Real-time data-intensive telematics functionalities at the extreme edge of the network: Experience with the PrEstoCloud Project

Salman Taherizadeh[1,2], Blaz Novak[1], Marija Komatar[2], Marko Grobelnik[1]
[1] Artificial Intelligence Laboratory, Jozef Stefan Institute, Ljubljana, Slovenia
[2] CVS Mobile d.d., Ljubljana, Slovenia

***Abstract*—In recent years, use of different sensors connected to vehicles is dramatically increasing in order to enhance transportation efficiency. The current Big Data technologies are predominantly used to store large amount of telematics data especially in the cloud, and they are only able to perform simple querying for the purpose of reporting. While all the data is stored in the cloud-centric datacenters, these telematics systems are not capable of exploiting other functionalities offered by advanced real-time analytics such as run-time anomaly detection. In this paper, we propose an advanced telematics system orchestrated upon an edge computing framework in the context of the PrEstoCloud (Proactive Cloud Resources Management at the Edge for Efficient Real-Time Big Data Processing) project. This telematics system is a real-time data-intensive application running at the extreme edge of the network for drivers' behavior profiling and triggering run-time alerts. Such functionalities may be useful in order to notify stakeholders for example drivers and logistic centers on situations where a possible accident may occur or attention is required.**

Keywords—edge computing; real-time analytics, logistic system

I. INTRODUCTION

Nowadays, vehicle telematics [1] such as fleet management systems, vehicle tracking solutions, location-based vehicle navigation technologies and traffic information services has gained increasing attention in the context of Big Data management paradigm. The current vehicle telematics systems produce big volumes of multimodal data [2], which is predominantly stored in cloud-based datacenters and processed by traditional analytics tools. Consequently, vehicle management is becoming a highly data-centric business.

The major challenge in this research area is not in collecting and storing the monitoring data, but rather in fast extraction of useful information from the data streams and triggering automated alerts at run-time to help stakeholders (e.g. driver, logistic center, insurance company or vehicle owner). In other words, a missing technology in fleet management is advanced real-time analytics on telematics data streams coming from vehicles and their contexts. Such modern solutions may contribute to the road safety, vehicle security, fleet business efficiency, logistics optimization, cleaner environment, efficient insurance strategies, and so on.

In the current setups used by traditional analytics tools, all actions to manage a vehicle and reduce costs are possible only after analyzing sensory data on the server side, typically not in real-time. By introducing real-time analytics processed on the nodes at the extreme edge of the network [3], and hence fast, safe communication with stakeholders such as driver, benefits (e.g. accident prevention or cost optimization) would be put to the advanced level with additional services enabled, not being possible with today's infrastructure.

To overcome above problems, this paper presents a new telematics system which provides a real-time data-intensive computing framework at the extreme edge of the network. The proposed edge computing framework is able to observe driving dynamics (e.g. acceleration, braking, turning, etc.), perform real-time analytics and trigger alerts to drivers and fleet managers on situations where new decisions should be made. Since not all of the data could be processed on the edge component itself, further demanding processing tasks (such as long-term storage of sensory data on the cloud and complete offline data analytics) would be performed on the cloud side, shown in Fig. 1.

The rest of the paper is organized as follows. Section II presents summary of related work supporting data-intensive telematics systems. Section III describes the use case. Section IV presents the architecture of our proposed edge computing approach, followed by implementation in Section V. Finally, conclusion appears in Section VI.

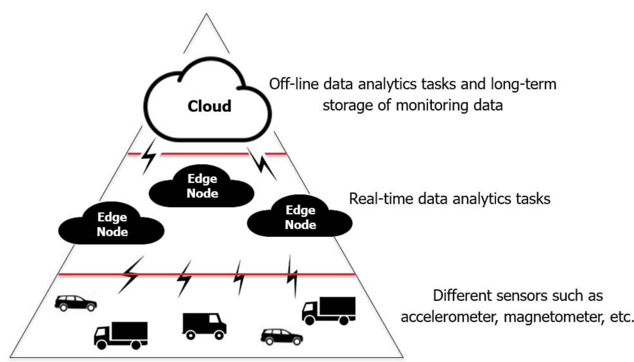

Fig. 1. Cooperation of both edge and cloud technologies

0730-3157/18/$31.00 ©2018 IEEE
DOI 10.1109/COMPSAC.2018.10288

II. RELATED WORK

Currently, road safety is considered as one of major societal issues. There is an estimation that each death on Europe's roads results in four persons with permanent injuries (e.g. the brain or spinal cord), and eight persons with serious injuries [4]. Therefore, telematics systems such as driver behavior monitoring and fleet performance management have gained increasing attention nowadays. However, in telematics applications, data stream processing for real-time analytics is a challenging issue since stream Big Data has high velocity, large volume and complex data types [5].

Current systems able to perform real-time analytics on telematics data may employ a set of constraints predefined according to previous empirical observations. For example, Bergasa *et al.* [6] proposed a system called DriveSafe capable of alerting inattentive drivers. They established three different levels (low, medium and high) for acceleration, braking and turning operations according to distraction event thresholds. Dai *et al.* [7] also presented an approach for abnormal movement detection that uses accelerometer monitoring data sensed by a mobile phone to recognize drunk driving patterns through violating a set of thresholds. These approaches are easy to implement, and they also have a low computational complexity. However, these approaches have their own intrinsic limitation since defining right, accurate values for necessary thresholds used in such systems is critical in order to reach a high rate of precision.

Moreover, various research works [8, 9, 10] have implemented machine learning algorithms to detect critical driving events such as hard braking and sudden acceleration. These approaches are able to provide continuous quality improvement in the results. However, they have to be trained prior to their use in real-world environments, and hence they require historical monitoring data and some time in order to converge towards an appropriate driven model. In this work, our proposed real-time data-intensive telematics system uses a combination of both threshold-based and machine learning approaches in order to take advantages of both mechanisms.

III. USE CASE

The data generated by vehicles' sensors (e.g. accelerometer and magnetometer, etc.) need to be processed at run-time to recognize important driving actions in an instant. This is because one of the major causes for vehicle accidents is inattention or distraction. In this work, edge nodes deployed in vehicles are employed to analyze data at run-time next to the location where the sensory data are generated.

Simultaneously, some of tasks such as highly computing-intensive analytics on telematics data can be directly forwarded to the cloud. In this way, the edge node also operates as intermediary to transmit the data to the cloud for further analytics, shown in Fig. 2. Moreover in another situation, if the edge node cannot appropriately execute computing operations (e.g. because it is running out of storage space), computing tasks provided on the edge node should be terminated and started on the cloud infrastructure.

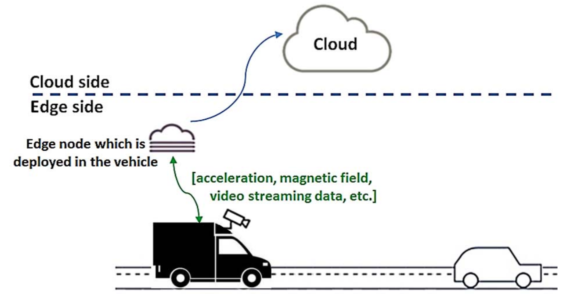

Fig. 2. Real-time data-intensive telematics system

A. Edge side

In this work, an edge node is a platform called MACH (Motorhome Ai Communication Hardware), shown in Fig. 3, which is developed in the OPTIMUM project [11].

Currently, MACH is successfully installed in prototype motorhome vehicles of the Adria Mobil Company [12]. Therefore, all sensors such as accelerometer and magnetometer are connected to MACH at the edge of the network. MACH includes one compute module (Raspberry Pi 3 model B [13]) and a custom extension (VESNA) [14] which is able to communicate with different hardware devices through various protocols such as RS232, RS485, LTE, Bluetooth, GPIO, SPI, I2C, CAN, LIN, CI-BUS. Raspberry Pi 3 model B is the third-generation Raspberry Pi which is a powerful credit-card sized single board computer with these features: Quad Core 1.2 GHz Broadcom BCM2837 64bit CPU, 1GB RAM, 4 USB 2 ports, 4 Pole stereo output and composite video port, etc. VESNA is also a fully flexible, modular, high-performance platform for the implementation of Wireless Sensor Networks (WSNs) developed by the SensorLab at the Jozef Stefan Institute.

B. Cloud side

All MACHs are connected to whether a private or public cloud with higher storage and processing capacity. This private or public cloud infrastructure provides a long-term storage of possibly anonymized monitoring information and performs offline data analytics based on actual data and related metadata. All machines used in this work on the cloud side belong to a non-profit cloud-based infrastructure provider called ARNES (the Academic and Research Network of Slovenia).

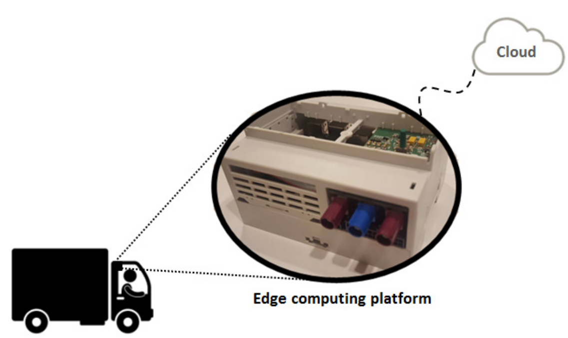

Fig. 3. Edge computing platform (MACH) which is deployed in the vehicle

IV. ARCHITECTURE

The proposed architecture to run a real-time data-intensive telematics system at the extreme edge of the network, shown in Fig. 4, includes various components at different layers namely Edge Infrastructure Layer, Cloud Infrastructure Layer, Cloud-Edge Communication Layer, and Control Layer. These components shown in Fig. 4 operate as follows.

A. Edge computing application

This component is the application which is deployed on the edge node. The edge computing framework itself is application agnostic in that this framework is not dedicated to a single type of software system or purpose [15]. In order to offer a fast response time for the service and provide efficient resource utilization, the application deployed on the edge node closer to the end-users is preferred over a centralized approach using faraway cloud-based infrastructures. In our logistic system, edge application may provide different services such as real-time anomaly detection, driver behavior profiling, and so on.

B. Monitoring Probe running on the edge node

A set of metrics related to both edge-based infrastructure and application deployed on the edge node is measured continuously by the Monitoring Probe. Monitoring Probe periodically sends the monitoring data in the form of messages to the Message Broker from which it can be distributed to any other components such as the Autonomic Resource Manager. Infrastructure-related metrics are CPU, memory, disk, etc. Moreover, application-related parameters present information about the situation of the application deployed on the edge resource. Monitoring of edge nodes used in the proposed telematics solution is critical to achieve efficient resource utilization and avoid any issues in delivering provided services.

C. On/Offloading Agent

An On/Offloading Agent is installed on every edge resource. On/Offloading Agent is responsible for registering the edge node in the Cloud and Edge Resources Repository through On/Offload Processing component. Moreover, it responds to the Mobile On/Offload Processing component's requests for the on/offloading tasks which can be start, stop or migration of application between edge nodes and datacenters. Such communication between the Mobile On/Offload Processing component and On/Offloading Agent is made via the Message Broker.

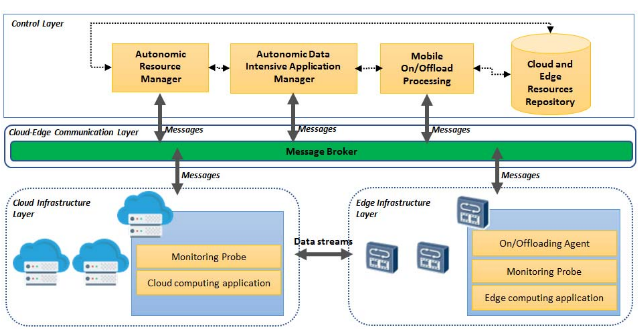

Fig. 4. Proposed architecture for real-time data-intensive logistic system

D. Cloud computing application

This component is the application which is deployed on the cloud. The capability of the application running on the cloud would be using cloud-based infrastructure to run further demanding processing tasks such as long-term storage and more data processing operations that the edge node may not be able to perform.

E. Monitoring Probes running on the cloud

A set of metrics related to both cloud-based infrastructure and application deployed on the cloud is measured continuously by the Monitoring Probe [16]. Monitoring Probe periodically sends the monitoring data in the form of messages to the Message Broker.

F. Mobile On/Offload Processing component

The Mobile On/Offload Processing component provides the interface for registration and monitoring of edge nodes by communicating with On/Offloading Agent installed on every edge node. It also maintains a database called Edge Resources Repository to store the current state of all registered edge nodes and their features.

G. Message Broker

Message Broker is a scalable, central part of the distributed, logistic system which has a sender and one or more receivers for a single message. It decouples data providers and data consumers. Message Broker can be implemented using open source solutions such as RabbitMQ [17].

H. Autonomic Resource Manager & Cloud and Edge Resources Repository

The Autonomic Resource Manager collects monitoring data (sent by Monitoring Probes) about all available cloud resources and edge nodes, and submits this information to the Cloud and Edge Resources Repository.

I. Autonomic Data Intensive Application Manager

This component knows the specifications and the desired placement of computing tasks (e.g. data analytics jobs) on the cloud. The Autonomic Data Intensive Application Manager is capable of recognizing situations where new resources must be acquired, or existing resources must be released. There are conditions where an edge node is not able to provide computing operations. For example if it is overloaded due to an increase in the number of sensors connected to the vehicle during execution and hence there are no spare cycles or storage capacity on the edge resource, or the edge node cannot improve the application QoS anymore. In such conditions, the Autonomic Data Intensive Application Manager sends reconfiguration instructions to the Mobile On/Offload Processing component which translates them into platform-dependent deployment instructions. Then, these platform-dependent deployment instructions (e.g. start, stop or migration of telematics application) will be issued to the On/Offloading Agent running on the edge node.

V. Implementation and Results

The proposed data-intensive telematics system extracts important information (such as sudden acceleration/braking or aggressive left/right turn) through real-time analytics. In order to perform real-time anomaly detection and extract driving events from input sensory data at run-time, the pattern matching approach [18] has been chosen as the baseline method which was innovatively extended in this work to diminish computing overhead on the edge node. Our proposed new approach, shown in Fig. 5, also considers a set of predefined constraints to detect abnormal situations. Using such constraints (e.g. thresholds for different metrics such as acceleration and magnetic field of vehicle, etc.) to detect an abnormal situation reduces the runtime computing overhead. This is because the majority of computation will be performed only when any of predefined thresholds is violated. Offering a low overhead system is a significant requirement in an edge computing framework since edge nodes in practice have their own resource restrictions such as limited computing power.

A. Input sensory data

The input data used by the extended pattern matching approach is taken by different sensors such as accelerometer and magnetometer, etc. In this work, the LSM303DLHC device [19] integrated with MACH is employed to provide 3-axis acceleration and magnetic field.

Accelerometer provides measurements of the vehicle's acceleration at run-time. In this regard, we focus on vehicle's movements along the longitudinal and lateral axes which respectively represent forward-and-backward and side-to-side movements. Longitudinal acceleration allows us to detect vehicle braking and accelerating events. Lateral acceleration is also used to recognize left/right turning and lane changing events. Magnetometer provides measurements of direction at which the vehicle is pointing towards. The data measured by magnetometer sensor is applied as an extra indicator of driving events in lateral aspect. The accelerometer and magnetometer measurements are sampled at the rate of 5 Hz where each sample is recorded every 200 ms in order to create a time series of data points (e.g. acceleration and magnetic field of vehicle).

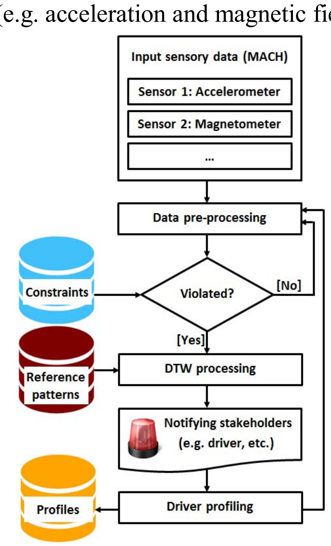

Fig. 5. Extended pattern matching approach

B. Data pre-processing

Raw input sensory data collected from different sensors such as accelerometer and magnetometer should be pre-processed since the negative effect of undesired noises in measurements needs to be smoothed out. To this end, the simple moving average method [20] which is a widely used approach to smooth out noisy volatility is employed. Moreover, a low-pass filter with a 1 Hz cutoff may be used since the raw signals could be very noisy due to vehicle interior vibrations. Alternatively, the row signals could be cleaned using a Kalman filter with a state vector formed by for example three-axis components of accelerometer or magnetic field.

C. Constraints checking

Constraints database includes all associated conditions predefined for driving events that need to be continuously checked. For example, an absolute value of acceleration at the rate of more than 0.3 G can be considered as sudden acceleration/braking event [21]. Or an observation of a change in instantaneous direction of heading that is more than 30 deg/s may refer to an aggressive left/right turn or aggressive lane change [22]. If a speed detector would be used to measure the speed of vehicle running on the road, constraints database should also consist of road speed limit. It should be noted that the minimum time periods during which these values have to be exceeded are also determined in this database.

On occurrence of an identified abnormal situation when attention is required, the Dynamic Time Warping (DTW) [23] algorithm as a machine learning processing task should be performed in the next step for the fast extraction of required information.

D. DTW procssing

Reference patterns stored in a separate database should be predetermined for all driving events. In order to construct reference patterns for each driving event (such as sudden acceleration/braking or aggressive left/right turn), sufficient data set should be collected in advance before the system starts working. In other words, a reference pattern needs to be made for each driving event for the pattern matching algorithm. Generated reference patterns [18] are then employed as templates to match incoming signals taken from sensors such as accelerometer, shown in Fig. 6, Fig. 7 and Fig. 8.

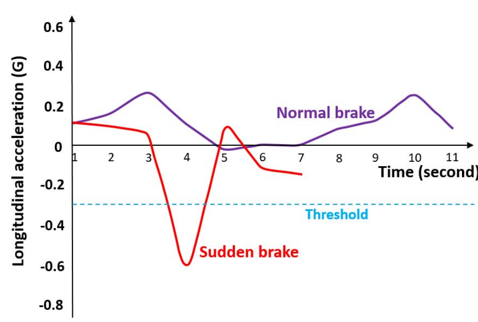

Fig. 6. Reference pattern for sudden brake

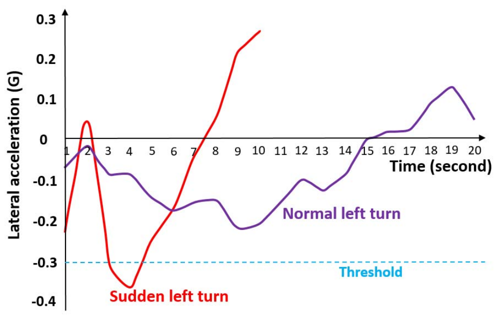

Fig. 7. Reference pattern for sudden left turn

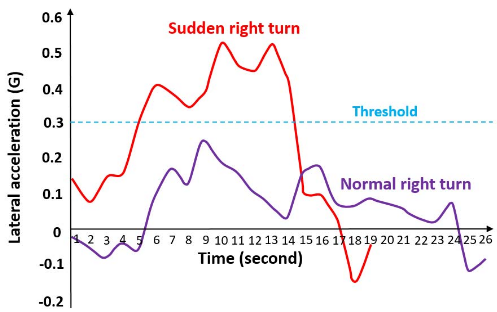

Fig. 8. Reference pattern for sudden right turn

The DTW algorithm is applied to find patterns in time series of data points. A DTW-based similarity checking mechanism measures the difference between two time series, namely the incoming sensory data and the reference patterns. DTW is an algorithm to provide a similarity measure between two signals that also allows for stretched and compressed sections of two given sequences. In other words, the main advantage of DTW is its ability to automatically cope with time deformations of two signals corresponding to movements performed at two different speeds. For example, similarities in movements could be detected, even if the vehicle is moving faster or slower than the reference pattern, or if there are accelerations and decelerations during the course of an observation.

E. *Triggering alerts*

This step is aimed at reacting on recognized events and reporting actionable alerts in real-time. Fig. 9 shows the real-time anomaly detection scenario by which in addition to the driver, these alerts can be sent to other entities such as logistic center, vehicle owner, insurance company, etc.

All stakeholders such as logistic center and vehicle owner are able to access to the graphical interface (shown in Fig. 10) which represents real-time information about current alerts and other data associated with the vehicle's status, acceleration history, driving direction, current location, etc.

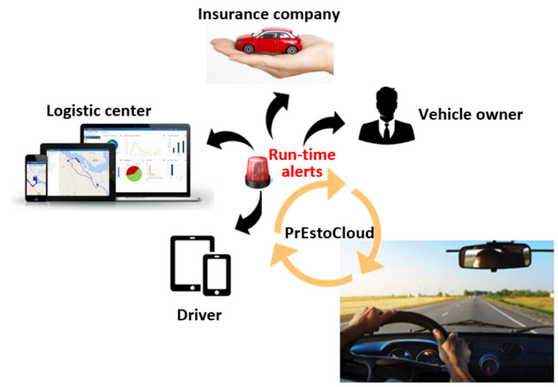

Fig. 9. Capablity of triggering run-time alerts to different stakeholders of telematics system


Fig. 10. Graphical interface showing real-time alerts and other information

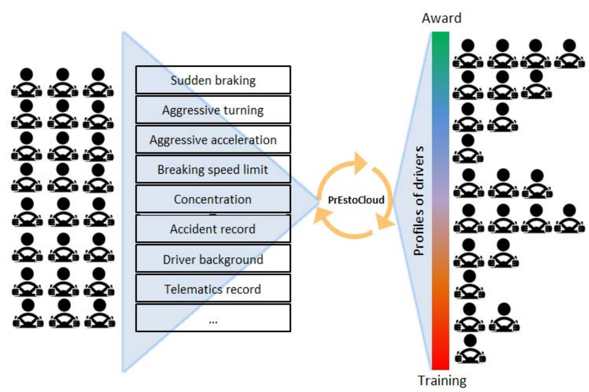

Fig. 11. Driver profiling

F. *Driver profiling*

Different characteristics of drivers can be considered to find out how they drive on the road, such as the rate of braking, vehicle acceleration, lane change patterns and so on. For example, it should be noted that an unsafe driver performs hard acceleration, sudden braking and steering maneuvers more frequently than a safe and also moderate driver. All these features performed by the driver can be analyzed and allow us to form a profile which represents the driver's behavior. Drivers' profiles stored in a specific database may be applied to show useful information on how safe their driving could be, how economic they drive in terms of fuel consumption, how much they care about vehicle maintenance, how efficient their driving is in terms of environmental impact, etc. Such a system can provide feedback to different stakeholders such as driver and logistic center. Moreover, driver profiling scenario can also be used to generate whether a training system or an award system which may motivate drivers to keep trying to attain high standards of driving excellence (see Fig. 11).

VI. CONCLUSION

This paper presents a new telematics system in the context of advanced real-time data-intensive analytics at the extreme edge of the network. This system extracts important information through real-time analytics on transport-logistic data streams (e.g. acceleration, magnetic field of vehicle, etc.) coming from different sensors. This new system is able to trigger alerts at run-time and notify stakeholders such as drivers and logistic centers on situations where new decisions should be made or attention is required (e.g. dangerous maneuvering, sudden lane change, etc.). Moreover, this telematics system orchestrated upon edge computing framework proposes a method for drivers' behavior profiling, where the major outcome is detailed understanding of their behavior with possible feed-back loop to improve drivers' activities. In this case, results as driver's profile will be provided back to the driver through in-vehicle screen (e.g. via a tablet or mobile phone) as well as to the logistic center (e.g. via a Web-based graphical interface showing real-time alerts and other information).

We have begun extending our proposed method towards using visual information processing. The primary scenario will be inclusion of visual information, where the major outcome to recognize and detect situations is not possible with the rest of the in-vehicle sensory information. In particular, the most interesting scenario will be the security of a vehicle which is often undetectable due to the limited observation power of traditional methods. However, introduction of one or more cameras with a real-time video feed to the edge will significantly add complexity to the use case.

Our work is included in the advanced software solution of an ongoing European Horizon 2020 project: PrEstoCloud [24]. The main objective of the PrEstoCloud project is to create substantial research contributions in the cloud and edge computing system environments and real-time Big Data technologies in order to provide a dynamic, distributed architecture for proactive resource management. To this end, the PrEstoCloud solution reaches the extreme edge of the network for efficient Big Data processing.

ACKNOWLEDGMENT

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No. 732339 (PrEstoCloud project: Proactive Cloud Resources Management at the Edge for Efficient Real-Time Big Data Processing).

REFERENCES

[1] J. Wahlström, I. Skog, and P. Händel, "Smartphone-based vehicle telematics: A ten-year anniversary," IEEE Trans. Intell. Transp. Syst., 18(10), 2017, pp. 2802-2825.

[2] D. Rao, M. De Deuge, N. Nourani–Vatani, S.B. Williams, and O. Pizarro, "Multimodal learning and inference from visual and remotely sensed data," International Journal of Robotics Research, 36(1), 2017.

[3] S. Taherizadeh, A.C. Jones, I. Taylor, Z. Zhao, and V. Stankovski, "Monitoring self-adaptive applications within edge computing frameworks: A state-of-the-art review," Journal of Systems and Software, 136, 2018, pp. 19-38.

[4] European Commission - Mobility and Transport, https://ec.europa.eu/transport/road_safety/specialist/statistics_en

[5] C.L. Philip-Chen and C.Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on Big Data," Inf. Sci., vol. 275, 2014, pp. 314-347.

[6] L.M. Bergasa, D. Almería, J. Almazán, J.J. Yebes, and R. Arroyo, "DriveSafe: An app for alerting inattentive drivers and scoring driving behaviors," in Proceedings of 2014 IEEE Intelligent Vehicles Symposium Proceedings. Dearborn, USA, 2014, pp. 240-245.

[7] J. Dai, J. Teng, X. Bai, Z. Shen, and D. Xuan, "Mobile phone based drunk driving detection," in Proceedings of 2010 4th International Conference on Pervasive Computing Technologies for Healthcare. Munich, Germany, 2010, pp. 1-8.

[8] F. Tango, L. Minin, F. Tesauri, and R. Montanari, "Field tests and machine learning approaches for refining algorithms and correlations of driver's model parameters.," Applied ergonomics, vol. 41, no. 2, 2010, pp. 211-24.

[9] J.F. Júnior, E. Carvalho, B.V. Ferreira, C. de-Souza, Y. Suhara, A. Pentland, and G. Pessin, "Driver behavior profiling: An investigation with different smartphone sensors and machine learning," PLoS one, 12(4), 2017.

[10] T. Liu, Y. Yang, G.B. Huang, Y.K. Yeo, and Z. Lin, "Driver distraction detection using semi-supervised machine learning," IEEE Trans. Intell. Transp. Syst., vol. 14, no. 4, 2016, pp. 1108-1120.
DOI: 10.1109/TITS.2015.2496157

[11] The Optimum project, http://www.optimumproject.eu/

[12] The Adria Mobil Company, http://www.adria-mobil.com/

[13] Raspberry Pi 3 model B, https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

[14] The VESNA platform, http://sensorlab.ijs.si/hardware.html

[15] S. Taherizadeh and V. Stankovski, "Auto-scaling Applications in Edge Computing: Taxonomy and Challenges," in Proceedings of the ACM International Conference on Big Data and Internet of Things (BDIOT 2017), London, United Kingdom, 2017, pp. 158-163.

[16] S. Taherizadeh and V. Stankovski, "Dynamic multi-level auto-scaling rules for containerized applications," The Computer Journal, Oxford University Press, 2018, pp. 1-24. doi: 10.1093/comjnl/bxy043

[17] A. Videla and J. Williams, RabbitMQ in action: Distributed messaging for everyone. Manning, 1st ed., Shelter Island, NY, April 2012.

[18] C. Saiprasert, T. Pholprasit and W. Pattara-atikom. "Detecting driving events using smartphone," in Proceedings of the 20th ITS World Congress, 2013.

[19] LSM303DLHC, https://github.com/ControlEverythingCommunity/LSM303DLHC

[20] B. Chikhaoui, B. Ye, and A. Mihailidis, "Aggressive and agitated behavior recognition from accelerometer data using non-negative matrix factorization," Journal of Ambient Intelligence and Humanized Computing, 2017, pp. 1-15.

[21] P. Chaovalit, C. Saiprasert, and T. Pholprasit, "A method for driving event detection using SAX on smartphone sensors," in Proceedings of 13th Int. Conf. ITS Telecommun., Tampere, Finland, Nov. 2013, pp. 450-455.

[22] S. Peters and K. Iagnemma. "Mobile robot path tracking of aggressive maneuvers on sloped terrain," in Proceedings of the Int. Conf. on Intelligent Robots and Systems, Nice, France, 2008, pp. 242-247.

[23] T. Zhang, G. Vosselman, and S.O. Elberink, "Vehicle Recognition In Aerial Lidar Point Cloud Based On Dynamic Time Warping," in ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Wuhan, China, 2017, pp. 193-198.

[24] The PrEstoCloud project, http://prestocloud-project.eu/