

# Contextualised Service Delivery in the Internet of Things

## Parking Recommender for Smart Cities

Ali Yavari <sup>\*</sup>, Prem Prakash Jayaraman <sup>†</sup>, Dimitrios Georgakopoulos <sup>†</sup>

<sup>\*</sup>RMIT University, Melbourne, Australia

<sup>†</sup>Swinburne University of Technology, Melbourne, Australia

ali.yavari@rmit.edu.au, pjayaraman@swin.edu.au, dgeorgakopoulos@swin.edu.au

**Abstract**— The Internet of Things (IoT) plays an important role in the development of smart cities. In this paper we focus on the development of IoT-based smart services for solving urban problems that involve IoT-enabled Observation, Orientation, Decision, and Action (OODA) loops. We also focus on how to efficiently support such OODA loops in situations where such loops involve internet-scale data. More specifically, IoT supports Observation via the discovery of sensors and the integration of their data. It supports Orientation via a contextualisation process that refines such data to include only those that are relevant to the situation and/or activities of each specific individual or group. As IoT contextualisation potentially involves internet-scale data, performing this process efficiently allows for fast decision making, and this in turn permits carrying out a timely Action. In this paper we propose an approach and related techniques for performing internet-scale data contextualisation. In particular, we propose IoT-based contextualisation techniques that effectively consider the entire range of data that is being collected in smart cities and use such data to provide hyper-personalised information to each user, i.e., information that best suits the context of each user in the Smart City. We exemplify the proposed contextualisation solution in a smart parking space recommender application/service, and provide an experimental evaluation of this service to illustrate the benefits of our solution.

**Index Terms**—Internet of Things; Context, Contextualisation; Scalable Recommendation; Big Data;

### I. INTRODUCTION

The Internet of Things (IoT) [1] is expanding rapidly (by 2020 there will be more than 50 billion IoT devices on the Internet [2]), and it is transforming major cities around the world by enabling the introduction of smart services [3] that help improve transportation, enhance health and well-being, provide green energy and reduce emissions, increase public safety, and enable better shopping experience. Such smart services lead to better quality of living, and they are currently fuelling the development of Smart Cities around the world.

In this paper we focus on Smart City services that involve *Observation* - collecting Smart City information, *Orientation* - contextualising such information to personal or group needs, *Decision* - making appropriate decisions, and *Action* - act based on the decisions made.

The primary focus of this paper is efficient support for such OODA loops, and in particular by providing solutions for the contextualisation of smart city data that is efficient, scalable, and effectively considers each user's context. We define contextualisation as a process of identifying the data relevant to an entity (in this case a person or a city) based on the entity's contextual information [4].

To explain contextualisation further, consider a user looking for a parking space in a Smart City. Currently there are many parking recommendation solutions that direct drivers to empty parking spaces, and some even provide an estimate of average waiting time to park. Unlike such existing solutions that only consider the parking information (e.g., available parking spaces and their location in the parking), the contextualisation approach we propose in this paper takes into account each driver's context that may include: the driver's preferences (e.g., park in a covered parking space), driving experience (e.g., avoid narrow parking spaces), the car's location (e.g., collected from the driver's smart phone), the vehicle's properties (e.g., vehicle type, length, height, etc.), and other parking information provided (e.g., the properties of parking spaces, such as shaded, covered, etc.).

The principle challenge in delivering even such basic contextualised services lies in the ability to contextualise internet-scale IoT data (based on available contextual information [5]) and do this efficiently and in near real-time. To this end, this paper proposes:

- An approach to represent and contextualise data from IoT devices.
- A technique for efficiently querying contextualised IoT data.
- A sample smart parking space recommender application that illustrates how such a contextualisation mechanism can be employed.
- An experimental evaluation of the proposed contextualisation technique using synthetic data generated from Melbourne city datasets<sup>1</sup>.

---

<sup>1</sup> <https://data.melbourne.vic.gov.au/>

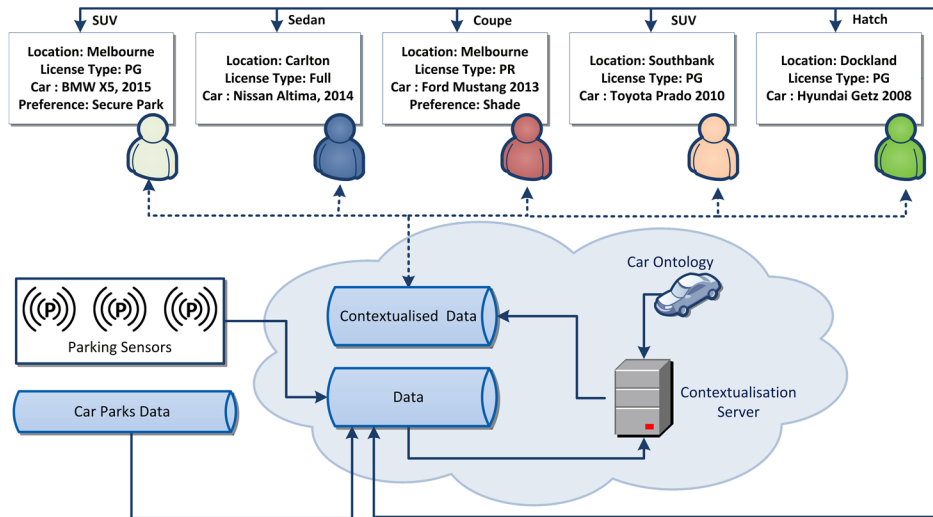


Figure 1 - Contextualisation Architecture

## II. CONTEXTUALISATION OF IoT SERVICES

*Context* is information (1) about all entities (i.e., persons, places, or things) that are relevant to a given IoT service, and (2) it can be used to contextualize IoT data.

Contextualisation can be used to effectively reduce the complexity of data processing and the amount of reasoning required for decision making within one or more IoT services. Contextualization of Internet-scale IoT data is a hard problem and requires algorithms that can process large volumes of heterogeneous data arriving at very high velocity. A closely related problem that also needs to be solved is mapping and scheduling contextualisation tasks on high performance data processing resources, such as cloud services, to provide on demand stability.

Contextualisation of IoT [4] services involves the following:

1. *Context Collection (and deduction)* - User context information can be collected from user's smart phone, wearable devices, or manually provided by the user. Moreover, cloud services can help to deduce new context information from the collected context. For example, an IoT service equipped with car ontology and related data can deduce the user's car size by knowing the car manufacturer and model.
2. *Contextualisation* – The contextualisation of IoT data we propose in this paper is based on two main operations including *Contextual Filter* and *Contextual Aggregation*. Contextual Filter, filtrates the data originating from IoT devices and services based on a given current context. For example, data received from a parking sensor located in a particular location (e.g., a parking space in a Melbourne suburb) can be excluded from further data processing and related queries whenever there is no particular user in that particular location. Contextual Aggregation, combine potentially filtered data based on the contextual similarities and relevance. For example, if all the current users searching for parking spots in Melbourne have SUV

vehicles then we can aggregate SUV and Melbourne and treat it as a new context.

3. *Dissemination of the contextualised data* – Delivery of contextualised data to a service must consider context of the target service.

## III. CONTEXTUALISED SMART PARKING RECOMMENDER

In this section, we present the architecture [Figure 1] of the *smart parking recommender* application that incorporates the proposed contextualisation and processing of IoT data. We use the smart parking recommender to exemplify the need to contextualise IoT data and the advantages it provides. Parking is becoming an expensive resource in any major city and finding the most appropriate parking space is always regarded as a challenge. Existing solutions [6]–[9] focus on the following approaches to make parking recommendations:

- Using IoT data from parking facilities (e.g., from their parking sensors).
- Using vehicle-provided data (e.g., from on board accelerometers) to compute empty parking spots based on each vehicle's kinetic state (i.e., moving, stopped, etc.) and location (Such information is typically crowdsourced [10]).
- Using machine learning models to predict queue length for parking in shopping centres.

Most of these approaches do not consider further context(s). Unlike such existing approaches, our solution provides the following:

- Allows IoT services to take into consideration multiple contexts originating from each driver, his/her car, and a Smart City section of interest (e.g., the parking in a shopping centre),
- Permits instant response to common parking queries by continuously (re) contextualising, and
- Combines the contexts of multiple drivers to efficiently answer parking queries.

#### IV. PROOF-OF-CONCEPT IMPLEMENTATION AND EVALUATION

To implement the smart parking recommender, we employ semantic web standards such as RDF<sup>2</sup> and Linked Data<sup>3</sup> for modelling and integrating IoT data. The RDF models and stores data as N-triples. A triple is a statement that describes data in form of <Subject, Predicate, Object>. Subject is the identifier of the entity that the data is describing; Object is the description of the Subject in terms of the relation described in Predicate. For example a triple such as <Parking1, hasType, closed>, describes that Parking1 is a closed parking. By contextualisation we convert relevant N-triples<sup>4</sup> to N-quads<sup>5</sup> by adding another field to the triples describing the relevant context.

To perform a Contextual Filter operation, we initially create a set of available contexts from drivers in each particular city area. Next, we assigned a *unique context identifier* (a prime number in this paper) to each context. Finally, we search for available parking spots in the selected city area, and we convert the resulting triples from N-triples to N-quads by adding the unique context identifier to the N-triples. If the triple is already N-quad (indicating that the triple is already contextualised for other contexts) we update the context identifier field by multiplying the prime number of the context identifier with the number the triple already has. For instance, if a secure parking such as Parking1 has a context with identifier 11, we convert the triple of Parking1 as follows:

< Parking1, hasType, closed> → < Parking1, hasType, closed, 11 >

Similarly, if Parking2 has already another context with identifier 7, we convert its triple as follows:

< Parking1, hasType, closed, 7> → < Parking1, hasType, closed, 77 >

After performing Contextual Filter operation, all triples that remain in form of N-triples are not relevant. To perform Contextual Aggregation, we merge multiple contexts and we assign a new prime number. For example, if in one particular location all the drivers are searching for parking spots with one context followed by another then these two contexts aggregable.

##### A. Data Set

We generated a synthetic dataset representing 50,000 users searching for parking spots in Melbourne Central Business District and surrounding suburbs as follows: (1) We used a real parking dataset provided by City of Melbourne to generate the dataset of the parking spots [Figure 2], and (2) we have added synthetic parking sizes and descriptions to (1).

Finally, we considered the following information:

##### Diver information

- License Type (Full for experienced, P Green for 1year experience, Red P for beginner).
- Car Specification (e.g., BMW X5 2015)
- Preferences (e.g., secure car park, shady car park, etc.)

- Location (Suburbs)
- Parking information
  - Location (suburbs)
  - Type (closed, opened)
  - Space (number of available spots)
  - Acceptable car body types (SUV, hatch, sedan etc.)
  - Weather condition (sunny, rainy, snowy and so forth)

##### B. Implementation

We implemented the proposed contextualisation methodology in Java. We used Apache Jena<sup>6</sup>, which is a widely used semantic triple store, and the ARQ [11] Query Engine for SPARQL to query the contextualised data. The parking dataset is converted to N-Quads after matching relevant contexts. For evaluation, we deployed the system on an Amazon EC2 [12] “M3 General Purpose” instance with 30 GB ram and 8 vCPU.

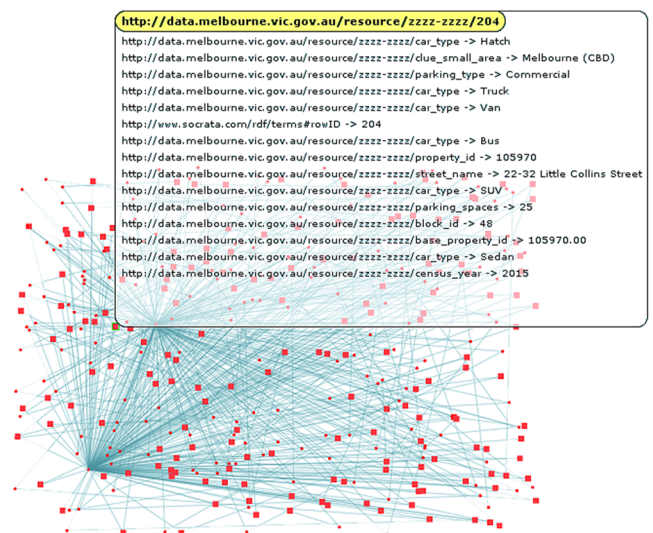


Figure 2 - Parking Spots Graph

##### C. Evaluation

The objective of the evaluation is to validate the following hypothesis: *Contextualisation helps process driver queries faster and can handle Internet-scale IoT data sets*. Since we keep track of shared context among multiple users, each individual query is independent of another (different from typical query caching mechanisms). For instance, a user may share his contextual preference e.g. shaded spot with one user while another contextual preference e.g. acceptable car type is shared with another user.

Figure 3, presents the percentage of shared contexts among 85 random users. As in many real-world situations, the shared contexts among users increase rapidly as the users increase, and this also drives a similar increase in unique contexts. These results also illustrate the fact that the number of shared contexts among users of IoT services, such as the smart parking locator service, is generally larger than the number of unique contexts.

Figure 4, shows the percentage of active user’s contexts relevant to the parking data at any given moment. For example,

<sup>2</sup> <https://www.w3.org/RDF/>

<sup>3</sup> <http://linkeddata.org>

<sup>4</sup> <https://www.w3.org/TR/n-triples/>

<sup>5</sup> <https://www.w3.org/TR/n-quads/>

<sup>6</sup> <https://jena.apache.org/>

with 85 driver contexts in the system and 3368 parking data triples, the total number of users whose contexts match the parking data triples (e.g., finding an enclosed parking in Melbourne CBD) is 27.5%.

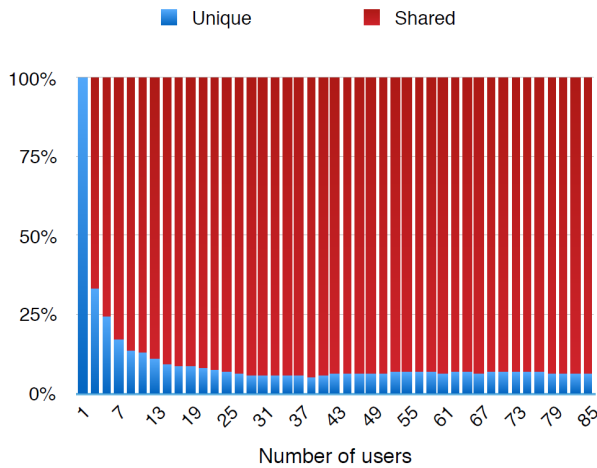


Figure 3 – Percentage of unique contexts

In Figure 5, we present the total query processing time with and without contextualisation process. Please note that as the similarity among context preferences of drivers increase (as in Figure 3), contextualisation can greatly help by significantly reducing the query processing time. This is due to the fact that our contextualisation solution has already resolved the queries of users with similar contexts. Hence in order to satisfy a request from a new user whose contexts match existing users in the system, contextualisation simply maps responses of existing user queries by matching the relevant contexts to the new requests.

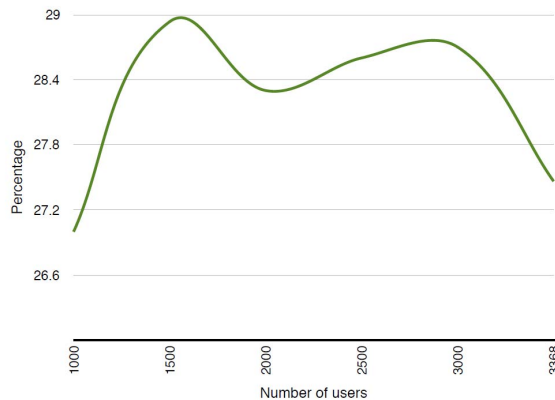


Figure 4 - Percentage of the relevant data

Finally, in Figure 6, we present the total time for the contextualisation process which includes converting drivers' and location (Smart City) information to relevant N-Quads and resolving driver queries. As the result indicates, after 10,000 users the process doesn't increase significantly due to the fact that shared contexts are the major contexts among all the users. This result validates our hypothesis, i.e., that contextualisation performs much better when the amount of data increases towards the Internet scale.

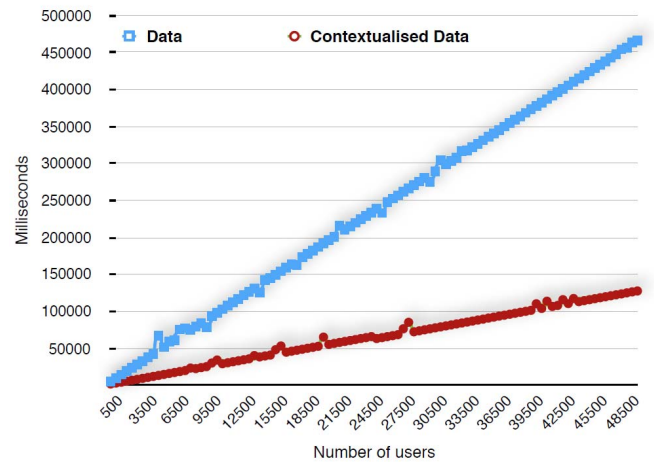


Figure 5 - Query Time Comparison

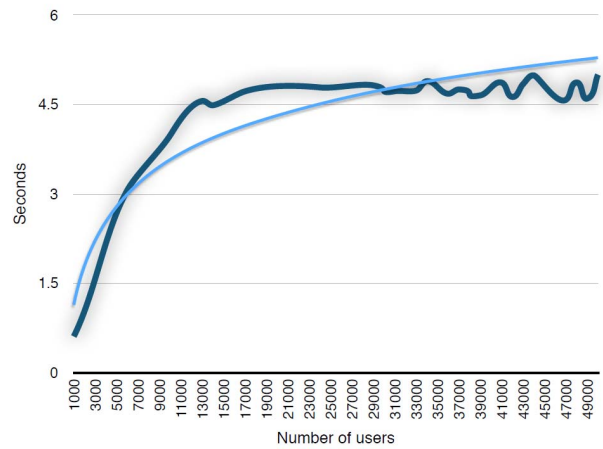


Figure 6 – Contextualisation Processing Time

## V. RELATED WORK

In [13], [14], the authors exemplify the current infrastructure embedded in smart cities that allows the development of smart parking solutions, considered a major problem in developing and developed cities. With the advent of technologies such as IoT, Cloud Computing and Big data analytics tools, there has been a number of recent works focusing on smart parking management approaches. Most of these approaches focus on the following two dimensions for estimating parking availability: (1) Development of new IoT devices for different parking situations (e.g. garages, shopping centres, etc.) [6], [9], [15]–[18], and (2) developing algorithms and methodologies in particular machine learning and queuing theory approaches [7], [8], [19]–[23].

In [9], the authors propose a VANET-based smart parking scheme using vehicular communications through road-based infrastructure (road side units). It provides real-time parking navigation in large spaces by ensuring user privacy. In [8], [17] the authors describe a smartphone-based crowd-sensing approach to provide parking place recommendation. They build statistical models of sensor data obtained from mobile smart phones to detect events, such as parking into a spot, driving out

of a spot, breaking (e.g. using accelerometer data), etc. In [15], the authors propose a parking estimation system using Arduino-based ultrasound sensors. In [18], the authors make use of existing IoT infrastructure deployed in parking spaces to provide a cloud-based parking space finder service. The focus of this work is on the middleware required to deliver the parking recommendation service. Similarly, in [6] the authors present an architecture for parking management in smart cities. This system makes use of custom-developed IoT hardware in particular retractable bollard, magnetic loop to detect occupancy, RFID reader and ZigBee-based wireless transceiver. In [23], the authors propose a vision-sensor (cameras)-based approach for estimating and recommending empty parking spots.

In [16], the authors propose a technique to predict parking spaces. Their approach is to identify key features that best describe parking availability and use various machine learning algorithms such as regression trees, support vector regression to determine the strength of these algorithms. In [19], the authors use anomaly detection and clustering to detect interesting patterns, such as heavily used parking spots, and to compare pricing vs. security. In [22], the authors employ mixed-integer linear programming to solve the same problem. Their solution reduces the overall time required to find a parking spot.

The solution proposed in [21], uses an online demand management model to provide parking spot recommendation to electric vehicles, while [20] proposes the use of contextual information from user and smart parking infrastructure to make more precise recommendations. However, this work does describe well how the context is represented and used.

Most of the above solutions focus on using IoT data to provide recommendations with either no or very little consideration for contexts available from drivers or the Smart City. Moreover, most of these approaches are tailor made to work for closed garages or certain shopping malls. There is no consistent way of representing the parking and driver data, and they all use different architectures. On the contrary, our proposed approach provides a unified solution for representing IoT data obtained from sensors, cars, wearables, smart phones, etc., and also to efficiently query all such data. To the best of our knowledge, this is the first time contextualisation is employed to exploit IoT data in order to provide a hyper-personalised service.

## VI. CONCLUSION

Scalable and real-time contextualisation of IoT data has the potential to significantly improve data processing for large scale IoT applications in Smart Cities. In this paper we proposed an approach to contextualise and query Internet scale IoT data and we exemplify the approach via a smart parking space recommender application for Smart Cities. The experimental scenario in this paper illustrates that contextualisation of IoT data reduces query times for IoT services (such as a smart parking space recommender) by more than 3 times in comparison with a situation where the query is contextualisation agnostic.

## REFERENCES

- [1] D. Georgakopoulos and P. P. Jayaraman, "Internet of things: from internet scale sensing to smart services," *Computing*, vol. 98, no. 10, pp. 1041–1058, 2016.
- [2] J. Rivera and R. van der Meulen, "Gartner Says The Internet Of Things Installed Base will grow to 26 Billion Units By 2020," *Stamford Conn Dec.*, vol. 12, 2013.
- [3] L. Sanchez *et al.*, "SmartSantander: IoT experimentation over a smart city testbed," *Comput. Netw.*, vol. 61, pp. 217–238, 2014.
- [4] A. Yavari, P. P. Jayaraman, D. Georgakopoulos, and S. Nepal, "ConTaaS: An Approach to Internet-Scale Contextualisation for Developing Efficient Internet of Things Applications," in *Hawaii International Conference on System Sciences HICSS-50*, 2017.
- [5] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "CA4IOT: Context Awareness for Internet of Things," in *Green Computing and Communications (GreenCom), 2012 IEEE International Conference on*, 2012, pp. 775–782.
- [6] R. E. Barone, T. Giuffrè, S. M. Siniscalchi, M. A. Morgano, and G. Tesoriere, "Architecture for parking management in smart cities," *IET Intell. Transp. Syst.*, vol. 8, no. 5, pp. 445–452, 2014.
- [7] P. R. de Almeida, L. S. Oliveira, A. S. Britto, E. J. Silva, and A. L. Koerich, "PKLot—A robust dataset for parking lot classification," *Expert Syst. Appl.*, vol. 42, no. 11, pp. 4937–4949, 2015.
- [8] A. Koster, A. Oliveira, O. Volpato, V. Delvequio, and F. Koch, "Recognition and recommendation of parking places," in *Ibero-American Conference on Artificial Intelligence*, 2014, pp. 675–685.
- [9] R. Lu, X. Lin, H. Zhu, and X. Shen, "SPARK: a new VANET-based smart parking scheme for large parking lots," in *INFOCOM 2009, IEEE*, 2009, pp. 1413–1421.
- [10] P. P. Jayaraman *et al.*, "Here-n-now: A framework for context-aware mobile crowdsensing," in *Proc. of the Tenth International Conference on Pervasive Computing*, 2012.
- [11] A. Seaborne, "ARQ-A SPARQL Processor for Jena," *Obtained Internet Httpjena Sourceforge NetARQ*, 2010.
- [12] E. Amazon, "Amazon elastic compute cloud (Amazon EC2)," *Amaz. Elastic Compute Cloud Amaz. EC2*, 2010.
- [13] E. Polycarpou, L. Lambrinos, and E. Protopapadakis, "Smart parking solutions for urban areas," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a*, 2013, pp. 1–6.
- [14] P. Neirotti, A. De Marco, A. C. Cagliano, G. Mangano, and F. Scorrano, "Current trends in Smart City initiatives: Some stylised facts," *Cities*, vol. 38, pp. 25–36, 2014.
- [15] R. Grodi, D. B. Rawat, and F. Rios-Gutierrez, "Smart parking: Parking occupancy monitoring and visualization system for smart cities," in *SoutheastCon 2016*, 2016, pp. 1–5.
- [16] Y. Zheng, S. Rajasegarar, and C. Leckie, "Parking availability prediction for sensor-enabled car parks in



- smart cities,” in *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2015 IEEE Tenth International Conference on*, 2015, pp. 1–6.
- [17] J. Cherian, J. Luo, H. Guo, S.-S. Ho, and R. Wisbrun, “Poster: ParkGauge: Gauging the Congestion Level of Parking Garages with Crowdsensed Parking Characteristics,” in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, 2015, pp. 395–396.
- [18] Z. Ji, I. Ganchev, M. O’Droma, L. Zhao, and X. Zhang, “A cloud-based car parking middleware for IoT-based smart cities: design and implementation,” *Sensors*, vol. 14, no. 12, pp. 22372–22393, 2014.
- [19] Y. Zheng, S. Rajasegarar, C. Leckie, and M. Palaniswami, “Smart car parking: temporal clustering and anomaly detection in urban car parking,” in *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*, 2014, pp. 1–6.
- [20] J. Rico, J. Sancho, B. Cendon, and M. Camus, “Parking easier by using context information of a smart city: Enabling fast search and management of parking resources,” in *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*, 2013, pp. 1380–1385.
- [21] E. Akhavan-Rezai, M. F. Shaaban, E. El-Saadany, and F. Karray, “Online intelligent demand management of plug-in electric vehicles in future smart parking lots,” *IEEE Syst. J.*, vol. 10, no. 2, pp. 483–494, 2016.
- [22] Y. Geng and C. G. Cassandras, “New ‘Smart Parking’ system based on resource allocation and reservations,” *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1129–1139, 2013.
- [23] L. Baroffio, L. Bondi, M. Cesana, A. E. Redondi, and M. Tagliasacchi, “A visual sensor network for parking lot occupancy detection in smart cities,” in *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*, 2015, pp. 745–750.