Smart Driving Behavior Analysis Based on Online Outlier Detection: Insights from a Controlled Case Study

Igor Vasconcelos[1,2], Rafael Oliveira Vasconcelos[1,2], Bruno Olivieri[1], Markus Endler[1]
[1]*Department of Informatics*
Pontifical Catholic University of Rio de Janeiro (PUC-Rio)
Rio de Janeiro, Brazil
e-mail: {*ivasconcelos, rvasconcelos, bolivieri, endler*}*@inf.puc-rio.br*
[2]*Department of Informatics*
University Tiradentes (UNIT)
Aracaju, Brazil

Methanias Colaço Júnior
Department of Informatics
Federal University of Sergipe (UFS)
Aracaju, Brazil
e-mail*: mjrse@hotmail.com*

***Abstract*— This paper presents an approach for online outlier detection over multiple data streams based on Complex Event Processing (CEP) to enable driving behavior classification. Driving is a daily task that allows people to move around faster and more comfortably. However, more than half of fatal crashes are related to recklessness behaviors. Reckless maneuvers can be detected with good accuracy by analyzing data relating to the driver-vehicle interaction, abrupt turnings, acceleration and deceleration, for instance. In this paper, we investigate if off-the-shelf smartphones can be used to an online detection of driving behavior. To do so, we have adapted the Z-Score algorithm, a classical outlier detection algorithm, to perform online outlier detection as a data stream processing model, which receives the smartphone and in-vehicle sensors data as input. The evaluation of the approach was carried out in a case study to assess the algorithm. Our results indicate that the algorithm's performance is fairly good in a real world case study since the algorithm's accuracy was 84% and the average processing time was 100 milliseconds.**

Keywords-Online Outlier Detection; Complex Event Processing; In-Vehicle Sensing; Online Driving Behavior Detection; Smartphone.

I. INTRODUCTION

Driving is a daily task that allows people to travel more quickly and comfortably. However, a study on road safety conducted by the American Automobile Association [1] reported that 56% of fatal crashes between 2003 and 2007 involved one or more unsafe behaviors typically associated with reckless driving, such as speeding, improper lane changes, making improper turns and weaving in and out of traffic [2][3]. Reckless driving is a particular type of driving behavior defined by Tasca [3] as a behavior that "*deliberately increases the risk of collision and is motivated by impatience, annoyance, hostility or an attempt to save time*".

Nonetheless, current Intelligent Transportation Systems (ITS) still rely on an infrastructure composed of static sensors and cameras installed on roads, making it difficult to collect, aggregate and analyze the data, especially in real time [4]. Moreover, due to the high cost of installation and maintenance, they are usually restricted to certain roads or neighborhoods. In contrast, Internet of Things (IoT) aims to pervasively connect billions of things or smart objects, such as vehicles, sensors, actuators and smartphones. IoT poses an even more complicated challenge in a multi-stream environment where multiple data streams are competing for the available memory and processing resources, especially in resource-constrained systems, such as sensors and mobile devices [5].

Current approaches, such as [2][6][7], use the smartphone to understand and evaluate the driver's driving behavior. Mobile Sensor Platform for Intelligent Recognition Of Aggressive Driving (MIROAD) [2] is a driving style recognition platform that uses only the smartphone as a data source and processing unit. MIROAD uses the Dynamic Time Warping algorithm, originally developed for speech recognition, to classify the maneuvers (driver events) performed by the driver. Join Driving [6] proposes a scoring mechanism to quantitatively evaluate maneuvers and passenger comfort level based on ISO 2631-1-1997 [8]. Quintero, Lopez and Pinilla. [7] use Fuzzy Logic and Neural Networks - the output Fuzzy variables are inserted in a neural network properly trained to classify the behavior of the drivers. However, as the neural network is on a remote server, all Fuzzy variables outputs need to be sent to the remote server that performs the offline analysis and classification of driver behavior. These approaches use models/techniques (e.g., in Neural Network, Fuzzy Theory and Hidden Markov) with good accuracy [9]. In addition, they were not designed for data stream processing [10], and according to Lin et al. [9], they have low processing performance, need a long training phase, require artificial assumptions or require prior knowledge to formulate rules. Moreover, since these approaches are static, they have difficulty to recognize the parameters quick and accurately [9], for instance, neural networks have subjective methods for adjusting their topology (numbers of layers and neurons) and requires a fixed number of input parameters.

The rest of this paper is organized as follows. Section II presents an overview of the key concepts. Section III highlights the definitions and planning of the study case. Section IV addresses the study case operation. Finally, Section V reviews and discusses the central ideas presented in this paper, and proposes lines of future work on the subject.

II. FUNDAMENTALS

This section presents the main concepts about complex event processing, as well as outlier detection algorithms.

A. *Complex Event Processing*

Complex Event Processing (CEP) is a set of techniques and tools that provides an in-memory processing model upon the asynchronous data stream, in real time (i.e., minimum delay) for online detection of situations of interest [11]. CEP offers [11]: (i) situation awareness through the usage of continuous queries that correlate data from different sensor data streams, (ii) context awareness by subdividing data streams into different views, such as temporal windows or key partitions, and (iii) flexibility since it can specify events at any time, that is, the specification of events may be changed dynamically while the system is running (i.e., on-the-fly).

The CEP central concept is a declarative Event Processing Language (EPL) to express the event processing rules (continuous queries and patterns). These rules are based on event-condition-action triad, and use operators (e.g., logic, counting and temporal) on the input events, searching for correlations, exceptional conditions and pattern occurrence. The CEP central task is to provide mechanisms for Event Pattern Matching, i.e., from hundreds or even thousands of events, to identify significant patterns in the application domain [12]. The event processing and pattern detection are made by so-called event processing agents (EPAs) that process the events' stream. Basically, an EPA filters, separates, aggregates, transforms and synthesizes new complex events from simple events. To be able to detect the pattern of the maneuver, it is necessary to use an important CEP concept called Time Window (or just window). A window is a temporal context that defines which portions of the input data stream are considered during the EPL rule execution [13], e.g., events in the last 30 seconds.

B. *Outlier Detection*

Commonly, outlier detection technics typically assume that outliers in data are rare when compared to normal instances and when outliers do occur, these are observations that deviate significantly from the rest of the sample [14]. However, "meaningful" constitutes a subjective judgment to consider an instance as outlier. For instance, the main grouping pattern are extremely similar in Figure 1(a) and (b). However, there are significant differences outside these major groups. In Figure 1(a), the point *A* clearly deviates significantly from the rest of the points and therefore it is an outlier. However, Figure 1 (b) is much more subjective, since point *A* lies in a sparse region of data. Thus, it becomes more difficult to state with confidence that this data differs significantly from the other points. It is quite likely that this data point represents randomly distributed noise. This is because point *A* seems to fit a pattern represented by other randomly distributed points.

Therefore, although noise detection/removal is important for several application domains, it is not always possible to classify an instance as normal, noise or outlier precisely and choices depend on the specific criteria of each application. In this way, noise can be modeled as the semantic limit between normal and anomalous instances [15]. Thus, some authors use the term weak outlier (noise) and strong outlier to distinguish them [16]. In this paper, the term outlier refers to an instance that can be considered an abnormality or noise.

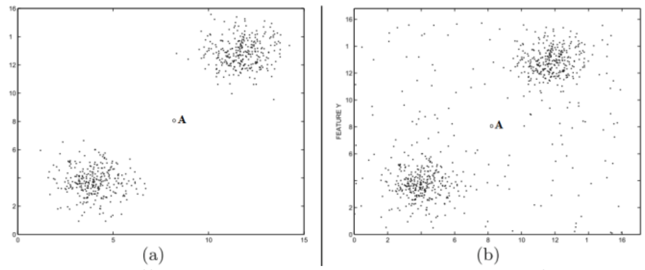

Figure 1. Difference between noise and anomaly. Adapted from [15].

Most anomaly detection algorithms uses scores for measuring the "outlierness", such as density, nearest neighbor clustering or statistical inference [15][17]. Thus, outliers usually have a higher score than the noises [15]. However, Aggarwal and Yu [16] emphasize that approaches based on clustering and density have an expensive computation and are not appropriate for data stream environments.

Statistical approaches were the earliest algorithms used for outlier detection and they assume that normal instances occur in high probability regions, while outliers occur in low probability regions of a stochastic model. The Standard Score (commonly referred as Z-Score) is a simple statistical technique that enables one-pass computation over a data stream to identify outliers. Z-Score describes raw score's location in terms of how far above or below the mean is when measured in standard deviation. A Z-Score of zero means that raw data instance is equal to the mean. Z-Score computation creates a unitless score that is no longer related to the original units (e.g., km/h and m/s^2) as it measures number of standard deviation units and therefore can more readily be used for comparisons. After computing the Z-Score for each data instance, the algorithm calculates the Z-distribution, i.e., the relative frequency of the raw Z-Scores of a population or sample.

III. DEFINING AND PLANNING OF THE CASE STUDY

In this section, the case study is presented with the focus on the definition and planning of the objective.

A. *Drivers' and route selection*

Due to the difficulty of recruiting drivers and the costs associated with assessing driving behavior, the process of driver selection was a matter of convenience and sampling was completed by quota. However, it was tried to establish a sample that represented the universe of drivers, preserving the same behavioral characteristics. Thus, 25 drivers were chosen for the study. Sixteen were male and nine female, theirs age ranged from 20 to 60 years. Another important

 ISBN: 978-1-61208-555-5

issue is the drivers' experience. In our sample, drivers' experience ranged from 2 to 42 years. Finally, all drivers were familiar with local traffic condition and regulations. This is important so that during the assessment of the drivers, the behaviors reflect the daily behaviors.

Regarding route selection, we defined a paved route comprising streets and avenues ranging from one to three lanes with approximately 14.5 km in Aracaju-SE, Brazil. In addition, the route contains roundabouts, traffic lights, pedestrian crossings and turns. The speed limit on the route was 60 km/h. A pilot study was conducted on the chosen route, and this provided insights about driver's behaviors.

B. Instrumentation

The instrumentation process started with the implementation of the Z-Score algorithm, through CEP rules. The algorithm was implemented in EPL, an structured query like language (SQL-like) where streams replace tables as the source of data with events replacing rows as the basic unit of data for running in ASPER, a CEP processing engine based on ESPER and adapted for Android. ESPER is an open source complex event processing engine.

A Brazilian version of manual Citröen C3 was equipped with a Samsung Galaxy SIII 1.4 GHz Quad Core with 1GB of RAM and Bluetooth OBD-II device. Our prototype was installed on the smartphone running the online Z-Score algorithm.

C. Measurement Metrics

We calculate five performance metrics, shown in Table I, which will be used to evaluate the algorithm. In addition, as quality metric, we use the **average execution time**, that is, the arithmetic mean of execution times for a given algorithm and algorithms' **average resources consumption**, i.e., Central Processing Unit (CPU) usage and memory consumption.

TABLE I. PERFORMANCE METRICS

Metric
Accuracy is the percentage of instances (evidences) correctly classified.
Recall is the percentage of instances that were correctly classified as positive.
Precision is the percentage of instances classified as positive (evidence) that are actually positive.
F Measure is the harmonic mean of precision and recall, that is, it combines the precision and recall.
Error Rate is the proportion of instances that are incorrectly classified.

IV. OPERATION OF THE CASE STUDY

This section describes the preparation and execution of the real world case study.

A. Preparation

For the evaluation, we used the open dataset provided by Bergasa [18]. The dataset provides three axis accelerometer data labeled (as cautious and reckless) based on thresholds given by Paefgen, Kehr, Zhai and Michahelles [19] for acceleration, braking and turning. This dataset contains driving data of six different drivers and vehicles in two different routes, one is 25km in a road with normally 3 lanes on each direction and 120km/h of maximum allowed speed, and the other is around 16km in a secondary road of normally one lane on each direction and around 90km/h of maximum allowed speed. For each driver data, a 3-fold cross-validation was performed, where each driver's data are randomly divided into two pieces of 35% for training and one piece of 30% for testing and checking the subset of data that generated the best results the algorithm. For the Online Z-Score algorithm, the best results were achieved considering a data instance as outlier based on the threshold (Z-score greater than the modulus of three) proposed by Chandola, Banerjee and Kumar [17].

Aiming to be operational in a mobile device, applications need to vary data rates based on available computation resources. Therefore, the Online Z-Score needs to adapt its behavior to perform the outlier detection with good accuracy. Thus, based on this scenario, we define two setups. First, we set the sensor data sample rate to be h = 100Hz and time window Δ = 10s (setup 1). Second, we set the sensor data sample rate to be h = 50Hz and time windows Δ = 20s (setup 2).

B. Intrinsic Evaluation of The Knowledge Model

In this subsection, we present the results for the training using the open dataset aforementioned. We repeated each evaluation 5 times and the confidence level for all results is 95%. Table II shows the average Z-Score Online performance. Despite the good overall performance, the Online Z-Score stood out with an average accuracy greater than 98%. In respect to the precision, the Online Z-Score had an expressive result with an average precision of 99.31%. This means that Online Z-Score classifies correctly cautious data instances that are really cautious in an average of 99.31%. Online Z-Score achieved an excellent recall performance in both setups. This means that, on average, Online Z-Score reached true positive rates greater than 99% in setup 1 and greater than 98% in setup 2. Regarding the F Measure, Online Z-Score stood out with the average F Measure greater than 99% in both setups.

TABLE II. OLINE Z-SCORE PERFORMANCE METRICS

Metric	Setup 1	Setup 2
	h = 100Hz and Δ = 10s	*h = 50Hz and Δ = 20s*
Accuracy	98.07%	98.70%
Precision	98.72%	99.90%
Recall	99.33%	98.78%
F-Measure	99.02%	99.33%

Table III shows the Online Z-Score quality metrics. In order to check the algorithms' resource consumption in both setups, we firstly verify the smartphone's memory (in megabytes) and CPU (in percentage) usage in two situations: in standby and collecting data from smartphone's own sensors and OBD-II device, however, without processing them. Through the Table III, it is possible to note that only

 ISBN: 978-1-61208-555-5

collecting data increases memory consumption by 12.54% and CPU usage by 60.83%. However, the Online Z-Score consumed only 6.15 and 6.40 MB of the 830 MB available on the smartphone. Disregarding CPU usage for *collecting*, Online Z-Score used only 3.01% and 3.86% of the CPU in setups 1 and 2, respectively. Moreover, a larger time window resulted in a higher memory consumption and processing.

TABLE III. Z-SCORE OLINE QUALITY METRICS

Metric	Setup 1			Setup 2		
	RAM (MB)	*CPU (%)*	*Time (s)*	*RAM (MB)*	*CPU (%)*	*Time (s)*
Standby	4.53	1.41	-	4.53	1.41	-
Collecting	5.18	3.60	-	5.18	3.60	-
Z-Score Online	6.15	6.61	101.6	6.40	7.46	99.58

C. *Execution*

The smartphone was installed in the center of the vehicle windshield. The OBD-II reader device was connected to the OBD-II port of the vehicle and reads a variety of data from the vehicle bus. The OBD-II device sends the data streams via Bluetooth to the smartphone. Thus, the execution of the case study consisted of performing the process of outliers' detection from driving data streams of each driver volunteer.

The driver behavior data were collected in seven sunny days and the drivers drove between 9am and 8pm. Each driver made one trip on the chosen route. Thus, a total of 362.5 km were covered comprising 12.5 hours of driving.

Then, the chosen route was explained in detail and it was asked to the driver drives as usually. The driver volunteer also was informed that a driver expert with 15 years of experience would follow him\her during the case study – similar to an expert-based test administered in initial tests to judge driver performance – but we emphasize that our goal was to analyze and classifies driver's behavior in cautious or reckless and not approve or disapprove him/her. This classification served as a ground truth.

The prototype collects data from smartphone sensors (i.e., accelerometer, gyroscope, magnetic compass and GPS) and from vehicle sensors (i.e., speed, revolutions per minute and throttle position in percentage) through OBD-II device. These sensor data streams are sent to the smartphone via Bluetooth connection. The connection between the smartphone and OBD-II device was performed using a generic mobile middleware [20] for short range communication.

D. *Extrinsic Evaluation of the Knowledge Model*

Unlike the results obtained by Hong, Margines and Dey [21], both cautious and reckless drivers have substantial differences regarding speed. Through an online analysis, it is possible to identify reckless maneuvers that result in significant changes in Z-distribution, as shown in Figure 2. Cautious maneuvers follow the normal distribution.

Figure 3 shows the average revolutions per minute (rpm) Z-distribution. There is a notable Z-distribution difference in reckless maneuvers while performing an online analyzes. For instance, in the maneuver one and two, the total of outlier evidences are 17% and 23.5% respectively. The Online Z-Score algorithm identified quite a different distribution for reckless maneuvers, as shown in Figure 4. For instance, the maneuvers 1 and 2 had respectively 32.67% and 31.14% of evidences classified as outliers.

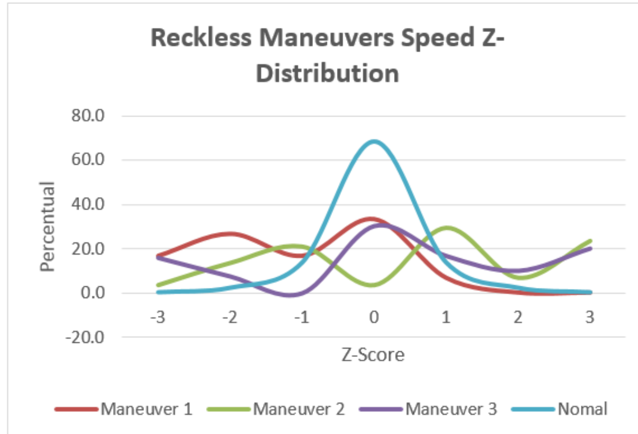

Figure 2. Speed maneuvers Z-distribution comparison

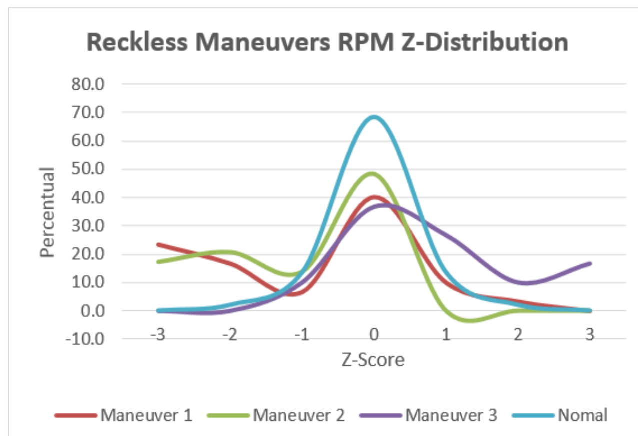

Figure 3. RPM maneuvers Z-distribution comparison

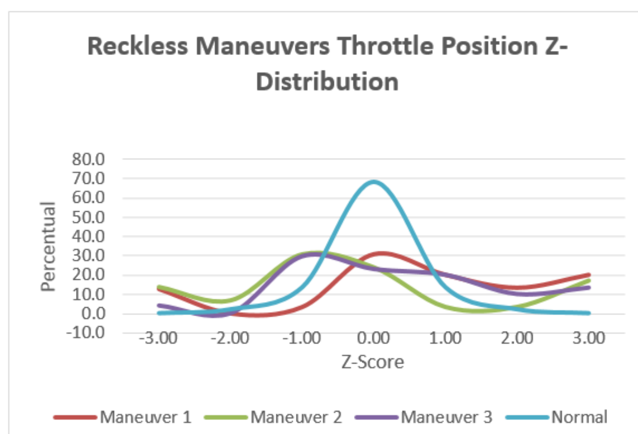

Figure 4. Throttle position maneuvers Z-distribution comparison

Analyzing 3-axis accelerometer data it should be noted that aggressive events Z-distribution is practically equal to the normal curve, as shown in Figure 5. Unlike other studies that consider only the lateral and longitudinal acceleration [21], we decided to consider the 3-axis since in Brazil many of the roads have poor quality. Thus, we believe that an analysis considering the 3-axis depicts more faithfully the Brazilian scene. However, to our surprise and going against the results of several studies, such as [21], evaluating the driver behavior, it was not noticed considerable changes in acceleration Z-distribution in reckless maneuvers as shown in Figure 5. Unlike the data aforementioned, maneuvers 1 and 2 had, respectively, only 9.37% and 7.86% of evidences classified as outliers. However, based on parameters established in ISO 2631-1-1997, which evaluates the effects of human exposure under acceleration, to measure the level of passengers comfort/discomfort. Considering that in our case study a stopped vehicle had acceleration equal to 9.8 m/s², so passengers felt comfortable while acceleration was within a range from 8.9 to 11.2 m/s². Nevertheless, reckless drivers' events, such as sudden lane changes, abrupt accelerations/deceleration and jerks, generated much more uncomfortable feelings for passengers once 75% of the outliers were out of comfortable this range.

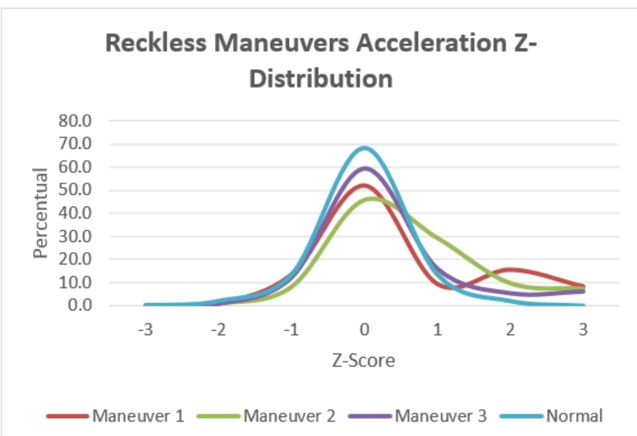

Figure 5. Acceleration Z-distribution comparison

E. Scoring Driving Behaviors

In order to score the drivers' behavior, it is necessary to consider that (i) sensors have different acquisition rates. For instance, in this case study, the OBD-II device and the smartphone's accelerometer average acquisition rate was 8 Hz and 140 Hz, respectively. Thus, during the data stream processing, we will have 17.5 times more evidences of acceleration than speed and (ii) certain evidences may have little power for discriminating driver behavior. To this end, we adapted a statistical mechanism used in document mining to evaluate how important a word is to a document in a collection, called inverse document frequency [22], to identify how important a outlier in a data stream.

We defined an outlier frequency (of_d) as the number of outliers that occurs in a dimension d. Furthermore, we defined the inverse outlier frequency (iof_d) of a data instance in dimension d as shown in (1).

$$iof_d = \log\left(\frac{N}{of_d}\right) \tag{1}$$

Thus, the iof_d of a rare outlier evidence is high, whereas the iof_d of a frequent outlier evidence is likely to be low. In order to weighting each outlier evidence in a time window, we combine the definition of outlier frequency and inverse outlier frequency (*ofiof*) as given by □2□, where e is the outlier evidence value and d is the dimension. Therefore, a driver' trip score is given through the weighted average of sum of all *ofiof*, as shown in □3□, where t is the number of time windows during the trip.

$$ofiof_{e,d} = of_{e,d} * iof_d \tag{2}$$

$$Score = average\left(\sum_{i=1}^{t}\left(ofiof_{e,d}\right)\right) \tag{3}$$

Figure 6 shows the drivers' score. For this case study, drivers with scores greater than 50 were classified as reckless. This threshold was chosen by analyzing data from six other drivers. These drivers drove on the same chosen route, but for three of them were asked to drive cautiously and the others recklessly. The maximum score for cautious drivers was 35 and the minimum one for the reckless was 65. Therefore, we consider the threshold of 50 as the upper bound in the classification of the cautious drivers and as the lower bound in the classification of reckless one. Comparing the algorithm classification with the ground truth, it should be noted an excellent performance, as shown Table IV.

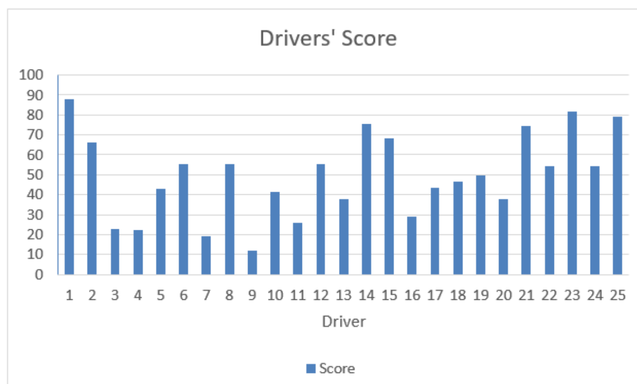

Figure 6. Drivers' score during case study

TABLE IV. OLINE Z-SCORE PERFORMANCE

Metric	Value
Accuracy	84.00%
Recall	76.47%
Precision	100.00%
F-Measure	86.67%
Error Rate	16.00%
RAM	6.35 MB
CPU	7.24%

V. CONCLUSION AND FUTURE WORK

In this paper, we introduced an online outlier detection for driver behavior detection approach. Unlike many works that aims to provide a faster outlier detection and to adapt algorithms to perform a distributed processing, our proposal performs an online outlier detection in mobile devices, such as smartphone, with limited computational resources. The main contributions of this paper are (i) a classical offline outlier detection algorithm adapted to perform online outlier detection. In addition, this algorithm is operational on mobile devices and able to adapt their behavior based on available computational resources, that is, change sensors' refresh rate and time window without varying the algorithm accuracy, (ii) a prototype to identify driver behavior based on online outlier detection and (iii) assessment that validates and demonstrates the performance of our proposal.

More research is still needed with this approach. However, considering the encouraging performance results, we are confident that our approach can be used in several others IoT scenarios. For the future, we expect to advance our work along the following lines: (i) perform a comparison with other related works, (ii) adapt the algorithm to scenarios where energy consumption is critical, (iii) perform a distributed online outlier detection, (iv) analyze the effect that different types of windows may generate in the correctness of algorithms and devices' resource consumption and (v) regarding to driver behavior, identify behaviors that precede accidents and to identify the relationship between driving behavior, fuel consumption and air pollution.

REFERENCES

[1] Safety Foundation for Traffic, "Aggressive driving: Research update," 2009. [Online]. Available: https://www.aaafoundation.org/sites/default/files/AggressiveDrivingResearchUpdate2009.pdf. [Accessed: 28-Apr-2017].

[2] D. A. Johnson and M. M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2011, pp. 1609–1615.

[3] L. Tasca, *A review of the literature on aggressive driving research*. Ontario Advisory Group on Safe Driving Secretariat, Road User Safety Branch, Ontario Ministry of Transportation, 2000.

[4] V. Moosavi and L. Hovestadt, "Modeling urban traffic dynamics in coexistence with urban data streams," in *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing - UrbComp '13*, 2013, p. 1.

[5] Q. Xie, J. Zhu, M. A. Sharaf, X. Zhou, and C. Pang, "Efficient buffer management for piecewise linear representation of multiple data streams," in *Proceedings of the 21st ACM international conference on Information and knowledge management - CIKM '12*, 2012, p. 2114.

[6] H. Zhao, H. Zhou, C. Chen, and J. Chen, "Join driving: A smart phone-based driving behavior evaluation system," in *2013 IEEE Global Communications Conference (GLOBECOM)*, 2013, pp. 48–53.

[7] C. G. Quintero M., J. O. Lopez, and A. C. Cuervo Pinilla, "Driver behavior classification model based on an intelligent driving diagnosis system," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, 2012, pp. 894–899.

[8] I. O. for Standardization, "ISO 2631-1-1997: Mechanical vibration and shock-Evaluation of human exposure to whole-body vibration-Part 1: General requirements," 1997.

[9] N. Lin, C. Zong, M. Tomizuka, P. Song, Z. Zhang, and G. Li, "An Overview on Study of Identification of Driver Behavior Characteristics for Automotive Control," *Mathematical Problems in Engineering*, vol. 2014, pp. 1–15, 2014.

[10] M. Čermák, D. Tovarňák, M. Laštovička, and P. Čeleda, "A performance benchmark for NetFlow data analysis on distributed stream processing systems," in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, 2016, pp. 919–924.

[11] D. C. Luckham, *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001.

[12] J. Dunkel, "On complex event processing for sensor networks," in *2009 International Symposium on Autonomous Decentralized Systems*, 2009, pp. 1–6.

[13] G. Cugola and A. Margara, "Processing Flows of Information: From Data Stream to Complex Event Processing," *ACM Comput. Surv.*, vol. 44, no. 3, p. 15:1--15:62, 2012.

[14] V. J. Hodge and J. Austin, "A Survey of Outlier Detection Methodologies," *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, Oct. 2004.

[15] C. C. Aggarwal, *Outlier Analysis*. New York, NY: Springer New York, 2013.

[16] C. C. Aggarwal and P. S. Yu, "Outlier detection for high dimensional data," *ACM SIGMOD Record*, vol. 30, no. 2, pp. 37–46, Jun. 2001.

[17] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, Jul. 2009.

[18] L. M. Bergasa, D. Almeria, J. Almazan, J. J. Yebes, and R. Arroyo, "DriveSafe: An app for alerting inattentive drivers and scoring driving behaviors," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, 2014, pp. 240–245.

[19] J. Paefgen, F. Kehr, Y. Zhai, and F. Michahelles, "Driving Behavior Analysis with Smartphones: Insights from a Controlled Field Study," in *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*, 2012, p. 36:1--36:8.

[20] L. E. Talavera, M. Endler, I. Vasconcelos, R. Vasconcelos, M. Cunha, and F. J. Da Silva E. Silva, "The Mobile Hub concept: Enabling applications for the Internet of Mobile Things," in *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, 2015, pp. 123–128.

[21] J.-H. Hong, B. Margines, and A. K. Dey, "A smartphone-based sensing platform to model aggressive driving behaviors," in *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*, 2014, pp. 4047–4056.

[22] C. D. Manning, P. Raghavan, and H. Schütze, *An Introduction to Information Retrieval*, Online edi. Cambridge, England: Cambridge University Press, 2008.