ACTA IMEKO
ISSN: 2221-870X
December 2020, Volume 9, Number 4, 88 - 95

Performance evaluation of a vehicular edge device for customer feedback in Industry 4.0

Marianne Silva[1], Gabriel Signoretti[1], Ivanovitch Silva[1], Paolo Ferrari[2]

[1] *Federal University of Rio Grande do Norte – Postgraduate Program in Electrical and Computer Engineering (PPgEEC), Natal, Brazil*
[2] *University of Brescia, Brescia, Italy*

ABSTRACT
Industry 4.0 is the term used to specify the current industrial revolution, not only from a technological point of view but also from economical, sociological and strategical points of view. The revolution involves several traditional economic sectors, as is the case with the industrial ecosystem. The main benefits are related to creating value during the entire product lifecycle and in terms of customer feedback, which is particularly relevant to the automotive industry. Its disruptive diffusion is due to various enabling technologies, such as the Internet of Things (IoT), and, as such, it is a vision rather than a technological step forward. Thus, this paper investigates a performance evaluation of an Edge OBD-II device, which collects data from vehicles in an autonomous way in order to provide customer feedback and tracking. The metrics evaluated were different sets of OBD-II Parameter IDs (PIDS), responsiveness, driver behaviour and CO_2 pollution estimates. The experiments were performed using three vehicles in urban and highway areas in the city of Natal, Brazil. For validation purposes, the results obtained from the vehicles were compared with an OBD-II Emulator, which demonstrated the accuracy of the experiments.

Section: RESEARCH PAPER

Keywords: Industry 4.0; vehicles; OBD-II; edge computing; pollution

Citation: Marianne Silva, Gabriel Signoretti, Ivanovitch Silva, Paolo Ferrari, Performance Evaluation of a Vehicular Edge Device for customer feedback in Industry 4.0, Acta IMEKO, vol. 9, no. 4, article 12, December 2020, identifier: IMEKO-ACTA-09 (2020)-04-12

Section Editor: Francesco Bonavolonta, University of Naples Federico II, Italy

Received September 26, 2019; **In final form** December 1, 2019; **Published** December 2020

Corresponding authors: Marianne Silva, e-mail: dinizmarianne@edu.ufrn.br, Ivanovitch Silva, e-mail: ivan@imd.ufrn.br

1. INTRODUCTION

Various industries have been undergoing several modifications since the advent of the Internet of Things (IoT) concept, which is changing the way human beings interact with the world. This is accomplished through the information exchange between interconnected physical objects and platforms. These can communicate with each other as well as with users through smart sensors and software that transmit data through a given network [1], [2]. It essentially presents an emergent paradigm as it comprises hardware infrastructure, software, and services that enable physical objects, known as 'things', to be connected to the Internet [3], [4].

Due to the new logistical challenges faced within the industrial ecosystem, the concept of the Industrial Internet of Things (IIoT) has emerged, which is considered to be one of the main stimulus mechanisms of the current revolution in the field of industrial automation [5]]-[7]. This concept, in turn, relates to the notion of Industry 4.0, a term introduced in 2011 as an initiative of the German government with a focus on improving the efficiency in the manufacturing industry [8], [9].

Industry 4.0 disseminates how business objectives, intelligent algorithms, analytics, predictive technologies, and cyber-physical systems should be combined to establish a new way of thinking about production management and factory transformation [10]. Furthermore, as [11]]-[13] note, Industry 4.0 is aimed at developing intelligent products that inherently collect and store information throughout their lifecycle, providing answers that can be used to improve the development and production processes.

It is notable that the various mechanisms that make up the industrial and commercial matrix are becoming more intelligent. Here, the data generated at all levels of the production process can be used to improve product quality, flexibility and productivity [5], [9].

In this scenario, recent years have seen the automotive industry focus on the development of smart vehicles equipped with various sensors [14], with interfaces also available to automatically capture and extract digital information through the sensors and the communication protocols present in the cars themselves [3]. It is expected that the vast majority of vehicles manufactured up until 2035 will have the capacity for wireless communication [14], with the vehicles connected via various types of devices (e.g. sensors, smartphones, cameras), wireless communication protocols and media broadcast systems [15]. These advancements can be categorised as the development of a new paradigm, that is, the IoIV paradigm [16], [17].

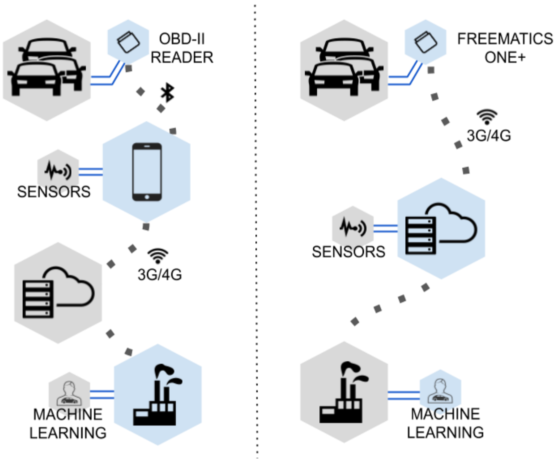
As a result, the extracted data from the various vehicle systems can be used for interactions within a new processing and communication context, leading to a revolution in the way vehicles are used [3], [18], [19]. As such, a combination of the IoIV concept and Industry 4.0 can enable a more efficient and accurate decision-making process for vehicle manufacturing organisations [20]. As Industry 4.0 is focused on exchanging and collecting information throughout the entire product lifecycle [12], these results can be accomplished by promoting a feedback loop from the final users (the vehicle owners) and the manufacturers (i.e. a constructive return of the vehicle information to the industry).

In order to enable services such as those previously mentioned, a number of solutions have been proposed in the existing literature [3], [18], [19], [21], which include the means to capture vehicular data within the context of the IoIV. These studies adopt a smartphone or a Raspberry Pi as a communication module alongside the vehicle's on-board diagnostics (OBD-II) system, a self-diagnosis system available on most vehicles currently in circulation [22] (Figure 1a). In this work, the Freematics One+ is adopted,[1] which is a device that does not depend on a smartphone to communicate with the OBD-II interface, as shown in Figure 1b. In fact, the device operates autonomously with access to the vehicle's engine control unit (ECU), making it possible to perform computations on the device itself before sending data to the cloud.

The Freematics One+ is an Arduino-based programmable vehicle telematics prototyping platform in the form of an OBD dongle. It plugs into a vehicle’s OBD port and works as a standalone device with access to the vehicle’s ECU, the high-rate Global Navigation Satellite System (GNSS), the 9-DOF (Degrees of Freedom) motion sensor and possibly a number of external sensors. The collected data can be processed in real time, stored in an internal Flash or micro SD card and transmitted via Bluetooth, WiFi or 3G connections.

The autonomous operating mode of the Freematics One+ presents a direct benefit for the implementation of edge computing, a technology applied to devices that performs data processing and can ensure maximum security and reliability [23], [24], while it also provides the capacity of local storage and computing [23], meaning the device can operate even when not connected to the Internet.

Due to these properties, an important factor that must be taken into account when using edge computing is the response time of the requests. Given that some of the current applications require a response time of around 20 ms, a substantial delay can cause serious damage to the driver or the machine [25], [26]. Within this context, this paper aims to evaluate the response time of the OBD-II parameter ID (PID) requests and proposes a state machine to model the driver behaviour while evaluating the CO_2 emission estimation. All the data were collected via an experiment involving vehicles that were selected in the city of Natal, Brazil for convenience. The results indicated that the request time of the PIDs depends on each vehicle and not on the type or quantity of the PIDs themselves.



(a) TRADITIONAL OBD-II (b) OBD-II FREEMATICS ONE+

Figure 1. Architecture overview.

The remainder of the paper is organized as follows. First, section 2 outlines the OBD-II system. Section 3 then discusses the related works, while the experimental evaluation methodology is described in section 4. The results are provided in section 5, while section 6 outlines the issues with the validity of the study. Finally, the conclusions and the future directions are provided in section 7.

2. OBD-II

The OBD-II is a hardware that is connected to the vehicle's ECU, as shown in Figure 2. It provides access to the status of the various vehicle subsystems, allowing the reading and transmission of the data measured by the sensors and actuators of the vehicle in real time [27].

This interface allows for connection to the vehicle through various different protocols. However, generally, each vehicle implements only one of these protocols. The most commonly used protocol is the controller area network (CAN) created by Bosh in 1980. The main rationale behind its creation was to allow communication between different ECUs. Meanwhile, a variety of other widely used protocols exist, including SAE J1850 PWM, SAE J1850 VPW, ISO 9141-2 and ISO 14230 KWP2000 [28].

The OBD-II technology implements 10 operating modes, with each mode designed for a specific purpose. For every mode, there are distinct sets of available commands that are used to obtain data from the sensors and actuators of the vehicles [3]. Here, the data is requested using codes known as parameter IDs or PIDs. However, manufacturers are not required to implement all modes in their vehicles. Therefore, prior to any data request, it is necessary to check which commands are supported by the vehicle.

The data collection and analysis system (OBD-II) is plugged into the vehicle’s OBD-II port and is subsequently configured (through its internal connection or via smartphones) to send the recorded data to the cloud database through mobile networks.

[1] https://freematics.com/pages/products/freematics-one-plus/

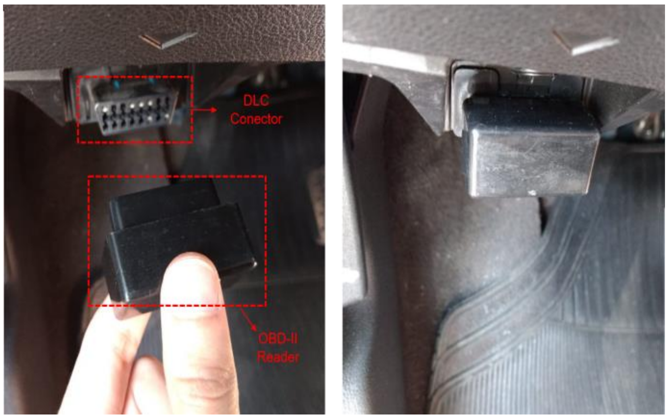


Figure 2. Communication interface between a typical OBD-II device and a vehicle.

3. RELATED WORKS

Following a systematic review of the literature, several papers were identified that influenced the research carried out and, therefore, contributed to the development of the proposed solution.

A fleet management system based on the OBD-II was implemented in [29]. Here, an OBD-II reader was designed to measure the speed and mass air flow (MAF) from which the distance and fuel consumption were also calculated. Many tests were performed to assess the system functionalities, while none were related to the system performance. Similarly, a customer feedback platform for the automotive industry within the context of Industry 4.0 is described in [3]. Such a platform is capable of collecting and analysing data from sensors available in the vehicles through an OBD-II scanner. This platform aims to assist in the management, prevention, and mitigation of different vehicular issues. What differs from the proposed solution is that, much like with the previously mentioned work, the response time of the PIDs requests was not evaluated.

Working along similar lines, a graph database solution to Industry 4.0 was proposed in [30] to manage the large amount of data generated by the sensors disseminated inside and around the vehicle in the urban environment. A number of benefits were recognised through the geo-localisation of the vehicles and the identification of specific events. However, this solution differs from the proposed solution in that it only evaluates the performance of the vehicle based on the data collected.

Elsewhere, a platform was proposed in [22] with the aim of estimating the amount of CO_2 emissions based on the readings of the vehicles sensors in order to monitor vehicular pollution. However, it was not applied to the context of Industry 4.0. Furthermore, a mobile measurement system for urban pollution monitoring that can be installed on public ground transportation vehicles was proposed in [31]. This system includes several important features that make it suitable for the specific purpose, namely, the low cost, the reduced dimensions, the autonomous power supply, and the measurement strategy capable of minimising the measurement uncertainty. However, it differs from the proposed work as it does not make use of the OBD-II.

In fact, it is clear that the OBD-II reading performance in terms of all sensors has yet to be evaluated, which means there remains a gap in the literature that must be explored to promote the development of new solutions incorporating OBD-II and IoIV technology.

4. EXPERIMENT

This section describes the methodology used to plan and execute the experiments.

4.1. Goal Definition

The main objective of this paper was to evaluate the response time of the OBD-II PIDs requests. In addition, a state machine for modelling driver behaviour was proposed in order to clarify the results, while estimations of CO_2 emissions were also performed.

4.2. Planning and Evaluation Design

This subsection details all the evaluation design aspects. In terms of context selection, the evaluation targeted vehicles manufactured in Brazil from 2010 on, since this was the year when the implementation of the OBD-II system in vehicles became mandatory. Meanwhile, in terms of research context, the questions we are attempting to address are as follows: **1)** what is the driver behaviour during the test?; **2)** what is the time distribution for the request of all supported PIDs in the vehicles, considering both real and simulated scenarios?; **3)** what is the performance of the request response time when considering the same set of PIDs for each vehicle?; **4)** what is the difference between the response time of the PIDs requests in relation to the CAN?; and **5)** what is the estimated CO_2 emission rate during the test?

Finally, in terms of sample selection, this was based on convenience and the availability of the drivers and specific vehicles, as described in Table 1.

4.3. Experimental Design

The experiment was designed in relation to a route with urban and highway sections in the city of Natal, Brazil. The pre-shift route was 30 km long and the drivers took the same route in two schedules (13 h and 16 h) on different days. The experiment was conducted according to these schedules in order to obtain more variability in the data, given that the traffic in the urban section at 16 h is heavier than at 13 h.

4.4. Instrumentation

The process was initiated with the configuration of the environment for the experiment and the planning of the data collection procedure. The settings were coded into the Freematics One+ and were defined as follows: connection type to 3G, server hosting on the Amazon Web Service, PID set request interval configured to 500 ms, and the interval for sending data to the server set to 5 s intervals.

4.5. Execution

The experiment was performed according to the state machine presented in Figure 3. The initial state is with the engine off (‘Start’). Immediately after turning the vehicle on, the status of the state machine can switch to ‘Stopped’ or ‘Moving’. Ultimately, the driver can step on the brake at certain points without stopping the vehicle, only reducing the speed. In addition, the driver can stop due to a semaphore, a traffic jam or

Table 1. Availability of vehicles.

Model	Year	Motor	Fuel	PIDs
Ford Ka	2019	1.5	flexible	21
Nissan Kicks	2017	1.6	flexible	22
Chevrolet Onix	2015	1.4	flexible	22

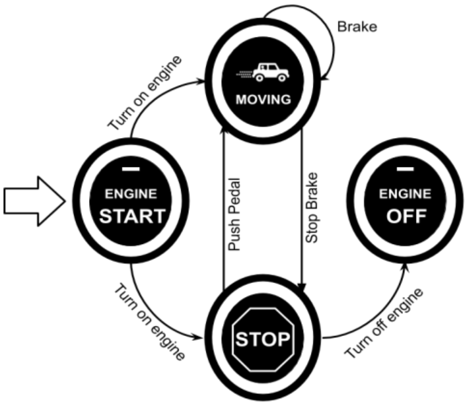


Figure 3. State machine modelling the driver behaviour during the experiment.

other similar situations. Finally, the execution of the experiment is completed when the car is turned off.

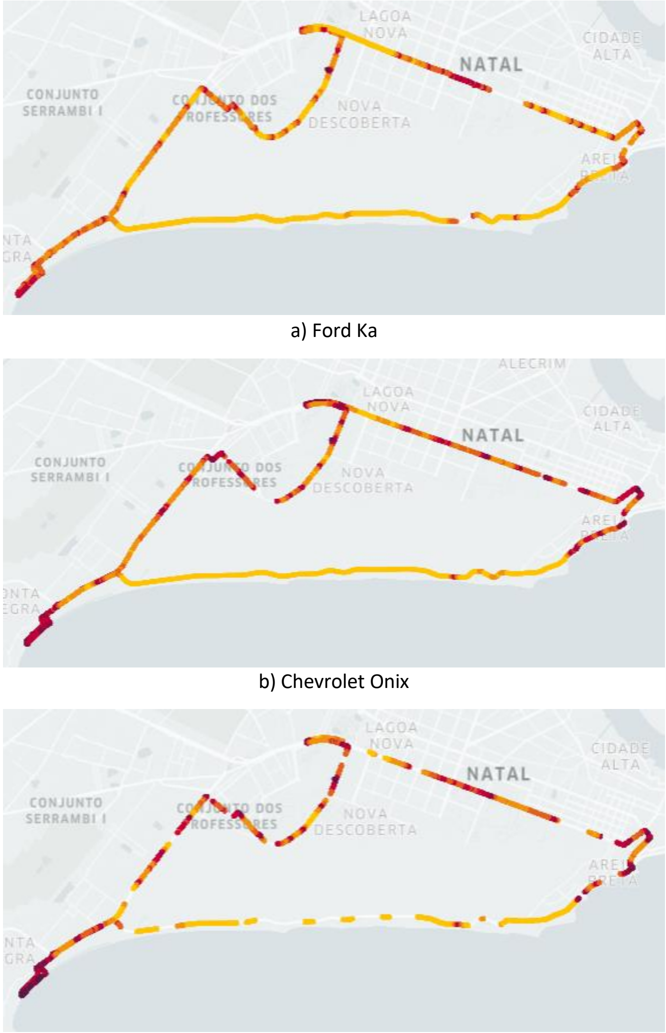
Following this process, data was collected for around 180 km, with 30 km for each round of the experiment per vehicle. Of these, 60 % was collected on urban roads and 40 % on highways. The experiments totalled (after the cleaning process) 1.4 MB (8,557 samples) for the Ford Ka, 370 KB (2,152 samples) for the Nissan Kicks and 1.7 MB (10,230 samples) for the Chevrolet Onix. OBD-Emulator MK2[2] hardware was adopted for the simulation scenario. The simulation period was configured to run for 60 minutes, which was the average time of the real experiments. The collected samples were approximately 9,678, 11,448, and 10,680 records, respectively, which corresponds to 1.6 MB, 1.9 MB and 1.8 MB.

5. RESULTS AND DISCUSSION

This section aims to answer the research questions previously proposed in section 4.

As noted in subsection 4.3, the experiment established a default route to be performed by all the drivers in their respective vehicles. Figure 4 shows the map of the first trip of each vehicle. It should be noted that the points are colour graded based on the speed of the vehicle at that specific time, red being slower and yellow being faster. It is also possible to see areas with large gaps in the readings, especially on the route performed by the Nissan Kicks, as can be observed in Figure 4b. These gaps may have been caused by a poor internet connection, resulting in the data not being sent to the remote server (hosted on the Amazon Web Service). In this specific case, the connection flaws were likely caused by the large rain precipitation at the time of the experiment, which may have interfered with the communication network.

First, as Figure 5 shows, it was possible to compare different driving styles (e.g. stopped ratio) to address our first question. In addition, the average values of five PIDs were also analysed: speed, revolutions per minute (RPM), engine load, throttle position and battery voltage. With these six variables, we built two radar plots (one for the routes performed at 13 h and one for those performed at 16 h) to compare the drivers and the routes. Figure 5a summarises the results of the profiles of the drivers in relation to the route taken at 13 h. The drivers had virtually the same profile in terms of RPM and battery voltage. The stopped ratios for both the Ford Ka and Chevrolet Onix vehicles were around six times larger than that for the Nissan Kicks. This clearly indicates the heavier traffic encountered by the drivers of the former. The experiment using the Nissan Kicks was carried out on a Saturday, when the traffic is generally lighter than on weekdays. The throttle position (position of the accelerator pedal) indicated that the Chevrolet Onix's driver had a more aggressive driving style, while the speed of the Nissan Kicks' driver was faster due to the lighter traffic. The increased engine load presented by the Ford Ka is justified by the fact that it was a brand-new vehicle, which means the engine is still adjusting.



a) Ford Ka

b) Chevrolet Onix

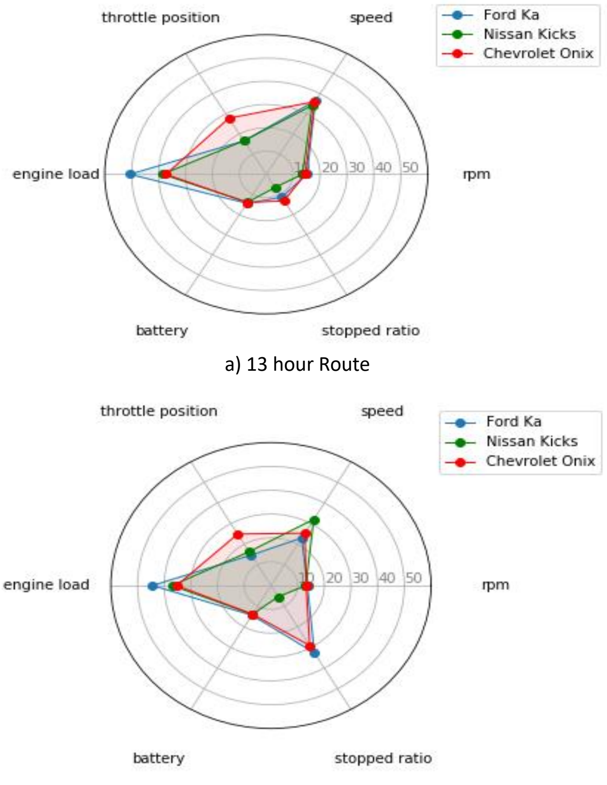
c) Nissan Kicks

Figure 4. Route performed by each vehicle.

In terms of the 16 h route shown in Figure 5b, we obtained a profile similar to the one in Figure 5a, albeit with one notable difference: the stopped ratio for both the Chevrolet Onix and the Ford Ka were almost triple the value of the previous observation. This clearly indicates the heavier traffic encountered by the drivers at that time compared to the earlier routes. Meanwhile, there was no significant difference in the time for the Nissan Kicks to complete the route. This is due to the fact that the experiment using this car was, as noted above, carried out on a Saturday with lighter traffic.

Regarding **question 2**, Figure 6 describes the reading time required to collect all PIDs supported by the vehicles, namely,

[2] https://freematics.com/products/freematics-obd-emulator-mk2/



a) 13 hour Route

b) 16 hour Route

Figure 5. Driver behaviour modelling by route.

maximum value, first quartile (*Q1*), mean (red diamond), median (grey line), third quartile (*Q3*), minimum value, outliers (grey diamond). According to Table 1, each vehicle used in the experiments has a different set of supported PIDs. The Ford Ka had an average response time for all 21 PIDs of 630 ms. These results are around 44 % greater than the simulation performance (438 ms). As expected, the simulation scenario had a better result given that it did not consider internal delays in the vehicle bus or communication errors, etc. The same behaviour was also verified in tests using the Nissan Kicks (22 PIDs, 39 % ratio) and the Chevrolet Onix (22 PIDs, 19 % ratio).

A further interesting result obtained from Figure 6 is related to the dispersion of the response time in order to collect all PIDs. The interquartile ranges (IQR) for the Ford Ka, Nissan Kicks and Chevrolet Onix vehicles were very close: a) real scenario – 7 ms, 7 ms, 4 ms, b) simulation scenario – 35 ms, 36 ms, 35 ms. The poorer result for the simulation scenario was due to the additional delay imposed by the serial connection between simulator and computer. These results demonstrated the response times involved a small variation and that they were concentrated around the mean. It is important to stress that during the experiments, transient communication failures occurred between the Freematics One+ and the vehicles. These delays generated a number of outliers in the results, while these were mainly related to the Nissan Kicks.

In addition to the average reading time of the PIDs, the standard deviation for each car and for the simulator was also investigated. The obtained results indicate that the standard deviation of the Nissan Kicks was the largest at 27.9 ms. This implies that the reading time of the PIDs for the Nissan Kicks varied more between each sample than those for the Ford Ka and the Chevrolet Onix, which were 7.9 ms and 3.6 ms, respectively. Here, the reading times of each sample were closer to the respective mean for both cars. With these values, we calculated the minimum and maximum reading times of all PIDs per car and obtained the amplitude. As expected, the results exhibited the same trend as with the standard deviation values. The reading time amplitude for the Nissan Kicks, Ford Ka, and Chevrolet vehicles was 145 ms, 83 ms, and 29 ms, respectively. In contrast, the standard deviation for the simulator was around 15.2 ms. In addition, the simulator presented a reading time amplitude for the PIDs of the Ford Ka of 42 ms, while for the Nissan Kicks and the Chevrolet Onix, we obtained a value of 43 ms.

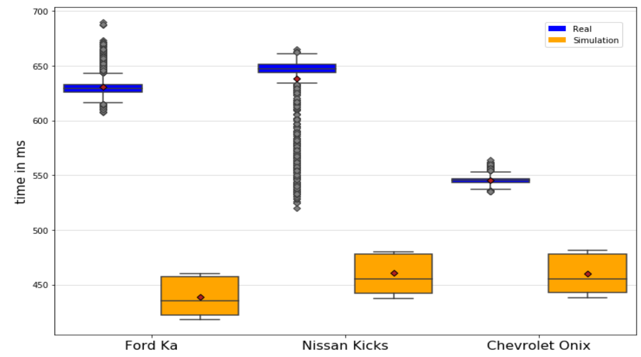


Figure 6. Response time distribution for all supported PIDs.

A further point that must be highlighted is related to the PID/server request interval of 500 ms and 5 s, respectively. The experiments verified that it was only possible to achieve the metric of the PID request interval in the simulator. For clarity, let us look at the specific case of the Nissan Kicks, which had the longest PID request time of around 638 ms. As such, only eight packets could be sent to the server in the 5 s interval range. However, it was possible to send 10 packets in the same interval range with the simulator.

Regarding **question 3**, we analysed the data response time for only the intersection of the PIDs supported by the Ford Ka, Nissan Kicks and Chevrolet Onix vehicles. This intersection resulted in a list of 17 PIDs. Figure 7 shows the cumulative distribution of response time for this set of PIDs. Even with using the intersection of the PIDs, the response times for the real scenarios differed and were greater than those achieved by the simulator. The response times obtained for the Ford Ka, the Nissan Kicks and the Chevrolet Onix were 513 ms, 492 ms and 421 ms respectively. As expected, the simulation obtained the lowest response time (356 ms). Meanwhile, the average response rate was 30 ms for the Ford Ka, 28 ms for the Nissan Kicks and 24 ms for the Chevrolet Onix. The average simulator response time was around 20 ms. Hence, it can be concluded that the

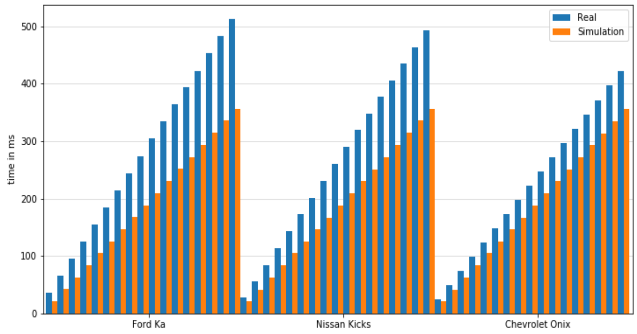


Figure 7. Cumulative distribution of response time for the PID set intersection among the Ford Ka, Nissan Kicks and Chevrolet Onix vehicles.

response time of the PIDs depends on each vehicle and not on the type or quantity of the PIDs.

To address **question 4**, we aimed to investigate whether or not there was a difference in the reading time of the PIDs with respect to the CAN. Here, we verified that the CAN for all the cars is ISO15765-4 CAN11/500, while it was observed that there were different response times even though all cars operate under the same protocol. As such, the CAN protocol may not be a relevant factor for the response time.

Finally, to address **question 5,** we analysed the $\mathrm{CO_2}$ emission rate estimates. To perform the required calculation, it is necessary to ascertain the amount of air that enters the engine at any given time. This value can be obtained directly by the MAF PID. However, this sensor is not available in some vehicles. In such cases, the value can still be estimated using the approach used in [26].

The approach makes use of the following variables: ***P***, which represents the pressure in the combustion chamber and can be obtained through the intake manifold absolute pressure (MAP) sensor in the unit kPa; ***V***, which is the total volume of the vehicle's cylinder capacity in litres or cm³; ***R***, which is the ideal gas constant with a value of approximately 8.3145 J/(mol·K); ***T***, which is the intake air temperature and can be obtained by the absolute temperature (IAT) sensor in K; **air molar mass**, which is a constant value equal to 28.87 g/mol (in the following equations, the air molar mass will be abbreviated to *mma*); and ***RPM*,** which is the engine revolutions per minute obtained by the OBD-II. In addition to these variables, the **volumetric efficiency** (*VE*) is required, which is the ratio of the air-fuel mixture volume that each cylinder admits in relation to the nominal volumetric capacity of the cylinder.

Thus, the calculation of mass air flow (*MAF*) can be written as in Eq. (1), which corresponds to a value equivalent to that obtained directly by the MAF sensor when it is available.

$$MAF = \frac{P \times V \times VE \times RPM \times mma}{120 \times R \times T \times 1000} \tag{1}$$

After obtaining the *MAF* value, the estimate of the CO_2 emission rate emitted by the vehicle could be calculated. For this, it was first necessary to calculate the fuel volume (V_{fuel}) according to Eq. (2):

$$V_{\mathrm{fuel}} = \frac{MAF}{AFR \times density} \tag{2}$$

The values of the constants used in Eq. (2) vary depending on the type of fuel used, as described in Table 2.

Finally, Eq. (3) could be used to calculate the estimated CO_2 emission rate of the vehicle. In this formula, the V_{fuel} value is multiplied by the carbon dioxide mass generated after burning one litre of the fuel in question (CO_2 *PL*)

$$CO_2\left(\frac{g}{s}\right) = V_{\mathrm{fuel}} \times CO_2\ PL\,. \tag{3}$$

Table 2. Fuel conversion factors.

Fuel	$\mathrm{CO_2}$ Per Liter	Air-Fuel Ratio (AFR)	Density ($\mathcal{P}$)
Gasoline	2310 g/L	14.7:1	737 g/L
Diesel	2660 g/L	14.6:1	850 g/L
Ethanol	1510 g/L	9.0:1	789 g/L

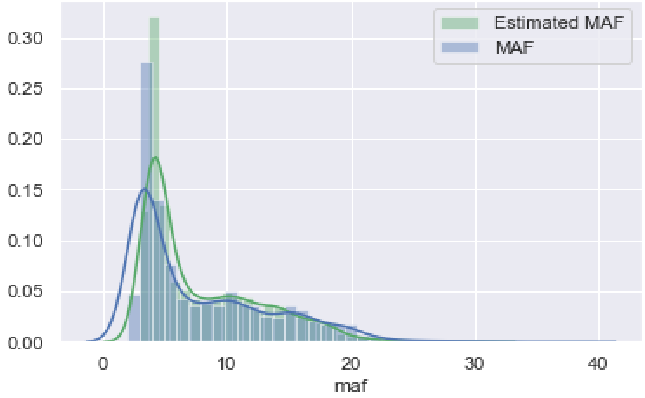


Figure 8. MAF estimate.

With these equations at hand, it was possible to perform embedded processing in the Freematics One+ to obtain the results in real time.

As stated above, not all vehicles have the same set of sensors and, as such, may lack the MAF of the MAP. Therefore, to validate the MAF estimation using the other vehicle sensors, we used data from the Chevrolet Onix, as it was the only vehicle analysed that had both sensors (MAF and MAP). The comparison between the real MAF sensor reading and the MAF estimation results can be seen in Figure 8.

In addition, Figure 9 shows the comparison between the emission rate calculated directly from the MAF sensor and that calculated with the estimated MAF. Here, it is clear that the results are very close to each other, as was the case for the comparison presented in the previous figure.

Finally, Figure 10 allows us to identify the places with the highest incidence of $\mathrm{CO_2}$ emission during the routes. The darkest points represent the places where the emission rate was higher.

These analyses can contribute to identifying the emission indices presented by the vehicles during normal use and can help verify whether they are equivalent to those issued by the automakers.

6. LIMITATIONS

During the use of the OBD-II (Freematics One+), the chosen route incorporated urban areas as well as highways. Given that, it was noted that in certain sections of the route, the hardware did not capture the information related to the geolocation of the

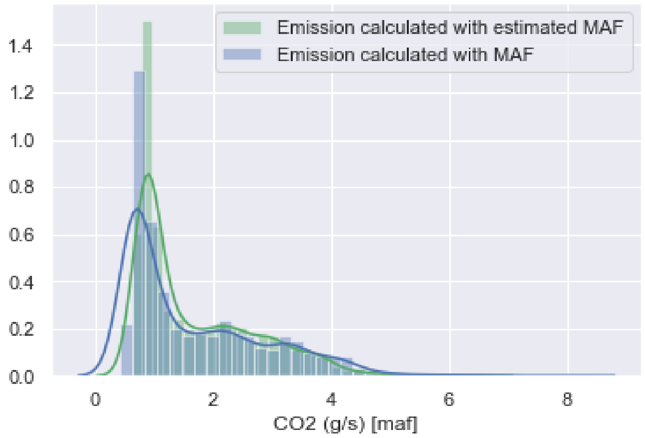
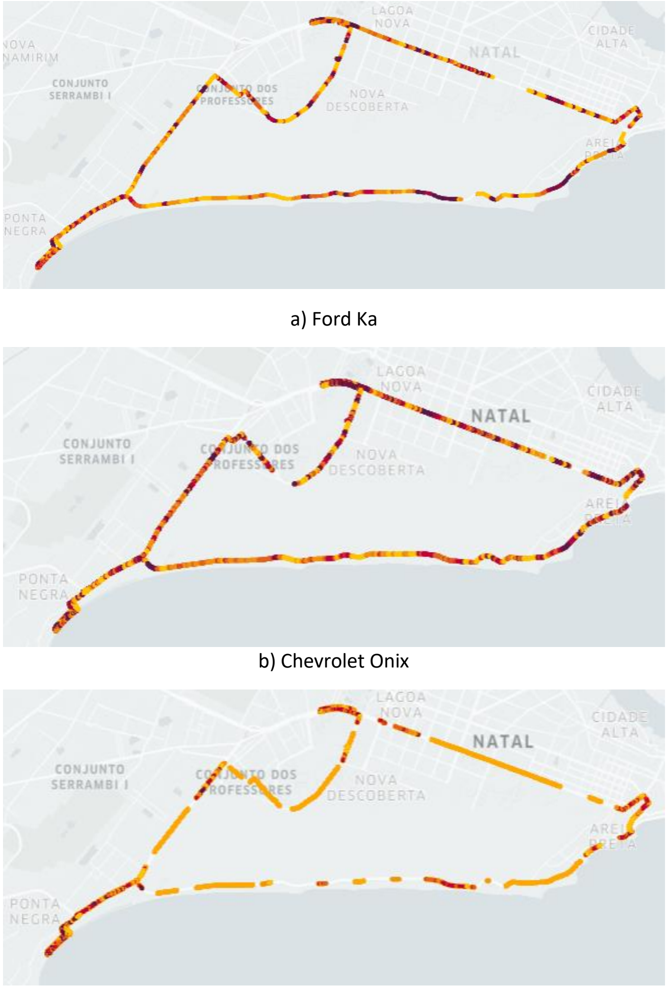


Figure 9. $\mathrm{CO_2}$ pollution estimate.



a) Ford Ka

b) Chevrolet Onix

c) Nissan Kicks

Figure 10. CO_2 pollution estimate by route.

vehicles. This was due to the lack of GPRS or 3G signal, making it impossible to send the data to the cloud. In addition, further difficulties were encountered during the implementation of the application used in the study, including how, since it was not a controlled environment (as in the simulator), the connection was prone to errors, which is a factor that must be addressed.

7. VALIDITY ISSUES

The threats to the validity of the present study are as follows:

a. **Internal validity**, driver behaviour: each driver has a unique driving profile, which may have affected the total time needed to complete the route.
b. **External validity**, **connection**: connection failures can cause communication error with the server, resulting in a timeout that could have led to a loss of data. A further external threat relates to the varying traffic conditions, which could have increased the time needed to complete the route.
c. **Construction validity**, appropriate instrumentation: the vehicles were evaluated on different days, since it was not intended to make any type of comparison regarding the day of the week, only a verification of the performance of the OBD-II.

8. CONCLUSIONS

The aim of this paper was to conduct a performance evaluation of an Edge OBD-II device (Freematics One+). The main requirements were driver behaviour modelling, PID request interval evaluation and CO_2 emission estimation. For this, experiments were performed in terms of both real and simulation scenarios. The experiments involving the real vehicles present a differential contribution to the existing literature since they have the potential to highlight important issues within a customer feedback context (e.g. bottlenecking, delays, amount of data, profiles) which is a core part of Industry 4.0. The first requirement (driving profile) was satisfied using the proposed state machine, where, with the use of only five PIDs, aggressive and/or moderate behaviours were identified. The results related to the response time of the PIDs demonstrated that this depends on the specifications of each vehicle and not on the type or quantity of the PIDs required. In fact, this aspect is crucial to determining the amount of data that will be used in a given customer feedback vehicular application. In terms of the final, complementary, requirement (CO_2 emission estimation), it was possible to identify the pollution and MAF estimations for vehicles that do not support the sensor.

Future work should include investigating the dependability issues in order to address the faults of the OBD-II device and the communication issues with the cloud infrastructure. In addition, the following should be included: increasing the sample size, analysing why no data was written to the SD card at the point that it failed to communicate with the Internet, implementing a data compression algorithm (such as the 'swinging door' algorithm), identifying the optimal request time for PIDs, and making use of Big Data and machine learning techniques to identify patterns in the collected data.

ACKNOWLEDGEMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, with grateful thanks going to the Brazilian fostering agency CNPq (National Council for Scientific and Technological Development), Process No. 435683/2018-7.

REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of things (iot): A vision, architectural elements, and future directions, Future generation computer systems 29(7) (2013), pp. 1645-1660. DOI: https://doi.org/10.1016/j.future.2013.01.010

[2] Z. Marafie, K. Lin, Y. Zhai, J. Li, Proactive fintech: Using intelligent IoT to deliver positive insurTech feedback, Proc. of the IEEE 20th Conference on Business Informatics (CBI), vol. 02, Vienna, Austria, 11-14 July 2018, pp. 72–81. DOI: https://doi.org/10.1109/CBI.2018.10048

[3] M. Silva, E. Vieira, G. Signoretti, I. Silva, D. Silva, P. Ferrari, A customer feedback platform for vehicle manufacturing compliant with industry 4.0 vision, Sensors 18(10) (2018), 3298, 24 pages. DOI: https://doi.org/10.3390/s18103298

[4] E. Balestrieti, L. De Vito, F. Lamonaca, F. Picariello, S. Rapuano, I. Tudosa, Research challenges in measurement for Internet of Things systems, Acta IMEKO, 7 (2019) 4, pp. 82-94. DOI: https://doi.org/10.21014/acta_imeko.v7i4.675

[5] M. S. Rocha, G. S. Sestito, A. L. Dias, A. C. Turcato, D. Brandão, P. Ferrari, On the performance of OPC UA and MQTT for data exchange between industrial plants and cloud servers, Acta IMEKO 8 (2019) 2, pp. 80-87. DOI: https://doi.org/10.21014/acta_imeko.v8i2.648

[6] A. Schutze, N. Helwig, T. Schneider, Sensors 4.0 - smart sensors and measurement technology enable industry 4.0, Journal of Sensors and Sensor Systems 7(1) (2018), pp. 359-371. DOI: https://doi.org/10.5194/jsss-7-359-2018

[7] W.-C. Wu, H.-T. Liaw, The next generation of internet of things, in: Frontier Computing. J. C. Hung, N. Y. Yen, L. Hui (editors). Springer, Singapore, 2018, ISBN-13 978-981-10-7398-4, pp. 278-282.

[8] S. Weyer, M. Schmitt, M. Ohmer, D. Gorecky, Towards industry 4.0-standardization as the crucial challenge for highly modular, multivendor production systems, Ifac-PapersOnLine 48(3) (2015) pp. 579-584. DOI: https://doi.org/10.1016/j.ifacol.2015.06.143

[9] J. Lee, B. Bagheri, H.-A. Kao, A cyber-physical systems architecture for industry 4.0-based manufacturing systems, Manufacturing Letters 3 (2015) pp. 18-23. DOI: https://doi.org/10.1016/j.mfglet.2014.12.001

[10] J. Lee, H.-A. Kao, S. Yang, Service innovation and smart analytics for industry 4.0 and big data environment, Procedia CIRP 16 (2014) pp. 3-8. DOI: https://doi.org/10.1016/j.procir.2014.02.001

[11] W. Scheidel, I. Mozgova, R. Lachmayer, Structuring information in technical inheritance with pdm systems, Proc. of the 21st Int. Conf. on Engineering Design (ICED 17), Vancouver, Canada, 21-25 August 2017, Vol 6: Design Information and Knowledge, ISBN: 978-1-904670-94-0, pp. 217-226.

[12] P. Ferrari, A. Flammini, E. Sisinni, S. Rinaldi, D. Brandão, M. S. Rocha, Delay estimation of industrial IoT applications based on messaging protocols, IEEE Transactions on Instrumentation and Measurement 67(9) (2018) pp. 2188-2199. DOI: https://doi.org/10.1109/TIM.2018.2813798

[13] F. Ferreira, J. Faria, A. Azevedo, A. Marques, Product lifecycle management enabled by industry 4.0 technology, Advances in Transdisciplinary Engineering 3 (2016) pp. 349-354. DOI: https://doi.org/10.3233/978-1-61499-668-2-349

[14] J. Contreras-Castillo, S. Zeadally, J. A. G. Ibanez, A seven-layered model architecture for internet of vehicles, Journal of Information and Telecommunication 1(1) (2017) pp. 4-22. DOI: https://doi.org/10.1080/24751839.2017.1295601

[15] F. Yang, S. Wang, J. Li, Z. Liu, Q. Sun, An overview of internet of vehicles, China Communications 11 (2014) pp. 1-15. DOI: https://doi.org/10.1109/CC.2014.6969789

[16] T. T. Dandala, V. Krishnamurthy, R. Alwan, Internet of vehicles (iov) for traffic management, Proc. of the International Conference on Computer, Communication and Signal Processing (ICCCSP), 2017, pp. 1-4. DOI: https://doi.org/10.1109/ICCCSP.2017.7944096

[17] H. Qiu, M. Qiu, Z. Lu, G. Memmi, An efficient key distribution system for data fusion in v2x heterogeneous networks, Information Fusion, vol. 50, (2019), pp. 212-220. DOI: https://doi.org/10.1016/j.inffus.2019.02.002

[18] M. Da Silva, E. Vieira, I. Silva, D. Silva, P. Ferrari, S. Rinaldi, D. Fernandes Carvalho, A customer feedback platform for vehicle manufacturing in industry 4.0, Proc. of IEEE Symposium on Computers and Communications (ISCC), Natal, Brazil, 25-28 June 2018, pp. 1249-1254. DOI: https://doi.org/10.1109/ISCC.2018.8538554

[19] D. G. Costa, A. Damasceno, I. Silva, Cityspeed: A crowdsensing-based integrated platform for general-purpose monitoring of vehicular speeds in smart cities, Smart Cities 2(1) (2019) pp. 46-65. DOI: https://doi.org/10.3390/smartcities2010004

[20] D. Lin, C. Lee, H. Lau, Y. Yang, Strategic response to industry 4.0: an empirical investigation on the Chinese automotive industry, Industrial Management & Data Systems, 2018, ISSN: 0263-5577.

[21] G. S. Thirunavukkarasu, B. Champion, B. Horan, M. Seyedmahmoudian, A. Stojcevski, IoT-based system health management infrastructure as a service, Proc. of the ACM International Conference on Cloud Computing and Internet of Things, Singapore, Singapore, 29-31 October 2018, pp. 55-61 DOI: https://doi.org/10.1145/3291064.3291070

[22] M. Silva, G. Signoretti, J. Oliveira, I. Silva, D. G. Costa, A crowdsensing platform for monitoring of vehicular emissions: A smart city perspective, Future Internet 11(1) (2019) 13, 20 pages. DOI: https://doi.org/10.3390/fi11010013

[23] Y. Mao, C. You, J. Zhang, K. Huang, K. B. Letaief, A survey on mobile edge computing: The communication perspective, IEEE Communications Surveys Tutorials 19 (2017) pp. 2322-2358. DOI: https://doi.org/10.1109/COMST.2017.2745201

[24] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, W. Wang, A survey on mobile edge networks: Convergence of computing, caching and communications, IEEE Access 5 (2017) pp. 6757-6779. DOI: https://doi.org/10.1109/ACCESS.2017.2685434

[25] J. Liu, G. Shou, Y. Liu, Y. Hu, Z. Guo, Performance evaluation of integrated multi-access edge computing and fiber-wireless access networks, IEEE Access 6 (2018) pp. 30269-30279. DOI: https://doi.org/10.1109/ACCESS.2018.2833619

[26] I. M. D. Silva, L. A. Guedes, F. Vasques, Performance evaluation of a compression algorithm for wireless sensor networks in monitoring applications, Proc. of the IEEE International Conference on Emerging Technologies and Factory Automation, Hamburg, Germany, 15-18 September 2008, pp. 672-678. DOI: https://doi.org/10.1109/ETFA.2008.4638468

[27] G. Signoretti, M. Silva, A. Dias, I. Silva, D. Silva, P. Ferrari, Performance evaluation of an Edge OBD-II device for Industry 4.0, Proc. of the II IEEE Workshop on Metrology for Industry 4.0 and IoT (MetroInd4. 0&IoT), Naples, Italy, 4-6 June 2019, pp. 432-437. DOI: https://doi.org/10.1109/METROI4.2019.8792885

[28] S. H. Baek, J.-W. Jang, Implementation of integrated OBD-II connector with external network, Information Systems 50 (2015) pp. 69-75. DOI: https://doi.org/10.1016/j.is.2014.06.011

[29] R. Malekian, N. R. Moloisane, L. Nair, B. Maharaj, U. A. Chude-Okonkwo, Design and implementation of a wireless OBD II fleet-management system, arXiv preprint arXiv:1701.02160, IEEE Sensors Journal, Volume 17, Issue 4, February 2017, pp. 1154-164. DOI: https://doi.org/10.1109/JSEN.2016.2631542

[30] A. Pieroni, N. Scarpato, M. Brilli, Industry 4.0 revolution in autonomous and connected vehicle a now-conventional approach to manage big data, Journal of Theoretical & Applied Information Technology 96(1) (2018). pp. 10-18. Online [Accessed 02 December 2020] http://www.jatit.org/volumes/Vol96No1/2Vol96No1.pdf

[31] A. Bernieri, D. Capriglione, L. Ferrigno, M. Laracca , Design of an efficient mobile measurement system for urban pollution monitoring, ACTA IMEKO 1(1) (2012) pp. 77-84. DOI: https://doi.org/10.21014/acta_imeko.v1i1.27