 2016 24th Mediterranean Conference on Control and Automation (MED)*, pp. 1133–1139, IEEE, Athens, Greece, June 2016.

[15] F. Giannotti, L. Gabrielli, D. Pedreschi, and S. Rinzivillo, "Understanding human mobility with big data," in *Solving Large Scale Learning Tasks. Challenges and Algorithms*-Springer, Berlin, Germany, 2016.

[16] E. Tonnelier, N. Baskiotis, V. Guigue, and P. Gallinari, "Smart card in public transportation: designing a analysis system at the human scale," in *Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1336–1341, IEEE, Rio de Janeiro, Brazil, November 2016.

[17] E. Manley, C. Zhong, and M. Batty, "Spatiotemporal variation in travel regularity through transit user profiling," *Transportation*, pp. 1–30, 2016.

[18] J. Ghosh, M. J. Beal, H. Q. Ngo, and C. Qiao, "On profiling mobility and predicting locations of wireless users," in *Proceedings of the 2nd International Workshop on Multi-Hop Ad Hoc Networks: from Theory to Reality*, pp. 55–62, ACM, Florence, Italy, 2006.

[19] J. Zhang, F.-Y. Wang, K. Wang et al., "Data-driven intelligent transportation systems: a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1624–1639, 2011.

[20] N. Bicocchi and M. Mamei, "Investigating ride sharing opportunities through mobility data analysis," *Pervasive and Mobile Computing*, vol. 14, pp. 83–94, 2014.

[21] N. Agatz, A. Erera, M. Savelsbergh, and X. Wang, "Optimization for dynamic ride-sharing: a review," *European Journal of Operational Research*, vol. 223, no. 2, pp. 295–303, 2012.

[22] B. Toader, F. Sprumont, S. Faye, M. Popescu, and F. Viti, "Usage of smartphone data to derive an indicator for collaborative mobility between individuals," *ISPRS International Journal of Geo-Information*, vol. 6, no. 3, p. 62, 2017.

[23] L. Montini, N. Rieser-Schüssler, A. Horni, and K. W. Axhausen, "Trip purpose identification from GPS tracks," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2405, no. 1, pp. 16–23, 2014.

[24] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[25] D. Heggie and P. Hut, *The Gravitational Million-Body Problem: A Multidisciplinary Approach to Star Cluster Dynamics*, Cambridge University Press, Cambridge, UK, 2003.

[26] M. Winkel, R. Speck, H. Hübner, L. Arnold, R. Krause, and P. Gibbon, "A massively parallel, multi-disciplinary Barnes-Hut tree code for extreme-scale N-body simulations," *Computer Physics Communications*, vol. 183, no. 4, pp. 880–889, 2012.

[27] H. Samet, "The quadtree and related hierarchical data structures," *ACM Computing Surveys*, vol. 16, no. 2, pp. 187–260, 1984.

[28] K. Yamaguchi, T. Kunii, K. Fujimura, and H. Toriya, "Octree-related data structures and algorithms," *IEEE Computer Graphics and Applications*, vol. 4, no. 1, pp. 53–59, 1984.

[29] S. J. Redmond and C. Heneghan, "A method for initialising the *k*-means clustering algorithm using *k*-*d*-trees," *Pattern Recognition Letters*, vol. 28, no. 8, pp. 965–973, 2007.

[30] C. D. Toth, J. O'Rourke, and J. E. Goodman, *Handbook of Discrete and Computational Geometry*, CRC Press, Boca Raton, FL, USA, 2004.

[31] G. Qian, Q. Zhu, Q. Xue, and S. Pramanik, "Dynamic indexing for multidimensional non-ordered discrete data spaces using a data-partitioning approach," *ACM Transactions on Database Systems*, vol. 31, no. 2, pp. 439–484, 2006.

[32] D. Kolbe, Q. Zhu, and S. Pramanik, "Efficient *k*-nearest neighbor searching in nonordered discrete data spaces," *ACM Transactions on Information Systems*, vol. 28, no. 2, p. 7, 2010.

[33] "DataThings: greycat framework," 2017, https://github.com/datathings/greycat.

[34] F. Francois, G. Nain, B. Morin et al., "Kevoree Modeling Framework (KMF): efficient modeling techniques for runtime use," 2014, http://arxiv.org/abs/1405.6817.

[35] B. Toader, A. Moawad, F. Fouquet, T. Hartmann, M. Popescu, and F. Viti, "A new modelling framework over temporal graphs for collaborative mobility recommendation systems," in *Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, IEEE, 2017.

[36] B. Toader, G. Cantelmo, M. Popescu, and F. Viti, "Using passive data collection methods to learn complex mobility patterns: an exploratory analysis," in *Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 993–998, IEEE, Maui, HI, USA, November 2018.

[37] Online profiler tool link, 2018, https://mobilab.lu/profiler-demo/.

[38] Microsoft Research Asia: Geolife GPS Trajectories, 2018, https://www.microsoft.com/en-us/download/details.aspx?id=52367.

[39] Y. Zheng, Y. Chen, X. Xie, and W. Y. Ma, "Geolife 2.0: a location-based social networking service," in *Proceedings of the Tenth International Conference on Mobile Data Management: Systems, Services and Middleware*, pp. 357-358, IEEE, Taipei, Taiwan, 2009.

[40] Guido Cantelmo Bogdan Toader, C. A. F. V.: Inferring Urban Mobility and Habits from User Location History, 2019.

[41] S. Breß, M. Heimel, N. Siegmund, L. Bellatreche, and G. Saake, "GPU-accelerated database systems: survey and open challenges," in *Transactions on Large-Scale Data-and Knowledge-Centered Systems XV*Springer, Berlin, Germany, 2014.

[42] T. Mostak, *An Overview of Mapd (Massively Parallel Database), White Paper*, Massachusetts Institute of Technology, Cambridge, MA, USA, 2013.

[43] M. Litwintschik, "1.1 billion taxi rides with mapd 8 nvidia tesla k80s," 2018, http://tech.marksblogg.com/billion-nyc-taxi-rides-nvidia-tesla-mapd.html.

[44] Transportation Research Board, *Transit Capacity and Quality of Service Manual-TCRP Report 165*, 3rd Edition, Washington DC, USA, 2003.

[45] P. van der Waerden, H. Timmermans, and M. de Bruin-Verhoeven, "Car drivers' characteristics and the maximum walking distance between parking facility and final destination," *Journal of Transport and Land Use*, vol. 10, no. 1, pp. 1–11, 2017.