2017 6th IEEE International Conference on Advanced Logistics and Transport (ICALT)

Optimizing V2X Data Collection and Storage for a Better Cost and Quality Trade-off

Mohamed Ben Brahim†§, Hamid Menouar†
† Qatar Mobility Innovations Center (QMIC), Qatar University, Doha, Qatar
§ HANA Research Lab, University of Manouba, Manouba, Tunisia
Email: {mohamedb, hamidm}@qmic.com

***Abstract*—Future vehicles will be equipped with advanced communication capabilities and a multitude of sensing devices. Vehicle-to-vehicle and to Infrastructure (V2X) is one of these future technologies. V2X-technology-enabled vehicles are expected to become a great source of big data. This data, if gathered in the right time and processed in the right way, can enable an interesting number of existing and new applications. This can be a challenging task, taken into account the considerable size of the data that will be gathered. One of the challenges is to find a good balance between the number of data to filter out and the quality of the end data. This contribution tackles this specific challenge, by studying data storage cost reduction and evaluating its impact on the data quality. The proposed solution compares three approaches of treating the collected data at the road-side unit after taking out unnecessary information details. This solution has been tested and validated through simulations that show promising results.**

***Keywords*—V2X wireless communication, vehicular edge computing, data storage, data quality, dimension reduction, data sampling.**

I. Introduction

The deployment of intelligent sensing devices amongst vehicles manufacturers is currently in active stage. Several sensors are constantly measuring the vehicle status parameters and provide them to an embedded control board. Some of these sensors' outputs serve for in-vehicle driver assistance, for example by providing engine status, battery voltage level, pre-crash warnings, and braking system status. Besides, the camera-based and the GPS sensors underlay many routing and travel planning applications. Despite the important role of these embedded systems in increasing the driver awareness and enhancing the traffic safety, their efficiency is still limited. Indeed, cameras and laser or ultrasonic sensors for pre-crash warning are useful only within short range and in non obstructing space. In non-controlled blind intersections, for example, the radar-based or camera-based warnings for collision avoidance could be raised only when the collision becomes inescapable. Due to all these reasons, the V2X wireless communication between vehicles and with road side infrastructure is proposed as the next generation of Intelligent Transport Systems (ITS). In this emerging technology, a vehicle is equipped with an on-board unit (OBU) which collects data from the in-vehicle systems and sensors. The OBU holds a set of automotive and ITS applications that aggregate the locally collected data and exchange them over the air with the other stations around through well defined communication policies. A Neighbor station could be either a peer vehicle that analyses in real-time the received data and raises the relevant warning to the driver or generates a command to the concerned actuator. The road side unit (RSU) communicates directly with OBUs and normally interfaces with the traffic control centers or other central units. The RSU is the network edge in the vehicular networks. Through a reliable link with the backend servers and service providers, the RSU can provide many services to the passing by vehicles. It is also a collection point i.e. sink, for gathering data from other mobile ITS stations i.e. passing by OBUs.

The data messages generated by road users could be either proactive or reactive i.e. event-triggered. Proactive messages are time and motion-based periodic where the inter packet delay is bounded by a predefined duration and controlled by the motion evolution of the ITS station (ITS-S). A very common example of this kind of applications is the Cooperative Awareness (CA) basic service [1] in the European ITS standards and the Basic Safety Message (BSM) [2] in US standards. The reactive messages are numerous. A good example is the Decentralized Environment Notification (DEN) basic service [3]. The DEN messages are transmitted by different kind of ITS stations to inform other road users about traffic events' details. The data gathered by the RSU from the other ITS stations could undergo some early pre-processing routines. In fact, the raw data often present one or more of the following proprieties: redundant information, missing data chunks, and over needed information for the application requirements. Performing a data pre-processing treatment in the network edge level, known as vehicular edge computing, will reduce the storage and processing cost in the remaining phases of the data pipeline. Particularly, a data sampling from the intensive amount of received data while preserving a comprehensive level of data accuracy is useful to save the required data storage space. Henceforth, we tackle the gain in storage space through three different approaches and their respective accuracy in representing the original raw data upon data offline processing. The three approaches are summarized below:

- The time-fixed sampling approach (TF) is a simple algorithm in which a message container is created and filled with the last received information before being transmitted to the backend database. The study investigates different sampling period values and their performance in terms of data size reduction and compares resulting data accuracy with respect to the original data.

- The vehicle dynamics (kinematics) based approach (KM) is a motion-aware sampling algorithm that considers different factors such as latitude, longitude, driving direction or heading, speed variation, and road lane number. A simulation-based inference of the vehicle dynamics thresholds for data sampling is carried out. Then, an algorithm is developed using these determined parameters to reduce the overall data size while ensuring a comprehensive data quality.

• The dimension reduction based approach picks some messages periodically and tries to reduce the data of the messages that have been received between two bounding messages. The content of the successive packets are organized in the columns of a matrix of offsets and a singular-value decomposition (SVD) is performed [4] to substantially reduce stored data.

To illustrate the details of the proposed approaches and highlight their impact on the storage space and the data quality, the remainder of this paper is organized as follows: Section II provides an overview of the related work where data mining for correlated data in vehicular networks and data sampling are presented. The vehicular data packets stream is detailed as well. Section III details the proposed approaches, and Section IV presents the simulation work and the obtained performance results. Finally, Section V concludes the paper and highlights future works.

II. Related work

The recent mobile networks are characterized with a big amount of generated data by mobile terminals. This trend raises several challenges in terms of managing the shared wireless bandwidth or other domain-specific resources like energy in sensor networks [5]. The literature includes several works which tackled the possibility of mining streams of generated data using techniques in the mobile terminals or in the network edge level.

A. Data Mining for Correlated Data in V2X

Other than conventional compression techniques which have known data reduction ratios based on the type of the compressed data and the underlying compression algorithm, different big data techniques exist which aim at summarizing the data or transforming the data to reduce their dimensions. In [6], authors tackled the problem of data dimensionality reduction through usage of data fusion approach based on Fisher and deep auto-encoder learning. The proposed approach helps in reducing the complexity and in improving the data processing phase. Indeed, the classification algorithms outperform when applying the data fusion than without data fusion. Another work is presented in [7] where a distributed data mining system was built and tested. The mobile distributed system collects and aggregates data in the vehicle scale to perform most of the processing tasks locally for monitoring vehicle and driver characterization. This ubiquitous solution makes improvements particularly in the bandwidth usage and computation efficiency. Although this approach is interesting in mining exchanged data, it doesn't fit our case requirements. Indeed, the safety constraints don't allow us to reduce the frequency of the traffic and motion awareness data specifically for in-network real-time interaction. Hence, working on the RSU level doesn't disturb the network performance. Moreover, making a hard computation in the vehicle to reduce the data to be routed to backend servers may end with a lost packet. Thus, we think that offloading the computation and aggregation tasks to the edge of the reliable part of the network is more reasonable. Another ubiquitous approach based on compressive sensing to minimize the amount of transmitted data over the air [8]. This technique leverages the spatial correlation within data and the data decoding relies on this correlation to accurately recover the original data. This technique is important from bandwidth perspective, however it has almost the same limitations as the previous technique for our case.
Data sampling is also another way to do for reducing the data storage space. Compressive sampling is largely used technique in image and signal processing domains [9] [10] [11] [12]. They have the advantage of reducing the data size and preserving a good approximation of the original data. In this work, we will leverage the sampling technique and the data dimensionality reduction to lower the required storage cost and maintain a similarity between the original data and the pre-processed (i.e, sampled) data. Three algorithms will be described and mutually compared.

B. CAM Data Stream Description

In the European ITS standards, the Cooperative Awareness basic service is the heartbeat protocol of the network in ITS. Every ITS-station running this basic service is expected to send at least one packet every second to inform its neighbors about its presence and current state. The vehicle state is well described through the formatted CA message (CAM). An overview of CAM is presented below.

1) CAM Packet Structure Overview: CA message is described by the ETSI Technical Committee Intelligent Transport Systems [1]. The exchange of CA message is intended to increase the awareness of the road users about each other and enhance the traffic safety. Each CAM originating station includes the recently collected values of station position, dynamics, and other attributes and broadcasts them wirelessly in one single hop range. The CA packet is composed of an ITS PDU header and four different data containers Basic and High Frequency Containers are mandatory and should be present in every transmitted packet. Low Frequency and Special vehicle containers are, however, conditional and might be omitted from some packets. This structure of CAM allows us to distinguish between two types of CAM: The basic CAM (B-CAM) which is the reduced form containing the mandatory containers and the extended CAM (E-CAM) which includes the mandatory and the optional containers. Hereinafter, we adopt these names to refer to extended and basic forms of CAM vehicular data.

2) CAM Generation frequency: One of the important aspects of the CA basic service is the management of packet transmission frequency. Indeed, different factors are affecting the inter-packet time and packets are generated at a frequency ranging from 1Hz to 10Hz. These factors include the originating ITS station dynamics, the radio channel congestion control process, and eventually application requirements. The stream of generated packets is an alternation of E-CAM and B-CAM. According to [1], if the speed, driving heading, or position varies above a pre-defined threshold, a CA message is generated if the the other factors constraints are in favor. After all, a message should be sent every second at least even the vehicle is in stopping status to ensure awareness to peer road users.

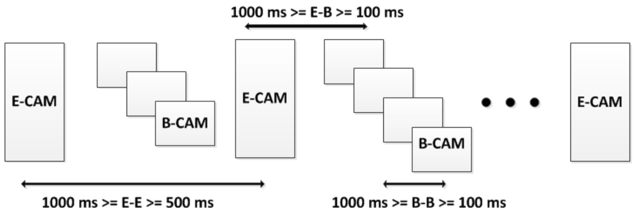

Fig. 1: CAM packets stream

Figure 1 illustrates the inter-packets time constraints in the CA basic service. At a metropolitan deployment scale, we could imagine the amount of data collected by deployed RSUs and the required storage space to handle it. Facing this challenge, we propose to reduce the stored data while keeping a fair accuracy of tracking information. These approaches are explicitly presented in the next section.

III. Proposed Approaches

A. Assumption and preliminaries

In the scope of this work, it is assumed that the main source of vehicular data is the CA messages. In addition, Since the original CAM generation algorithm considers different time and vehicle motion based parameters, which are imposed by different tiers components of the communication stack (application, management), we will simplify the simulation by considering only the vehicle motion-related parameters, namely, speed, position, and driving heading variations.

Singular-value decomposition: The singular value decomposition (SVD) of a real or a complex matrix $M(m \text{ x } n)$ consists on its factorization as follows:

$$M = USV^* \tag{1}$$

where

- U is a m x n, unitary matrix,
- S is a diagonal m x n matrix with non-negative real numbers on the diagonal representing the singular values (s_1, s_2, ...s_m), and
- V^* is a n x n, unitary matrix. It's the conjugate transpose of the n x n unitary matrix, V.

The k-truncated SVD consists on considering the k most important singular values and vectors of the representation to produce more compact approximation with a desired number of dimensions.

B. Problem statement

The problem faced in this paper is the management of the enormous amount of real-time data received by the RSU before transferring it to the traffic center database. For a caching period of time within the RSU, the stream of CA messages could be seen as a sequence of timestamped vectors of data values of E-CAM and B-CAM types. Assuming that the RSU could get the application requirements regarding the data granularity and accuracy, the RSU should find a trade-off to ensure a maximum storage saving while preserving a comprehensive data quality.

C. Approaches description

To tackle this challenge, we opted for three different algorithms. The first obvious attempt is through the usage of a fixed period sampling. The second approach is based on various dynamic parameters of the vehicle. And the last last approach is through leveraging the dimension reduction concept using k-truncated singular value decomposition.

1) TF Algorithm: Based on a predefined sampling period, the time fixed sampling algorithm III.1 keeps caching the last updated information retrieved from the received CA messages, being either under basic format or extended format until the expiration of the sampling period. At that time, the resulting E-CAM is pushed to the backend database and a new sampling period starts.

Algorithm III.1 TF_Algorithm

$C \leftarrow (e_1, b_1, b_2, b_3, e_2...)$ { A cache of received CAMs at RSU}
$S\text{-}CAM$ {sampled CAM container of E-CAM type}
for $cam\text{-}entry \in C$ **do**
  $S\text{-}CAM(timestamp) \leftarrow cam\text{-}entry(timestamp)$
  if $S\text{-}CAM(i) != cam\text{-}entry(i)$ **then**
    $S\text{-}CAM(i) \leftarrow cam\text{-}entry(i)$
  end if
end for
if $isSamplingTime() == TRUE$ **then**
  return $S\text{-}CAM$ {Send the sampled CAM to backend DB}
end if

The performance of this algorithm depends only on the sampling period. Feedback from the application will be used to continuously adjust the sampling time in order to better fill the application requirements.

2) KM Algorithm: To overcome the limitations of the exclusive time based approach, and in order to adapt the sampling frequency to the variation of different vehicle motion parameters, an adaptive sampling algorithm is proposed. It's obvious that when the vehicle is moving straight in a road segment and with a constant speed for a long period of time, which is usually the case for example in the highway traffic, it will be useless to save too many information to track the positions. Instead, far fewer samples of messages could derive the missing CAMs through a low-cost processing. In the opposite scenario, if a vehicle is approaching and crossing an intersection or driving across a road curve, increasing the sampling rate becomes useful to precisely track the variation of the vehicle dynamic parameters. To get rid of the influence of the small measurement fluctuations, the thresholds of vehicle dynamics are considered in the filtering of CAM. Indeed, if the variation is below the $optimal$ thresholds, the parameter is considered to be invariant. These threshold values are empirically inferred and considered in developing algorithm III.2.

The performance of this algorithm depends on several parameters. Feed-backs from the application are expected to continuously tune the sampling parameters thresholds in order to meet the application requirements.

Algorithm III.2 KM_Algorithm

$C \leftarrow (e_1, b_1, b_2, b_3, e_2...)$ {RSU received CAMs cache}
$thresh\_Speed \leftarrow optimal\_speed$
$thresh\_Distance \leftarrow optimal\_distance$
$thresh\_Heading \leftarrow optimal\_heading$
$sampling\_maxTime \leftarrow delta\_time$
$S\text{-}CAM$ {sampled CAM container of E-CAM type}
for $cam\text{-}entry \in C$ **do**
  $S\text{-}CAM(timestamp) \leftarrow cam\text{-}entry(timestamp)$
  if $S\text{-}CAM(i)! = cam\text{-}entry(i)$ **then**
    $S\text{-}CAM(i) \leftarrow cam\text{-}entry(i)$
  end if
end for
if $delta\_speed >= thresh\_Speed \ ||$ $delta\_heading >= thresh\_Heading \ ||$ $delta\_distance >= thresh\_Distance \ ||$ $delta\_time >= sampling\_maxTime$ **then**
  return $S\text{-}CAM$ {Send the sampled CAM to traffic center DB}
end if

Algorithm III.3 SVD_Algorithm

$C \leftarrow (e_1, b_1, b_2, b_3, e_2...)$ {RSU received CAMs cache}
$M \leftarrow [\ ]$ {Matrix of B-CAMs offsets as columns}
$U \leftarrow [\ ]$ {Matrix of SVD resulting left singular vectors}
$S \leftarrow [\ ]$ {Matrix containing SVD resulting singular values}
$V \leftarrow [\ ]$ {Matrix containing SVD resulting right singular vectors}
$S\text{-}CAM \leftarrow C(1)$ {Save the reference E-CAM}
for $i \in 2 \,..\, size(C)$ **do**
  $M(.,i) \leftarrow C(i) - C(i-1)$
  if $isSamplingTime() == TRUE$ && $type(C(i+1)) == E\text{-}CAM$ **then**
    $[U, S, V] \leftarrow svd(M)$ {Perform SVD on matrix M}
    $[U_k, S_k, V_k] \leftarrow k\text{-}truncate(U, S, V)$ {Perform k-truncation on U, S, and V matrices}
    Push $S\text{-}CAM$, U_k, S_k, and V_k to backend DB
    $M \leftarrow [\ ]$
    $S\text{-}CAM \leftarrow C(i+1)$ {Update the reference E-CAM}
  end if
end for

3) SVD Algorithm: The previous two algorithms try to pick a subset of the received CAMs that better approximate the original data but with lower data storage cost. Looking to the problem from another perspective, the problem could be resolved differently. Indeed, the sequence of received CA messages is formed of an alternation of E-CAM and B-CAM. Between two E-CAM, zero, one or many B-CAM messages are generated as shown in Figure 1. the data elements update could be at any part of the message. Hence, we propose to define a sampling time period. Once expired, an E-CAM is picked from the cache and persisted into the backend DB. In between, the received B-CAM are cached in the RSU and appended as columns of a matrix M. Instead of storing the original data values in the matrix, the data value offsets are saved. This will result most of the time in sparse matrices since only few parameters often change between two successive CAMs. At this phase, a k-truncated SVD is performed and only required data is stored as illustrated in algorithm III.3. For example, if four singular values from the matrix S are to be considered, then we save only these four values as well as four columns from matrices U and V. When the matrix M is sparse, we could save space and keep a high accuracy of data. The resulting performance gain will be further discussed later in the performance analysis section.

IV. Performance Analysis

This section describes the simulation environment and discusses the performance results of the proposed algorithms.

A. Simulation Setup

An urban traffic region is considered under the form of a regular grid traffic network. The intersections are controlled with traffic signals and an RSU is deployed in the central junction of the studied area. The mobility traces were generated using the Simulator of Urban Mobility (SUMO) [13] and the algorithms were implemented and tested with Matlab. Since the proposed algorithms are hosted in the RSU rather than in the mobile terminals, we didn't consider the related wireless communication issues and we focus exclusively on the data management aspect. Table I illustrates the configuration parameters that have been used to generate the mobility traces by SUMO, and to run the simulations using Matlab.

TABLE I: Parameter Description

Parameter	Value/Description
Number of simulated vehicles	500
Simulation area dimensions	1km x 1km
Number of junctions	5 x 5
Road segment length	250 m
Number of lanes per segment	3
Simulation time	100 s
RSU placement	Central junction
RSU/OBU DSRC communication range	750 m
Traffic control strategy	Traffic light (default SUMO TLlogic)
Maximum speed	80 km/h
CAM (Basic) packet size	100 byte
CAM (Extended) packet size	600 byte

B. Performance results

In this section, the impacts of the algorithms on the data size and data quality are explored.

1) Storage space saving: The first motivation behind this work is to deal with the exponential increasing size of the collected data. Hence, reducing the storage space is the main goal. Figure 2 shows the overall data size meant to be stored as a function of the sampling period in the TF algorithm. It's clear the drastic decrease of data size with the increase of sampling period. It's worthy to notice that is starting from $1s$ as a sampling period, which is the minimum CAM generation frequency, we could save up to 50% of the storage space compared to the raw data size.

Considering now the vehicle mobility related parameters, the sampling becomes better aware of the changes in the transmitted successive CAMs. Similar to time-based sampling, we want to define the optimal values which play the role of thresholds to different vehicle dynamic

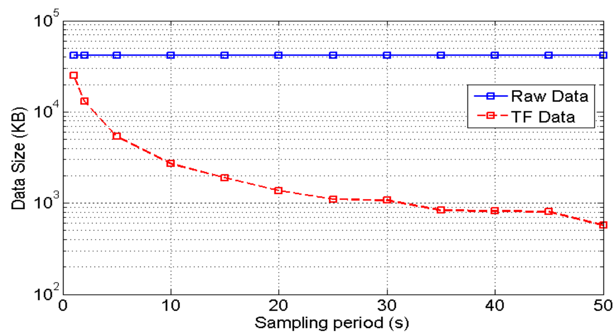

Fig. 2: TF algorithm: data size for different sampling periods

parameters. The thresholds of parameters' variations are meant to make a trade-off between the data size reduction and the data quality preservation. Although this decision is purely application-dependent, we empirically infer the values that should be considered as good-enough. Figures 3, 4, and 5 depict both the data size evolution as well as the resulting data quality as functions of speed variation, driving heading variation, and vehicle position variation, respectively. From these graphs we set the parameters' thresholds which ensure substantial data size reduction around 50% or more, and preserve data quality as higher as possible. The data quality measurement is quantified as the conjugate of the *normalized root mean squared deviation* ($NRMSD$) given by:

$$DataQuality(F) = 1 - NRMSD(F), \quad (2)$$

$$NRMSD(F) = \frac{RMSD(F)}{F_{max} - F_{min}}, \quad (3)$$

$$RMSD(F) = \sqrt{E((F_m - F_e)^2)}, \quad (4)$$

where F is a given feature, F_m and F_e are respectively the measurement and the estimation values of the feature F, and $E(.)$ is the statistical variance function.

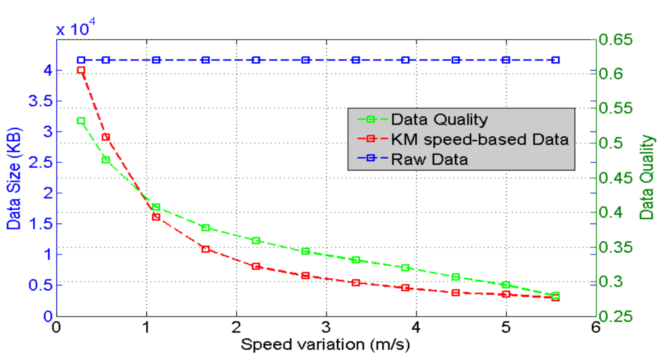

Fig. 3: KM speed-based sampling and data quality

From hereinafter, we consider the following parameters' threshold values to be used in the algorithm III.2.

- $thresh\_Speed = 0.825\ m/s$
- $thresh\_Heading = 20\ degree$

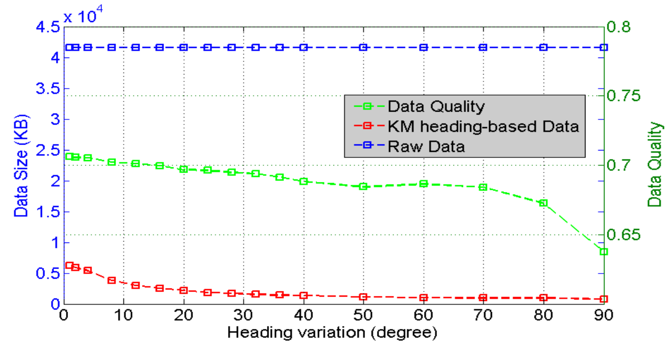

Fig. 4: KM heading-based sampling and data quality

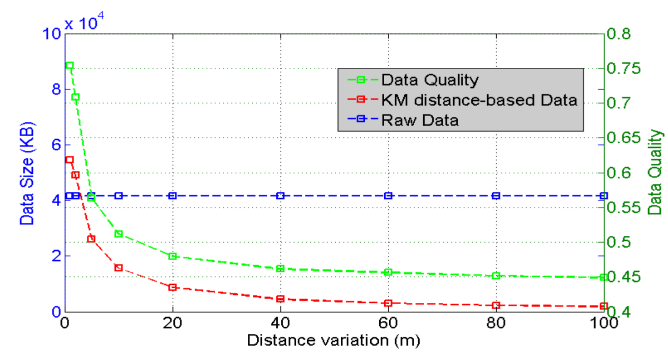

Fig. 5: KM position-based sampling and data quality

- $thresh\_Distance = 15\ m$

Setting $thresh\_Speed$ yields around 44% data quality and 47% data size reduction as illustrated in Figure 3. The $thresh\_Heading$ value setup ensures around 70% data quality and 94% data size reduction depicted by Figure 4. Finally, assigning the $thresh\_Distance$ a value of $15m$ leads to 50% data quality and 70% data size reduction as could be seen in Figure 5.

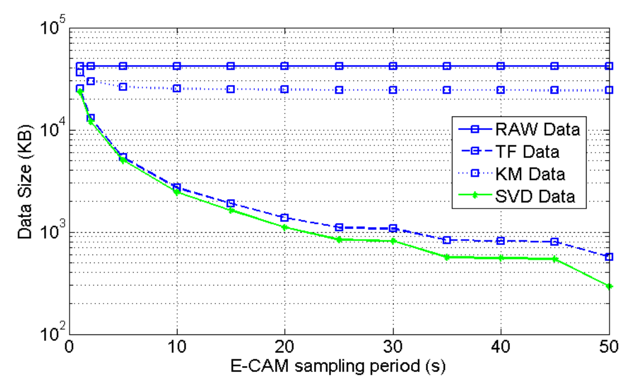

Fig. 6: Data storage size of different approaches

Once the parameters *optimal* values are set and the inferred thresholds values are injected into the KM algorithm, the outcomes of the three algorithms in term of data size saving are measured and illustrated in Figure 6. For all the algorithms, the only variable parameter is the sampling time period, since

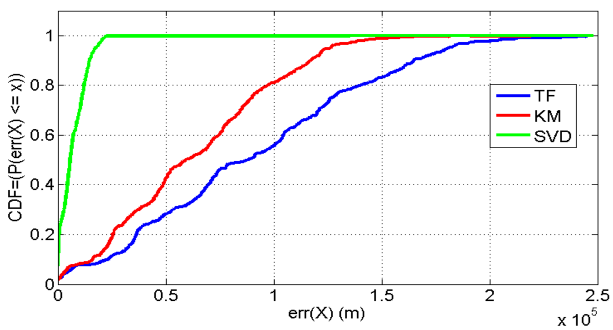

Fig. 7: X-coordinate data accuracy

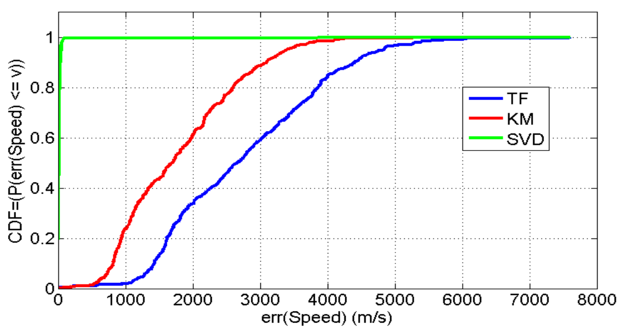

Fig. 8: Speed data accuracy

it is the common parameter impacting the performance of the three algorithms. The curves show the behavior of the SVD-based algorithm versus other candidates. It is understood why SVD algorithm performs better than KM algorithm. In fact, starting from a given sampling time period, around $10s$, the increase of the sampling window will have no impact on KM algorithm performance. This is because the decision of whether to save or not the data becomes controlled by other vehicle motion parameters rather than simple time-based sampling frequency. However, for TF based algorithm, the relative extra data size versus the SVD algorithm is resulting from the fact that the former is always saving the extended format of the data message which adds more data containers to memorize the last collected values between two successive sampling hits. This behavior is enhanced in the SVD algorithm through the k-truncated SVD to reduce the storage space.

2) Data accuracy: Figures 7 and 8 illustrate the cumulative distribution function of the error of the x-coordinate (e.g. Latitude) and speed of the tracked vehicles, respectively. The graphs present different outcomes regarding the data quality. The graphs confirm the higher performance of the SVD algorithm against the two other algorithms. It is also clear that the KM algorithm outperforms the TF algorithm. This is due to its adaptive behavior to the vehicle motion. Nevertheless, both KM and TF algorithms still far low accurate than the SVD one.

V. CONCLUSION

Vehicle to vehicle and to Infrastructure (V2X) is about to head the market, and it is expected to be a real big data source. A large amount of data will be generated by each vehicle, and this data can be extremely useful if it is well captured and analyzed. No doubt, managing such a large amount of data will be challenging. Therefore, this contribution proposes three algorithms to reduce the size of this data by aggregating the gathered information while keeping the information quality and completeness at an acceptable level. The proposed solutions have been validated by simulations, and the obtained results are highly interesting. A future extension will compare the performance of the SVD-based algorithm with the compressive sampling approach, largely adopted in the imaging and signal processing scope.

ACKNOWLEDGMENT

This publication was made possible by NPRP grant #NPRP8-2459-1-482 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

REFERENCES

[1] "ETSI EN 302 637-2 V1.3.2(2014-11) - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service," November 2014.

[2] "SAE J 2735 (2016-03) - Dedicated Short Range Communications (DSRC) Message Set Dictionary ," March 2016.

[3] "ETSI EN 302 637-3 V1.2.1(2014-09) - Intelligent Transport Systems (ITS);Vehicular Communications;Basic Set of Applications;Part 3: Specifications of DecentralizedEnvironmental Notification Basic Service," http://www.etsi.org/deliver/etsi_en/302600_302699/30263703/01.02.01_30/en_30263703v010201v.pdf, September 2014, accessed: 2016-11-28.

[4] "Dimensionality Reduction," http://infolab.stanford.edu/~ullman/mmds/ch11.pdf, accessed: 2016-11-27.

[5] C. Alippi, G. Anastasi, M. D. Francesco, and M. Roveri, "An adaptive sampling algorithm for effective energy management in wireless sensor networks with energy-hungry sensors," *IEEE Transactions on instrumentation and measurement*, vol. 59, no. 2, pp. 335 – 344, February 2010.

[6] X. Tao, D. Kong, Y. Wei, and Y. Wang, "A big network traffic data fusion approach based on fisher and deep auto-encoder," *Information*, vol. 7, no. 2, p. 20, 2016. [Online]. Available: http://www.mdpi.com/2078-2489/7/2/20

[7] H. Kargupta, R. Bhargava, K. Liu, M. Powers, P. Blair, S. Bushra, and J. Dull, "VEDAS: A Mobile and Distributed Data Stream Mining System for Real-Time Vehicle Monitoring," *Proceedings of the 2004 SIAM International Conference on Data Mining*, p. 12, April 2004.

[8] C. Liu, C. Chigan, and C. Gao, "Compressive Sensing based Data Collection in VANETs," *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1756 – 1761, April 2013.

[9] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, March 2008.

[10] T. Blu, P.-L. Dragotti, M. Vetterli, P. Marziliano, and L. Coulot, "Sparse sampling of signal innovations," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 31–40, March 2008.

[11] V. K. Goyal, A. K. Fletcher, and S. Rangan, "Compressive sampling and lossy compression," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 48–56, March 2008.

[12] J. Rombergn, "Imaging via compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 14–20, March 2008.

[13] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, December 2012.