Real-Time Road Quality Assessment Using Smartphones and Cloud Lambda Architecture

Marcin Badurowicz[1,*], and *Tomasz* Cieplak[2]

[1]Lublin University of Technology, Faculty of Electrical Engineering and Computer Science, Institute of Computer Science, Nadbystrzycka 36B, 20-618 Lublin, Poland
[2]Lublin University of Technology, Faculty of Management, Nadbystrzycka 38A, 20-618 Lublin, Poland

Abstract. In this paper, the authors are proposing a computer system built in the cloud-computing fashion for the collection of data from smartphones to achieve a crowdsensing-based quality assessment of roads, as well as detection, identification, and assessment of road artefacts (potholes, speed bumps, *etc*). The proposed information processing methods were based on Lambda architecture and the integration of different types of crowdsourced data to finally calculate the singular value of the human-readable road quality. Finally, the prototype implementation of the system for gathering, processing, integration, and processing the road quality data is also presented.

1 Introduction

The problem of locating potential road hazards, including speed breakers or different road artefacts, such as potholes or surface degradation points, is a key problem for road users to solve in order to know where to expect them and to facilitate immediate reaction. The same issue is also crucial for governmental organisations [1], pinpointing locations of routine road maintenance, especially in the situation of limited funds. The defined problem can be divided into smaller aspects: the first concerns locating problems on the road to warn users as they are approaching them in the car; the second is to describe the overall state of the road on a simple scale for road users to know if this particular fragment of the road should be avoided, speed lowered or care is advised; the third is to use both data about traffic and assessed road quality or about significant road artefacts to indicate whether a given fragment should be included in the maintenance process.

The Internet of Things (IoT) paradigm, where ubiquitous computing is used for real-time monitoring, smart environment and finally smart cities, is a perfect candidate for this solution. With tiny, always-on and always-connected devices, which may be included in our daily lives, it appears viable to build a wireless sensor network solution to monitor the state of the road – similar large-scale deployments for different civil infrastructure systems have been already developed [2]. Deployment of numerous fixed sensors is however very costly, even including public funding, and a serious challenge based on the number of roads already being active and being built at the moment. However, there is a possibility of inverting the sensor network – what if we could track every car movement over the road surface using the car itself and send it to the central aggregation system? Such data, being collected from cars, may be processed in search of road artefacts or signs of deteriorating quality.

In this paper, we propose a robust solution for real-time road quality assessment and monitoring using the aggregated data, extended by the possibility of pinpointing surface problems, as accurately as possible, using aggregated data from smartphones, other IoT devices, as well as Global Navigation Satellite System (GNSS).

2 Related works

2.1 Smartphones as urban sensors

There are over two billion smartphone users in the world, and over the course of the last five years, the percentage of smartphone users has risen dramatically [2].

These smartphones are already equipped with a set of sensors: accelerometers, microphones, gyroscopes, video cameras and more. Devices and data from them have been already used in different community sensing ideas, starting from urban road usage patterns [2], already available also in commercial products, as well as detection of potholes, and even earthquakes [3].

2.2 Assessment of the road quality using accelerometers

Our previous works on road artefact detection with commonly used devices such as smartphones, as well as works from several researchers [4], from different countries [5] in the last ten years [6], concentrated on detection of single road artefacts during a one-time drive over the road fragment. Different proposals have been introduced to detect potholes based on the

* Corresponding author: m.badurowicz@pollub.pl

accelerometers mounted in cars, as well as using smartphones of the road users, including: basic thresholding [7], neural networks of different kinds and machine learning [8], Bag of Words [9], as well as digital fingerprinting [10].

The other side of the problem is the crowdsensing aspect, which also has been discussed earlier [11]. Data aggregation is rising in complexity when more and more cars with monitoring devices are added to the system, and relatively small is told about how to aggregate the data into something human-readable with a possibility of observing changes over time. Some papers discuss this topic into the creation of IRI [12] values, which are not representative of the general population. The move from raw numeric data into a simple scale has been already discussed by the authors [13], but there is still a lack of holistic approach, which we would like to provide in this paper.

3 The proposed solution

The goal of the study is to provide an integrated solution for road artefact detection and road quality monitoring by means of smartphone devices used by drivers in their cars where the system in exchange for their acceleration data contribution will provide them with a set of warnings about possible road quality changes through the application. Such a system, dubbed "Community Road Artefacts Detection, Identification and Assessment" (CRADIA) is in the final stages of implementation, and the prototype implementation is presented in this paper.

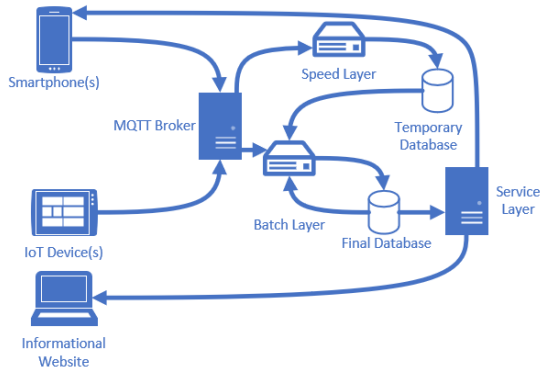

Fig. 1. The CRADIA system overview diagram.

The system is divided into three major parts, presented in Fig. 1 – the first one is the smartphone or embedded (Internet of Things) devices section, which are the real devices performing data acquisition and transfer to the system.

Second is the data processing part, implemented in the form of Lambda architecture [11]: there are stream processing and batch processing layers in the cloud computing system.

The third part, and probably the most important for end users, is the view part presenting current processed information, historical data, as well as notifying users about the problems on the road.

3.1 The sensing part

This part consists of a smartphone/tablet application, implemented for Windows 10 and Android platforms, which records acceleration data when mounted in a stable way in a car. Apart from the acceleration data, the application records current position and speed, as well as GPS system coordinates and the current time.

While the user may position the device in multiple ways, and the most important data for the research is recorded in the vertical axis acceleration, the authors are using the orientation sensor for the transformation of recorded acceleration data from the local to the global coordinate system [15]. There are no other requirements regarding the position of the device in the car apart from mounting it in a stable way, so that the device will not fall during the ride,.


```
{
    "X": 0.00049,
    "Y": 0.04102,
    "Z": -0.9752,
    "N": -0.18242,
    "E": -0.10243,
    "Z2": -0.95337,
    "Time": "2018-07-10T11:22:23.395",
    "Longitude": 22.5567806884646,
    "Latitude": 51.2236101273447,
    "Speed": 7.75,
    "Course": 0.0,
    "Accuracy": 0.0
}
```


Fig. 2. Example of the data package.

The smartphones are sending their data directly when recording them, using the MQTT protocol to the central MQTT broker. Every packet is a JSON object, as presented in Fig. 2, consisting of: acceleration in X, Y and Z axis (in g, which is a Earth's acceleration value multiplication), acceleration in N (global north), E (global east) and Z2 (global perpendicular to the Earth's surface) axis (after transformation to global coordinate system), current device timestamp, current GNSS location data (latitude and longitude), speed (in meters per second), course (in degrees in relation to the geographic north), GNSS accuracy (in meters) and device unique ID (collected using the device APIs). However, the proposed system does not require any kind of user registration, to avoid privacy issues – the authors are using device IDs only to differentiate between devices during speed layer analysis and this field is discarded just after data aggregation step.

While cars are different and accelerometer readings of different cars for even the same road artefact will be different, we propose using the correction technique based on preparing a correction/normalisation vector calculated on the "known" part of the road and applied then to every other measurement sent to the central system. This technique has been described earlier by the authors of [10] and is employed before data processing in the system, thus the processing part does acquire any

specific data concerning the devices sending data to the system.

3.2 The processing system part

The massive amount of data – as they are recorded multiple times per second, as well as the necessity for connecting multiple devices, forces the use of the Big Data techniques and the cloud computing system. In the case of the solution presented here, the Lambda architecture was used for different techniques for analysing the acquired data. The architecture consists of three layers: batch, speed, and service.

3.2.1 The speed layer

The speed layer objective is not to store data but to process them in real time - to perform streaming analytics over a narrow window on incoming streams of data. Streams are analysed in data windows based on device IDs, so there is one data stream for every device, and in different windows, based on the algorithm employed. Two main kinds of windows are used: time and distance-based.

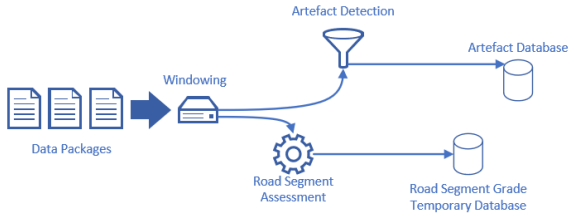

Fig. 3. The Speed Layer overview.

In the presented solution already tested and documented algorithms for a pothole, speed bump and other road artefacts detection and classification algorithms are employed; at the moment, there are two methods used simultaneously, resulting in a sequence of datasets about possible road artefacts – positions of speed bumps, positions of potholes *etc*. There is a classification of road artefacts into several types we have [10] already implemented in previous works.

The first algorithm used is the modified Z-THRESH algorithm (MOD-Z-THRESH) [12], where every data point with a Z acceleration over a certain threshold (T) is classified as possible road artefact, while the threshold is adaptive, as is a multiplication of the standard deviation of the currently processed road fragment (σ) times the experimentally calculated value x:

$$T = x \cdot \sigma \tag{1}$$

Our previous works have indicated that the best results were achieved using the x value of 4.3.

The use of the relative threshold means that the outlier detection is adapting itself automatically to the changes in the road surface acceleration readings, allowing for better outlier detection than the globally defined threshold. While a simple and pretty effective measure [13], MOD-Z-THRESH does not allow to determine the type of road artefact.

To try to recognise the artefact, different methods are discussed in the literature, and the presented system is currently based on SCC-DFP algorithm, where acceleration in the N, E and Z2 axis and speed vector are being cross-correlated to the known patterns of the road artefacts with a set of defined weights, and then cross-correlated to negative patterns, where the first correlation value should be over the defined threshold and the second correlation value should be lower than the defined threshold. The SCC-DFP algorithm is a novel approach to the problem and is currently in the final stages of development, but its full description is beyond the scope of this paper.

However, the speed layer architecture allows also to plug in additional methods of processing, so there is a possibility to use a different set of algorithms, for the creation of a better dataset for end users. Because of the usage of multiple detection approaches on the same dataset, every algorithm has weight and finally, in the micro-clustering phase in the batch layer, its final score is calculated as a weighted average.

Finally, for every detection of road artefact, essential information is saved to the temporary Artefact Database:

$$(\lambda, \phi, \varepsilon, \rho, \chi, \alpha) \tag{2}$$

The presented tuple (2) elements are: the position of the road artefact, with current latitude (λ) and longitude (ϕ); GNSS accuracy (ε), type of road artefact (if algorithm allows to differentiate) (ρ), and its score (χ) as defined by the algorithm, and finally, identifier of the algorithm (α).

The temporary database is saved to be processed later by the batch layer for micro-clustering and aggregation.

The score for the road artefact defines how "important" is the road artefact and is normalised to values between 0 and 1. The calculation of the score is defined by the algorithm implementation.

The second part of speed layer analysis is to calculate the wider quality index, and not to find single road artefacts. To achieve this, the system is performing the calculation of two more quality indices on larger windows. While outlier detection and road artefact classification are using the fixed window size (100 data points and 50 data points, respectively), here a window size is based on the physical distance covered. This allows waiting for more data when the user is stuck in a traffic jam or is driving slowly. The defined window size is 50 meters at the moment.

In our solution, the overall road quality fragment of the defined size is calculated using the Road Relative Unevenness Index (RRUI) method [13], as a difference between roughness between the road fragment and averaged roughness of the base road fragment, which is a fragment of a very good quality road.

$$RRUI = (\sigma_Z - A) \cdot 10^2 \tag{3}$$

The second parameter calculated on the wide window is the Normalised Dispersion (ND) [1], defined as

a difference between the maximum and minimum acceleration values for the processed road fragment:

$$ND = \max(Z) - \min(Z) \quad (4)$$

Both RRUI and ND are saved to the database along with geographical positions of the starting point and the final point of the road fragment, as well as the average speed of the particular user and current timestamp, which allows as defining it later as the urban or suburban environment and track changes in both indices over time.

3.2.2 The batch layer

The first element of the batch layer is the micro-clustering of recognised road artefacts - results of the different methods for outlier detection and different methods for road artefacts classification are organised into microclusters based on their location.

Firstly, if there are points at the same location detected by multiple algorithms, they are grouped into one and their final score is the weighted average:

$$\chi = \frac{\sum_{i=1}^{i=n} \chi_i \cdot b_i}{\sum_{i=1}^{i=n} b_i} \quad (5)$$

Where χ is the i-th algorithm score for current point normalised to value from 0 to 1 and b is the i-th algorithm weight, which allows introducing better algorithms as more important.

Secondly, to cope with GNSS inaccuracy, the proposed solution is to merge every detected road artefact in the radius of two times of current GNSS accuracy, as detected by the device, into one.

$$(\lambda, \phi, \varepsilon, \rho, \delta, \tau, \chi) \quad (6)$$

The final road artefact information, which is a tuple (5) consisting of latitude (λ), longitude (ϕ), current GNSS accuracy (in meters) (ε), type of road artefact (ρ) and number of artefacts grouped for the geographic point (δ), as well as current timestamp (τ) and final average score (χ) are processed by one more step, an interesting addition in terms of improving the accuracy of the final positions of the road artefacts: the final clustered positions are also tested in the database with the already existing data obtained from previous analyses.

If the data are in the same geographical position (in the radius of a circle starting, but with lower GNSS inaccuracy value, the position of the cluster is updated and the radius is also lowered, again to the size of two times of lower GNSS inaccuracy, and timestamp is updated. If there is no data on road artefact in the final artefacts database, the tuple is simply added to the database. This enables the eventual improvement of the location of the road artefacts on condition that new devices and data are available.

The batch layer is also using the wider quality data from the speed layer in the next step of processing, where simple road fragments are re-grouped. The first problem here is the quality assessment of overlapping road fragments. If two overlapping fragments are detected, the overlapping fragment is subtracted from both fragments, if it is longer than 20 meters, and its final RRUI is a weighted average:

$$RRUI_F = \frac{\frac{RRUI_1 \cdot M^{-T} + RRUI_2}{1+M^{-T}} + l \cdot x}{1+x} \quad (7)$$

Where $RRUI_1$ and $RRUI_2$ are RRUI values for both road fragments, l is the length of the overlapping fragment, x is a constant weight of a data about the length of the intersection, and M^{-T} is the weight for reducing the impact of old data, where M is a constant defined by the implementation and T is a number of days divided by 30 (roughly number of months) passed between calculation of first and second RRUI values.

Otherwise, if the intersection is short, it is added to the shorter fragment, to achieve balance. However, if the grade difference between the overlapping fragment and the referenced fragment is very high (> 10 in case of RRUI), the overlapping fragment is not added but left as one. The similar technique is applied to the ND parameter.

Finally, the final grade on all combined road segments is computed, which is called the Road Quality Index (RQI). The RQI for the defined road segment is defined as:

$$RQI = RRUI \cdot x_1 + \frac{n}{l} \cdot x_2 + \left(\sum_{i=1}^{i=n} (x_{\rho i} + \chi_i) \cdot \delta_i \right) \cdot x_3 + ND \cdot x_4 \quad (8)$$

Where: n is the number of road artefacts over the assessed road fragment, l is the length of road fragment (in meters), x_ρ is the road artefact class weight, δ is the number of road artefacts grouped in the position, χ is the score of the road artefact as assessed in the batch layer and finally x_i are the weights of parameters, which are defined in the implementation. It has been resolved that for the time being the classification of roads into urban/suburban environment is not pursued; however, it may be implemented in the future when different quality index are prepared.

This means that the road quality is defined by overall road fragment unevenness, the number of road artefacts and their classes – for example, simple potholes may be classified lower than speed bumps, but very large pothole will still be ranked higher than a slight speed bump. The final RQI is a numerical value for a road fragment and lower means better road quality.

The important part is, however, that final results calculation of road segments quality is also saved to the historical database with a timestamp, allowing to analyse changes in quality (for example, its deterioration) over time. Thus, two road quality databases are actually used: current and historical.

3.3 The service part

The service part is the final foundation of the system allowing the users to see the final results of the system in the graphical form.

This is implemented as the web application with the REST service, allowing the operations presented in Table 1.

The rest service may be used by the mobile applications, which are asking the service layer periodically about incoming road quality and artefacts on the predicted user course, so during the drive users may be alerted of imminent road artefact encounter or road quality deterioration.

Alternatively, there is a possibility to build a web application presenting collected information in the form of an overlay on a map, allowing a user to check out the general data, as well as to browse the road quality change over time using collected historical information.

Table 1. REST operations implemented in the system.

Operation URI	Operation Definition
/artefacts/{latitude,longitude,range} (GET)	Returns all the road artefacts saved in the system at the current position and in requested range.
/quality/{latitude,longitude} (GET)	Returns the final road quality at the current position.
/quality/{latitude,longitude,course,speed}	Returns the final road quality about the current road and roads on defined course – the amount of data is also based on current speed.

The third possibility is to prepare an application of automatic alerts: when the system notices a decrease in the calculated road quality over a road segment it may be used to automatically notify the corresponding territorial governmental units.

3.4 Additions to the traditional Lambda architecture

In contrast with the traditional Lambda architecture, not all data are sent to the system using only speed and batch layers. The system allows to "plug in" the IoT Edge paradigm [14], where devices are processing data on the fly on the device itself, to reduce the amount of data sent to the system. In this case, devices feed the processing system with data already processed on the device, allowing bypassing the speed layer. The data from the IoT Edge processing are defined as a tuple consisting of geographical position of the recognised road artefact, the GNSS inaccuracy and current time, and are being processed only by the batch layer for clustering data from the same device to calculate δ parameter and then to eventually aggregate data with the data already existing in the final artefacts database.

The authors are also willing to propose second modification in the speed layer - where there is a match for detected road artefacts using MOD-Z-THRESH or SCC-DFP, raw data of this processed window is saved to the separate patterns database, where it can be used later as a new pattern to match the signal window with, again in the speed layer, or to be used as training data for new algorithms research in the future.

4 The prototype setup and preliminary results

The prototype setup was prepared using the set of data already acquired during the set of over 220 drive sessions using three different cars and different acquisition devices (Nokia Lumia 920, Nokia Lumia 1020 and Microsoft Lumia 950) in 2015-2017, mostly over the territory of Lublin, Poland. The coverage of data over the map of Lublin is presented in Fig. 4 and, including multiple drives over the same road fragments, covers more than 190 km.

Fig. 4. The overall view of preliminary data.

In the prototype setup instead of using MQTT broker receiving data from the devices; such solution was mocked using streaming reading from already registered data files.

Streaming analytics were simulated using Python scripts, read in streaming fashion from the data files, and producing temporary JSON files (as presented in Fig. 5) as well as saving data to the MariaDB database for future usage in the service layer.


```
{
    "start": {
        "lat": 51.225005714222796,
        "lng": 22.5369538832456
    },
    "end": {
        "lat": 51.224708408117294,
        "lng": 22.5367738399655
    },
    "rrui": -1.7939573266734634,
    "class": "A",
    "nd": 0.16395000000000004,
    "speed": 4.445876288659794,
    "set": "#346e16",
    "timestamp": "2015-03-31"
}
```


Fig. 5. The Speed Layer road segment example data.

The batch layer was prepared as a set of Python scripts, working on data from both data files, as well as on data saved in the temporary database by speed layer.

The service layer was not implemented fully, but instead only a limited set of proposed operations was

implemented in C# and JavaScript for presentation purposes – the most important being the map, where RQI values are translated to the 8-color scale from lime (best) through yellow and orange to maroon (worst) – as presented on the map, along with road artefacts positions and classifications presented in the form of pushpins on the same map. The example of the map processed by the prototype system is presented below, in Fig. 6 (different scale) and Fig. 7 (pushpins marking road artefacts).

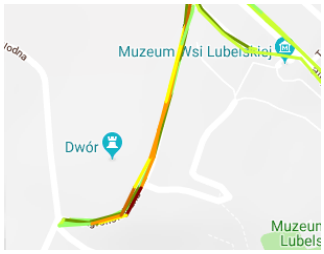

Fig. 6. Example final quality results in colour scale.

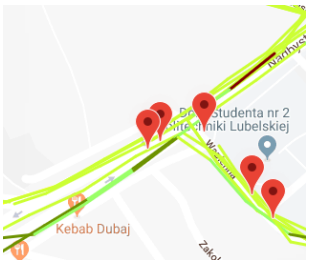

Fig. 7. Example pushpins marking positions of potholes.

During the analysis, in the speed layer stage, over 4500 road fragments were analysed, resulting in over 800 potholes and other road artefacts mapped.

The method for overriding data with new information over time was also proven to be successful: for example, one of the road segments with RRUI of 4.09 (very good) was later overridden with new data after 4 months with value 9.87 (worse). The calculated new RRUI of 7.49 was based mostly on newer data and may indicate that road fragment degraded over this time.

5 Summary and future works

The paper presented a solution to unify different techniques for road artefact detection and road quality assessment in the cloud computing Lambda architecture, by employing a plug-in analysis over windowed data streams in the speed layer. In the batch layer, the recalculation of RRUI and ND parameters was introduced during the drive over the same road fragment, and was based on the time difference to ensure that recent data is has higher priority than the old data, allowing to maintain a historical database of the road quality changes. The similar technique may be employed also for additional road quality indicators. Finally, the Road Quality Index was proposed as a unified method of assessing road quality in a human-readable way.

The prototype system, CRADIA, was implemented and tested over a set of already collected data. The prototype implementation was tested, successfully validating the concept, however, the final implementation of the system is not yet ready.

The important element which was not employed in this research, but should be considered, is merging the data between acceleration data from the smartphone devices and data about existing roads, for example, road centrelines, as now in the prototype system there may be a possibility of the data calculation slightly near the real existing road due to GNSS inaccuracy. This is one of the factors the authors are willing to add to the CRADIA system in the future.

References

1. M. Badurowicz, J. T. Montusiewicz, T. Cieplak, *Rocz. Kol. Anal. Ekon.*, 13–22 (2017)
2. T. Matarazzo, M. Vazifeh, S. Pakzad, P. Santi, C. Ratti, *Procedia Eng.* **199**, 966–971 (2017)
3. Q. Kong, R. M. Allen, L. Schreier, Y.-W. Kwon, *Sci. Adv.* **2** (2016), doi:10.1126/sciadv.1501055
4. A. Mukherjee, S. Majhi, *Eur. Transp. Res. Rev.* **8**, 1–12 (2016)
5. F. Kalim, J. Jeong, M. U. Ilyas, *IEEE Access*. **PP**, 8317–8326 (2016)
6. T. S. Brisimi, C. G. Cassandras, C. Osgood, I. C. Paschalidis, Y. Zhang, *IEEE Access*. **4** (2016), doi:10.1109/ACCESS.2016.2529562
7. V. Astarita, R. Vaiana, T. Iueleand, V. C. Maria, V. P. Giofre, F. De Masi, *Procedia - Soc. Behav. Sci.* **111**, 242–251 (2014)
8. A. Allouch, A. Koubaa, T. Abbes, A. Ammar, *IEEE Sens. J.* **17**, 4231–4238 (2017)
9. L. C. Gonzalez, R. Moreno, H. J. Escalante, F. Martinez, M. R. Carlos, *IEEE Trans. Intell. Transp. Syst.*, 1–13 (2017)
10. M. Badurowicz, J. Montusiewicz, *ITM Web Conf.* **15** (2017) (available at https://doi.org/10.1051/itmconf/20171502008)
11. D. Sun, G. Zhang, W. Zheng, K. Li, 193–214 (2014)
12. M. Badurowicz, J. Montusiewicz, in *Information and Software Technologies, ICIST 2015 [WOS]*, R. Damasevicius, G. Dregvaite, Eds. (Springer, Berlin, Germany, 2015; http://link.springer.com/10.1007/978-3-319-24770-0_43), pp. 503–514
13. M. Badurowicz, T. Cieplak, J. Montusiewicz, in *Computer Networks: 23rd International Conference, CN 2016, Brun{ó}w, Poland, June 14-17, 2016, Proceedings*, P. Gaj, A. Kwiecień, P. Stera, Eds. (Springer International Publishing, Cham, 2016; http://dx.doi.org/10.1007/978-3-319-39207-3_31), pp. 360–369
14. R. Talluri, Why edge computing is critical for the IoT (2017), (available at https://www.networkworld.com/article/3234708/internet-of-things/why-edge-computing-is-critical-for-the-iot.html)