Big Data for Mobility Tracking Knowledge Extraction in Urban Areas

# SoA – Big Data Analytics

This section first explains knowledge discovery in big data. Several topics and techniques are discussed. For each case, research papers covering both moderately sized datasets as well as big data are discussed. First, we discuss trajectory 'clustering' for a given time period and then consider the problem to discover groups of objects moving together. Next, we focus on sequential pattern mining. Pattern growth algorithms are less computationally expensive the Apriori- based algorithms and better suited to be transformed to versions that allow big data being processed in parallel taking advantage of the MapReduce model. Hot spot analysis discretizes space-time and identifies cells for which a particular attribute takes a statistically significant value. In the mobility domain, the number of moving objects in a space-time cell can be counted and analyzed using the classical Getis-Ord statistic. Ongoing and recent research on finding hot spots in big data is discussed. Future location prediction takes a history of movements (visited location sequences) and tries to predict the sequence of visited locations for a given time horizon. 'Pattern-based prediction' is based on the history of sets of moving objects (as opposed to individual histories). We discuss several ways to represent the mined patterns. Predicting object movements on a network from streaming data is of high importance to mobility science (travel guidance). Location prediction is extended to 'Trajectory prediction': most related research applies to aircraft movements and is not network bound. Current challenges include the adaptation of the methods discussed for use on big data. Beyond that, the recent concept of predictive queries applied to network bound movements deserves attention because of its relevance to travel guidance. Finally, data may not be available for specific regions. Hence, 'transfer learning' comes into play when trying to apply models and model parameters sets in regions different from where they were mined. This is a particularly challenging topic for complex phenomena like urban mobility because they depend on spatially dependent habits.

What follows is a review of methods on complex network analytics. The use of complex networks is briefly introduced and examples from community detection and (information) diffusion are presented. Sample real world problems show the need to study network topology dynamics. Mobility data includes several relations which leads graph theoretical and complex network problem formulations. The example of Individual Mobility Networks (IMN) that can be mined from big data is explained in detail.

This chapter concludes with a discussion on Complex Event Recognition (CER) techniques. CER or event pattern matching applies to continuous data flows origination from several sources. Tools and methods to define and represent complex events are discussed along the processes leading to CER. Main research topics are: event pattern specification, uncertainty handling, the challenges posed by CER in big data streams and techniques for both supervised and unsupervised learning of event

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Grant Agreement No 780754.

patterns. Finally, the importance of CER in big data is illustrated for mobility data (both road based and maritime).

# 1 Knowledge Discovery in Big Data

In this section, several issues related to knowledge discovery in Big Data analytics are discussed. Specifically, related works and state-of-the-art are presented for the core tasks of clustering, sequential pattern mining, future location prediction and trajectory prediction. Additionally, a short literature review is presented for other related tasks and methods, including predictive query processing and transfer learning with mobility data analytics.

## 1.1 Clustering

Location aware devices, such as mobile phones, tablets and automobiles carry numerous networked sensors, which create huge amounts of data that represent some kind of mobility. In addition, the massive participation of individuals on location based social networks will continue to fuel exponential growth in the production of this kind of data. This enormous volume of data has posed new challenges in the world of mobility data management in terms of storing, querying, analyzing and extracting knowledge out of them in an efficient way.

One of these challenges is cluster analysis. The typical approach is to either transform trajectories to vector data, in order for well-known clustering algorithms to be applicable, or to define appropriate trajectory similarity functions, which is the basic building block of every clustering approach. For instance, CenTR-IFCM (Pelekis et al. 2014) builds upon a Fuzzy C-Means variant to perform a kind of time-focused local clustering using a region growing technique under similarity and density constraints. For each time period, the algorithm determines an initial seed region (that corresponds to the sub-trajectory restricted inside the period) and searches for the maximum region that is composed of all sub-trajectories that are similar with respect to a distance threshold *d* and dense with respect to a density threshold $\vartheta$. Subsequently, the growing process begins and the algorithm tries to find the next region to extend among the most similar sub-trajectories. The algorithm continues until no more growing can be applied, appending in each repetition the temporally local centroid. In the same line of research, having defined an effective similarity metric, TOPTICS (Nanni et al. 2006) adapts OPTICS (Ankerst et al. 1999) to enable whole-trajectory clustering (i.e. clustering the entire trajectories), TRACLUS (Lee et al. 2007) exploits on DBSCAN (Ester et al. 1996) to support sub-trajectory clustering, while T-Sampling (Panagiotakis et al. 2012, Pelekis et al. 2010), introduces trajectory segmentation (aiming at temporal-constrained sub-trajectory clustering (Pelekis et al. 2017), by taking into account the neighborhood of a trajectory in the rest of the dataset, yielding a segmentation that is related only on the number of neighboring segments that vote for the line segments of a trajectory as the most representatives. All the above trajectory clustering approaches they are capable of identifying trajectory clusters and their densities but do not tackle the issue of statistical significance in the space-time they take place.

A branch of related works aim to discover several types of collective behavior among moving objects, forming a group of objects that moves together for a certain time period. Among the most related to this work, in (Laube et al. 2005a; Laube et al. 2005b), the authors define various mobility behaviors around the idea of the flock pattern, such as the meeting, convergence and encounter patterns. The discovery of a meeting in a time interval *I* of at least *k* timepoints, consists of at least *m* objects that stay within a stationary disk of radius *r* during *I*. There are two variants of meetings: either the same *m* entities stay together during the entire interval (fixed-meeting), or the entities in the meeting region may change during the interval (varying meeting). On the other hand, the convergence pattern describes trajectories that converge to the same location, coming not necessarily from the same origin.

Inspired by this idea, the notion of a moving cluster was introduced in (Kalnis et al. 2005), which is a sequence of clusters {$c_1$, …, $c_k$}, such that for each timestamp *i*, $c_i$ and $c_{i+1}$ share a sufficient number of common objects. There are several related works that emanated from the above ideas, like the approaches of convoys, swarms, platoons, traveling companion, gathering pattern (Zheng et al. 2015). There are several other methods that try to identify frequent (thus, dense) trajectory patterns. In case where moving objects move under the restrictions of a transportation network, (Sacharidis et al. 2008) proposed an online approach to discover and maintain hot motions paths while (Chen et al. 2011) tackled the problem of discovering the most popular route between two locations based on the historical behaviour of travellers. In case where objects move without constraints, (Cao et al. 2006) proposed a method to discover collocation patterns.

However, all of the aforementioned approaches are centralized and in order to meet with the challenges posed in the Big Data Era, one should think beyond the centralized paradigm and start examining how solutions to such problems could be implemented in a way that would meet with these challenges. A line of research is to adapt well-known solutions to trajectory datasets. In this context, (Deng et al. 2015) introduces a scalable GPU-based trajectory clustering approach which is based on a scalable density-based clustering approach for point data (POPTICS) (Patwary et al. 2013). As to finding flock patterns in large trajectory databases, (Valladares et al. 2013) presented a GPU-based approach for finding extremal sets within a family *F* of *k* finite sets, which has no restrictions on the input. (Fort et al. 2014) studied the problem of finding flock patterns in trajectory databases and presented some parallel algorithms based on GPU for reporting all maximal flocks, the largest flock and the longest flock. Moreover, (Jinno et al. 2012) attempts to discover frequent movement patterns from the trajectories of moving objects. More specifically, they propose a MapReduce-based approach to trajectory pattern mining using a hierarchical grid with quadtree search in order to identify complex patterns involving different levels of granularity.

(Moussalli et al. 2015) and (Moussalli et al. 2013) presented FPGA- and GPU-based solutions for parallel matching of variable-enhanced complex patterns by stream-mode (single pass) filtering. Both implementations are able to process the trajectory data in a single pass when handing pattern queries with no more than one variable or no wildcards with two or more variables but result in false positive matches when two or more variable occur in a pattern query alongside wildcards. The parallel solutions can outperform the current state-of-the-art CPU-based approaches by two or three orders of magnitude at certain circumstances and shows very good scalability with regard to pattern complexity. Similarly, in (Lan et al. 2017) a streaming environment is assumed, however, here, a new concept is proposed, that of evolving group pattern that captures the interesting group patterns over streaming trajectories that cannot be captured by the current group pattern detection techniques.

An approach that defines a new generalized mobility pattern is presented in (Fan et al. 2016). In more detail, the general co-movement pattern (GCMP), is proposed, which models various co-movement patterns in a unified way and can avoid the loose-connection anomaly. Further, the GCMP detector is deployed on a modern MapReduce platform (i.e., Apache Spark) to tackle the scalability issue. On the other hand, in (Ding et al. 2018) an efficient and flexible platform for an open-ended range of trajectory data management and analytics techniques, called UlTraMan, is proposed. Within this system, the GCMP detector is implemented. Moreover, all the necessary preprocessing tasks that are not covered in (Fan et al. 2016) can be supported efficiently in UlTraMan, hence avoiding unnecessary data transfer.

## 1.2 Sequential Pattern Mining

Sequential pattern mining discovers subsequences that appear in a sequence database with frequency no less than a user-specified threshold. A sequence database stores a number of records, where all records are ordered sequences of events, with or without concrete notions of time. Sequential pattern mining is an important data mining problem with broad applications, such as mining customer purchase patterns, identifying outer membrane proteins, automatically detecting erroneous sentences, discovering block correlations in storage systems, identifying copy-paste and related bugs in large-scale software code, API specification mining and API usage mining from open source repositories, and Web log data mining.

This problem was defined as follows: Given a set of sequences, where each sequence consists of a list of elements and each element consists of a set of items, and given a user-specified *min_support* threshold, sequential pattern mining is to find all frequent subsequences, i.e., the subsequences whose occurrence frequency in the set of sequences is no less than *min_support* (Agrawal et al. 2014).

Generally, sequential pattern mining algorithms can be categorized into two major classes: Apriori-based approaches and pattern growth algorithms. The first class of algorithms (i.e., Apriori-based approaches) form the vast majority of algorithms proposed in the literature for sequential pattern mining. They depend mainly on the Apriori property, which states the fact that any super-pattern of an infrequent pattern cannot be frequent and are based on a candidate generation and- test paradigm proposed in association rule mining (Agrawal et al. 1993). These methods have the disadvantage of repeatedly generating an explosive number of candidate sequences and scanning the database to maintain the support count information for these sequences during each iteration of the algorithm, which makes them computationally expensive.

To alleviate these problems, pattern growth approach for efficient sequential pattern mining adopts a divide-and-conquer, pattern growth paradigm as follows, sequence databases are recursively projected into a set of smaller projected databases based on the current sequential pattern(s), and sequential patterns are grown in each projected database by exploring only locally frequent fragments (Han et al. 2000). The frequent pattern growth paradigm removes the need for the candidate generation and prune steps that occur in the Apriori-based algorithms and repeatedly narrows the search space by dividing a sequence database into a set of smaller projected databases, which are mined separately.

In the era of Big Data, where huge amounts of data are available, algorithm and implementation of sequential pattern mining has to re-designed and re-implemented under a distributed computing framework as traditional approaches are not designed to handle massive amounts of data. In recent years research has been done for finding sequential patterns in parallel and distributed areas like Hadoop, Grid, Cloud, etc.

In Parallel Transaction Decomposed Sequential Pattern Mining (PTDS) (Wang et al. 2010) transactions are decomposed to mine the sequential patterns and pattern growth approach is greatly accelerated to improve the efficiency of large-scale data. First, PTDS sorts the sequences and plan the sequences with identical or similar prefix, which is considered as first transaction of each sequence. The input sequence is split in to two parts one is the first transaction and other is the remaining part of transaction in the sequence. PTDS collects sequences with equal prefix, decompose the prefix and applies serial sequential pattern mining method on the set of subsequences; each one contains the remaining transactions of the raw sequence, and finally merges the mining results together. PTDS is implemented using MapReduce framework on Apache Hadoop environment which greatly accelerate pattern growth approach and improves the performance and efficiency of parallel sequential pattern algorithm on large scale data.

Following collaborative pattern mining for distributed information system (CLAP) (Zhu et al. 2011), mining of data is divided into three parts: first, identify locally important patterns on individual database; second, determine major patterns after combining distributed database into single view; third, find patterns which follow special relationship across different data collection. This algorithm makes use of pattern mining for query processing to satisfy user specified query constraints to discover patterns from distributed databases. In existing system pattern pruning is based on single database, so to solve this problem cross-database pruning concept is used for distributed sequential pattern mining. CLAP encourage pattern discovery in distributed approach where each distributed site carries pattern pruning in collaboration with its peers by employing bloom filter-based pattern switching mechanism. A bloom filter is space efficient data structure which contains k hash functions, and binary array of m bits. Patterns like $x_1$,$x_2$,…,$x_n$ can be added into the bloom filter to check whether pattern exist in bloom filter or not by using all k hash functions to map $x_t$ to k positions. CLAP system consists of mainly two parts as one construction of FP-tree and bloom filter for each local site and second CLAP cross database pruning and pattern growth. CLAP only focuses on frequent itemset mining.

Recently, many applications are moved to cloud infrastructure. Sequential pattern mining on cloud (SPAMC) (Chen et al. 2013) adapts is developed for mining sequential patterns on MapReduce model on cloud. SPAMC is a cloud-based version of sequential pattern mining algorithm consisting of two phases: scanning phase, and mining phase. In the scanning phase, high performance is achieved by distributing tasks on multiple computers by using MapReduce programming model to proceed in parallel by distributing sub-tasks to independent machines. Each mapper scans and transforms a partitioned database, and reducers are used to count the frequency of each item and eliminate infrequent items. The bitmap information of frequent items will be stored into a distributed hash table (DHT) that can be accessed in the mining phase. After that, in the mining phase, the sequential pattern mining tasks are processed in parallel by distributed machines. Main task of the mining phase is to construct the complete lexical sequence tree, and then all patterns can be derived. Additionally, to achieve better load balancing, depth first search strategy is used to bring out the steps of sequence and itemset extension with limited sub-tree depth. This strategy effectively improves the situation like mapper may stand and wait for a long time. In such a context, each MapReduce round will complete two levels of lexical sequence sub tree construction. On the other side, reducers efficiently integrate output results from mappers and do the support counting to generate frequent sequential patterns of the current sub-tree.

## 1.3 Hot-spot Analysis

The data wealth, produced by the proliferation of GPS technology, the widespread adoption of smartphones, social networking, as well as the ubiquitous nature of monitoring systems, contributes to the ever-increasing size of what is recently known as Big spatial (or spatio-temporal) data (Eldawy et al. 2016), a specialized category of Big data focusing on mobile objects. Analyzing spatio-temporal data has the potential to discover hidden patterns or result in non-trivial insights, especially when its immense volume is considered. To this end, specialized parallel data processing frameworks (Alarabi et al. 2017a, Alarabi et al. 2017b, Hagedorn et al. 2017, Tang et al. 2016) and algorithms (Doulkeridis et al. 2017, Fang et al. 2016, Whitman et al. 2017, Xian et al. 2016) have been recently developed aiming at spatial and spatio-temporal data management at scale.

In this context, a useful data analysis task is Hot spot analysis, which is the process of identifying statistically significant clusters. However, there is practically no work on hot spot analysis for Big trajectory data. One of the main challenges is focused on discovering hot spots in the maritime domain, as this relates to significant challenging use-case scenarios (Claramunt et al. 2017), such as identifying different types of activities in a region of interest, estimating fishing pressure, environmental fingerprint, etc. Similarly, in the aviation domain the predicted presence of a number of aircrafts above a certain threshold results in regulations in air traffic, while in the urban domain such a presence accompanied with low speed patterns implies traffic congestions. Thus, the effective discovery of such diverse types of hot spots is of critical importance for our ability to comprehend the various domains of mobility.

Hot Spot discovery and analysis is usually based on spatio-temporal partitioning of the 3D data space in cells. The identification of cells that constitute hot spots includes having high concentration of mobile objects and in statistically significant densities. One of these methods is the Getis-Ord statistic (Ord et al. 1995), a popular metric for hot spot analysis, which produces z-scores and p-values. A cell is considered as a hot spot if it is associated with high z-score and low p-value. Unfortunately, the

Getis-Ord statistic is typically applicable in the case of 2-D spatial data, and even though it can be extended to the 3-D case, it has been designed for point data.

The problem of Trajectory hot spot analysis can be formulated by taking into account the contribution of a moving object's trajectory to a cell's density, which is proportional to the time spent by the moving object in the cell. To this end, the Getid-Ord statistic can be adapted (Nikitopoulos et al. 2018) to capture this approach for the case of trajectory data and the algorithm can be designed for parallel and scalable processing for computing hot spots in terms of spatio-temporal cells produced by grid-based partitioning of the data space under consideration.

Similar approaches can be adapted and applied in the urban environment, especially designed for Big mobility data. Hot spot analysis will be a very important aspect of detecting points of high density, bottlenecks and points of interest, which can be combined with efficient identification of mobility patterns.

## 1.4 Future Location Prediction

The problem of Future Location Prediction (FLP) can be informally described as follows: Given the recent spatio-temporal history of N previous data points of a moving object, i.e., consisting of its time-stamped locations recorded at N past time instances, and an integer look-ahead value L, predict the anticipated future locations of the object for the next L time instances. The main factors for any FLP algorithm are size of the history (N), the extent of the prediction window (L) and the way these two are combined together in a predictive model.

The FLP problem finds two broad categories of application scenarios. The first scenario involves cases where the moving entities are traced in real-time to produce analytics and compute short-term predictions, which are time-critical and need immediate response. Short-term FLP can be extremely important in domains where safety, adaptiveness and responsiveness out outmost importance and a decision-making process. The second scenario involves cases where long-term FLP is important to identify cases which exceed regular mobility patterns, detect anomalies, and determine a position or a sequence of positions of special interest at a given time interval in the future. In this case, although response time may not be a critical factor per se, it is still crucial in order to identify correlations between historical mobility patterns and patterns that are expected to appear, e.g. approach to a restricted area.

There are two main directions when dealing with the FLP problem: (a) *vector-based* prediction or the the spatial database management approach and (b) *pattern-based* prediction or the data mining & Machine Learning approach. Each has its own advantages and drawbacks and, most importantly, it is based on different assumptions regarding the data and their organization used as the input.

The vector-based approaches, inspired by the spatial database management domain, aim to model current locations (and perhaps a short history) of objects as *motion functions*, in order to be able to predict future locations by some kind of extrapolation. In practice, they take into consideration space and time and predict future locations of moving objects within a given time interval using a mathematical or probabilistic model, which aims to simulate the anticipated movement. First- or second-degree physics models of movement are commonly used, employing extrapolation with velocity or velocity and acceleration components, respectively, to estimate the evolution of movement, provided that these can be assumed to be constant in a short-term look-ahead time window.

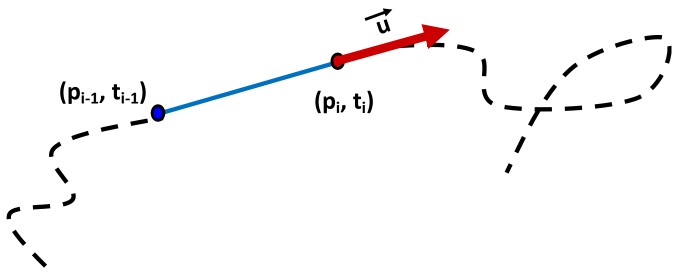

Figure 1: The future position of a moving object as the result of a linear motion function.

The constant-speed assumption is also very useful in the development of proper transformations of the input space that enable time-invariant representations, e.g. via the Hough-X transform (Jagadish et al. 1990). Essentially, the evolving position of a moving object remains a stationary point in dual space as soon as it does not change its velocity vector, thus it can be efficiently indexed in a spatial access method. This is the main concept behind the family of *predictive query processing* techniques for FLP that introduces various state-of-the-art methods including PMR Quadtree (Tayeb et al. 1998; Samet 1990), TPR*-tree (Tao et al. 2003), Bx-tree (Jensen et al. 2004) and STP-tree (Tao et al. 2004).

The pattern-based approaches, inspired by the spatial data mining domain, identify and exploit motion patterns by analyzing historic data of moving objects, i.e., classification models, repetitive patterns, clusters of "similar" movements, etc, based upon a history of movements. An important difference with respect to the vector-based approaches is that in this case the models are built upon the history of movements, not only of the object of interest, but also of the other objects moving in the same area; therefore, they are able to build better models and use them for addressing the FLP task in a more generic and data-driven way. Particularly, extensive surveys on vehicle motion prediction models have been presented by (Lef`evre et. al. 2014) and (Zhan et. al. 2018)

Techniques based on Hidden Markov Models (HMM), Neural Networks (NN) and other data-driven approaches have been extensively used to address the FLP problem. (Ishikawa et al. 2004) introduce an algorithm that extracts mobility statistics from indexed spatio-temporal datasets for interactive analysis of huge collections of moving object trajectories. In the maritime domain, (Zorbas et al. 2015) introduce a machine learning model using a NN that exploits geospatial time-series surveillance data generated by sea-vessels, in order to predict future trajectories with real-time constraints with a look-ahead time window of 5 minutes. In a different domain, that of aviation, (Hamed et al. 2013) propose a method for predicting the altitude change of an aircraft within a predefined look-ahead time window of 10 minutes. Also, (Pecher et. al. 2016) employed various methods, including NNs, to predict taxi-drivers' trajectories, by dividing the road network into a two dimensional grid, while in (Choi et.al. 2019) an RNN-based method for urban vehicle trajectory prediction is proposed, where the urban traffic network is partitioned into a grid area composed of cells. In the literature there are, also, works that correspond to short-term prediction. Particularly, in (Park et.al. 2018), (Kim et.al. 2017), (Ma et.al. 2019), (Hou et.al. 2019), (Altché et.al. 2017) and (Wu et.al. 2017), the employed maximum prediction horizons are 2sec, 2sec, 3sec, 5sec 10 sec and 90sec, respectively.

Due to the advancement in the field of Deep Neural Networks (DNNs) there are works that employ DNNs to improve FLP. More specifically, (Wang et.al. 2019) proposed an LSTM model for trajectory prediction, which can first make a single-step prediction after one-hour of observation. Also, in (Fan et.al. 2018), the DLNLP was proposed, which predicts vehicle's next location given its trajectories and related contextual information. Furthermore, (Wang et.al. 2020) proposed a hybrid Encoder-Decoder DNN model in order to predict objects' future locations moving in free space.

There are also pattern-based techniques that are based on association rules or frequent mobility patterns. These include methods like the Mobility Patterns (Yavas et al. 2005), TrajPattern algorithm for pattern groups (Yang et al. 2006), Spatio-Temporal Association Rules (STARs) (Verhein et al. 2006), WhereNext for trajectory patterns (T-patterns) (Monreale et al.2009), as well as state-of-the-art methods in this area like NextLocation (Gomes et al. 2013) and MyWay (Trasarti et al. 2017).

There is also a relatively new category of *semantic-aware* approaches that involves semantics or *enrichments* extracted by the surrounding environment, e.g. stops, hot-spots, etc. Then, patterns are built upon this knowledge of enriched spatio-temporal data and then used for predicting the next location(s). As an example, (Ying et al. 2011) are the first who exploit both geographic and semantic features of trajectories. Their approach is based on a novel cluster-based prediction method, which estimates a mobile user's future location by exploiting frequent patterns in similar users' behavioral activities.

In other works, a set of motion patterns is exploited for optimally designed `codebook' of motion functions that is used to fit the recent history of an object's movement and then extrapolate upon them within a specific look-ahead time window. Such an approach is the LeZi-Update adaptive on-line algorithm (Bhattacharya et al. 1999), incorporating dictionary updates as in the Lempel-Ziv algorithm family (Liv et al. 1978) for lossless compression.

## 1.5 Trajectory Prediction

Typically, the trajectory of a moving object is defined as a sequence of spatio-temporal data points of length N, consisting of its time-stamped locations recorded at N past time instances, chronologically ordered. In principle, the spatial dimension D of the data points is arbitrary, but the most common cases are moving objects on a surface (D=2, e.g. maritime or land) and in a volume (D=3, e.g. aviation). Additionally, in order to simulate continuous movements, we usually make an assumption of interpolation in-between two consecutive data points; the most popular is linear interpolation, although other functions may be used as well (B-splines, etc.).

Similarly to FLP, the Trajectory Prediction (TP) task can be informally described as follows: Given the recent history of S previous trajectories of one or more moving objects, i.e., each consisting of its time-stamped data points recorded in the past, predict the anticipated future trajectory of the same or “similar” objects, based on some common reference initialization (e.g. starting point, time frame, region of interest, etc). The main factors for any TP algorithm are size of the history (N) and how it is exploited by a predictive model.

In principle, the TP problem can be approached as a generalization of the FLP problem (Hamed et al. 2013; Theodoridis et al. 2008; Zheng et al. 2015), which is the task of predicting the next spatio-temporal position(s) of a moving object based on its previous track, most commonly in the short-term context (up to few minutes). On the other hand, the TP problem is to predict the anticipated track of the moving object given a set of constraints and/or historic data. A FLP method could be transformed to address the TP problem, given a specific granularity upon which the same method is applied iteratively. However, in that case the prediction errors are accumulated with each step (e.g. via multi-step linear regression), thus making the next predicted points increasingly error-prone. In contrast, ‘pure’ TP methods aim to forecast the trajectory itself from the start, thus making each predicted point equally error-prone.

Recently, there has been plenty of work on location and trajectory prediction in the mobility (Pelekis et al. 2014). The proposed approaches include systems-engineering view (Sip et al. 2003) balancing TP accuracy and processing speed, stochastic approaches other than HMM, splitting the flight phases (Gong et al. 2004), collaborative TP via Conflict Avoidance & Resolution (CA&R) (Chen et al. 2011; Matsuno et al. 2015; Vouros et al. 2018), anomaly detection (Di Ciccio et al. 2016), etc. Not surprisingly, the vast majority of methods are domain-specific (with most of them in the aviation domain) and this is in order to take advantage of the properties of the moving objects under consideration. The issue of exploiting additional data or enrichments in TP have created the notion of semantic-aware TP or

Semantic Trajectory Prediction (STP), which enables better estimations for departure and arrival times and, hence, more robust scheduling and logistics, especially in the congestion points.

During the last few years, there is a mainstream trend of using stochastic models for retrieval, with HMM approach being the most popular (Rabiner et al. 1989), as it has proved its efficiency in modeling a wide range of sequences of observations. In general terms, a system is assumed to have the Markovian property if its future situations depend only upon its current state. Exhibiting high accuracy in modeling sequential data, the HMM approach has given rise to a wide range of applications, such as speech recognition, music retrieval, human activity recognition, consumer pattern recognition, etc. Consequently, it is a clear opportunity to apply them in the domain of mobility data analysis. In the context of trajectory prediction, the flight route and all the associated information (weather, semantic data, etc), are usually encoded into discrete values that constitute the HMM states; then, the trajectory itself is treated as an evolution of transitions between these states, using the raw trajectory data of a large set of flights for training, plus spatio-temporal constraints (locality) to reduce the dimensionality of the problem.

(Ayhan et al. 2016) introduce a novel stochastic approach to aircraft trajectory prediction problem, which exploits aircraft trajectories modeled in space and time by using a set of spatio-temporal data cubes. They represent airspace in 4-D joint data cubes consisting of aircraft's motion parameters (i.e., latitude, longitude, altitude, and time) enriched by weather conditions. They use Viterbi algorithm (Viterbi 1967) to compute the most likely sequence of states derived by a HMM, which has been trained over historical surveillance and weather conditions data. The algorithm computes the maximal probability of the optimal state sequence, which is best aligned with the observation sequence of the aircraft trajectory. In their experimental study, they demonstrate that their methodology efficiently predicts aircraft trajectories by comparing the prediction results with the ground truth aligned trajectories, with the error being reasonably low for one-hour flights.

Two of the most widely explored approaches in TP is regression and clustering, separately or in combination, some also exploring the use of weather or other data. These include methods based on Generalized Linear Model (GLM) (de Leege et al. 2013), multi-stage clustering (Yang et al. 2015), typical regression-based short/mid-term TP (Krumm et al. 2003; Tastambekov et al. 2014), combination of clustering and Kalman filters (Song et al. 2012), etc. Neural networks have also been used successfully for the climb/vertical TP (Le Fablec et al. 1999) or in relation to the air traffic flows (Cheng et al. 2003) for Estimated Time of Arrival (ETA). Recently, (Rathore et.al. 2019) proposed a scalable clustering and Markov chain based hybrid framework, called Traj-clusiVAT-based TP, for both short-term and long-term trajectory prediction, which can handle a large number of overlapping trajectories in a dense road network.

Regarding en route climb TP, one of the major aspects of ATM decision support tools, (Coppenbarger et al. 1999) discusses the exploitation of real-time aircraft data, such as aircraft state, aircraft performance, pilot intent and atmospheric data for improving ground-based TP. The problem of climb TP is also discussed in (Thipphavong et al. 2013) as it constitutes a very important challenge in ATM. In another work by (Ayhan et al. 2016), the authors investigate the applicability of the HMM for TP on only one phase of a flight, specifically the climb after takeoff. A stochastic approach such as the HMM can address the TP problem by taking environmental uncertainties into account and training a model using historical trajectory data along with weather observations. There are also numerical approaches to the problem of climb-phase TP, e.g. (Hadjaz et al. 2012).

## 1.6 Other Challenges

As described in the previous sections, both FLP and TP problems have been studied extensively in the last few years. Some of the proposed approaches are compatible with Big data applications and some are not. Mobility data are in the core of various Big data modalities and approaches in addressing analytics and predictive modelling tasks in a wide range of contexts. Thus, it is imperative that such approaches are scalable and parallelizable, in order to handle data of very large volume, velocity, veracity and variety.

A more recent approach for addressing predictive modelling tasks via mobility patterns comes from the area of Predictive Queries (PQ) (Hendawi et al. 2012b, Zhang et al. 2012), which is one of the most exciting research topics in spatio-temporal data management. In many location-based services, including traffic management, ride sharing, targeted advertising, etc., there is a specific need to detect and track mobile entities within specific areas and within specific time frames. In Range Queries (RQ), the task is focused on identifying POIs and mobility patterns related to the current locations of moving objects. Instead, Predictive Range Queries (PRQ) address the same task but for future time frames. This is a typical use case in aviation, when one or more airplanes need to be checked in some spatial context in the future, e.g. for proximity (collision avoidance), scheduling (takeoff/landing), airspace sectorization (avoid overload and/or delays), etc.

In the context of PRQ and most commonly in the RQ task, various approaches can be used for checking arrivals/departures of airplanes to/from specific regions of interest, including optimized k-nearest-neighbour (k-nn) variants that employ spatio-temporal index trees. Similarly, a reverse k-nn query can be used to detect moving objects that are expected to have the query region as their nearest neighbour, e.g. for assigning moving objects to their "nearest" tracking node. Indexing can be implemented by very efficient data management structures, such as R-trees (time-parameterized, a.k.a. TPR/TPR*-trees), variants of B-trees, kd-trees, Quad-trees, etc (Hendawi et al. 2012b, Hendawi et al. 2015b). The predictive model itself can be linear or non-linear and it is most commonly based on historical data in the same spatio-temporal context, in the short- or the long-term w.r.t. time frame. The uncertainty of the prediction is addressed by either model-based approaches, which determine a representative model for the underlying mobility pattern, or pure data-driven approaches, which "learn" and index movements from historic data (Zhang et al. 2009).

Another important aspect especially in FLP is the ability to employ such models in streaming data, i.e., using "live" sources of mobility data as they become available. This task can also be addressed by PRQ approaches, more specifically the continuous PRQ algorithms. The difference between a "snapshot" predictive query and a continuous one is that the second can be continuously re-evaluated with minimal overhead and optimal efficiency. As an example, the Panda system (Hendawi et al. 2012a, Hendawi et al. 2015b), designed to provide efficient support for predictive spatio-temporal queries, offers the necessary infrastructure to support a wide variety of predictive queries that include predictive spatio-temporal range, aggregate (number of objects), and k-nn queries, as well as continuous queries. The main idea of Panda is to monitor those space areas that are highly accessed using predictive queries. For such areas, Panda pre-computes the prediction of objects being in these areas beforehand.

Similar approaches exist in various domains, such as the iRoad (Hendawi et al. 2013), which is employed for tracking vehicles in urban areas. More specifically, the system supports a variety of common PQs including point query, range query, k-nn query, aggregate query, etc. The iRoad is based on a novel tree structure named reachability tree, employed to determine the reachable nodes for a moving object within a specified future time T. By employing spatial-aware pruning techniques, iRoad is able to scale up to handle real road networks with millions of nodes and it can process heavy workloads on large numbers of moving objects. Since flight routes of civilian and cargo flights are also conditioned by various constraints, e.g. by submitted flight plans (aviation domain) or common ship routes (maritime domain), such road-based approaches can be adapted for a wide variety of problems (Jeung et al. 2010, Hendawi et al. 2015b).

In the context of scalability and the Big data aspect, there are very recent and promising approaches such as the UITraMan (Ding et al. 2018), which addresses the scalability, the efficiency, the persistence and the extensibility of such frameworks. More specifically, it extends Apache Spark w.r.t. data storage and computing by employing a key-value store and enhances the MapReduce paradigm to allow flexible optimizations based on random data access. Another approach for PQs in Big data is presented by Panda* (Hendawi et al. 2017), which is a scalable and generic enhancement of Panda (Hendawi et al. 2012a), applied to traffic management. More specifically, Panda* is a generic framework for supporting spatial PQs over moving objects, introducing prediction function when there is lack of historic data, isolation of the prediction calculation from the query processing and control over the trade-off between low latency responses and use of computational resources. For both UITraMan and Panda*, experimental results on large-scale real and synthetic data sets in other domains, which include comparisons with the state-of-the-art methods in this area, show promising results and hints of successful application to the aviation domain too.

It should be noted that there are also other types of PQs, more advanced than the ones presented above, such as the predictive pattern queries (PPQ), which check conditions muc more complex than simple presence or not of a moving object within a specific spatio-temporal frame. Such advanced PPQs can be considered as a link between data management and data analytics, which can be very valuable in the context of the aviation domain.

## 1.7 Geographical Transfer Learning and Mobility Data

Most machine learning and data mining methods work on the expectation that the context where the models and patterns were extracted is similar (i.e. has the same dependencies between variables) to the one where we want to deploy them. However, in several problems that is not the case, either because the samples in the two contexts are not homogeneous (e.g. the distributions of some variables are different) or because the data available in the second context is poorer. In such cases, transferring the knowledge from one context to the other can be challenging but also extremely useful, since would avoid the set-up of a completely new analysis process, including expensive data collection and labelling. This problem is called transfer learning, or knowledge transfer, and gained a large attention from the research community the latest years.

Transfer learning has been deeply studied in the general context of machine learning (Pan and Yang 2010, Tsung et al. 2017), yet transferring models across different geographical contexts has been only sparsely explored, especially in relation to human mobility.

Some basic, geography-related example of knowledge transfer is given in (Wei, Zhang and Yang 2010), where mobility-based models for estimating air quality are transferred from a city where there exist sufficient multimodal mobility data and labels to cities with insufficient data. Similarly, (Liu at al. 2017) aim to identify the combinations of landscape metrics (inferred from satellite images) that correspond to the presence of urban villages. The technical issue, here, is that the relations between the two phenomena vary in space due to the presence of different geographical factors, and therefore the models must be adapted to the different contexts;

A common problem is the geospatial transfer of models describing physical or social phenomena, such as house prices and seismic movements, across regions having different variable distributions or correlations, as studied in (Bussas et al. 2017). Similarly, the work by (Jun 2010) deals with the problem of classifying spatial data (specifically hyperspectral data) through spatially adaptive model parameters for Gaussian process models and presents various solutions to infer the parameters locally to each area. Finally, (Iddianozie and McArdle 2019) tackle the problem of learning to classify street types on a city and apply it to a different location. The provided methodology is based on statistical multi-measures that allow to ascertain the spatial similarities of cities, making the prediction (based on random forest models) more robust and transferrable.

The work in (Wang et al. 2017) considers a slightly different problem: how to transfer models from one set of mobility modes (taxis and buses) to a different one (ridesourcing cars, like Uber and similar services), although in the same geographical area. The main problem, in this case, is to understand how to map (mobility) features across the different modalities.

Finally, various works try to transfer models (i.e. model parameters) for various kinds of recognition tasks from one place to another one that might show slightly different conditions. An example on human activity recognition across different buildings is provided in (Kasteren et al. 2010).

Despite the various examples discussed above, very little has been done so far on the transfer of complex models, such as trajectory patterns, mobility profiles or mobility forecasting models. This is a challenging and very promising direction of research that Track&Know will pursue.

# 2 Complex Network Analysis in Big Data

Complex Networks (Newman, 2003) are popular mathematical tools commonly used to describe and analyze interaction phenomena that occur in the real world. Social ties formation, economic transactions, face to face communications, the unfolding of human mobility are examples of events usually described by semantic rich Big Data often investigated using instruments borrowed from Graph Theory. Thanks to such heterogeneous analytical context, during the last decades several problems have been modeled and approached leveraging the framework offered by Complex Networks. Among the network related tasks addressed to extract meaningful information from real data, Community Discovery, Link Prediction, Spreading and Epidemic modeling are certainly the most famous ones.

The concept of a “community” in a (web, social, technological, biological or informational) network is intuitively understood as a set of entities that have some latent factors in common with each other, and thus play a specific role in the overall function of the complex system (Fortunato, 2010). Traditional approaches to discover such mesoscale topologies assume that latent factors drive network connectivity; thus, finding sets of nodes with a high edge density among each other and a low edge density with the rest of the network effectively detects the functional modules of the network. Community discovery is then a network variant of data clustering, where proximity is replaced with edge connectivity. Communities are often used as a pre-processing step to enable complex analysis on top of network structures. For instance, they are often used to relate topological structures with external information – as in (Rossetti et al. 2016) where densely connected sets of Skype/Google+/Last.fm users were used to providing a characterization of the overall service usage.

Since network topologies are expected to change as time goes by, forecast the appearance and vanishing of the entities (nodes as well as edges) composing them represents a crucial task to address. In this scenario, Link Prediction (Liben-Nowell et al. 2007, Lu et al. (2011)) focuses on the analysis of network historical data to provide insights on the future evolution of the network topology. Several Link prediction methodologies where proposed with the aim of identifying future friendships in social graphs (Jalili et al. 2017), collaborations in scientific/professional networks, interactions in protein-protein networks as well as future co-locations of individuals (Wang et al. 2011).

Generally, a dynamic process can describe not only graph topology perturbation but also the diffusion of some kind of content upon such complex structure. Commonly, when we use the word “spreading” we think to contagious diseases caused by biological pathogens, like influenza, measles or sexually transmitted diseases. However, a plethora of phenomena can be linked to the concept of epidemic: the spread of computer viruses (Szor 2004), the spread of mobile phone virus (Wang et al. 2013), the diffusion of knowledge, innovations, products in an online social network, etc. Several network models were designed to approach the complex task of modelling and forecasting diffusive phenomena, often leveraging data-driven analysis of real-world phenomena. As an example, in (Bakshy et al. 2012) the authors examine the role of information diffusion in the sharing habits of 235 million Facebook users. They study the role of weak and strong ties in information diffusion showing that the propagation of novel information is mostly due to the abundance of weak ties. The authors of (Leskovec et al. 2007) studied a corpora of weblogs (composed by 45,000 blogs and 2.2 million blog-posting) for two months.

In their paper, they show that blog posts do not have bursty behavior and that post popularity drops as a function of time. In (Cha et al. 2009) a Flickr dataset of 33 million photos marked as "favorite" from 2.5 million users of the service is analyzed. The authors observed that most of the markings do not spread widely throughout the network: even the more popular photos have limited popularity outside the immediate neighborhood of the original uploader.

Indeed, both network topological dynamics – as the ones studied by Link Prediction approaches – and dynamics that occur on top of network structures are often interdependent. Such dualism has lead in recent years to the rising of the dynamic network analysis field (Holme et al. 2012). In a dynamic scenario, all the network problems defined and studied on top of static data are extended to allow a fine-grained time-aware analysis. Community Discovery, as an example, is revised to tracking network substructures as time goes by (Rossetti et al. 2018): such life-cycle analysis allows not only to profile group of entities involved in a networked structure but also to understand how their profile changes as the phenomenon the network describe evolves.

The massive amount of mobility data available from different sources requires intensive analysis in order to extract useful models and patterns. The challenge is not only the computational aspect, but also the representation of this data in a meaningful and semantically rich way allowing classical and new methodologies algorithms to be applied. In particular the network (or, equivalently, graph) representation of this data gives a flexible way to define relations (edges) among basic concepts (nodes). In literature we can consider three different approaches considering what the nodes represent.

The first class of works has the **users as nodes** (Hossmann et al. 2011; Wang et al. 2011), in both the cases the edges are weighted links representing the spatio-temporal co-location of them, i.e. the possible contacts, and the authors uses this graph to discover communities of users, connectivity measures and to predict future social ties.

The second approach has **user's locations as nodes** and the edges represent the trips between them (Gonzalez et al. 2002; Rinzivillo et al. 2014) – the link weight being proportional to the frequency of the trip. In this case the main analytical objectives are finding spatio-temporal regularities and patterns in user mobility or classify the purpose of the user's visit.

The third approach, the most used one, is to consider **global locations as nodes**. In this case, current analysis methods in the literature follow various different ways of defining edges between such nodes:

- A link if there is an Infrastructure (e.g. streets, railways, etc.) connecting the locations;
- A link if there is a collective service (i.e. taxi, bus, etc) connecting the locations;
- Weighted links representing the number of users moving between the two locations.

These different ways of building the graph are used for a large variety of analytical objectives, which include: trips simulation (Tian et al. 2002), evaluating the resilience of the road/transport network (Woolley-Meza et al. 2011) and simulating diseases spreading (Brockmann et al. 2009) for the first group; studying network and traffic evolution (Xia et al. 2018) and detecting traffic anomalies (Chawla et al. 2012) for the second group; inferring communities of locations (Brilhante et al. 2012), optimizing traffic (Zhang et al. 2018), comparing the structure of cities (Saberi et al. 2017), inferring new local borders within a country (Thiemann et al. 2010) and nowcasting air quality (Zheng et al. 2013) for the case of weighted links.

From a global location perspective, a very popular task involving mobility and networks is the modelling and prediction of traffic flows between areas. The problem of estimating human flows between locations in a geographical space has been first addressed by (Wilson 1971) through a family of spatial interaction models and subsequently extended by (Fotheringham and O'Kelly 1989). Spatial interaction models, extensively used to estimate human mobility flows and trip demand between locations as a function of the location features, have become an acknowledged method for modelling geographical mobility in transportation planning (Erlander and Stewart 1990, de Dios Ortuzar and

Willumsen 2011), commuting (McArthur et al 2011), and spatial economics (Patuelli et al 2007). The spatial interaction models are usually calibrated via an Ordinary Least Squares (OLS) regression, which assumes normally distributed data. However, OD flows are usually not distributed normally, are count data, and contain a large number of zero flows. This makes the setting incompatible with OLS estimation and requires either a Poisson model or, in the presence of over-dispersion, a Negative Binomial Regression (NB) model (Zhang et al 2019).

More recently, machine learning, particularly a Random Forest approach, has shown promising results in reconstructing inter-city OD flow matrices (Spadon et al 2019). However, its performance on intra-urban flow data remains to be tested. The problem of estimating OD flows has also been addressed with neural network methods (Mussone and Matteucci 2013). As flows are most naturally modelled by graphs, most work has focused on the use of graph neural networks for flow estimation. An early neural network model for graph structured data has been suggested in (Scarselli et al 2009). Later work has specifically focused on generalising Convolutional Neural Networks from the domain of regular grids to the domain of irregular graphs (Defferrard, Bresson and Vandergheynst 2016). One of the most commonly used graph neural network models is the Graph Convolutional Neural Network (GCN) proposed in (Kipf and Welling 2017). Graph neural networks have previously been applied to urban planning tasks. In (Chai, Wang and Yang 2018), they have been used to predict the flow of bikes within a bike sharing system. Here, flows are modelled as node-level features, which requires a different neural network model and does not allow to predict flows between specific pairs of nodes. Although (Wang et al 2019) uses graph neural networks to predict flows between parts of a city, their model operates on spatio-temporal data and focuses on the temporal aspect of the data. Beyond flow prediction, in (Zhu and Liu 2018), a graph neural network model has been proposed for building site selection. A broader overview of machine learning methods applied to the task of urban flow prediction is given in (Xie et al 2019).

A sample analysis framework of particular interest for the study of mobility at the level of single individuals is the work in (Rinzivillo et al. 2014), where the Individual Mobility Networks (IMNs) are defined. IMNs describe the individual mobility of an individual through a graph representation of her locations and movements, grasping the relevant properties of individual mobility and removing unnecessary details. Formally, the Individual Mobility Network of an individual u is a directed graph Gu = (V, E), where V is the set of nodes and E is the set of edges. On nodes and edges the following functions are defined:

- ω : E → N returns the weight of an edge (i.e. the number of travels performed by u on that edge);
- τ : V → N returns the time spent by the individual in a given location;
- pe : E × T → [0, 1] estimates the probability pe (e, t) of observing an individual u moving on edge e at time t;
- pl : V × T → [0, 1] estimates the probability pl(v, t) of observing an individual u at location v at time t.

Nodes represent locations and edges represent movements between locations. We attach to both nodes and edges statistical information by means of structural annotations: edges provide information about the frequency of movements through the ω function; nodes provide an estimation of the time spent in each location through the τ function. To clarify the concept of IMN, let us consider the network in Figure X. It describes the IMN extracted from the mobility of an individual who visited 19 distinct locations. Location “a” has been visited a total of 18 time units (days in the example), i.e. τ(a) = 18. The edge e = (a, b) has weight ω(e) = ω(a, b) = 20, indicating that the individual moved twenty times from location a to location b.

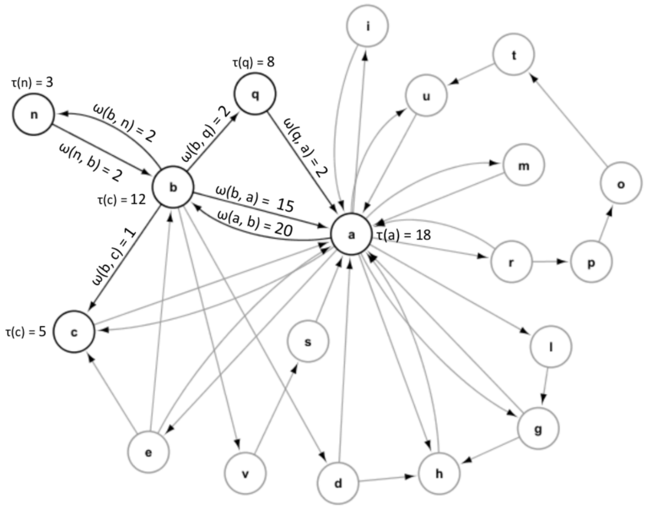

Figure 2: The IMN extracted from the mobility of an individual. Edges represent the existence of a trip between two locations. Function ω(e) is the number of trips performed along edge e, τ(x) the total time spent in location "x".

In (Rinzivillo et al. 2014) the analytical objective is to build a classifier for the purpose of the visits of a user. This work demonstrated that abstracting the mobility data of the user from the geography provided a suitable representation layer for performing a classical data mining task to discover semantically rich models and patterns. Also, the work exploited the explicit relations encoded in each network, which allow, for instance, to propagate information from one node to the others (in the specific application, the activities performed in a location have an impact on the activities performed in adjacent [in terms of network topology] locations).

# 3 Complex Event Recognition in Big Data

Complex Event Recognition (CER) — event pattern matching — applications detect various events of interest in continuous, high-velocity data flows originating from a multitude of distributed sources, by timely providing responses to complex queries. CER plays an important role in Track & Know project, aiming to allow for real-time intelligence in the big data analytics toolbox that will be developed in the project. We review the state-of-the-art in CER with respect to key objectives related to research and development in Track & Know. We begin with an overview of the main CER languages and formalisms, including a brief description of representative systems for each such formalism and their ability to handle the variety of big data. We next discuss uncertainty handling in CER, crucial for addressing the lack of veracity of such streams and continue with important issues related to scaling CER systems to the volume and velocity of big data. We also present some existing techniques for machine learning event patterns from data and conclude with a discussion of CER approaches for mobility applications, which are highly relevant to Track & Know project.

## 3.1 Event Pattern Specification languages

In principle, an event is any time-stamped piece of information. CER systems accept as input simple events, i.e. non-decomposable event occurrences, and they recognize complex events, i.e. event patterns of special significance, which are defined in terms of simple events and potentially other complex events and contextual knowledge. A variety of languages and formal methods for CER have been proposed in the literature - see (Artikis et al. 2012; Cugola and Margara 2012; Artikis et al. 2017) for overviews. Existing approaches have been developed within the database, distributed systems, and artificial intelligence communities. They all have a common goal - express event patterns and match such patterns in the input data - but due to the diversity of their origins, they differ in their architectures, data models, pattern languages, and processing mechanisms.

One family of CER systems relies on automata-based approaches. Event patterns in such systems are compiled into some form of automaton, such as non-deterministic finite automata (NFA) (Mozafari et al. 2013) or finite state machines (FSM) (Schultz-Møller, Migliavacca, and Pietzuch 2009). Such representations are used to provide the semantics of the event pattern language, as well as an execution framework for the event recognition task. Examples of such systems include Cayuga (Brenna et al. 2007), SASE (Mozafari et al. 2013), SASE+ (Zhang, Diao, and Immerman 2014) and TESLA (Cugola and Margara 2010). Some of these approaches use automata both as an event pattern specification formalism and as an execution framework for event recognition (Brenna et al. 2007; Mozafari et al. 2013; Zhang, Diao, and Immerman 2014), while others use an ad-hoc event pattern specification language and then translate such patterns into an automata-based representation, which is eventually used for event recognition (Cugola and Margara 2010). Automata-based methods are well-suited for CER, since they are able to match event sequences in an input string, similarly to strings of characters recognized by regular automata. However, CER automata are more powerful than traditional finite state automata that recognize regular expressions, since they operate on rich event representations consisting of attributes, relations and constraints, they are capable of storing previously observed events in registers, to allow for temporal reasoning between events and they produce output rather than simply deciding whether a string is matched or not.

Another family of CER systems relies on tree-based models. In a tree-based event pattern, leaf nodes in the tree represent event attributes and inner nodes represent event operators, where an operator node is parent to two or more attribute nodes or other operator nodes, thus defining a hierarchy of event operators. Event operators may include e.g. sequencing, negation, conjunction, disjunction, Kleene closure (iteration) etc. Realizations of such models for event pattern specification are ZStream (Mei and Madden 2009) and Esper1. The recognition process in tree-based systems is based on assigning buffers to all nodes in the tree. For leaf nodes, these buffers store the input events as they arrive, whereas the buffers of non-leaf nodes store intermediate results that are assembled from sub-tree buffers. To perform even recognition, tree-based CER models start from the leaves of the tree where the input data are loaded, and they traverse the tree in a bottom-up fashion assembling match results based on the semantics of the event operators in the tree.

A third family of CER systems are logic-based. They are characterized by a formal semantics expressed in some form of logic, in contrast to other types of CER systems that often present an informal or procedural semantics (Artikis et al. 2012). In some cases, logic-based CER systems encode rules using logic programming and use inference to detect complex events (Anicic et al. 2011). Prominent logic-based approaches are based on chronicle recognition (Dousson and Maigat 2007) and the event calculus (Artikis, Sergot, and Paliouras 2015). Chronicle recognition relies on temporal logic and encodes event occurrences using logical predicates that define the time of occurrence and the content (attributes) of each event. Complex events are defined starting from primitive ones linked together with contextual and temporal constraints. Event calculus builds on fluents, which are properties that have different values at different points in time. In event calculus-based CER approaches, an event specification consists of rules that define the event occurrences, the effects of events, and the values of fluents.

Logic-based approaches have a number of important advantages as compared to automata-based and tree-based formalisms. In addition to their formal declarative semantics, they also allow to express and reason with complex relations between events and utilize rich domain knowledge in the recognition process. On the other hand, non-logical CER approaches are in general more efficient than

[1] URL: http://www.espertech.com/esper/.

logic-based ones. This is not the case with RTEC (Artikis, Sergot, and Paliouras 2015), a recent, event calculus-based CER engine, which relies on reasoning over time intervals, windowing techniques and several other runtime optimizations to scale to massive data volumes and compete in efficiency with non-logical CER approaches.

## 3.2 Uncertainty Handling in Complex Event Recognition

CER systems operate on noisy data streams. In addition to data uncertainty (e.g. missing, or erroneous input), due to the lack of veracity in big data streams, an additional source of uncertainty in CER is pattern uncertainty, i.e. cases where the employed complex event patterns are imprecise or incomplete. The ability to handle erroneous and uncertain input, as well as uncertain event patterns is an important aspect of CER research. A number of CER techniques that can handle uncertainty have been proposed, based mainly on automata, probabilistic graphical models and logic (Alevizos et al. 2017).

Automata-based approaches are usually probabilistic versions of crisp CER systems. For instance, in the probabilistic version of SASE, the goal is to recognize complex events with some probability, via considering alternative "event histories" and calculating a probability for a complex event based on the number of such histories that actually result to the recognition of the complex event and those that do not. Lahar (Ré et al. 2008) is another automata-based approach, in particular, a probabilistic version of the Cayuga CER engine. Lahar handles uncertainty via modelling events by first-order Markov processes, thereby being capable of probabilistic complex event recognition.

Another line of research is based on probabilistic graphical models, with Markov Logic Networks (MLN) being the most prominent example of using such approaches for CER (Alevizos et al. 2017). Complex event patterns in MLN are represented as weighted first-order logic formulae. Patterns with larger weights are "stronger", while patterns with smaller weights express conditions that are unlikely, but not impossible. Given a set of constants (representing e.g. time-stamps and event attributes) the formulae of an MLN specify a ground Markov network and standard inference methods from the field of probabilistic graphical models may be used to recognize complex events (Tran and Davis 2008; Liu, Deng, and Li 2017; Skarlatidis, Paliouras, et al. 2015). Other probabilistic graphical models-based formalisms have been used in a CER context as well. For instance, in (Cugola et al. 2015), the authors present an approach where the logical event pattern specification language of the TESLA CER system is embedded into a probabilistic framework based on Bayesian networks. In the field of logic programming, the ProbLog language, a probabilistic version of Prolog has been used as a basis for specifying uncertainty-handling event specifications (Skarlatidis, Artikis, et al. 2015).

## 3.3 Complex Event Recognition in Big Data Streams

Scaling CER systems to massive, high-velocity data streams is an important research topic in the event processing community. A comprehensive survey of related methods and techniques may be found in (Flouris et al. 2017). Such techniques seek to optimize the event recognition task w.r.t. a number of performance metrics, the most important of which are throughput, i.e. the number of events processed by time unit, as well as recognition time. In addition to these metrics, used mainly in cases where the entirety of the data is delivered to a single processing node, approaches based on parallel or distributed CER try to balance the cost of communication between processing nodes and the detection latency, i.e. the time between the occurrence of a complex event and its detection from a central node whose role is to continuously monitor a multitude of geographically distributed streams. Finally, memory management is another important aspect of optimizing CER systems for processing big data streams.

In centralized approaches the goal is to achieve high throughput with low recognition time and a small memory footprint. To this end, a number of techniques are utilized in an attempt to cope with the volume and velocity big data event streams. The most important of these techniques are query rewriting, predicate-related optimizations and memory management. Query rewriting is an optimization technique that allows a suboptimal query expression to be rewritten in a form that is more efficient to execute. The goal is for the rewritten query to produce exactly the same results as the original one, while exhibiting improved performance w.r.t. the optimization objectives. Most approaches to query rewriting use a set of operators that allow to translate an event pattern into a semantically equivalent form, which allows for more efficient execution. Predicate-related optimizations use early event predicate evaluation to optimize the execution of queries for matching event patterns on the input stream. This is achieved by properly partitioning the input stream and filtering the selected events that will actually be part of complex event detection based on the query. Memory management techniques focus on optimizing event buffers by e.g. removing pieces of information stored multiple times across different buffers. We refer to (Flouris et al. 2017) for a detailed presentation of such techniques.

The aforementioned techniques for scaling-up the CER process are generic, i.e. applicable to all CER approaches discussed earlier in this section (i.e. automata-based, tree-based or logic-based). In addition to such generic techniques, different CER approaches use special techniques to further increase their efficiency. For instance, automata-based approaches, which typically use non-deterministic automata, need to store at runtime all possible candidate runs, where each run depends on non-deterministic choices such as ignoring or consuming an event, and different runs result in different outputs. The maintained set of runs rapidly becomes very large (Zhang, Diao, and Immerman 2014), since it grows exponentially with the number of events in the temporal window under consideration. To cope with that, automata-based systems store the set of candidate runs in a compressed form, by e.g. factoring-out commonalities between different runs (Mozafari et al. 2013), or storing only so-called maximal runs from which other runs can be efficiently computed (Zhang, Diao, and Immerman 2014). Logic-based CER systems also resort to specialized techniques to tame the complexity of logical inference mechanisms, by e.g. translating rules into more efficient structures to perform incremental recognition as new events become available. Examples include temporal constraint networks (Dousson and Maigat 2007) and automata (Cugola and Margara 2010). Limiting the scope of inference via windowing techniques is also used in logic-based approaches, such as RTEC (Artikis, Sergot, and Paliouras 2015). Specialized techniques towards enhancing performance are also used to scale-up probabilistic CER systems, where the event recognition task is typically harder than in crisp CER systems. An overview of such techniques may be found in (Flouris et al. 2017).

Distributed CER consists of two main approaches. The first is to centralize the monitoring of the stream and distribute the complex event processing to multiple sites, as proposed in (Schultz-Møller, Migliavacca, and Pietzuch 2009). The second is to distribute the monitoring of the stream to multiple sites (where each cite receives one input stream) and centralize the processing effort, as proposed by (Akdere, Çetintemel, and Tatbul 2008). The first of these approaches seeks to improve throughput, as well as memory management. Optimizing throughput is achieved thanks to the fact that the total number of input events is distributed across multiple nodes, thus overall the system processes more events per time unit. Memory management is also improved in this processing model, since distributed processing allows for dealing with larger time windows. In the second approach to distributed CER (Akdere, Çetintemel, and Tatbul 2008) multiple input event streams are received at multiple sites and a coordinator node communicates with all sites to detect complex events. In this strategy the goal is to optimize the tradeoff between the latency in detecting complex events and the cost of communicating with the coordinator node. An example of such an approach is presented in (Akdere, Çetintemel, and Tatbul 2008), where the authors use pareto-optimality theory to generate monitoring plans for the distributed processing that conform to particular communication cost and latency constraints.

## 3.4 Machine Learning for Complex Event Recognition

Manual authoring of complex event patterns is a difficult task that requires significant effort. Moreover, event patterns need frequent updating to cope with the drifting nature of streaming data. Therefore, machine learning techniques that are able to extract event patterns from data or revise existing ones as new observations become available are highly desirable. Both supervised and unsupervised techniques have been employed to automatically construct and adapt event definitions. Widely used unsupervised learning techniques include frequency-based analysis of sequences of events (Vautier, Cordier, and Quiniou 2007), or clustering of such sequences (Lee and Jung 2017). Such approaches are promising for discovering unknown events but are limited to propositional learning, therefore they cannot be used to learn complex event patterns expressing relations between events or attributes thereof. Moreover, these techniques are hard to adapt towards learning the structure of complex events that are not frequent in the data – for instance in cases where the goal is to learn event patterns of abnormal behaviour.

A few approaches to supervised learning of complex event patterns have been proposed. In (Margara, Cugola, and Tamburrelli 2014) the authors propose a combination of techniques for learning patterns in the TESLA language (Cugola and Margara 2010), however, their approach is relatively ad-hoc, it is hard to evaluate in more mainstream machine learning settings and has limited support for the incorporation of background knowledge in the learning process. In (Mousheimish, Taher, and Zeitouni 2017) the authors use an existing method for shapelet learning (extracting patterns from time-series data) and propose a technique for temporally combining the extracted shapelets to form event patterns over multiple streams. Patterns learnt with this approach have limited expressive power, while background knowledge is also hard to utilize.

A common feature of all the above-mentioned techniques is that they assume a batch learning setting, where the training data are available before learning begins and the generated models cannot be updated in the face of new data that stream-in. Given the streaming nature of big data flows in CER, machine learning techniques for learning complex event patterns must be capable of learning in an online fashion.

A different line of work towards machine learning of event patterns has been put forth in logic-based CER approaches. Using logical formalisms as a basis for CER allows access to well-established machine learning techniques from the fields of Inductive Logic Programming (ILP) (Raedt 2008) and Statistical Relational Learning (Raedt et al. 2016), which allow to learn patterns expressing arbitrarily complex relations and constraints between events and event attributes, while easily utilizing rich domain knowledge in the process. For instance, in (Carrault et al. 2003) the authors use an off-the-shelf ILP system to learn complex event patterns in the chronicle formalism (Dousson and Maigat 2007). Moreover, online learning techniques have been proposed in event calculus-based CER approaches (Katzouris, Artikis, and Paliouras 2016; Michelioudakis et al. 2016).

## 3.5 Complex Event Recognition for Mobility Data

CER techniques are becoming increasingly important in a wide range of applications involving mobile objects, where real-time situational awareness is a requirement. Traffic/transport monitoring in intelligent transportation systems (Dasarathy 2011) is a prominent application domain. To give but a few examples, in (Terroso-Saenz et al. 2012) the authors use sensor data from a vehicular network, in addition to environmental and weather data to detect different levels of traffic jams with an event processing methodology, while in (Michelioudakis, Artikis, and Paliouras 2016) data from on-vehicle sensors and sensors mounted on road segments are used to learn complex event patterns for the early detection of traffic jams. In (Terroso-Saenz et al. 2015) a CER-based approach is proposed that allows to detect interesting situations related to the passengers' comfort and security, from data originating from sensors installed in different parts of the vehicle. Related approaches are presented in (Artikis et al. 2013, 2014), where the authors propose CER-based techniques towards the detection of events related to congestion and quality of service in intelligent transport management applications.

Maritime surveillance is another CER application domain related to mobility data. In (Patroumpas et al. 2017) the authors propose a system for online monitoring of maritime activity over streaming positions from numerous vessels sailing at sea. The system employs an online tracking module for detecting important changes in the evolving trajectory of each vessel across time, and thus can incrementally retain concise, yet reliable summaries of its recent movement. In addition, thanks to a CER module, this system is also capable for offering instant notification to marine authorities regarding emergency situations, such as suspicious moves in protected zones, or package picking at open sea. A related approach is put forth in (Boubeta-Puig et al. 2012) where the authors propose a CER-based methodology for detecting vessel communication hijacking or failure, engine malfunction or ship collision. In (Terroso-Saenz, Valdés-Vela, and Skarmeta-Gómez 2016), CER is used to detect illegal and/or dangerous activities in the maritime domain, such as collisions, smuggling or human trafficking.

Distributed processing is of utmost importance in mobility-related applications, such as those addressed in T&K. In such applications, massive data volumes are collected at different sites (e.g. moving vehicles) and much of the processing needs to take place in situ, since moving data around for centralized analysis incurs excessive communication costs. Equally important is the development of machine learning techniques for extracting and updating interesting complex event patterns from data, in order to e.g. discover abnormal mobility patterns, which domain experts have not yet identified. The requirement is for distributed, online machine learning, capable of handling the volume and velocity of data streams in mobility-related applications.

# 4 Location Allocation Problems

Location-allocation problems typical deal with provisioning of resources between facilities based on historic demand. The p-median approach is one such model that aims to minimise the total demand-weighted distance between the demand points and the facilities. This NP-Hard problem aims to locate p facilities to serve n demand, by minimising the total demand-weighted distance between the facilities and the demand. Given the computational complexity of the p-median, several approaches have been proposed to solve problems in polynomial time. These solutions include using trees (Goldman, 1971) and heuristics (metaheuristics (Mladenović et. al, 2007), Lagrangian heuristics (Daskin, 2013)). Several approaches using genetic algorithms have also been proposed to leverage the power of AI in solving the p-median problem in polynomial time (Bozkaya et. al, 2002; Alp et. al, 2003).

The formulation of the p-median problem by ReVelle and Swain provides a robust framework for solving location allocation problems. This is an unconstrained formulation of the problem as it does not take facility capacity into account when making location-allocation decisions(ReVelle and Swain, 1970). Alp, Erkut and Drezner (2003) provide a Genetic Algorithm approach to solving Revelle and Swain's model in polynomial time (Alp et. al, 2003).

Genetic algorithms can also be parallelised to shorten execution time. To fit in with a big data ecosystem Maqbool et. al, (2019) propose an approach to parallelisation. The Scalable Genetic Algorithm (S-GA) implementation that has been developed under the Boost 4.0, LAMBDA, SLIPO, and QROWD projects, provides a parallelisation strategy that utilises the Apache Spark framework (Maqbool et. al, 2019).

# 5 References

Agrawal, R., Imielinski, T., Swami, A. (1993) Mining association rules between sets of items in large databases. Proceedings of ACM SIGMOD.

Agrawal, R., Srikant, R. (2014) Mining sequential patterns. Proceedings of ICDE.

Akdere, M., U. Çetintemel, and N. Tatbul (2008) Plan-Based Complex Event Detection across Distributed Sources. PVLDB 1 (1): 66–77.

Alarabi, L., Mokbel, M.F. (2017) A Demonstration of ST-Hadoop: A MapReduce Framework for Big Spatio-temporal Data. PVLDB 10(12), 1961–1964.

Alarabi, L., Mokbel, M.F., Musleh, M. (2017) ST-Hadoop: A MapReduce Framework for Spatio-Temporal Data. Proceedings of SSTD.

Alevizos, E., A. Skarlatidis, A. Artikis, and G. Paliouras (2017) Probabilistic Complex Event Recognition: A Survey. ACM Comput. Surv. 50 (5): 71:1–71:31.

Alp, O., Erkut, E. and Drezner, Z., (2003). An efficient genetic algorithm for the p-median problem. Annals of Operations research, 122(1-4), pp.21-42.

Altché, F., and de La Fortelle, A. (2017). An LSTM network for highway trajectory prediction. In Proceedings of the 20th IEEE International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, pp. 353-359.

Anicic, D., P. Fodor, S. Rudolph, R. Stühmer, N. Stojanovic, and R. Studer (2011) ETALIS: Rule-Based Reasoning in Event Processing. In Proceedings oReasoning in Event-Based Distributed Systems, 99–124. Studies in Computational Intelligence. Springer, Berlin, Heidelberg.

Ankerst, M., Breunig, M. M., Kriegel, H. P., & Sander, J. (1999). OPTICS: ordering points to identify the clustering structure. ACM SIGMOD record, 28(2), 49-60.

Artikis, A., A. Margara, M. Ugarte, S. Vansummeren, and M.s Weidlich (2017) Complex Event Recognition Languages: Tutorial. In Proceedings of the 11th ACM International Conference on Distributed and Event-Based Systems, DEBS 2017, Barcelona, Spain, June 19-23, 2017, 7–10. ACM. https://doi.org/10.1145/3093742.3095106.

Artikis, A., A. Skarlatidis, F. Portet, and G. Paliouras (2012) Logic-Based Event Recognition. Knowledge Eng. Review 27 (4): 469–506. https://doi.org/10.1017/S0269888912000264.

Artikis, A., M. J. Sergot, and G. Paliouras (2015) An Event Calculus for Event Recognition. IEEE Trans. Knowl. Data Eng. 27 (4): 895–908. https://doi.org/10.1109/TKDE.2014.2356476.

Artikis, A., M. Weidlich, A. Gal, V. Kalogeraki, and D. Gunopulos (2013) Self-Adaptive Event Recognition for Intelligent Transport Management. In Proceedings of the 2013 IEEE International Conference on Big Data, 6-9 October 2013, Santa Clara, CA, USA. https://doi.org/10.1109/BigData.2013.6691590.

Artikis, A., M. Weidlich, F. Schnitzler, I. Boutsis, T. Liebig, N. Piatkowski, C. Bockermann, et al. (2014) Heterogeneous Stream Processing and Crowdsourcing for Urban Traffic Management. In Proceedings of the 17th International Conference on Extending Database Technology, EDBT 2014, Athens, Greece, March 24-28, 2014. https://doi.org/10.5441/002/edbt.2014.77.

Ayhan, S., Samet, H. (2016) Aircraft Trajectory Prediction Made Easy with Predictive Analytics. Proceedings of ACM SIGKDD 2016.

Ayhan, S., Samet, H. (2016) Time Series Clustering of Weather Observations in Predicting Climb Phase of Aircraft Trajectories. Proceedings of IWCTS 2016.

Bakshy, E., I. Rosenn, C. Marlow, and L. Adamic (2012) The role of social networks in information diffusion. WWW 2012 - Session: In- formation Diffusion in Social Networks April 16-20, 2012, Lyon, France, pages 519–528.

Bhattacharya, A., & Das, S. (1999) LeZi-update: an information-theoretic approach to track mobile users in PCS networks. Proceedings of ACM MobiCom.

Boubeta-Puig, J., I. Medina-Bulo, G. Ortiz, and G. Fuentes-Landi (2012) Complex Event Processing Applied to Early Maritime Threat Detection. In Proceedings of the 2Nd International Workshop on Adaptive Services for the Future Internet and 6th International Workshop on Web APIs and Service Mashups, 1–4. WAS4FI-Mashups ’12. New York, NY, USA. https://doi.org/10.1145/2377836.2377838.

Bozkaya, Burcin. Zhang, Jianjun. and Erkut, Erhan. (2002) An efficient genetic algorithm for the p-median problem. Facility location: Applications and theory, pages 179–205.

Brenna, L., A. J. Demers, J. Gehrke, M. Hong, J. Ossher, B. Panda, M. Riedewald, M. Thatte, and W. M. White (2007) Cayuga: A High-Performance Event Processing Engine. In Proceedings of the ACM SIGMOD International Conference on Management of Data, Beijing, China. https://doi.org/10.1145/1247480.1247620.

Brilhante, I. R., M. Berlingerio, R. Trasarti, C. Renso, J. A. F. d. Macedo and M. A. Casanova (2012) ComeTogether: Discovering Communities of Places in Mobility Data. Proceedings of IEEE 13th International Conference on Mobile Data Management, Bengaluru, Karnataka. doi: 10.1109/MDM.2012.17

Brockmann, D. (2009) Human mobility and spatial disease dynamics. Reviews of nonlinear dynamics and complexity, pp. 1–24. Wiley-VCH, Weinheim.

Bussas, M., Sawade, C., Kühn, N. et al. Machine Learning Journal (2017). 106: 1419. https://doi.org/10.1007/s10994-017-5639-3

Cao, H., Mamoulis, N., & Cheung, D. W. (2006). Discovery of collocation episodes in spatiotemporal data. Proceedings of 6th IEEE Conference on Data Mining.

Carrault, G., M.-O. Cordier, R. Quiniou, and F. Wang (2003) Temporal Abstraction and Inductive Logic Programming for Arrhythmia Recognition from Electrocardiograms. Artificial Intelligence in Medicine 28 (3): 231–263. https://doi.org/10.1016/S0933-3657(03)00066-6.

Cha, M., A. Mislove, and K. P. Gummadi (2009) A measurement- driven analysis of information propagation in the flickr social network. In Proceedings of the 18th international conference on World wide web, pages 721–730. ACM.

Chai D, Wang L and Yang Q (2018) Bike flow prediction with multi-graph convolutional networks. In Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 397–400, ACM.

Chawla, S., Y. Zheng, and J. Hu (2012) Inferring the root cause in road traffic anomalies. In Proceedings of the 12th IEEE International Conference on Data Mining. IEEE, 141-150.

Chen, C.C., Tseng, C.Y., Chen, M.S. (2013) Highly Scalable Sequential Pattern Mining Based on MapReduce Model on the Cloud. Proceedings of IEEE International Congress on Big Data.

Chen, X.W., Landry, S.J., Nof, S.Y. (2011) A framework of enroute air traffic conflict detection and resolution through complex network analysis. Computers in Industry 62, 8 (2011), 787–794.

Chen, Z., Shen, H. T., & Zhou, X. (2011). Discovering popular routes from trajectories. Proceedings of IEEE Conference on Data Engineering.

Cheng, T., Cui, D., Cheng, P. (2003) Data mining for air traffic flow forecasting: a hybrid model of neural network and statistical analysis. Proceedings of ITSC 2003.

Choi, S., Kim, J., and Yeo, H. (2019). Attention-based Recurrent Neural Network for Urban Vehicle Trajectory Prediction. Procedia Computer Science, 151, 327-334. https://doi.org/10.1016/j.procs.2019.04.046

Claramunt, C., Ray, C., Camossi, E., Jousselme, A.L., Hadzagic, M., Andrienko, G., Andrienko, N., Theodoridis, Y., Vouros, G., Salmon, L. (2017) Maritime data integration and analysis: recent progress and research challenges. Proceedings of EDBT.

Coppenbarger, R.A. (1999) En Route Climb Trajectory Prediction Enhancement Using Airplane Flight-Planning Information. American Institute of Aeronautics and Astronautics, AIAA-99-4147.

Cugola, G. (2012) Processing Flows of Information: From Data Stream to Complex Event Processing. ACM Comput. Surv. 44 (3): 15:1–15:62. https://doi.org/10.1145/2187671.2187677.

Cugola, G., A. Margara, M. Matteucci, and G. Tamburrelli (2015) Introducing Uncertainty in Complex Event Processing: Model, Implementation, and Validation. Computing 97 (2): 103–44. https://doi.org/10.1007/s00607-014-0404-y.

Cugola, G., and A. Margara (2010) TESLA: A Formally Defined Event Specification Language. In Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems, DEBS 2010, Cambridge, United Kingdom. https://doi.org/10.1145/1827418.1827427.

Dasarathy, B. V. (2011) A Special Issue on Intelligent Transportation Systems. Information Fusion 12 (1): 1. https://doi.org/10.1016/j.inffus.2010.06.009.

Daskin, M.S., (2013). *Network and discrete location: models, algorithms, and applications*. John Wiley & Sons. 2 ed.

de Dios Ortuzar J. and Willumsen L. G. (2011) Modelling transport. John Wiley & Sons.

de Leege, A., Van Paassen, M., Mulder, M. (2013) A machine learning approach to trajectory prediction. Proceedings of AIAA GNC 2013.

de Raedt, L. (2008) Logical and Relational Learning. Cognitive Technologies. Springer. https://doi.org/10.1007/978-3-540-68856-3.

de Raedt, L., K. Kersting, S. Natarajan, and D. Poole (2016) Statistical Relational Artificial Intelligence: Logic, Probability, and Computation. Synthesis Lectures on Artificial Intelligence and Machine Learning 10 (2): 1–189. https://doi.org/10.2200/S00692ED1V01Y201601AIM032.

Defferrard M, Bresson X and Vandergheynst P (2016) Convolutional neural networks on graphs with fast localized spectral filtering. In Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16.

Deng, Z., Hu, Y., Zhu, M., Huang, X., & Du, B. (2015). A scalable and fast OPTICS for clustering trajectory big data. Cluster Computing, 18(2), 549-562.

Di Ciccio, C., var der Aa, H., Cabanillas, C., et al. (2016) Detecting flight trajectory anomalies and predicting diversions in freight transportation. Decision Support Systems 88 (2016), 1–17.

Ding, X., Chenz, L., Gao, Y., Jensenz, C.S., Bao, H. (2018) UlTraMan: A Unified Platform for Big Trajectory Data Management and Analytics. Proceedings of VLDB'18.

Doulkeridis, C., Vlachou, A., Mpestas, D., Mamoulis, N. (2017) Parallel and Distributed Processing of Spatial Preference Queries using Keywords. Proceedings of EDBT.

Dousson, C., and P. Le Maigat (2007) Chronicle Recognition Improvement Using Temporal Focusing and Hierarchization. In Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI), Hyderabad, India. http://ijcai.org/Proceedings/07/Papers/050.pdf.

Eldawy, A., Mokbel, M.F. (2016) The Era of Big Spatial Data: A Survey. Foundations and Trends in Databases 6(3-4), 163–273.

Erlander S. and Stewart N. F., The gravity model in transportation analysis: theory and extensions, vol. 3. Vsp, 1990.

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of KDD.

Fan, Q., Zhang, D., Wu, H., & Tan, K. L. (2016). A general and parallel platform for mining co-movement patterns over large-scale trajectories. Proceedings of the VLDB Endowment, 10(4), 313-324.

Fan, X., Guo, L., Han, N., Wang, Y., Shi, J., and Yuan, Y. (2018). A deep learning approach for next location prediction. In Proceedings of the 22nd IEEE International Conference on Computer Supported Cooperative Work in Design ((CSCWD)), Nanjing, China, pp. 69-74. https://doi.org/10.1109/CSCWD.2018.8465289.

Fang, Y., Cheng, R., Tang, W., Maniu, S., Yang, X.S. (2016) Scalable Algorithms for Nearest-Neighbor Joins on Big Trajectory Data. IEEE Trans. Knowl. Data Eng. 28(3), 785–800.

Flouris, I., N. Giatrakos, A. Deligiannakis, M. N. Garofalakis, M. Kamp, and M. Mock (2017) Issues in Complex Event Processing: Status and Prospects in the Big Data Era. Journal of Systems and Software 127: 217–236. https://doi.org/10.1016/j.jss.2016.06.011.

Fort, M., Sellarès, J. A., & Valladares, N. (2014). A parallel GPU-based approach for reporting flock patterns. International Journal of Geographical Information Science, 28(9), 1877-1903.

Fortunato S. (2010) Community detection in graphs. Physics Reports. 486, 75–174.

Fotheringham A. S. and O'Kelly M. E. (1989) Spatial interaction models: formulations and applications. vol. 1. Kluwer Academic Publishers Dordrecht.

Goldman AJ. (1971) Optimal center location in simple networks. Transportation science, 5(2):212– 221.

Gomes, J., Phua, C., & Krishnaswamy, S. (2013) Where will you go? Mobile data mining for next place prediction. Proceedings of DaWaK.

Gong, C., McNally, D. (2004) A methodology for automated trajectory prediction analysis. Proceedings of AIAA GNC 2004.

Gonzalez, M.C., Hidalgo, C.A., Barabasi, A.L. (2008) Understanding individual human mobility patterns. Nature 453, 779–782.

Goo, J. (2010) Transfer learning for classification of spatially varying data. PhD dissertation at University of Texas at Austin, Department of Electrical and Computer Engineering. http://hdl.handle.net/2152/ETD-UT-2010-08-1962.

Hadjaz, A., Marceau, G., Saveant, P., et al. (2012) Online learning for ground trajectory prediction. CoRR abs/1212.3998.

Hagedorn, S., Räth, T. (2017) Efficient spatio-temporal event processing with STARK. Proceedings of EDBT.

Hamed, M., Gianazza, D., Serrurier, M., & Durand, N. (2013) Statistical prediction of aircraft trajectory: regression methods vs point-mass model. Proceedings of ATM.

Han, J., Pei, J. (2000) Mining frequent patterns by pattern-growth: methodology and implications. ACM SIGKDD Explor. Newsl., 2(2), 14–20.

Hendawi, A.M., Mokbel, M.F. (2012a) Panda: A Predictive Spatio-Temporal Query Processor. Proceedings of ACM SIGSPATIAL GIS'12.

Hendawi, A.M., Mokbel, M.F. (2012b) Predictive Spatio-Temporal Queries: A Comprehensive Survey and Future Directions. Proceedings of ACM SIGSPATIAL MobiGIS'12.

Hendawi, A.M., Bao, J., Mokbel, M.F. (2013) iRoad: A Framework For Scalable Predictive Query Processing on Road Networks. Proceedings of VLDB'13.

Hendawi, A.M., Bao, J., Mokbel, M.F., Ali, M. (2015a) Predictive Tree: An Efficient Index for Predictive Queries on Road Networks. Proceedings of ICDE 2015.

Hendawi, A.M., Ali, M., Mokbel, M.F. (2015b) A Framework for Spatial Predictive Query Processing and Visualization, Proc. of the 16th IEEE International Conference on Mobile Data Management.

Hendawi, A.M., Ali, M., Mokbel, M.F. (2017) Panda ∗: A generic and scalable framework for predictive spatio-temporal queries. GeoInformatica, 21(2), 175-208.

Holme, P., & Saramäki, J. (2012). Temporal networks. Physics reports, 519(3), 97-125.

Hossmann, T., T. Spyropoulos and F. Legendre (2011) A complex network analysis of human mobility. 2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Shanghai, pp. 876-881. doi: 10.1109/INFCOMW.2011.5928936

Hou, L., Xin, L., Li, S. E., Cheng, B., & Wang, W. (2019). Interactive trajectory prediction of surrounding road users for autonomous driving using structural-LSTM network. IEEE Transactions on Intelligent Transportation Systems. vol. 21, no. 11, pp. 4615-4625, Nov. 2020, doi: 10.1109/TITS.2019.2942089

Iddianozie C and McArdle G (2019) A transfer learning paradigm for spatial networks. SAC '19: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, April 2019, Pages 659–666.

Ishikawa, Y., Tsukamoto, Y., & Kitagawa, H. (2004) Extracting mobility statistics from indexed spatio-temporal datasets. Proceedings of STDBM.

Jagadish, H.V. (1990) On indexing line segments. Proceedings of VLDB.

Jalili, M., Orouskhani, Y., Asgari, M., Alipourfard, N., & Perc, M. (2017). Link prediction in multiplex online social networks. Royal Society open science, 4(2), 160863.

Jensen, C., Lin, D., & Ooi, B. (2004) Query and update efficient B+-tree based indexing of moving objects. Proceedings of VLDB.

Jeung, H., Yiu, M.L., Zhou, X., Jensen, C.S. (2010) Path prediction and predictive range querying in road network databases. VLDB Journal 19 (2010), 585-602.

Jinno, R., Seki, K., & Uehara, K. (2012). Parallel distributed trajectory pattern mining using MapReduce. Proceedings of IEEE Cloud Computing Technology and Science.

Kalnis, P., Mamoulis, N., & Bakiras, S. (2005). On discovering moving clusters in spatio-temporal data. Proceedings of International Symposium on Spatial and Temporal Databases.

Kasteren, T. L. M. van, Englebienne, G., Krose, B. J. A. (2010) Transferring Knowledge of Activity Recognition across Sensor Networks. Pervasive Computing pp 283-300.

Katzouris, N., A. Artikis, and G. Paliouras (2016) Online Learning of Event Definitions. TPLP 16 (5–6): 817–833. https://doi.org/10.1017/S1471068416000260.

Kim, B., Kang, C. M., Kim, J., Lee, S. H., Chung, C. C., and Choi, J. W. (2017). Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network. In Proceedings of the 20th IEEE International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, pp. 399-404, https://doi.org/10.1109/ITSC.2017.8317943.

Kipf T N and Welling M (2017) Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the 5th International Conference on Learning Representations, ICLR ’17.

Krumm, J., Horvitz, E. (2003) Predestination: inferring destinations from partial trajectories. Proceedings of UbiComp 2003.

Lan, R., Yu, Y., Cao, L., Song, P., & Wang, Y. (2017). Discovering Evolving Moving Object Groups from Massive-Scale Trajectory Streams. Proceedings of IEEE Conference on Mobile Data Management.

Laube, P., Imfeld, S., & Weibel, R. (2005a). Discovering relative motion patterns in groups of moving point objects. International Journal of Geographical Information Science, 19(6), 639-668.

Laube, P., van Kreveld, M., & Imfeld, S. (2005b). Finding REMO—detecting relative motion patterns in geospatial lifelines. Developments in Spatial Data Handling. Springer.

Le Fablec, Y., Alliot, J.M. (1999) Using neural networks to predict aircraft trajectories. Proceedings of ICIS 1999.

Lee, J. G., Han, J., & Whang, K. Y. (2007). Trajectory clustering: a partition-and-group framework. Proceedings of ACM SIGMOD International Conference on Management of Data.

Lee, O.-J., and J. E. Jung (2017) Sequence Clustering-Based Automated Rule Generation for Adaptive Complex Event Processing. Future Generation Comp. Syst. 66: 100–109. https://doi.org/10.1016/j.future.2016.02.011.

Lefèvre, S., Vasquez, D., and Laugier, C. (2014). A survey on motion prediction and risk assessment for intelligent vehicles. ROBOMECH journal, 1(1), 1-14. https://doi.org/ 10.1186/s40648-014-0001-z

Leskovec, J., M. McGlohon, C. Faloutsos, N. Glance, and M. Hurst (2007) Cascading Behavior in Large Blog Graphs. sdm, pages 1–21.

Liben-Nowell D, Kleinberg J (2007) The link prediction problem for social networks. J Am Soc Inform Sci Technol 58(7):1019–1031

Liu, F., D. Deng, and P. Li (2017) Dynamic Context-Aware Event Recognition Based on Markov Logic Networks. Sensors 17 (3): 491. https://doi.org/10.3390/s17030491.

Liu, H., Huang, X., Wen, D., Li, J. (2017) The Use of Landscape Metrics and Transfer Learning to Explore Urban Villages in China. Remote Sens., 9, 365.

Lu L, Zhou T (2011) Link prediction in complex networks: a survey. Phys A Stat Mech Appl 390(6):1150–1170

Ma, Y., Zhu, X., Zhang, S., Yang, R., Wang, W., and Manocha, D. (2019). Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, pp. 6120-6127. https://doi.org/10.1609/aaai.v33i01.33016120

Maqbool, F., Razzaq, S., Lehmann, J. and Jabeen, H., (2019), August. Scalable Distributed Genetic Algorithm Using Apache Spark (S-GA). In Proceedings of the International Conference on Intelligent Computing (pp. 424-435). Springer, Cham.

Margara, A., G. Cugola, and G. Tamburrelli (2014) Learning from the Past: Automated Rule Generation for Complex Event Processing. In The 8th ACM International Conference on Distributed Event-Based Systems, DEBS ’14, Mumbai, India. https://doi.org/10.1145/2611286.2611289.

Matsuno, Y., Tachiya, T., Wei, J., et al. (2015) Stochastic optimal control for aircraft conflict resolution under wind uncertainty. Aerospace Science and Technology 43 (2015), 77–88.

McArthur D. P., Kleppe G., Thorsen I. and Ubøe J. (2011) The spatial transferability of parameters in a gravity model of commuting flows. Journal of Transport Geography, vol. 19, no. 4, pp. 596–605.

Mei, Y., and S. Madden (2009) ZStream: A Cost-Based Query Processor for Adaptively Detecting Composite Events. In Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2009, Providence, Rhode Island, USA. ACM. https://doi.org/10.1145/1559845.1559867.

Michelioudakis, E., A. Artikis, and G. Paliouras (2016) Online Structure Learning for Traffic Management. In Inductive Logic Programming - 26th International Conference, ILP 2016, London, UK. Springer. https://doi.org/10.1007/978-3-319-63342-8_3.

Michelioudakis, E., A. Skarlatidis, G. Paliouras, and A. Artikis (2016) $$\mathtt {OSL}\alpha $$: Online Structure Learning Using Background Knowledge Axiomatization. In Machine Learning and Knowledge Discovery in Databases, 232–47. Lecture Notes in Computer Science. Springer, Cham. https://doi.org/10.1007/978-3-319-46128-1_15.

Mladenović, N., Brimberg, J., Hansen, P. and Moreno-Pérez, J.A., (2007). The p-median problem: A survey of metaheuristic approaches. European Journal of Operational Research, 179(3), pp.927-939.

Monreale, A., Pinelli, F., Trasarti, R., & Giannotti, F. (2009) WhereNext: a location predictor on trajectory pattern mining. Proceedings of ACM SIGKDD.

Mousheimish, R., Y. Taher, and K. Zeitouni (2017) Automatic Learning of Predictive CEP Rules: Bridging the Gap between Data Mining and Complex Event Processing. In Proceedings of the 11th ACM International Conference on Distributed and Event-Based Systems, DEBS 2017, Barcelona, Spain. ACM. https://doi.org/10.1145/3093742.3093917.

Moussalli, R., Absalyamov, I., Vieira, M. R., Najjar, W., & Tsotras, V. J. (2015). High performance FPGA and GPU complex pattern matching over spatio-temporal streams. GeoInformatica, 19(2), 405-434.

Moussalli, R., Vieira, M. R., Najjar, W., & Tsotras, V. J. (2013). Stream-mode fpga acceleration of complex pattern trajectory querying. Proceedings of International Symposium on Spatial and Temporal Databases.

Mozafari, B., K. Zeng, L. D’Antoni, and C. Zaniolo (2013) High-Performance Complex Event Processing over Hierarchical Data. ACM Trans. Database Syst. 38 (4): 21:1–21:39. https://doi.org/10.1145/2536779.

Mussone L and Matteucci M (2013) OD matrices network estimation from link counts by neural networks,” Journal of Transportation Systems Engineering and Information Technology, vol. 13, no. 4, pp. 84–92.

Nanni, M., & Pedreschi, D. (2006). Time-focused clustering of trajectories of moving objects. Journal of Intelligent Information Systems, 27(3), 267-289.

Newman, M. E. J. (2003) The structure and function of complex networks. SIAM Review. 45, 2, 167–256.

Nikitopoulos, P., Paraskevopoulos, A., Doulkeridis, C., Pelekis, N., Theodoridis, Y. (2018) Hot Spot Analysis for Big Trajectory Data. Proceedings of SSDBM’18 (submitted).

Ord, J. K., Getis, A. (1995) Local Spatial Autocorrelation Statistics: Distributional Issues and an Application. Geographical Analysis 27(4), 286–306.

Pan, S. J. and Yang, Q. (2010) A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering, Volume: 22, Issue: 10, pp. 1345-1359.

Panagiotakis, C., Pelekis, N., Kopanakis, I., Ramasso, E., Theodoridis, Y. (2012). Segmentation and sampling of moving object trajectories based on representativeness. IEEE Transactions on Knowledge and Data Engineering, 24(7), 1328-1343.

Park, S. H., Kim, B., Kang, C. M., Chung, C. C., and Choi, J. W. (2018). Sequence-to-sequence prediction of vehicle trajectory via LSTM encoder-decoder architecture. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), pp. 1672-1678. https://doi.org/10.1109/IVS.2018.8500658

Patroumpas, K., E. Alevizos, A. Artikis, M. Vodas, N. Pelekis, and Y. Theodoridis (2017) Online Event Recognition from Moving Vessel Trajectories. GeoInformatica 21 (2): 389–427. https://doi.org/10.1007/s10707-016-0266-x.

Patuelli R. et al (2007) Network analysis of commuting flows: A comparative static approach to german data. Networks and Spatial Economics, vol. 7, no. 4, pp. 315-331.

Patwary, M. M. A., Palsetia, D., Agrawal, A., Liao, W. K., Manne, F., & Choudhary, A. (2013). Scalable parallel OPTICS data clustering using graph algorithmic techniques. Proceedings of IEEE Int. Conf. on High Performance Computing, Networking, Storage and Analysis.

Pecher, P., Hunter, M., and Fujimoto, R. (2016). Data-driven vehicle trajectory prediction. In Proceedings of the ACM SIGSIM Conference on Principles of Advanced Discrete Simulation, New York, NY, USA, pp. 13-22. http://doi.acm.org/10.1145/2901378.2901407

Pelekis, N., Kopanakis, I., Kotsifakos, E. E., Frentzos, E., & Theodoridis, Y. (2011). Clustering uncertain trajectories. Knowledge and Information Systems, 28(1), 117-147.

Pelekis, N., Kopanakis, I., Panagiotakis, C., & Theodoridis, Y. (2010). Unsupervised trajectory sampling. Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases.

Pelekis, N., Tampakis, P., Vodas, M., Doulkeridis, C., & Theodoridis, Y. (2017). On temporal-constrained sub-trajectory cluster analysis. Data Mining and Knowledge Discovery, 31(5), 1294-1330.

Pelekis, N., Theodoridis, Y. (2014) Mobility Data Management and Exploration. Springer.

Rabiner, L.R. (1989) A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE 77, 2 (1989), 257–286.

Rathore, P., Kumar, D., Rajasegarar, S., Palaniswami, M., and Bezdek, J. C. (2019). A scalable framework for trajectory prediction. IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 10, pp. 3860-3874. https://doi.org/10.1109/TITS.2019.2899179

Ré, C., J. Letchner, M. Balazinska, and D. Suciu (2008) Event Queries on Correlated Probabilistic Streams. In Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada. ACM. https://doi.org/10.1145/1376616.1376688.

ReVelle, Charles S. and Swain, Ralph W. (1970) Central facilities location. Geographical analysis, 2(1):30–42.

Rinzivillo, S., Gabrielli, L., Nanni, M., Pappalardo, L., Pedreschi, D., Giannotti F. (2014) The purpose of motion: Learning activities from individual mobility networks. Proceedings of International Conference on Data Science and Advanced Analytics (DSAA).

Rossetti, G., & Cazabet, R. (2018). Community Discovery in Dynamic Networks: A Survey. ACM Computing Surveys (CSUR), 51(2), 35.

Rossetti, G., Pappalardo, L., Kikas, R., Pedreschi, D., Giannotti, F., Dumas, M. (2016) Homophilic network decomposition: a community-centric analysis of online social services. Social Network Analysis and Mining.

Saberi, M., Mahmassani, H.S., Brockmann, D. et al. A complex network perspective for characterizing urban travel demand patterns: graph theoretical analysis of large-scale origin–destination demand networks. Transportation (2017) 44: 1383. https://doi.org/10.1007/s11116-016-9706-6

Sacharidis, D., Patroumpas, K., Terrovitis, M., Kantere, V., Potamias, M., Mouratidis, K., & Sellis, T. (2008). On-line discovery of hot motion paths. Proceedings of the ACM 11th international conference on Extending database technology: Advances in database technology.

Samet, H. (1990) The Design and Analysis of Spatial Data Structures. Addison-Wesley.

Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G (2009) The graph neural network model. IEEE Transactions on Neural Networks, vol. 20, no. 1, pp. 61–80. https://doi.org/10.1109/TNN.2008.2005605

Schultz-Møller, N. P., M. Migliavacca, and P. R. Pietzuch (2009) Distributed Complex Event Processing with Query Rewriting. In Proceedings of the Third ACM International Conference on Distributed Event-Based Systems, DEBS 2009, Nashville, Tennessee, USA. ACM. https://doi.org/10.1145/1619258.1619264.

Sip, S., Green, S.M. (2003) Common Trajectory Prediction Capability for Decision Support Tools. ATM 5th USA/Europa R&D seminar, Budapest.

Skarlatidis, A., A. Artikis, J. Filipou, and G. Paliouras (2015) A Probabilistic Logic Programming Event Calculus. TPLP 15 (2): 213–245. https://doi.org/10.1017/S1471068413000690.

Skarlatidis, A., G. Paliouras, A. Artikis, and G. A. Vouros (2015) Probabilistic Event Calculus for Event Recognition. ACM Trans. Comput. Log. 16 (2): 11:1–11:37. https://doi.org/10.1145/2699916.

Song, Y., Cheng, P., Mu, C. (2012) An improved trajectory prediction algorithm based on trajectory data mining for air traffic management. Proceedings of IEEE ICIA 2012.

Spadon G. et al (2019) Reconstructing commuters network using machine learning and urban indicators. Scientific reports, vol. 9, no. 1, pp. 1–13.

Szor P. (2004) Fighting computer virus attacks. USENIX

Tang, M., Yu, Y., Malluhi, Q.M., Ouzzani, M., Aref, W.G. (2016) LocationSpark: A Distributed In-Memory Data Management System for Big Spatial Data. PVLDB 9(13), 1565–1568.

Tao, Y., Faloutsos, C., Papadias, D., & Liu, B. (2004) Prediction and indexing of moving objects with unknown motion patterns. Proceedings of ACM SIGMOD.

Tao, Y., Papadias, D., & Sun, J. (2003) The TPR*-tree: an optimized spatio-temporal access method for predictive queries. Proceedings of VLDB.

Tastambekov, K., Puechmorel, S., Delahaye, D., et al. (2014) Aircraft trajectory forecasting using local functional regression in Sobolev space. Transportation research part C: Emerging technologies 39 (2014), 1–22.

Tayeb, J., Ulusoy, Ö., & Wolfson, O. (1998) A quadtree-based dynamic attribute indexing method. The Computer Journal, 41(3), 185-200.

Terroso-Saenz, F., M. Valdés-Vela, C. Sotomayor Martínez, R. Toledo-Moreo, and A. F. Gómez-Skarmeta (2012) A Cooperative Approach to Traffic Congestion Detection With Complex Event Processing and VANET. IEEE Trans. Intelligent Transportation Systems 13 (2): 914–929. https://doi.org/10.1109/TITS.2012.2186127.

Terroso-Saenz, F., M. Valdés-Vela, F. Campuzano, J. A. Botía, and A. F. Gómez-Skarmeta (2015) A Complex Event Processing Approach to Perceive the Vehicular Context. Information Fusion 21: 187–209. https://doi.org/10.1016/j.inffus.2012.08.008.

Terroso-Saenz, F., M. Valdés-Vela, and A. F. Skarmeta-Gómez (2016) A Complex Event Processing Approach to Detect Abnormal Behaviours in the Marine Environment. Information Systems Frontiers 18 (4): 765–780. https://doi.org/10.1007/s10796-015-9560-7.

Theodoridis, S., Koutroumbas, K. (2008) Pattern Recognition (4/e). Academic Press.

Thiemann, C., F. Theis, D. Grady, R. Brune, D. Brockmann (2010) The Structure of Borders in a Small World. PLoS One. 5(11): e15422. https://doi.org/10.1371/journal.pone.0015422

Thipphavong, D.P., Schultz, C.A., Lee, A.G., et al. (2013) Adaptive Algorithm to Improve Trajectory Prediction Accuracy of Climbing Aircraft. Journal of Guidance, Control and Dynamics (JGCD) 36(1), 15–24.

Tian, J., J. Hahner, C. Becker, I. Stepanov and K. Rothermel (2002) Graph-based mobility model for mobile ad hoc network simulation. Proceedings 35th Annual Simulation Symposium. SS'2002, pp. 337-344.

Tran, S. D., and L. S. Davis (2008) Event Modeling and Recognition Using Markov Logic Networks. In Computer Vision – ECCV 2008, 610–23. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-88688-4_45.

Trasarti, R., Guidotti, R., Monreale, A., & Giannotti, F. (2017) MyWay: Location Prediction via mobility profiling. Information Systems, 64, 350-367.

Tsung, F., Zhang, K., Cheng, L, Song (2017). Statistical transfer learning: A review and some extensions to statistical process control, Quality Engineering, 30:1, 115-128, DOI: 10.1080/08982112.2017.1373810

Valladares Cereceda, I. (2013). GPU parallel algorithms for reporting movement behaviour patterns Proceedings of Spatiotemporal Databases.

Vautier, A., M.-O. Cordier, and R. Quiniou (2007) Towards Data Mining Without Information on Knowledge Structure. In Knowledge Discovery in Databases: PKDD 2007, 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, Warsaw, Poland. Springer. https://doi.org/10.1007/978-3-540-74976-9_29.

Verhein, F., & Chawla, S. (2006) Mining spatio-temporal association rules, sources, sinks, stationary regions and thoroughfares in object mobility databases. Proceedings of DASFAA.

Viterbi, A. (1967) Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory 13(2), 260–269.

Vouros, G.A., Vlachou, A., Santipantakis, G., et al. (2018) Big data analytics for time critical mobility forecasting: recent progress and research challenges. Proceedings of EDBT 2018.

Wang, C., Ma, L., Li, R., Durrani, T. S., and Zhang, H. (2019). Exploring trajectory prediction through machine learning methods. IEEE Access, 7, 101441-101452. https://doi.org/10.1109/ACCESS.2019.2929430

Wang D., Pedreschi, D., Song, C., Giannotti, F., & Barabasi, A. L. (2011). Human mobility, social ties, and link prediction. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 1100-1108). ACM.

Wang, L., Geng, X., Ke, J., Peng, C., Ma, X., Zhang, D., Yang, Q. (2017) Ridesourcing Car Detection by Transfer Learning. Eprint arXiv:1705.08409.

Wang, P., Gonzalez, M.C., Menezes, R., Barabasi, A.L. (2013) Understanding the spread of malicious mobile-phone programs and their damage potential. IJIS

Wang, X., Wang, J., Wang, T., Li, H., Yang, D. (2010) Parallel Sequential Pattern Mining by Transaction Decomposition. Proceedings of 7th Int. Conf. on Fuzzy Systems and Knowledge Discovery.

Wang X, Zhou Z, Tang Z, Liu Y, Peng C (2019) Spatio-temporal analysis and prediction of cellular traffic in metropolis. IEEE Transactions on Mobile Computing, vol. 18, no. 9, pp. 2190–2202.

Wang, Y., Zhang, D., Liu, Y., & Tan, K. L. (2020). Trajectory Forecasting with Neural Networks: An Empirical Evaluation and A New Hybrid Model. IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 10, pp. 4400-4409, https://doi.org/10.1109/TITS.2019.2943055

Wei, Y., Zheng, Y., and Yang, Q. (2016) Transfer Knowledge between Cities. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16).

Wilson, A. G. (1971) A family of spatial interaction models, and associated developments. Environment and Planning A, vol. 3, no. 1, pp. 1–32.

Whitman, R.T., Park, M.B., Marsh, B.G., Hoel, E.G. (2017) Spatio-Temporal Join on Apache Spark. Proceedings of ACM SIGSPATIAL GIS'17.

Woolley-Meza, O., Thiemann, C., Grady, D., Lee, J., Seebens, H., Blasius, B. and Brockmann, D., (2011), Complexity in human transportation networks: a comparative analysis of worldwide air transportation and global cargo-ship movements, The European Physical Journal B: Condensed Matter and Complex Systems, 84, issue 4, p. 589-600.

Wu, F., Fu, K., Wang, Y., Xiao, Z., & Fu, X. (2017). A spatial-temporal-semantic neural network algorithm for location prediction on moving objects. Algorithms, 10(2), 37. https://doi.org/10.3390/a10020037

Xia, F., J. Wang, X. Kong, Z. Wang, J. Li and C. Liu (2018) Exploring Human Mobility Patterns in Urban Scenarios: A Trajectory Data Perspective. IEEE Communications Magazine, vol. 56, no. 3, pp. 142-149. doi: 10.1109/MCOM.2018.1700242.

Xian, Y., Liu, Y., Xu, C. (2016) Parallel gathering discovery over big trajectory data. Proceedings of IEEE International Conference on Big Data.

Xie P, Li T, Liu J, Du S, Yang X, Zhang J (2019) Urban flows prediction from spatial-temporal data using machine learning: A survey. arXiv preprint arXiv: 1908.10218.

Yang, J., & Hu, M. (2006) TrajPattern: Mining sequential patterns from imprecise trajectories of mobile objects. Proceedings of EDBT.

Yang, Y., Zhang, J., Cai, K.Q. (2015) Terminal-area aircraft intent inference approach based on online trajectory clustering. The Scientific World Journal, 671360 (2015).

Yavas, G., Katsaros, D., Ulusoy, Ö., & Manolopoulos, Y. (2005) A data mining approach for location prediction in mobile environments. Data and Knowledge Engineering, 54(2), 121-146.

Ying, J.-C., Lee, W.-C., Weng, T.-C., & Tseng, V. (2011) Semantic trajectory mining for location prediction. Proceedings of ACM SIGSPATIAL.

Zhan, W., La de Fortelle, A., Chen, Y. T., Chan, C. Y., & Tomizuka, M. (2018). Probabilistic prediction from planning perspective: Problem formulation, representation simplification and evaluation metric. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Changshu, China, pp. 1150-1156, https://doi.org/10.1109/IVS.2018.8500697.

Zhang, D., He, T., Zhang, F., & Xu, C. (2018). Urban-Scale Human Mobility Modeling With Multi-Source Urban Network Data. IEEE/ACM Transactions on Networking. DOI: 10.1109/TNET.2018.2801598

Zhang, H., Y. Diao, and N. Immerman (2014) On Complexity and Optimization of Expensive Queries in Complex Event Processing. In International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA. ACM. https://doi.org/10.1145/2588555.2593671.

Zhang L., Cheng J.and Jin C. (2019) Spatial interaction modeling of OD flow data: Comparing geographically weighted negative binomial regression (GWNBR) and OLS (GWOLSR). ISPRS International Journal of Geo-Information, vol. 8, no. 5, p. 220.

Zhang, M., Chen, S., Jensen, C.S., Ooi, B.C., Zhang, Z. (2009) Effectively Indexing Uncertain Moving Objects for Predictive Queries. Proceedings of VLDB '09.

Zhang, R., Qi, J., Lin, D., Wang, W., Chi-WingWong, R. (2012) A highly optimized algorithm for continuous intersection join queries over moving objects, VLDB Journal 21 (2012), 561-586.

Zheng, Y., F. Liu, and H.-P. Hsieh (2013) U-Air: when urban air quality inference meets big data. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '13), New York, NY, USA, 1436-1444. ACM. DOI: https://doi.org/10.1145/2487575.2488188

Zheng, Y. (2015) Trajectory Data Mining: An Overview. Transactions on Intelligent Systems and Technology 6(3), 1–41.

Zhu D and Liu Y (2018) Modelling spatial patterns using graph convolutional networks. In Proceedings of the 10th International Conference on Geographic Information Science (GIScience 2018).

Zhu, X., Li, B., Wu, X., He, D., Zhang, C. (2011) CLAP: Collaborative pattern mining for distributed information systems. Decision Support Systems 52, pp. 40-51.

Ziv, J. & Lempel, A. (1978) Compression of individual sequences via variable-rate coding. IEEE Transactions on Information Theory, 24(5), 530–536.

Zorbas, N., Zissis, D., Tserpes, K., & Anagnostopoulos, D. (2015) Predicting object trajectories from high-speed streaming data. Proceedings of IEEE TrustCom/BigDataSE/ISPA.