A Resilient Smart Architecture for Road Surface Condition Monitoring

Article

Accepted version

V. Agate, F. Concone, and P. Ferraro

Proceedings of the 6th International Conference on Smart City Applications (SCA2021) – Part of the Lecture Notes in Networks and Systems book series
DOI:

Publisher: Springer

A Resilient Smart Architecture for Road Surface Condition Monitoring

Vincenzo Agate, Federico Concone, and Pierluca Ferraro

Department of Engineering, University of Palermo, Palermo, Italy,
{vincenzo.agate, federico.concone, pierluca.ferraro}@unipa.it

Abstract. Nowadays, road surface condition monitoring is a challenging problem that cannot be addressed with traditional techniques. In this paper we propose an architecture for monitoring the condition of road surfaces based on the paradigm of Mobile Crowdsensing. First, a surface detection module extracts high level features from raw data, indicating the presence of hazards. Then, in order to make the system resilient to attacks, the system exploits a reputation module to identify malicious users and filter out unreliable data. Finally, a truth discovery module aggregates the resulting information to obtain the desired truth values. Experiments carried out on a real world dataset prove the resilience of the proposed system to different attacks and the accuracy achieved.

Keywords: anomaly detection, road monitoring

1 Introduction

During their lifetime, road infrastructures are subject to continuous degradation and structural damage, thus the maintenance, preservation and renewal of sustainable and resilient road surfaces has been a major challenge in recent times [1]. Several factors pose threats to the good condition of the roadway system and thus to public safety. For example, the durability of road surfaces can gradually degrade due to aging under natural environmental conditions and due to the normal operating loads to which such roads are constantly subjected.

Natural phenomena such as earthquakes and strong winds can cause serious damage even when these events are not particularly severe. All of these factors lead to changes in various physical characteristics of the structures themselves, which in turn alter their structural behavior.

In this regard, researchers, engineers and city administrations have long recognized the importance of carrying out monitoring programs to check the health status of buildings and road infrastructures [2], in order to ensure their structural integrity before they require costly interventions or cause potentially catastrophic consequences. It is worth noticing that the currently adopted monitoring systems for the health of road infrastructures are generally based on the measurement and analysis of their dynamic response [3]. Although these approaches can provide highly accurate global information on the state of the structures, they require the installation of numerous fixed sensors and constant access to

electrical power. Clearly, considering the sheer extension of the road infrastructure, instrumentation and installation costs become prohibitive [4, 5]. Thus, most roads are not regularly monitored, leading to irreparable structural damage, or even collapses and casualties when technical interventions are delayed for too long. In this context, it is imperative to develop cost-effective solutions for rapid and continuous structural status assessment that can be deployed extensively on the majority of road infrastructures without additional cost [6].

We propose a smart architecture for road surface monitoring that aggregates user-submitted data to infer high-level information about the health status of road infrastructures, by embracing the Mobile Crowdsensing (MCS) paradigm. The system also leverages a reputation management system to give more importance to data submitted by trusted users and thus improve the overall QoI [7].

The remainder of the paper is organized as follows. Related work is outlined in Section 2. Our proposed architecture is described in Section 3, focusing on the most important aspects related to road surface detection, reputation assessment and truth discovery. We present our experimental evaluation in Section 4, and finally, we draw our conclusions in Section 5.

2 Related Work

MCS [8] is a paradigm which allows a large group of individuals to collect and share information about phenomena of public interest. This is made possible by the now overwhelming availability of mobile devices capable of measuring and analyzing event of interest in the physical world. Such devices generally incorporate a plethora of sensors of all kinds, e.g., accelerometers, gyroscopes, and GPS, which can be exploited to capture low level data on phenomena of interest [9]. For the purpose of road condition monitoring, data provided by accelerometers are of great interest, especially when they are geolocalized by GPS sensors. An active role in MCS systems is played by the user community, which is responsible for monitoring large-scale phenomena that would be impossible for a single entity to measure. The applications of this paradigm are innumerable, as testified by the large body of related work produced in recent years. Indeed, MCS is widely employed in areas such as environmental pollution monitoring [10], health monitoring, and smart environments [11]. Moreover, this paradigm can also provide a significant aid to the early detection of existing anomalies on the vehicular road network.

An important aspect of MCS systems concerns the benefits a user receives by participating, which must be tangible: users are more likely to collaborate in a project if they derive some form of gain from it, whether it be monetary or of a different nature [12, 13]. This, however, is a double-edged sword, since while it incentivizes user participation for better data collection, it also encourages some malicious users to send poor quality information to increase their participation level in the platform and, consequently, their payoff [14, 15]. Indeed, data analysis must take into account the reliability of users and weight each of their contributions according to their trustworthiness. In this regard, the use of

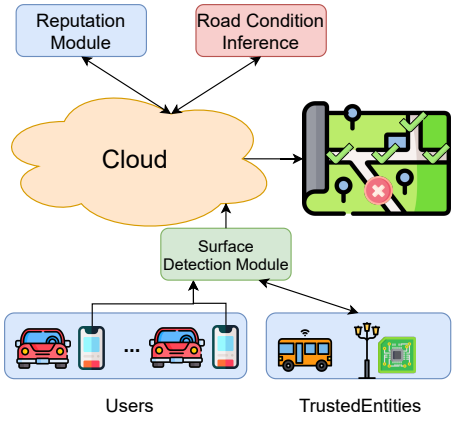

Fig. 1: Architecture of the system.

Algorithm 1 Road Condition Monitoring

Require: reports from users and TEs
Ensure: truth values for all sectors

```
for each report do
    if report can be validated by TE then
        update user reputation
    end if
end for
for each user do
    if user reputation < threshold then
        reject user reports
    end if
end for
for each sector do
    calculate sector truth value
end for
return truth values for all sectors
```

a reputation management system (RMS) [16] makes it possible to estimate the reliability level of each participant and use it in the data aggregation phase to infer a truth value that is as accurate as possible.

3 The Proposed Architecture

We propose a mobile crowdsensing system for road condition monitoring that is resilient to malicious user attacks. We envision a client-server architecture in which users collect raw sensory data, extract feature vectors from it, and share that intermediate information with the cloud server, which will use it to infer high-level information about road surface conditions. Figure 1 shows the architecture of the proposed system, which consists of three main modules, i.e., *Surface Detection*, *Reputation*, and *Road Condition Inference*.

We treat the sensing area of interest as being partitioned into s sectors, which can be of variable forms and dimensions depending on the sensing granularity of the application. At the lowest level, sensors equipped on users' smartphones continuously collect inertial data for each sector they pass through, and send them to the server for further processing. Starting from the raw data collected by users' smartphones, the *Surface Detection* module extracts a triplet of values that will be exploited by the higher layers of the architecture. In particular, the main information of interest, for each user report, are: the time interval in which the report is valid, its geographical location and an aggregate value indicating the presence or absence of hazards along the road.

The validity period depends on the context, and may vary depending on the granularity required by the application. Then, based on the geographical coordinates acquired by the GPS sensor, a reference sector is assigned to the report. Finally, the aggregate value of the report is derived by exploiting clustering techniques, as will be explained in the next section.

The *Reputation* module has the task of determining the trustworthiness of users, assigning them a reputation value that allows the system to clearly distinguish good users from malicious or unreliable ones. To this end, we adopt the concept of *Trusted Entity* (TE), which represent the root of trust of the system. TEs are trustworthy devices (such as black boxes equipped on buses and taxi cabs, or smart devices embedded in lamp posts) that periodically send reports that we consider always trustworthy. Other systems in the literature have also utilized the concept of trusted users in recent years, with favorable results [17].

These reports are used to validate the correctness and accuracy of the reports sent by ordinary users, as well as to update their reputation accordingly. TEs will only be a small fraction of the total number of users of the application, for obvious scaling and cost issues. However, their input is extremely valuable in providing feedback on users in their vicinity. If, on the other hand, a user is not in proximity of a TE, their report will be weighted according to their reputation.

Finally, the *Road Condition Inference* module accepts as input both the triplets of values produced by the *Surface Detection* module and the users' reputations. The latter are used to exclude, from the calculation of the final truth values, unreliable users, who have a reputation lower than a certain threshold. Then, starting from values sent by reliable users, the *Road Condition Inference* module uses truth discovery techniques to obtain a single truth value about the state of the road surface in each of the sectors considered. Algorithm 1 shows the pseudo-code explaining how the whole system works.

3.1 Surface Detection Module

The *Surface Detection* (SD) module aims to discover the road condition by using inertial data gathered from sensors in users' smartphones. The problem addressed by the SD can be solved by an anomaly detection algorithm that uses a machine learning approach to identify patterns in sensor data, whose strongly differ from the majority of data-points [18]. Hence, the first step of the SD is to continuously collect 3D values from both accelerometer and gyroscope sensors and then process them to extract the information of interest that will be sent to the cloud [19]. Peaks in acceleration over the three axes, in a vehicle that is moving on a road that should be in good conditions, may indicate the presence of a hole in the road surface or other hazards. Data retrieved from gyroscope not only enrich the information about the context of the vehicle but can also be employed to make the system independent from device orientation and placement [20]. Finally, GPS data is used to determine the corresponding sector in the sensing area.

To ensure a synchronization among different sensors, the SD is able to process input data within certain time windows in order to extract the features that will be used in the next classification stage. The temporal pattern produced by the accelometer and gyroscope sensors is processed into fixed-length windows of $m \times n$ samples, where m is the number of axes along which measurements are performed. Choosing the proper length for the acquisition window is essential because of the impact it could have on the whole system [21]. Short windows

may improve system performance, but may not contain enough information to properly capture the characteristics of the road. Vice versa, large windows may alter system performances since information about multiple sectors visited in sequence might be analyzed within a single window.

Raw data within the fixed-length time windows are then processed to obtain a very compact representation of input data. In particular, the feature vectors are built by considering five well-accepted values, i.e. maximum, minimum, mean, standard deviation, and root mean square over the three accelerometer and gyroscope axes. Therefore, each feature vector contains 30 elements, i.e., 15 values of acceleration and 15 values of angular velocity. Finally, the feature vectors are passed to the K-Means algorithm for the detection of surface conditions. Given the set of feature vectors $(f_1, f_2, \ldots, f_m)$, the most common application of the K-means algorithm consists in partitioning the m observations into k sets, $C = (C_1, C_2, \ldots, C_k)$, so as to minimize the intra-cluster error:

$$E = \sum_{k=1}^{K} \sum_{f_i \in C_k} ||f_i - \mu_k||^2 \tag{1}$$

where μ_k is the mean value of the k-th cluster C_k. Nevertheless, K-means can also be used for classification, i.e., supervised learning. Our algorithm outputs a binary value, where 0 indicates that the road surface is in good conditions, while 1 indicates it is not.

3.2 Reputation Module

Once the high-level information about the road surface condition has been stored in the cloud system, we can determine if this data has been sent by trustworthy users. In order to accomplish this task, we leverage a *Reputation module* which, by comparing the information that users collected and the known information about the same areas, is able to assess the trustworthiness of each participant. We can consider the task performed by users as a service rendered, where our system evaluates its quality by releasing feedback. In order to enable our system to release positive or negative feedback on the user's service, we need to compare at least some of the values provided by users with a ground truth. To this end, we introduce trusted entities (e.g., buses equipped with black boxes or smart lamp posts) that frequently monitor the condition of the road surface of given areas. This information is directly comparable with that coming from users, and will provide the system with the ground truth necessary to release feedback.

In our implementation, feedback is defined as $f_{ij}(t) \in \{-1, +1\}$, where i denotes the user who provided the information, j is the sector that was evaluated, and t is the timestamp. The value $+1$ indicates that the information provided by the user is equal to that known by the system and derived from a trusted entity, while the value -1 is stored in the opposite case.

Our module calculates user reputation according to a formulation of the reputation idea derived from Jøsang's Beta Reputation [14], one of the most widely

used RMSs in the literature. This RMS is based on the Beta probability density function and combines information about users' past transactions to obtain reputation scores. The original formulation requires that opinions can be released by all entities receiving a service, then each of them proceeds to calculate a local reputation and consequently, this information is merged to calculate a global reputation. To model the problem more closely to the original formulation, we could consider individual trusted entities capable of releasing feedback. However, since we currently assume these entities to be always reliable, considering them separately would complicate the formulation unnecessarily. Therefore, in our model of the RMS, users provide their service to our system, which is the only entity capable of issuing feedback on their behavior. This simplified approach allows us to obtain a single definition of reputation calculated according to the following formula:

$$r_i(t) = \frac{\sum_{t'=0}^{t} f_{ij}(t')\lambda_{Beta}^{(t-t')}}{2 + \sum_{t'=0}^{t} \left| f_{ij}(t')\lambda_{Beta}^{(t-t')} \right|}. \tag{2}$$

The reputation of a given user is therefore calculated by taking into account the number of positive and negative feedback, over the total number of services provided by the user. The factor λ_{Beta} is a *forgetting factor* and is defined in the range $[0, 1]$. This factor allows the system to assign lower weight to past feedback. From Equation 2 it is easy to see that the range of reputation values is in $[-1; 1]$ and that the default reputation is equal to zero when the user has not yet provided any information to the system.

3.3 Road Condition Inference

Prior to using truth discovery techniques, the *Road Condition Inference* module performs a pre-filtering phase on user data to obtain a first screening of the information to consider for the computation of the final truth values. During this phase, exploiting the reputation values and a threshold value τ, our system removes all those users whose reputation value r_i is lower than τ. Since the default reputation value is 0, a reasonable choice is to set the threshold as equal or less than 0, to include data from all those users, who have never actually been evaluated by trusted entities (TEs), but still contributed to collect valuable information. Although the inclusion of opinions from unknown users may seem risky at this stage, the *Road Condition Inference* module will properly weigh the contribution of these users and estimate the final truth value for each sector.

To obtain reliable information, the *Road Condition Inference* module exploits state-of-the-art truth discovery techniques based on CRH [22]. The input of the module consists of a set of triplets, namely the status of the road condition, the spatial coordinates, and a temporal coordinate. As previously discussed, we treat the area of interest as divided into s sectors of different forms and dimensions.

Let us suppose there are u users and the system runs in fixed time steps, so that only corresponding values of a specified time step are processed. The truth discovery algorithm aims to find a truth value t_j for each of the j sectors based on the values estimated by users and on the weights assigned to them by the system in order to appropriately merge collected data.

The algorithm consists of two phases, one of weight estimation and one of truth estimation that proceed iteratively, until some convergence criterion is reached (e.g., fixed number of iterations or when there are no more changes in the truth values). The first step entails assigning a weight w_i to each of the participants that varies based on how close their reported opinions are to the estimated truth values. We use the following formula to calculate the weights:

$$w_i = -log \frac{\sum_{j=0}^{s} d(v_{i,j}, t_j)}{\sum_{i'=0}^{u} \sum_{j=0}^{s} d(v_{i',j}, t_j)}. \tag{3}$$

There are several ways to calculate distances depending on the application context and the nature of the data (e.g., continuous or categorical data). Since our data is discrete, the distance function that best fits our case is defined as follows: $d(v_{i,j}, t_j) = 1$ if $v_{i,j} \neq t_j$, or 0 otherwise. Once the weights are obtained, the system can calculate the truth values using the following formula:

$$t_j \leftarrow \arg\max_{v} \sum_{i=1}^{u} w_i \cdot \delta(v, v_{i,j}), \tag{4}$$

where the function $\delta(x, y)$ returns 1 if x and y are equal and 0 otherwise. Equation 4 is equivalent to a weighted majority voting, where the final truth value is the one that is receives the most votes among all participants, considering that the vote of each user is weighted w_i. These two steps are repeated until the convergence criterion is reached. The truth values obtained for each sector are then used to create of a heatmap of road hazards which is available to all users.

4 Experimental Evaluation

To evaluate the effectiveness of our approach, we performed a series of experiments leveraging real-world mobility traces. The dataset we chose contains the mobility traces of about 500 taxi cabs, collected during a 30-day period in San Francisco, USA [23]. The evaluation of our system is carried out by simulating an application where participants (both normal users and TEs) periodically send reports about the state of the road surface, signaling the possible presence of hazards as a boolean value. In the experiments we consider a sensing area of about 4x4 km, divided into 500 sectors of variable size. The results that we present are the average of 100 runs performed, with 1000 users in an interval of 100 time steps. The paths followed by users are taken directly from the mobility traces. As for the values sent by each user, we have assumed that normal users send correct reports with a given probability (i.e., 90%) to take into account possible inaccuracies. Attackers, on the other hand, send false data more frequently.

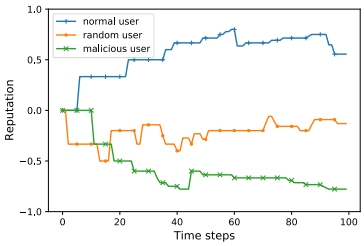

Fig. 2: Reputation trends.

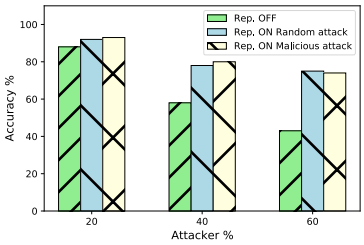

Fig. 3: Accuracy of the system.

In the following experiments, we modeled two types of attackers: the first type almost always sends false data (90% of the time) in order to maliciously change the truth values inferred by the system, while the second type sends data randomly in an attempt to gain incentives from the application without really participating. Figure 2 shows the reputation trend of three users, during 100 timesteps. As can be seen from the figure, the normal user's reputation grows steadily when reports are validated by TEs. On the other hand, when no TE is nearby, the user's reputation remains constant, which results in a stair-step trend. The malicious user who almost always sends false data is easily detected by the *Reputation* module, and his/her reputation tends to decrease over time. These kinds of attackers are easily removed during the filtering phase. Finally, the user who sends data randomly is obviously more difficult to categorize. Indeed, in this case, its reputation oscillates around the initial reputation value (i.e., 0).

Figure 3 summarizes the results obtained by the system as a whole, in terms of accuracy, as the number of attackers increases. The different bars show, respectively, the behavior of the system both with and without the *Reputation* module to pre-filter untrustworthy users. As expected, the impact of the *Reputation* module is directly proportional to the number of attackers. When attackers are a small portion of the total user base, the truth discovery system is able to infer accurate information on its own, as normal users outnumber attackers in the majority of sectors. When, on the other hand, there are a considerable number of attackers, the reputation system has a strong influence on the result. Indeed, without the pre-filtering phase, attackers would overwhelm normal users and manage to reverse the final outcome, as is generally the case in majority voting systems. However, the combined use of the reputation and truth discovery modules produces great results, as shown in Figure 3. In particular, we note how our system achieves similar results with both random and malicious attackers. This can be explained by the fact that users submitting random data are difficult for the *Reputation* module to detect, but cause less problems for the truth discovery system. Users that almost always send false data, instead, are easier to manage on the reputation side, but if they somehow pass the filtering phase they have a greater influence on the final truth calculation. The two effects are balanced, as shown in Figure 3, hence the similar results.

5 Conclusion

In this paper we studied the problem of efficiently and scalably monitoring the condition of road surfaces, inferring high-level information that allows technicians to intervene in a timely manner when maintenance is required.

To this end, we proposed a road surface monitoring system that aggregates user-submitted data by exploiting the Mobile Crowdsensing paradigm. The system we propose is resilient to attacks thanks to its reputation management system that tracks user behaviors and allow us to filter out untrustworthy users, improving the overall quality of information. The system collects raw data coming from users' smartphones, extracts feature vectors from it, and share that intermediate information with the cloud server. Our approach involves the combined use of a reputation management system and a truth discovery system to infer reliable high-level information [24]. Extensive experiments carried out by exploiting real world taxi cabs traces have demonstrated the viability of the system even when malicious users try to alter the information inferred.

Acknowledgment

This research is partially funded by the Project CrowdSense (PO FESR Sicilia 2014/2020).

References

1. Li, H.N., Ren, L., Jia, Z.G., Yi, T.H., Li, D.S.: State-of-the-art in structural health monitoring of large and complex civil infrastructures. Journal of Civil Structural Health Monitoring 6(1), 3–16 (2016)
2. Eriksson, J., Girod, L., Hull, B., Newton, R., Madden, S., Balakrishnan, H.: The pothole patrol: using a mobile sensor network for road surface monitoring. In: Proceedings of the 6th international conference on Mobile systems, applications, and services. pp. 29–39 (2008)
3. OBrien, E.J., Malekjafarian, A., González, A.: Application of empirical mode decomposition to drive-by bridge damage detection. European Journal of Mechanics-A/Solids 61, 151–163 (2017)
4. Laubis, K., Konstantinov, M., Simko, V., Gröschel, A., Weinhardt, C.: Enabling crowdsensing-based road condition monitoring service by intermediary. Electronic Markets 29(1), 125–140 (2019)
5. Lo Re, G., Peri, D., Vassallo, S.D.: Urban air quality monitoring using vehicular sensor networks. In: Advances onto the Internet of Things, pp. 311–323. Springer (2014)
6. Strazdins, G., Mednis, A., Kanonirs, G., Zviedris, R., Selavo, L.: Towards vehicular sensor networks with android smartphones for road surface monitoring. In: 2nd International Workshop on Networks of Cooperating Objects. vol. 11 (2011)
7. Agate, V., Khamesi, A.R., Silvestri, S., Gaglio, S.: Enabling peer-to-peer user-preference-aware energy sharing through reinforcement learning. In: ICC 2020 - 2020 IEEE International Conference on Communications (ICC). pp. 1–7 (2020)

8. Concone, F., Lo Re, G., Morana, M.: Smcp: a secure mobile crowdsensing protocol for fog-based applications. Human-centric Computing and Information Sciences 10(1), 1–23 (2020)
9. De Paola, A., Ferraro, P., Gaglio, S., Lo Re, G.: Context-awareness for multi-sensor data fusion in smart environments. In: Conference of the Italian Association for Artificial Intelligence. pp. 377–391. Springer (2016)
10. Becnel, T., Tingey, K., Whitaker, J., Sayahi, T., Lê, K., Goffin, P., Butterfield, A., Kelly, K., Gaillardon, P.E.: A distributed low-cost pollution monitoring platform. IEEE Internet of Things Journal 6(6), 10738–10748 (2019)
11. Agate, V., Concone, F., Ferraro, P.: Wip: Smart services for an augmented campus. In: 2018 IEEE International Conference on Smart Computing (SMARTCOMP). pp. 276–278 (2018)
12. Timilsina, A., Khamesi, A.R., Agate, V., Silvestri, S.: A reinforcement learning approach for user preference-aware energy sharing systems. IEEE Transactions on Green Communications and Networking (2021)
13. Restuccia, F., Ferraro, P., Silvestri, S., Das, S.K., Lo Re, G.: IncentMe: Effective mechanism design to stimulate crowdsensing participants with uncertain mobility. IEEE Transactions on Mobile Computing 18(7), 1571–1584 (2018)
14. Josang, A., Ismail, R.: The beta reputation system. In: Proceedings of the 15th bled electronic commerce conference. vol. 5, pp. 2502–2511 (2002)
15. Agate, V., De Paola, A., Gaglio, S., Lo Re, G., Morana, M.: A framework for parallel assessment of reputation management systems. In: Proceedings of the 17th International Conference on Computer Systems and Technologies 2016. p. 121–128. CompSysTech '16 (2016)
16. Agate, V., De Paola, A., Lo Re, G., Morana, M.: A simulation software for the evaluation of vulnerabilities in reputation management systems. ACM Trans. Comput. Syst. 37(1–4) (Jun 2021), https://doi.org/10.1145/3458510
17. Restuccia, F., Ferraro, P., Sanders, T.S., Silvestri, S., Das, S.K., Lo Re, G.: FIRST: A framework for optimizing information quality in mobile crowdsensing systems. ACM Transactions on Sensor Networks (TOSN) 15(1), 1–35 (2018)
18. Martorella, G., Peri, D., Toscano, E.: Hardware and software platforms for distributed computing on resource constrained devices. In: Advances onto the Internet of Things, pp. 121–133. Springer (2014)
19. Concone, F., Ferraro, P., Lo Re, G.: Towards a smart campus through participatory sensing. In: 2018 IEEE International Conference on Smart Computing (SMARTCOMP). pp. 393–398. IEEE (2018)
20. Guo, H., Chen, L., Chen, G., Lv, M.: Smartphone-based activity recognition independent of device orientation and placement. International Journal of Communication Systems 29(16), 2403–2415 (2016)
21. Concone, F., Gaglio, S., Lo Re, G., Morana, M.: Smartphone data analysis for human activity recognition. In: AI*IA 2017 Advances in Artificial Intelligence. pp. 58–71. Springer (2017)
22. Li, Y., Li, Q., Gao, J., Su, L., Zhao, B., Fan, W., Han, J.: Conflicts to harmony: A framework for resolving conflicts in heterogeneous data by truth discovery. IEEE Transactions on Knowledge and Data Engineering 28(8), 1986–1999 (2016)
23. Piorkowski, M., Sarafijanovic-Djukic, N., Grossglauser, M.: A parsimonious model of mobile partitioned networks with clustering. In: 2009 First International Communication Systems and Networks and Workshops. pp. 1–10. IEEE (2009)
24. Agate, V., Ferraro, P., Gaglio, S.: A cognitive architecture for ambient intelligence systems. In: AIC. pp. 52–58 (2018)