A Survey of Smart Cards Data Mining

Dmitry Namiot and Manfred Sneps-Sneppe

Faculty of Computational Mathematics and Cybernetics
Lomonosov Moscow State University; National Competence Center for Digital Economy
GSP-1, 1-52, Leninskiye Gory, Moscow, 119991, Russia
and Institute of mathematics and computer science
University of Latvia
Raina bulvaris 29, Riga, LV-1459, Latvia

Abstract. Smart cards are used worldwide in transport applications as a payment tool. So, cities will have big and constantly updated collections of transactions data from cards validation equipment. Of course, this data source could be used to study user behavior from collected observations, to detect movement patterns, to perform new route planning, etc. In this paper, we are going to provide a survey of data models used in such kind of analysis and describe practical questions (problems) that could be solved with histories of payment transactions collected by cities and transport companies. In our opinion, this information will be useful for smart cities applications because of the relative ease of collection of such data and their transparency. In the context of smart city users, mobility is one of the keys components.

Keywords: smart card, transport cards, data mining, mobility, smart city.

1 Introduction

Smart cards (transport cards) are popular payment tool in many countries (Fig. 1).

Typically, it is a pre-paid card. Users can purchase cards with some initial credits and any later buy a new card as well as buy new credits for existing card. With such cards, users can pay for transport, parking slots and sometimes other city-related services. In the most cases, pre-paid smart cards are anonymous, so there is no user-related information at all. Of course, each card has got unique ID but this ID is not linked with user data.

Smart Card validators are part of the fare collection system. They are devices that read smart cards and support the fare applications contained on them. So, they are devices for getting credits, recorded on the card [2]. The validator units can be contact, contactless or both, and can be stand-alone or integrated with the on-board computer. They can include some data storage.

This store can be downloaded via USB, Bluetooth, Wi-Fi. The validators can include SIM-cards and support GPRS. Also, they can be directly connected

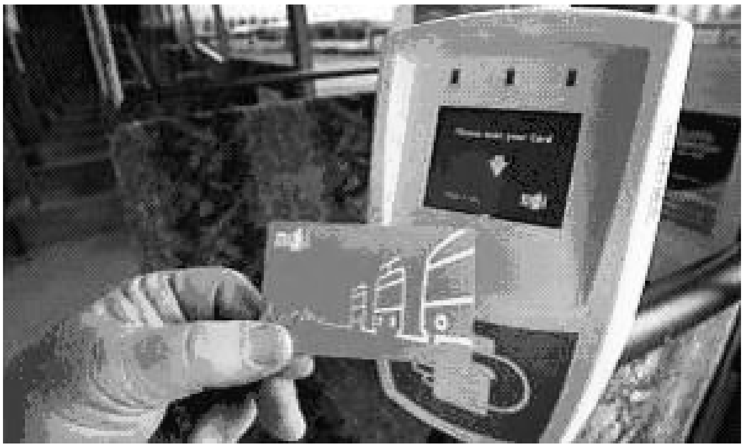

Fig. 1. Smart cards in Melbourne [1].

to some central computer (cloud, external data storage) via wireless LAN or Ethernet connection. So, what is important for us, validators collect logs for transactions. Each transaction contains card ID, validator ID, time stamp. Of course, there is also a sum of credit, but this value is not so important for our tasks.

For smart cities, logs from validators present another dataset which could be used in so-called digital urbanism [3]. In this paper, we would like to discuss data mining models for smart cards logs. The aim of the work is to present (to discover) algorithms and models that are suitable for the processing of transport card data used in Moscow. We will also present one new approach to the processing of transport data, which uses ideas from the processing of web sessions (web logs).
The rest of the paper is organized as follows. In Section II, we discuss data formats. In Section III, we describe data mining models. In this section, we also describe our own proposal for using web statistics analysis tools for transport card logs.

2 On Data Formats

In general, there are two types of payment models: flat fares and distance-based fares [4]. For a flat rate model, users deploy their smart cards on the card reader when entering (only check-in scans are necessary). For the distance-based scheme, riders need to swipe their smart cards twice: when checking-in and checking-out. The separate question is location info. Depends on the device, it could be saved too (for check-in and check-out in a case of distance-based fares).

Key information stored in the database, in general, includes smart card ID, validator ID, transaction time, remaining balance, transaction amount, boarding

stop, alighting stop.
Note that at each particular time point, any particular validator, of course, is placed in a specific bus (subway station, etc.). Accordingly, instead of the ID of the validator, we can use bus ID or a metro station ID (in both cases there could be several validators). This model (Fig. 2), for example, is used in the paper [5].

Field	**Name**
CardID	Unique Identifier of Smart Card
TrmnlID	Metro station ID or bus ID
TrnsctTime	Transaction time
TrnsctType	Transaction type (31 – bus boarding, 21 – metro-swiped-in, 22 – metro-swiped-out

Fig. 2. Smart card records in Shenzhen, China [5].

Note that a particular route in the city is served by the group of the buses. So, if we want to talk about smart cards transactions in terms of routes, we need to maintain a mapping for buses and routes. And because any bus in our model is a set of validators, it is a mapping between validators and routes. This external information with respect to transactions log (as the assignments of buses to the route, for example, are subject to change).
Accordingly, prior to processing, we can combine (join) the mapping with transactional data. This approach is used, for example, in the paper [4] (Fig. 3).

Chain ID	Card ID	Date	First boarding time	Last alighting time	Route sequence	Stop ID sequence
46388399	1000751018309337	20100705	07:08:45	07:47:28	00635 → 10 → 13	99964,99966 → 50258,50167
46388400	1000751018309337	20100705	18:15:24	18:53:10	13 → 10 → 00635	50192,50245 → 100013,100015
46388401	1000751018309337	20100706	07:19:21	08:01:13	00350 → 10 → 13	91267,91269 → 50258,50167
46388402	1000751018309337	20100706	17:56:08	18:49:50	13 → 10 → 00635	50192,50245 → 100013,100015
46388403	1000751018309337	20100707	07:10:43	07:49:21	00635 → 10 → 13	99964,99966 → 50258,50167
46388404	1000751018309337	20100707	18:29:00	19:06:47	13 → 10 → 00350	50192,50245 → 91276,91278
46388405	1000751018309337	20100708	21:13:58	21:40:10	5 → 10	50125,50246
46388406	1000751018309337	20100709	07:16:24	08:03:46	00635 → 10 → 13	99964,99966 → 50258,50167
46388407	1000751018309337	20100709	17:25:00	18:11:59	13 → 10 → 00635	50192,50245 → 100013,100015
46388408	1000751018309337	20100709	18:30:31	NULL	00031	NULL

Fig. 3. Beijing Comprehensive Transport Survey [4].

In [6], authors use the following data format (Canada, Quebec):

- Date and time of the boarding transaction;
- Card number and fare type;
- Route number and direction;
- Vehicle and driver numbers;
- Stop number at boarding

There are several other papers devoted to smart cards logs and additional attributes [7-10].

In our work, we are interested in the data representation with a single mark (check-in only). This form is a typical solution for smart cards used in Moscow, Russia (Troika card).

A separate question is location information associated with check-in marks. Technically, it could be implemented. On practice, requesting location information may cause delays in check-in (check-out) processes. So, it is rather the rare decision. But for any particular route, we can assume that the bus, for example, is on time schedule and assign (approximately) location according to time stamps.
Note, that we do not discuss here data engineering and by this reason do not touch practical questions related to smart cards databases.

3 On mining smart cards data

In this section, we would like to discuss data processing issues. From the literature analysis, the following problems were identified: traffic patterns, trips generation, and routes-based studies.

A. Transit patterns

A good review of transit smart card data processing provided in the paper [11]. In the paper [12], authors support the idea that individual daily movement uses only a small number of movement patterns. As per their research, these patterns, termed motifs, appear stable in many different cities. And the detection of motifs is rule-based [13]. Suppose, C is some central node (bus stop, etc.). Let T(n) denote a cyclical tour with n locations starting at C and ending there. Then a construction rule is a set of cycles in the motif starting at C. There are four predominant rules for the most frequent motifs:

I) T(1), T(n-2),
II) T(n-1),
III) T(2), T(n-3),
IV) T(1), T(1), T(n-3).

In plain English, the construction rule (I) consists of a long tour plus a short tour visiting an additional node not contained in the long tour, rule (II) presents some tour with no other nodes visited, rule (III) has one long and one short tour without joint locations, rule, and rule (IV) finally features two back and forth trips and a long cycle. As per [13], the most frequent motifs fall into these four construction rules.

In the paper [14], authors study the so-called spatial variability of transit use. It is examined through the enumeration of all the bus stops used for boarding. Then, the frequency of use of the bus stops is studied, in order to express a level of regularity. It allows detecting the number of bus stops which cover the main proportion of observed transit paths. And as the last step, a k-mean algorithm is used to partition the data set into a predefined number of clusters for the different types of smart cards (students, adult, etc.).

In the paper [4], authors use the Density-based Spatial Clustering of Applications with Noise (DBSCAN) algorithm for the identified trip chains in order to detect historical travel patterns for transit riders. A trip chain is defined as a series of trips made by a traveler on a daily basis. We can assume some fixed temporal threshold to link several smart card transaction records into a trip chain.
The following algorithm is used for clustering:
1. Sort the trip chain records and add visited/non-visited flag to the each record
2. Randomly select an unvisited trip. Flag this record as visited and form a cluster for this record
3. Check the boarding time difference between unvisited records and the last visited record. If the difference is greater than some predefined interval (it is 1 hour in that paper), repeat Step 2. This predefined interval let us separate connected trips from new independent trips. The same idea with predefined time weve used, for example, in our paper [15].
4. Check the spatial relationship between unvisited records and the last visited record. If a spatial relationship exists within some predefined radius (it is 200 meters in the above-mentioned paper), then this record is included in the cluster formed in Step 2 and flagged as visited.
5. If the total count of trips in cluster is less than some predefined threshold (it is 3 in in the above-mentioned paper), drop the cluster and mark records as noised.
6. Continue to process the unvisited records from Step2 through Step 5 until all the records are flagged as visited or noised.
7. The number of total clusters corresponds to the number of typical trip chains per day. The recurring route, boarding/alighting stops, and timings can be acquired by counting the most frequent pattern within each cluster.

And the following set of features could be used for the detecting regularity [4]:
A number of travel days. The more days a transit rider travels, the more likely it is that he is a frequent transit rider.
A number of similar first boarding times. Boarding time represents a riders temporal characteristics. If a rider begins an own trip at a similar time of day every weekday, then he is more likely to be a regular transit rider.
A number of similar route sequences. Route sequence represents a general spatial pattern for a rider. The number of similar route sequences followed during the week may indicate a repetitive travel pattern (e.g., home office).
A number of similar stop ID (end points) sequences. The stop ID sequence may contain detailed spatial similarity information. For example, two formally different stop IDs might be spatially adjacent.

Another example of clustering is presented in papers [16, 17]. In their study, passenger heterogeneity is investigated based on a longitudinal representation of each users multi-week activity sequence derived from smart card data. Authors propose a methodology leveraging this representation to identify clusters of users with similar activity sequence structure. In general, there are four categories for travelers: non-exclusive commuters, exclusive commuters, non-commuter residents, and leisure travelers [18]. In [17], authors propose a simple and effective algorithm for sampling selection. In order to identify users whose activity pattern can be inferred more completely from smart card data, cards were clustered based on their level of public transport usage. Each card is characterized by the number of days it was observed traveling over the 29-day analysis period, and by the spread of days between the first and last day it is observed. Using just these two variables and classical k-means clustering, 3 user clusters were identified: a group of non-recurrent users who are seen traveling few days concentrated over a short period, a group of occasional users who travel on few days spread over the analysis period and a group of frequent users who travel on many days spanning most of the analysis period. This simple approach is, probably, the best way for getting a quick snapshot for transit info in smart cards data.

In the paper [19] authors describe a transit passenger segmentation method based on a two-step DBSCAN algorithm. They use also k-means algorithm to distinguish frequent and infrequent transit users based on the number of travel days and journeys made [20].

B. Trip generation

It is about extracting (discovering) trips in log files. Summarizing the papers, the following technique could be used for extracting trips from cards data.

A trip is composed of a sequence of activities for a particular purpose. In our case, it is a sequence of transport card transactions. Classically, time thresholds are adopted to link these transactions. As the boundary data, we can use the maximum transfer times for the different types of travelers activities. Activity here is a type of trip. E.g., it could be metro (subway) only, metro and bus, bus and metro, etc. So, the transaction time differences that fall within the maximum transfer times could be extracted.

Some of the authors also suggested using 95th percentile transaction time difference for each transfer activity as the time threshold to form up a complete trip [11].

Now, for any individual transport card (for any individual traveler), if the transaction time difference for two consecutive card records exceeds any of the thresholds, then a trip is separated. For individual's trips, we could merge also multiple trips with short duration times as a single trip.

After that, we can reasonably assume, for example, that for any individual traveler the first trip (every day) is a home-to-work commute, and the last trip is a work-to-home travel. So, this pair of trips could be used for detecting activity patterns.

The regularity of commuting should be spatially and temporally measured [11]. The temporal patterns could be detected by the similarity of departure time and the number of traveling days. For spatial patterns, we can calculate the frequency of the most visited stops as well as the number of recurring travels on similar routes or lines.

C. Routes-based study

For this research, we can talk about models that are suitable for single swipe (check-in) only. It is the most interesting direction for our goal. Here we can list the following suggestions.

At the first hand, we should calculate the distribution of check-ins by boarding places. As the source information for this, we can use boarding counts for every weekday with 1-hour step, for example.
Obtaining such distribution lets us compare routes. Secondly, it lets us detect changes in loading (amount of travelers) for the particular days. It is about a statistically significant difference. And changes in the day distribution should be linked to some real life events (e.g., opening new mall, closing the offices, etc.). Also, we will be able to build time series for the route related check-ins (e.g., 30 minutes summary of check-ins for the particular bus). And in these time series, we can see outlines/trends.

In paper [21], authors describe another idea for the routes-based study: a Markov model to study stochastic behavior for travelers in the day-to-day route choice adjustment process. The model is described by two components: how often a passenger reconsiders a route choice (in other words, it is a route switching rate), and the probability to take a certain route (in other words, it is a route choice probability). All travelers make route choice today only depending on the limited road information available from yesterday (Markov's rule). It is illustrated in Fig 4.

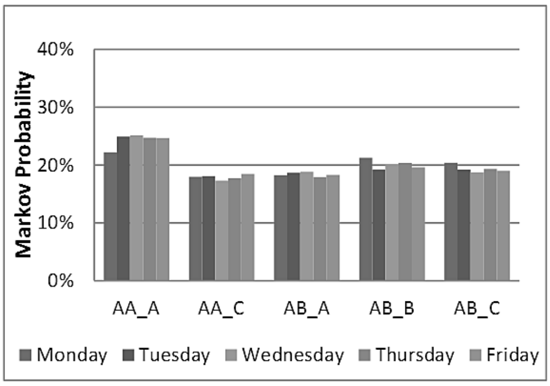

Fig. 4. Routes and probabilities during a week [21].

The next idea (it is our own proposal), follows the model used in our papers [22-24]. We can present (simulate) our transport system as a web statistics system. Cards swipe device (validator) is a web page. The set of such devices on a particular route is a web site. Cards ID is an analogue for visitor's IP address.

This approach should have a clean path for the visualization. We should be able to use many existing web statistics visualization and data mining applications (Fig. 5).

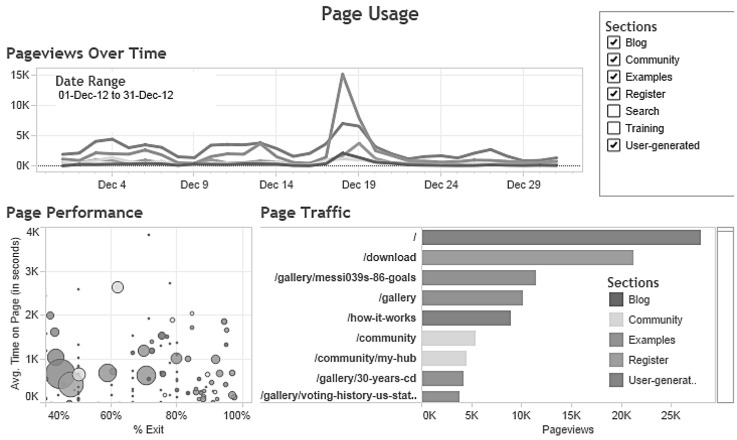

Fig. 5. Web statistics dashboard. A Pageview now is an individual card's usage, etc.

For testing, we can take a transport data set, build a mapping Card ID - IP address, present our dataset as a standard web log (IP address corresponds to Card ID) and use any available web log analyzer software as a proof of concept (in our case, we used the free Deep Log Analyzer).

Also, we can use web mining algorithms for the research and visualization [25]. It is so-called web log mining. There are several techniques that could be reused here [26]. For example, association rules could be used in order to discover the routes which are used together. It could help to discover movement patterns. The discovery of association rules in Web logs discussed in [27] for example. Other methods are sequence mining, topology patterns, Markov predictors, Ngram models.

Originally, the sequence mining for web logs can be used for discovering the Web pages which are accessed immediately after another. For our simulation, it will be a sequence of validators. It means an effective route. The typical examples for such methods are presented in the paper [28].

According to sequence mining, we can preprocess our log and create a database of sequences (routes). In this database, we can collect for the each card (cards ID, originally for the each IP address) the sequence of the visited validators (originally, it is the sequence of web accesses). Technically, such collection could be presented as some key-value store. A key here is a card's ID.

For the next explanation, we follow to the definitions in [29]. The event is a card's usage on some validator (originally, it is a visit for a web page). Given a set of events E, the route sequence (originally - access sequence) S can be represented as e_1 e_2 ... e_n sequence Here e_i is some validator (originally some web page). That means the access sequence is composed of a series of events, which are members of event set E (the whole set of validators). The repetitions of events are allowed in a sequence (e.g. a citizen used the same bus twice). A route (access sequence) R = e_1 e_2 ... e_l is called a sub-sequence of a route R1 = e_1 e_2 ... e_n, and R1 is a super-sequence of R, if and only if for every event e_j in R, there is an equal event e_k in R1, while the order that events occurred in R should follow the order of events in R1.

A frequent pattern is a route (originally, it is an access sequence) to be discovered during the mining process. The frequency here could be defined via so-called "support". The support of pattern S in database of sequences is defined as the number of sequences S_i , which contains the subsequence S, divided by a number of transactions in the whole database. Although events (card's transactions) can be repeated in a route (in an access sequence), a pattern can get at most one support count contribution from one access sequence. So, any frequent pattern should have a support that is higher than minimum support.

The minimum support for sequential pattern mining is the percentage value between 0 and 1. It could be set empirically to identify the frequent sequence. The problem of route mining (originally, web usage mining) is that of finding all patterns which have supports greater than some predefined minimum support threshold *d*.

There are two basic techniques for mining sequential patterns from web logs fall: Apriori and non-Apriori [30]. Apriori algorithm uses the fact that any super-pattern of non-frequent patterns is not frequent. The non-Apriori algorithms divide the original database into smaller partitions and solve them recursively. The most popular, according to the academic papers, is a non-Apriori method called WAP-tree mining [29]. This approach stores the web access patterns in a compact prefix tree (it is called WAP-tree). Since non-Apriori algorithms did not need to scan the database multiple times, they should be faster.

But we can see some drawbacks on this way too. Although detected routes (sequential patterns) include the order of the events, the time between the individual events is unknown. So, we should think about finding sequential patterns with time intervals. For example, authors in [31] describe the time-interval sequential pattern, which includes not only the order of the events but also the time intervals between successive events.

A time-interval sequential pattern provides more valuable information than a conventional sequential pattern. Consider the transport business and mobility as an example, with the assistance of the time-interval sequential pattern, the smart city not only learns the mobility patterns, but also links them with time of the day. There are several papers devoted to time-interval sequential patterns [32, 33].

With time-interval sequences the definitions for sub-sequence (super-sequence) should be changed. Now we should include not only the events inclusions, but the inclusion for time-intervals also. The modifications for Apriori algorithms in case of time-interval sequences are presented in the paper [30], for example.

As the next issue, we should mention stream processing for sequences data mining [34]. In general, a data stream processing has to satisfy the several constraints: new elements are generated continuously and should be processed as soon as possible, the data can be examined only once, memory usage is restricted. The final idea is to proceed data in real-time. For smart cities mobility, for example, it could be important to detect usage mode outlines in real-time. In the paper [34], authors propose the algorithm for sequence mining in data streams, which is based on sequences alignment for mining approximate sequential patterns in data streams. The similar task is studied by authors in [35] and [36].

4 Conclusion

In this paper, we present data models for transport cards data processing. We discuss transit patterns detection (it is rule based), trips discovering (it is clustering in the various forms, e.g. modified DBSCAN), routes-based studies (statistics, Markov's model). Additionally, we present a new approach where transport cards data log could be processed like a web server log. It opens the possibility to use many existing methods and tools for web statistics. For testing this approach, we tried to re-code one existing transport cards dataset [37] as a web log and used a free web log analyzer as a proof of concept.

References

1. Myki Travel Smart Card http://australiaforeveryone.com.au/melbourne/getting-around.html Retrieved: Feb, 2017
2. Toolkit on Intelligent Transport Systems for Urban Transport https://www.ssatp.org/sites/ssatp/files/publications/Toolkits /ITS%20Toolkit %20content /its-technologies/ electronic- fare-collection/smart-card-validator-and-display.html Retrieved: Feb, 2017
3. Namiot, Dmitry, and Elena Zubareva. "Data-driven Cities." International Journal of Open Information Technologies 4.12 (2016): 79-85.
4. Ma, Xiaolei, et al. "Mining smart card data for transit riders travel patterns." Transportation Research Part C: Emerging Technologies 36 (2013): 1-12.
5. Zhao, Juanjuan, et al. "Understanding temporal and spatial travel patterns of individual passengers by mining smart card data." Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on. IEEE, 2014.
6. Agard, Bruno, V. Partovi Nia, and Martin Trpanier. "Assessing public transport travel behaviour from smart card data with advanced data mining techniques." World Conference on Transport Research. Vol. 13. 2013.
7. Dinant, Jean-Marc, and Ewout Keuleers. "Data protection: multi-application smart cards: the use of global unique identifiers for cross-profiling purposes-Part II: towards a privacy enhancing smart card engineering." Computer Law & Security Review 20.1 (2004): 22-28.
8. Trepanier, Martin, Khandker MN Habib, and Catherine Morency. "Are transit users loyal? Revelations from a hazard model based on smart card data." Canadian Journal of Civil Engineering 39.6 (2012): 610-618.

9. Lee, Sanggu, and Mark D. Hickman. "Travel pattern analysis using smart card data of regular users." Transportation Research Board 90th Annual Meeting. No. 11-4258. 2011.
10. Min, Yun-Hong, et al. "Mining missing train logs from Smart Card data." Transportation Research Part C: Emerging Technologies 63 (2016): 170-181.
11. Ma, Xiaolei, et al. "Understanding commuting patterns using transit smart card data." Journal of Transport Geography 58 (2017): 135-145.
12. Schneider, Christian M., et al. "Daily travel behavior: Lessons from a week-long survey for the extraction of human mobility motifs related information." Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing. ACM, 2013.
13. Schneider, Christian M., et al. "Unravelling daily human mobility motifs." Journal of The Royal Society Interface 10.84 (2013): 20130246.
14. Morency, Catherine, Martin Trepanier, and Bruno Agard. "Measuring transit use variability with smart-card data." Transport Policy 14.3 (2007): 193-203.
15. Namiot, Dmitry. "Mining groups of mobile users." International Journal of Wireless and Mobile Computing 9.3 (2015): 211-217.
16. Goulet Langlois, Gabriel, Haris N. Koutsopoulos, and Jinhua Zhao. "Clustering the Multi-week Activity Sequences of Public Transport Users." Transportation Research Board 95th Annual Meeting. No. 16-6162. 2016.
17. Langlois, Gabriel Goulet, Haris N. Koutsopoulos, and Jinhua Zhao. "Inferring patterns in the multi-week activity sequences of public transport users." Transportation Research Part C: Emerging Technologies 64 (2016): 1-16.
18. Ortega-Tong, M.A., 2013. Classification of Londons Public Transport Users Using Smart Card Data, Thesis MIT. http://dspace.mit.edu/handle/1721.1/82844 Retrieved: Mar, 2017
19. Bhaskar, Ashish, and Edward Chung. "Passenger segmentation using smart card data." IEEE Transactions on intelligent transportation systems 16.3 (2015): 1537-1548.
20. Kieu, Le-Minh, Ashish Bhaskar, and Edward Chung. "A modified Density-Based Scanning Algorithm with Noise for spatial travel pattern analysis from Smart Card AFC data." Transportation Research Part C: Emerging Technologies 58 (2015): 193-207.
21. Kurauchi, Fumitaka, et al. "Variability of commuters bus line choice: an analysis of oyster card data." Public Transport 6.1-2 (2014): 21-34.
22. Namiot, Dmitry, and Manfred Sneps-Sneppe. "On the analysis of statistics of mobile visitors." Automatic Control and Computer Sciences 48.3 (2014): 150-158.
23. Namiot, Dmitry. "Mining Relationships in Proximity Movements." Applied Mathematical Sciences 7.144 (2013): 7173-7177.
24. Namiot, Dmitry, and Manfred Sneps-Sneppe. "Customized check-in procedures." Smart Spaces and Next Generation Wired/Wireless Networking (2011): 160-164.
25. Deshmukh, Sana M., and Krishnakant P. Adhiya. "A Review on Finding Users Navigation Behavior Using Web Mining Algorithm." (2016).
26. Ivncsy, Renta, and Istvn Vajk. "Frequent pattern mining in web log data." Acta Polytechnica Hungarica 3.1 (2006): 77-90.
27. Huang, Xiangji, and Aijun An. "Discovery of interesting association rules from livelink web log data." Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on. IEEE, 2002.
28. J. Pei, J. Han, B. Mortazavi-Asl, and H. Zhu, Mining access patterns efficiently from web logs, in PADKK 00: Proceedings of the 4th PacificAsia Conference on

Knowledge Discovery and Data Mining, Current Issues and New Applications. London, UK: Springer-Verlag, 2000, pp.396-407
29. Ezeife, Christie I., and Yi Lu. "Mining web log sequential patterns with position coded pre-order linked wap-tree." Data Mining and Knowledge Discovery 10.1 (2005): 5-38.
30. Agrawal, Rakesh, and Ramakrishnan Srikant. "Mining sequential patterns." Data Engineering, 1995. Proceedings of the Eleventh International Conference on. IEEE, 1995.
31. Chen, Yen-Liang, Mei-Ching Chiang, and Ming-Tat Ko. "Discovering time-interval sequential patterns in sequence databases." Expert Systems with Applications 25.3 (2003): 343-354.
32. Chang, Joong Hyuk. "Mining weighted sequential patterns in a sequence database with a time-interval weight." Knowledge-Based Systems 24.1 (2011): 1-9.
33. Chen, Yen-Liang, and TC-K. Huang. "Discovering fuzzy time-interval sequential patterns in sequence databases." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 35.5 (2005): 959-972.
34. Marascu, Alice, and Florent Masseglia. "Mining sequential patterns from temporal streaming data." Proc. of the 1st ECML/PKDD Workshop on Mining Spatio-Temporal Data (MSTD 2005). 2005.
35. Zihayat, Morteza, et al. "Mining high utility sequential patterns from evolving data streams." Proceedings of the ASE BigData & SocialInformatics 2015. ACM, 2015.
36. Jiang, Nan, and Le Gruenwald. "Research issues in data stream association rule mining." ACM Sigmod Record 35.1 (2006): 14-19.
37. Opal Trips - Bus https://opendata.transport.nsw.gov.au/dataset/opal-trips-bus Retrieved: May, 2017