Managing nonuniformities and uncertainties in vehicle-oriented sensor data over next generation networks

Stavros Nousias[1], Christos Tselios[1], Dimitris Bitzas[1], Olivier Orfila[2], Samantha Jamson[3],
Pablo Mejuto[4], Dimitrios Amaxilatis[5], Orestis Akrivopoulos[5], Ioannis Chatzigiannakis[6],
Aris S. Lalos[1] and Konstantinos Moustakas[1]

[1] University of Patras, Greece, {nousias, tselios, bintzas, aris.lalos, moustakas}@ece.upatras.gr
[2] IFSTTAR, France, {olivier.orfila}@ifsttar.fr
[3] University of Leeds, UK {s.l.jamson}@its.leeds.ac.uk
[4] CTAG, Spain {pablo.mejuto}@ctag.com
[5] Spark Works ITC Ltd, UK {d.amaxilatis, akribopo}@sparkworks.net
[6] Sapienza University of Rome, Italy {ichatz}@dis.uniroma1.it

***Abstract*—Detailed and accurate vehicle-oriented sensor data is considered fundamental for efficient vehicle-to-everything (V2X) communication applications, especially in the upcoming highly heterogeneous, brisk and agile 5G networking era. Information retrieval, transfer and manipulation in real-time offers a small margin for erratic behavior, regardless of its root cause. This paper presents a method for managing nonuniformities and uncertainties found on datasets, based on an elaborate Matrix Completion technique, with superior performance in three distinct cases of vehicle-related sensor data, collected under real driving conditions. Our approach appears capable of handling sensing and communication irregularities, minimizing at the same time the storage and transmission requirements of Multi-access Edge Computing applications.**

***Index Terms*—Graph Matrix Completion, V2X, MEC, Sensor Data.**

I. Introduction

As global telecommunication market shifts towards the 5G era and the fascinating technological advances it proclaims [1], it becomes obvious that new applications and verticals with a tremendous affect in our everyday lives are approaching. The deployment of a highly heterogeneous, always available, brisk and agile network, offering inherent support for billions of interconnected devices with less than 1 millisecond end-to-end latency [2], will transform almost every application from simple daily entertainment to autonomous private transportation and the automotive domain in general, inside Smart Cities which will provide the proper underlying infrastructure.

The automotive domain is somehow divided into two separate yet highly consolidating tracks, autonomous driving vehicles (ADV) and vehicle-to-everything (V2X) communication [3], with V2X act as a key technology enabler for ADV by allowing moving vehicles to approach each other more safely, thus enabling traffic flow optimization techniques and increased situational awareness. Vehicle-oriented notifications and alerts are propagated to nearby infrastructure, properly equipped pedestrians or other vehicles, rendering every involving entity capable of reacting fast and efficiently regardless of the situation [4] for instance, to the now hazardous cases of blind intersections, closed curves, lane switching and overtaking which often cause fatal car accidents worldwide. Smart Cities all around the globe are deploying dedicated infrastructure for facilitating V2X communication in a seamless manner, since citizen security and effective public/private transportation are considered top priorities.

Information retrieval, transfer and manipulation in real-time, requires a rapidly operating network with decreased end-to-end propagation delay, large throughput and low latency. In addition, for efficient and secure V2X communication applications, the same network should include proper interfaces for establishing robust connections with the heterogeneous devices involved, being adequately fault tolerant and able to deliver informative messages even with fragmented data.

In this paper, motivated by the necessity for fault tolerant systems that will extract information from incomplete datasets, as well as the omnipresent demand of limited data transfer through the rather congested Smart City networks that will be forced to operate partially based on the resource savvy MEC architecture, we present a method of managing nonuniformities and uncertainties in vehicle-related data. This solution will allow us to retrieve adequately detailed information from smaller or compromised datasets through an algebraic interpolation process that intends to effectively fill the missing information gaps without changing the overall essence of the actual dataset. Depending on the use case, the proposed solution could be used for analytics retrieved from artificially reconstructed data or simply as a method of lowering the threshold of the necessary amount of data needed for a specific decision.

The rest of this paper is organized as follows: Section II identifies our motivation and previous work on the topic. Section III describes the mathematical method used for the vehicle-oriented data reconstruction process. Section IV presents the testbed architecture while the necessary results are included in Section V. Finally, Section VI concludes the paper.

II. Motivation and Previous Work

Over the last few years, cloud computing evolved to a highly disruptive architectural paradigm which clearly dominated the global market of on-demand computational resources, allowing end users to efficiently lease them according their needs on any given time [5],[6]. However, most service providers deploy massive geographically isolated datacenters, leading to a large round trip delay, network congestion or service quality degradation, all highly compromising factors for real-time, latency-sensitive service requests and applications.

An initial approach for tackling issues such as increased delay and elevated latency was eliminating distance between the data source and the corresponding computational resources that handle them as proposed by [7]. This work introduced the notion of data processing in the network edge through dedicated nodes referred to as “edge devices”. Alas, despite its efficiency, this approach does not comply with the common deployment paradigms of cloud computing followed by every vendor, therefore an alternative solution was needed, however with respect to the notion of *edge computing* which was proved to be a step towards the right direction.

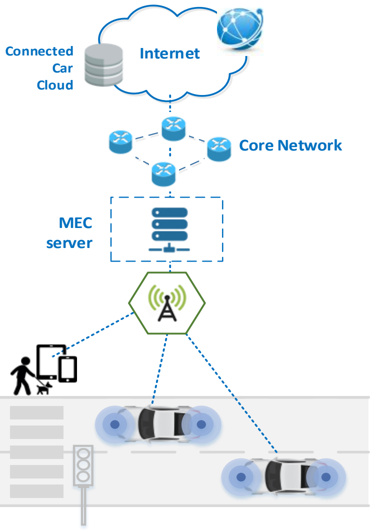


Fig. 1: V2X communications over Multi-access Edge Computing Infrastructure in a Smart City

The most recent architectural model for providing cloud computing capabilities paired with an IT service environment at the edge of the mobile network within the Radio Access Network (RAN) and in proximity to the service subscribers is Multi-access Edge Computing formerly known as Mobile Edge Computing (MEC) [8]. This novel framework is characterized by ultra-low latency, high bandwidth, real-time access to radio network and context information, location awareness, efficient network operation and service delivery, thus ensuring high quality of experience (QoE) for all interconnected users . According to the European 5G Public Private Partnership (5G-PPP)[1] MEC facilitates the actual transformation of the mobile broadband network into a programmable ecosystem and paves the way towards meeting the original standards of 5G in terms of expected throughput, latency, scalability and automation.

To further clarify the abilities of the proposed architecture, European Telecommunications Standards Institute (ETSI) created an industry specification group (ISG) which in [9] describes a few service scenarios under consideration that fully take advantage of MEC towards increasing performance compared providing such services through the cloud or core network servers. As expected, the Smart City-relevant, Vehicle-to-Infrastructure/Pedestrian/Vehicle (V2X) use case was among the dominant ones, since it uses MEC to extend the connected car cloud into the highly distributed mobile base station environment, and enable data and applications to be housed close to the vehicles. This can reduce the round trip time of data and enable a layer of abstraction from both the core network and applications provided over the internet. MEC applications can run on corresponding MEC servers which are deployed at the base station site to provide roadside functionality. The MEC applications are designed to receive local messages directly from vehicle and roadside sensors, analyze them and then propagate (with extremely low latency) hazard warnings and other latency-sensitive messages to all interconnected devices, as depicted in Figure 1. This enables for instance a nearby car to receive data in a matter of milliseconds, allowing the driver to immediately react.

However a serious issue still remains; what happens when, regardless the reason, the obtained vehicle-oriented sensor datasets are not adequate for proper analysis? Or similarly, which is the baseline above which problematic situations can be identified, taking into consideration that network conditions may compromise the transmission of the whole set of sensor information from the vehicles. This thorny issue could be efficiently tackled by employing Matrix Completion techniques [10] as analyzed in the following section.

III. Low rank approximations in the presence of missing data

A. *Preliminaries on Matrix Completion*

Assuming that we are given a low-rank incomplete matrix $\mathbf{V}$ with dimensions $m \times n$, then Matrix Completion

[1] https://5g-ppp.eu/

(MC) refers to the problem of reconstructing the values of the data matrix when only a small subset of its entries is available. More specifically, given $\mathbf{V} \in \mathbb{R}^{m \times n}$ with rank R and a set of known entries K , then the solution of the optimization problem

$$\begin{aligned} &minimize\ \tau \|\mathbf{Y}\|_* \\ &s.t.\ \mathbf{Y}_{ij} = \mathbf{V}_{ij}, i, j \in K \end{aligned} \tag{1}$$

where τ is a weighting parameter and $\|\mathbf{Y}\|_*$ is the nuclear norm of $\mathbf{Y}$ in (1) can precisely recover the content of matrix $\mathbf{V}$. The nuclear norm is formulated as

$$\|\mathbf{Y}\|_* = \sum_k \sigma_k(\mathbf{Y}) \tag{2}$$

where $\sigma_k(\mathbf{Y})$ is the k-th singular value of Y. As the authors in [11] state, for a matrix sampled with a random process with n entries, so that $n \geq c_1 N^{5/4} Rlog(N)$ then with a probability $1 - c_2 N^{-3} logN$ the solution of (1) is equal to $\mathbf{V}$ where c_1 and c_2 are constants. To perform the minimization of (1) the singular value thresholding (SVT) algorithm [12] is employed. The SVT operator $D_\tau(\mathbf{Z})$ is presented in [12]. Given $\mathbf{Z}$ the dual variable of the Lagrangian, the minimizer of (1) is evaluated via the iterative execution of the following equations:

$$\mathbf{Y}_l = \mathcal{D}_\tau(\mathbf{Z}_{l-1}) \tag{3}$$

$$\mathbf{Z}_l = \mathbf{Z}_{l-1} + \Omega \circ (V - Y) \tag{4}$$

for $l = 0, \ldots, l_{max}$. Moreover, matrix Ω is defined as

$$\Omega = \begin{cases} 1 & i, j \in K \\ 0 & otherwise \end{cases}$$

B. Application to one dimensional signals

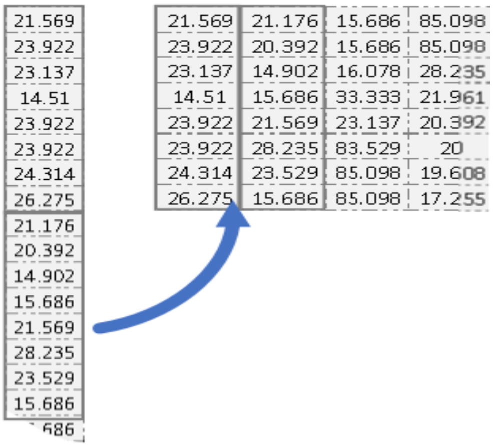
To be able to apply the aforementioned approach in the case of time-series, we initially rearrange the one dimensional data into square or rectangular matrices. To be more specific, we initially construct matrix columns by concatenating the measurements $v_1, v_2, \cdots, v_{n \cdot m}$ as presented in Figure 2,

$$\mathbf{V} = \begin{bmatrix} v_1 & v_{m+1} & \cdots & v_{(n-1)m+1} \\ v_2 & v_{m+2} & \cdots & v_{(n-1)m+2} \\ \vdots & \vdots & \ddots & \vdots \\ v_m & v_{2m} & \cdots & v_{n \cdot m} \end{bmatrix} \tag{5}$$

To be able to evaluate the reconstruction ability of the aforementioned MC approach to the current dataset, we need to identify the required number of singular values k that minimize the nuclear norm of $\mathbf{V}$

$$\sum_1^k \sigma_k(\mathbf{V}) \approx \sum_1^{min(n,m)} \sigma_k(\mathbf{V}) \tag{6}$$

so that



21.569
23.922
23.137
14.51
23.922
23.922
24.314
26.275
21.176
20.392
14.902
15.686
21.569
28.235
23.529
15.686
̇686

21.569	21.176	15.686	85.098
23.922	20.392	15.686	85.098
23.137	14.902	16.078	28.235
14.51	15.686	33.333	21.961
23.922	21.569	23.137	20.392
23.922	28.235	83.529	20
24.314	23.529	85.098	19.608
26.275	15.686	85.098	17.255

Fig. 2: Matrix formulation

$$\frac{\sum_1^k \sigma_k(\mathbf{V})}{\sum_1^{min(n,m)} \sigma_k(\mathbf{V})} \geq 0.98 \tag{7}$$

In order to utilize more effectively the temporal correlation of the vehicle sensor data measurements in the time domain we further impose Laplacian constraints in the initial optimization problem in order to move the missing sample v_i with index i to the weighted average of its known neighbors:

$$x_i - \frac{w_r}{w_l + w_r} x_{i-w_l} - \frac{w_l}{w_l + w_r} x_{i+w_r} \tag{8}$$

where the weight w_r is equal to the temporal distance between x_i and the next available entry and correspondingly w_l denotes the temporal distance between x_i and the previous available entry. The aforementioned operation can be expressed in matrix form by formulating the Laplacian matrix $\mathbf{L}$ of dimensions $nm \times nm$ defined as

$$\mathbf{L} = \begin{cases} 1 & i \notin K, j = i \\ -\frac{w_r}{w_l + w_r} & i \notin K, j = i - w_l \\ -\frac{w_l}{w_l + w_r} & i \notin K, j = i + w_r \\ 0 & otherwise \end{cases}$$

To minimize the distance between the i-th component of the reconstructed time series x_i and the weighted mean y_i of the available nearest neighbours we need to minimize the following expression:

$$\|\mathbf{L}vec(\mathbf{Y})\|_2^2 \tag{9}$$

C. Matrix completion with Laplacian constraints

Therefore, the nuclear norm minimization problem after the addition of the Laplacian constraint can be written as follows:

$$\min_{\mathbf{Y}} \|\Omega \circ (\mathbf{Y} - \mathbf{V})\|_F + \tau \|\mathbf{Y}\|_* + \mu \|\mathbf{L}vec(\mathbf{Y})\|_2^2 \tag{10}$$

where the first term minimizes the error between the known values and the estimated, the second term imposes the low rank constraint to the recovered matrix and the last term moves the estimated value to the weighted average of the nearest available neighbours. $\mathbf{Y}$ is the optimization variable, the parameter μ is the regularization parameter for the Laplacian constraint and $\|.\|_F$ corresponds to the Frobenius norm.

The Lagrangian of the splitting version of the optimization problem in (10) can be written as

$$\begin{aligned}\mathcal{L}(\mathbf{Y},\mathbf{U},\mathbf{Z}) = \frac{1}{2}\|\Omega\circ(\mathbf{U}-\mathbf{V})\|_F^2 + \tau\|\mathbf{Y}\|_* + \\ + \mu\|\mathbf{L}vec(\mathbf{U})\|_2^2 + \frac{\rho}{2}\|\mathbf{Y}-\mathbf{U}\|_F^2 \quad (11)\end{aligned}$$

where $\mathbf{U}$, $\mathbf{Z}$ are the dual variables and ρ is the penalty parameter. The solution of (11) can be efficiently obtained after using the Alternating Direction Method of Multipliers (ADMM) [13] which can be summarized into the following steps:

$$\mathbf{Y}(l+1) = arg\min_{\mathbf{Y}} \mathcal{L}(\mathbf{Y},\mathbf{U}(l),\mathbf{Z}(l)) \quad (12)$$

$$\mathbf{U}(l+1) = arg\min_{\mathbf{U}} \mathcal{L}(\mathbf{Y}(l+1),\mathbf{U},\mathbf{Z}(l+1)) \quad (13)$$

$$\mathbf{Z}(l+1) = \mathbf{Z}(l) + \rho(\mathbf{Y}(l+1)-\mathbf{U}(l+1)) \quad (14)$$

For equation (12) the minimizer is given by the SVT operator $\mathbf{Y}(l+1) = D_\tau(\mathbf{U}(l)-\rho^{-1}\mathbf{Z}(l))$.

The minimizer of equation (13), it can be easily shown that is given by the following formulation:

$$\Omega\circ(\mathbf{U}-\mathbf{V}) + \mu(\mathbf{L}^T\mathbf{L})\mathbf{U} + \rho(\mathbf{U}-\mathbf{Y}-\rho^{-1}\mathbf{Z}) = 0 \quad (15)$$

The solution of (15) is found after executing iteratively for $l = 0,\dots,l_{max}$ the following steps:

1) Equation (15) is formulated as

$$\begin{aligned}\left[D(vec(\Omega)) + \mu(\mathbf{L}^T\mathbf{L}) + \rho\mathbf{I}\right] u = \\ vec(\Omega\circ\mathbf{V} + \rho\mathbf{Y} + \mathbf{Z}) \quad (16)\end{aligned}$$

2) Compute

$$\mathbf{Y}(l+1) = D_\tau(\mathbf{U}(l)-\rho^{-1}\mathbf{Z}(l)) \quad (17)$$

3) Employ over-relaxation parameter γ so that

$$\mathbf{W} = \gamma\mathbf{Y}(l+1) - (1-\gamma)\mathbf{U}(l)) \quad (18)$$

4) Solve

$$\begin{aligned}\left[D(vec(\Omega)) + \mu(\mathbf{L}^T\mathbf{L}) + \rho\mathbf{I}\right] u = \\ vec(\Omega\circ\mathbf{V} + \rho\mathbf{W} + \mathbf{Z}(l)) \quad (19)\end{aligned}$$

5) Reshape $\mathbf{u}$ to matrix $\mathbf{U}$
6) Set $\mathbf{Z}$ equal to

$$\mathbf{Z}(l) + \rho(\mathbf{Y}(l+1)-\mathbf{U}(l+1)) \quad (20)$$

IV. System Architecture

For properly evaluating the aforementioned method through real data from vehicle sensors the system architecture of Figure 3 was deployed. Vehicle-oriented data collected by the embedded sensors of the car were obtained using an On-Board Diagnostics (OBD) module, supported by every major manufacturer following specific EU regulatory guidelines. Without the loss of generality, only data regarding vehicle speed, engine rounds-per-minute (RPM) and throttle position were collected for creating matrices stored in a per-trip fashion. The OBD module was connected over Bluetooth protocol to an Android smartphone, on which a specialized, tailor-made application was running[2]. This application acted as a data aggregator that accumulated sensor values, added a timestamp and created a .CSV file. After each trip, the application transmitted the .CSV file through throttled 802.11g/4G LTE connections for emulating network congestion into a specially designed [14] online repository for further process. The online data undergo the process described in the previous section for accurately evaluating the effect of Matrix Completion method.

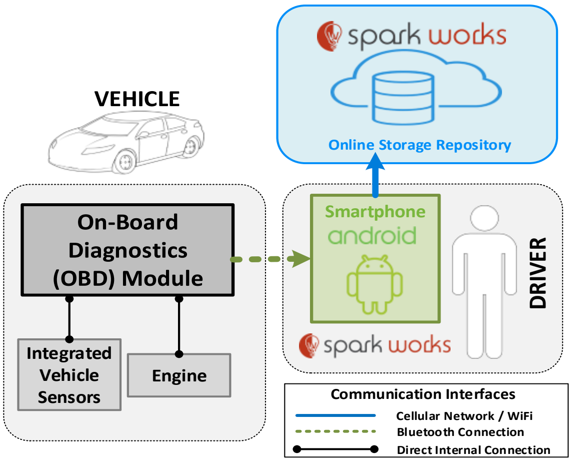


Fig. 3: System Architecture

V. Experimental Results

This section provides the performance analysis for the proposed method described in section III. Experiments were carried out after using real-world data of vehicle speed, engine RPM and throttle position. The matrices were filled with data corresponding to six driving sessions of the same driver over the same route. The time series of all driving sessions were synchronized according to the time the vehicle started, while the duration of each driving session was approximately 26 minutes.

In order to execute matrix completion, the time series of data were reshaped in a matrix of size 138 × 138 for all three types of sensor data namely speed, engine

[2] https://play.google.com/store/apps/details?id=app.gamecar.sparkworks.net.gamecardatalogger

RPM and throttle position. Each column of this matrix corresponds to approximately 1 minute of driving. The matrices with missing entries were obtained from the initial matrices of data after the removal of values at random. The missing entries percentage started from 5% and reached 60% with 5% step. For each percentage of missing values the proposed approaches are executed in 100 different permutations. The final result is the mean value of all permutations results per missing percentage value.

The reconstruction accuracy was evaluated using the Normalized Mean Square Error and the Normalized Root Mean Square Error for the different number of missing entries for the three aforementioned cases.

The first case study, related to the RPM measurements showed that the matrix of engine RPM data has high correlation thus resulting in a low rank matrix as can be easily seen by inspecting the eigenvalue distribution in Figure 4. This attribute explains also the accurate reconstruction of the matrix with even 60% missing entries, justified by the relatively small NMSE that is presented in Figure 5.

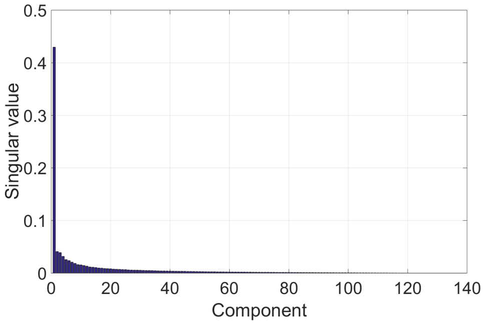
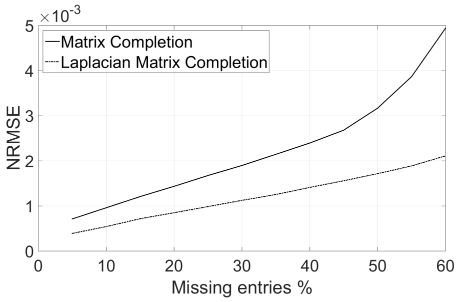


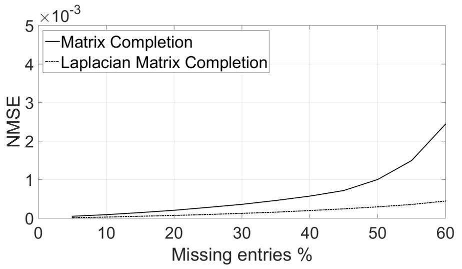
Fig. 4: Eigenvectors importance for matrix of engine RPM data.

The matrix completion approaches provide better reconstruction results in the case of vehicle speed data. The main reason is again the low rank of the vehicle speed matrix Figure 6. It is worth noting that the NMSE is close to zero even with 60% missing entries as shown in Figure 7.

The rank of the throttle position matrix is not as low as in the previous two cases of engine RPM and vehicle speed Figure 8. The results in this case are worse, but the difference between the regularized and the conventional MC approach remains obvious Figure 9.



(a)



(b)

Fig. 5: Comparison of proposed Laplacian based MC with conventional MC at various percentages of known entries for engine RPM dataset:(a) NRMSE (b) NMSE

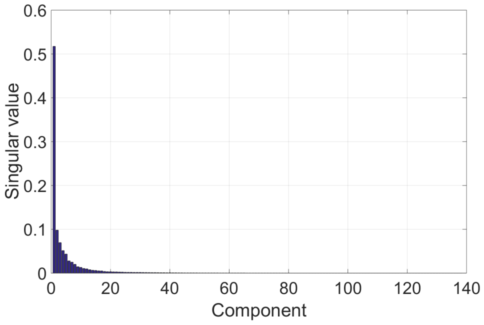


Fig. 6: Eigenvectors importance for matrix of vehicle speed data.

VI. Conclusions

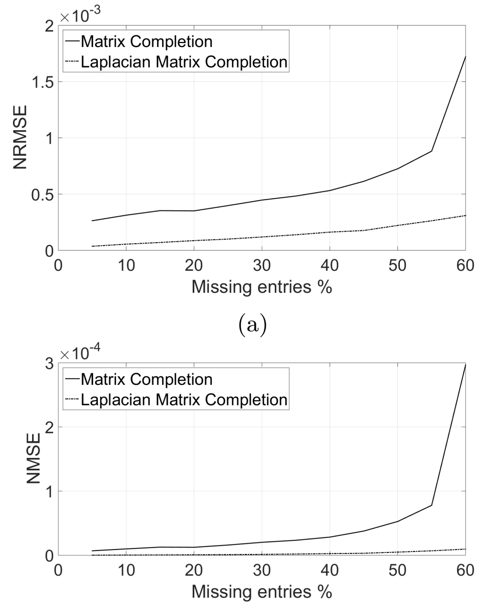
This paper introduced the Laplacian Matrix Completion method and demonstrated is superior performance compared to MC in three cases of vehicle-related sensor data. At 60% missing entries the Laplacian Matrix Completion reconstructs the matrices with half NMSE of conventional Matrix Completion.

By capitalizing on the low rank property of the sensor data retrieved by the OBD module, we permeat benefits from low rank matrix completion, allowing the reconstruction of the vehicle speed, RPM and throttle position from a small subset of the recorded measurements. The exploitation of this property could potentially allow the identification of data outliers using robust pca, since it also exploits the low rank property of the data.

The proposed approaches are expected to manage effectively several uncertainties that could be attributed to sensing and communication failures, minimizing at the same time the storage and transmission requirements of Multi-access Edge Computing applications.

Acknowledgement

This work was partly funded by European Union's Horizon 2020 research and innovation programme GamECAR under grant agreement No 732068



(a)

(b)

Fig. 7: Comparison of proposed Laplacian method with conventional MC at various percentages of known entries for vehicle speed dataset:(a) NRMSE (b) NMSE

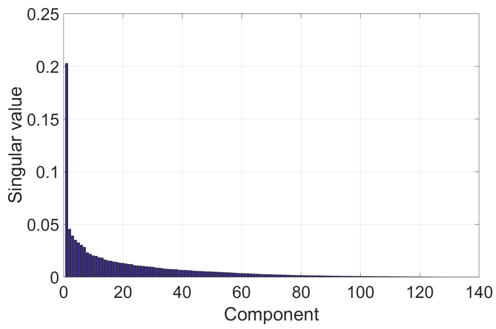
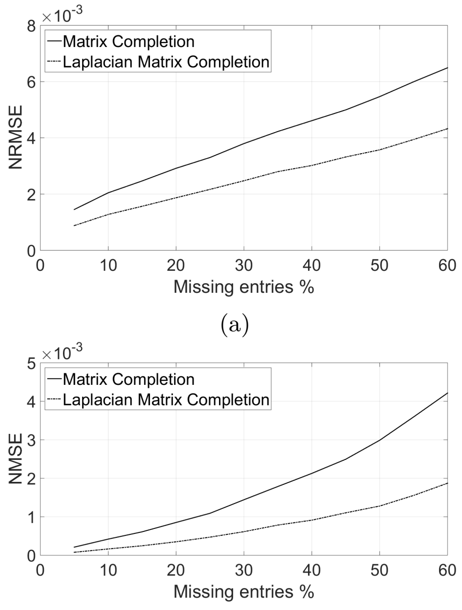


Fig. 8: Eigenvectors distribution for matrix of throttle position data.



(a)

(b)

Fig. 9: Comparison of proposed Laplacian with conventional MC at various percentages of known entries using the throttle position dataset:(a) NRMSE (b) NMSE

References

[1] G. Bianchi, E. Biton, N. Blefari-Melazzi, I. Borges, L. Chiaraviglio, P. Cruz Ramos, P. Eardley, F. Fontes, M. J. McGrath, L. Natarianni *et al.*, “Superfluidity: a flexible functional architecture for 5g networks,” *Transactions on Emerging Telecommunications Technologies*, vol. 27, no. 9, pp. 1178–1186, 2016.

[2] C. Tselios and G. Tsolis, “On QoE-awareness through Virtualized Probes in 5G Networks,” in *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2016 IEEE 21st International Workshop on*, 2016, pp. 1–5.

[3] E. Vlachos, A. S. Lalos, K. Berberidis, and C. Tselios, “Autonomous driving in 5g: Mitigating interference in ofdm-based vehicular communications,” in *2017 IEEE 22nd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, June 2017, pp. 1–6.

[4] L. Hobert, A. Festag, I. Llatser, L. Altomare, F. Visintainer, and A. Kovacs, “Enhancements of v2x communication in support of cooperative autonomous driving,” *IEEE Communications Magazine*, vol. 53, no. 12, pp. 64–70, Dec 2015.

[5] C. Tselios, I. Politis, V. Tselios, S. Kotsopoulos, and T. Dagiuklas, “Cloud Computing: A Great Revenue Opportunity for Telecommunication Industry,” in *FITCE Congress (FITCE), 51st, 6, Poznan, Poland*, 2012.

[6] C. Tselios, I. Politis, K. Birkos, T. Dagiuklas, and S. Kotsopoulos, “Cloud for multimedia applications and services over heterogeneous networks ensuring QoE,” in *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2013 IEEE 18th International Workshop on*, Sept 2013, pp. 94–98.

[7] P. Garcia Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, and E. Riviere, “Edge-centric computing: Vision and challenges,” *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 5, pp. 37–42, Sep. 2015.

[8] P. Mach and Z. Becvar, “Mobile edge computing: A survey on architecture and computation offloading,” *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1628–1656, Q3 2017.

[9] ETSI GS MEC 004, “Mobile Edge Computing (MEC): Service Scenarios,” V.1.1.1, March 2016.

[10] R. H. Keshavan, A. Montanari, and S. Oh, “Matrix completion from a few entries,” *IEEE Transactions on Information Theory*, vol. 56, no. 6, pp. 2980–2998, June 2010.

[11] E. J. Candès and B. Recht, “Exact matrix completion via convex optimization,” *Foundations of Computational mathematics*, vol. 9, no. 6, p. 717, 2009.

[12] J.-F. Cai, E. J. Candès, and Z. Shen, “A singular value thresholding algorithm for matrix completion,” *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[13] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[14] O. Akrivopoulos, I. Chatzigiannakis, C. Tselios, and A. Antoniou, “On the deployment of healthcare applications over fog computing infrastructure,” in *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2, July 2017, pp. 288–293.