infrastructures

MDPI

Article

Fuel Consumption Prediction for Construction Trucks: A Noninvasive Approach Using Dedicated Sensors and Machine Learning

Gonçalo Pereira [1,*], Manuel Parente [1], João Moutinho [1] and Manuel Sampaio [2]

[1] BUILT CoLAB, 4150-003 Porto, Portugal; manuel.parente@builtcolab.pt (M.P.); joao.moutinho@builtcolab.pt (J.M.)
[2] ISEP, 4200-072 Porto, Portugal; vilhenasampaio@gmail.com
* Correspondence: goncalo.pereira@builtcolab.pt

Abstract: Decision support and optimization tools to be used in construction often require an accurate estimation of the cost variables to maximize their benefit. Heavy machinery is traditionally one of the greatest costs to consider mainly due to fuel consumption. These typically diesel-powered machines have a great variability of fuel consumption depending on the scenario of utilization. This paper describes the creation of a framework aiming to estimate the fuel consumption of construction trucks depending on the carried load, the slope, the distance, and the pavement type. Having a more accurate estimation will increase the benefit of these optimization tools. The fuel consumption estimation model was developed using Machine Learning (ML) algorithms supported by data, which were gathered through several sensors, in a specially designed *datalogger* with wireless communication and opportunistic synchronization, in a real context experiment. The results demonstrated the viability of the method, providing important insight into the advantages associated with the combination of sensorization and the machine learning models in a real-world construction setting. Ultimately, this study comprises a significant step towards the achievement of IoT implementation from a Construction 4.0 viewpoint, especially when considering its potential for real-time and digital twins applications.

Keywords: cyberphysical systems; IoT; machine learning; construction machinery remote monitoring; fuel consumption

Citation: Pereira, G.; Parente, M.; Moutinho, J.; Sampaio, M. Fuel Consumption Prediction for Construction Trucks: A Noninvasive Approach Using Dedicated Sensors and Machine Learning. *Infrastructures* **2021**, *6*, 157. https://doi.org/10.3390/infrastructures6110157

Academic Editor: M. Amin Hariri-Ardebili

Received: 18 September 2021
Accepted: 2 November 2021
Published: 5 November 2021

1. Introduction

Fuel is one of the largest costs in construction and, in particular, in transportation infrastructure projects. Typically, decisions regarding heavy machinery allocation, scheduling, or performance evaluation are carried out using fuel consumption estimates based on experienced professionals or in generic specification documents and guides, such as the CATERPILLAR performance handbook [1]. These are usually a function of the machine type and the number of hours of work. On average, these estimations are good for most applications (budgeting, fuel stocking, etc.). However, when the objective is to improve processes in their efficiency (cost, time, resources, etc.), it becomes important to estimate fuel consumption in a more precise way and as a function of the input parameters such as the distance, slope, the vehicle's load, or others. As one can intuitively guess, fuel estimation may also be linked to the route characteristics, vehicle type, or driving behavior. One of the scientific challenges, approached here, relies on determining what characteristics influence the most fuel consumption in these vehicles. One key constraint for the success of this work is related to the fact that it is very difficult to acquire precise fuel consumption data from vehicles. Accurate fuel consumption is required to be used as the ground truth for Machine Learning (ML) algorithms, and therefore, it is critical to obtain these data reliably. In more modern trucks, for instance, the information is presented in the vehicle's

dashboard and fuel consumption is presented as instant consumption and/or average consumption (liters per kilometer or miles per gallon). On the one hand, this information is typically not reliable and the values presented are typically underestimated. On the other hand, extracting that visual information from the vehicles is a complex task, since the character's font and placement are specific to the manufacturer's system, rendering Optical Character Recognition (OCR) technology unfeasible to implement for such applications. Considering this, the considered alternatives are based on reading the vehicle's communication bus, in order to extract data concerning the flow of fuel for injection, the reservoir's level, or the fuel consumption sensor that some vehicles have. However, the access to the Controller Area Network (CAN) bus and to the On-Board Diagnostic (OBD) plug has been considered invasive and difficult to achieve in some vehicles. Therefore, it was decided that the approach should be noninvasive, thus compatible with every vehicle, and simple enough to rely on the drivers or operators in order to obtain reliable measurements and support. Yet, being able to indirectly estimate fuel consumption in such a way means that the necessary data must be acquired (sensors) and leveraged by resorting to prediction models (using machine learning).

The potential of machine learning applications in transportation infrastructures and geotechnics has been the target of considerable attention in the past decade [2]. Indeed, accompanying the ever-increasing development of remote monitoring and data warehousing technologies, successful machine learning applications in this field span several different areas, from earthworks productivity [3,4], slope safety [5], and jet grouting compressive strength [6], to pavement management and monitoring [7,8]. These can often address specific processes, such as the estimation of compaction work rate [9] or of excavator cycle time [10,11], as well as comprise an essential part of larger, more complex systems, such as fleet management and allocation systems [12] or pavement design and management systems [7,8].

The latter field has also seen several applications in the context of pavement condition assessment and maintenance [13–15]. A noteworthy aspect of these systems is related to the fact that they leverage concepts such as sensorization and digital twins to gather data, which, in turn, provides the basis for the training and testing databases. In other words, predictive models in these pavement management systems are trained on data stemming from different sensors placed in the field, either in maintenance vehicles [13,14] or in the pavement itself [7,8,15].

This notwithstanding, there has not been specific focus on the estimation of the costs associated with construction processes. In particular, the estimation of fuel consumption via a predictive machine learning model is a topic that, despite having had some developments in other fields such as logistics and long-haul truck routes [16–18], has not been given equivalent effort in the construction field. Thus, the availability of real-world sensor-based information allows the decision-making teams to properly manage their resources (both economic resources and physical ones, such as construction equipment) not only during the design and planning phases, but also during the construction phase itself, in which resource allocation can be adjusted as a function of restrictions and unpredictable occurrences (e.g., equipment malfunction, low productivity).

Taking into account the availability of data and the increasing affordability of remote sensors, this work aims to fill this gap in the literature, presenting the preliminary results of an ongoing field project carried out in association with a Portuguese road construction company. The project aims to study the viability of implementing a data-gathering remote-monitoring system to support data-driven models, which in turn attempt to estimate the fuel consumption of transportation equipment in a construction site as a function of aspects such as the transported cargo, the specifications of the equipment, and the characteristics of the chosen route. Naturally, the aforementioned study analyzes the requirements of the sensorization parameters for transportation equipment in construction environments (e.g., trucks, dumpers), since their activity is considerably different in comparison to other fields.

As such, this work is organized as follows: Section 2 depicts the methodology and prediction models that were adopted in this approach. Section 3 describes the experimental project that was developed in order to validate the methodology and the prediction models, but also to acquire data that will support the latter. Finally, the results and the associated discussion are presented in Section 4, while some conclusions and further work recommendations are drawn in Section 5.

2. Methodology and Prediction Models

The scope of this project is to implement an IoT-based sensing framework capable of retrieving field data and transmitting them in (near) real time, which, in turn, can be leveraged by machine learning algorithms to create a prediction model capable of estimating fuel consumption. The model is *trained* by resorting to both sensor data and a ground truth, which, in turn, can be used by applications to provide fuel estimations, as portrayed in Figure 1.

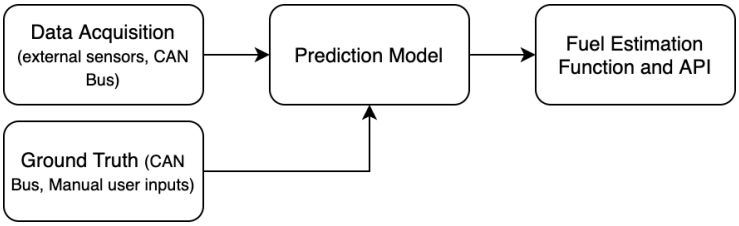


Figure 1. Fuel estimation workflow diagram.

Ultimately, one of the possible applications of this project is to provide a web application and an API that allow a user to input a given route, vehicle data, and carried load and retrieve fuel consumption estimations in order to assess the project cost based on real case scenario data. This platform could allow users to omit some parameters when not known, resulting in a prediction with a wider confidence interval, but still proving itself useful to predict the project's cost.

2.1. Data Acquisition

As previously mentioned, route characteristics, load, and vehicle classification seem, at first sight, relevant parameters to estimate fuel consumption. One can perceive that if the truck carries a larger load, consumption will increase substantially on steeper roads. Similarly, if the type of vehicle used is not adequate for the carried load, the fuel consumption will not be optimal. Thus, the system must use sensor data, together with the vehicle characteristics, to provide accurate fuel estimations. To be capable of providing fuel consumption estimations before the project is in course, the input system parameters must be available during the project's planning. However, in this work, to validate the hypothesis, we tracked real-time vehicle data to train the machine learning algorithm, which we discuss in detail below to define the requirements to assist sensor selection among several possibilities:

- Road grade: As one may imagine, a heavy vehicle, given its weight, can rapidly develop speed on a descending road without needing the engine and oppositely requiring a considerable amount of mechanical energy to climb ascending roads. Thus, road grade, which can be monitored by tracking the inclination of the vehicle, together with vehicle moving speed or acceleration, is a promising indicator of the engine need. Thus, since vehicle inclination empirically oscillates between 5 Hz and 15 Hz, it is desired to configure a low-pass filter to attenuate road irregularities and roughness at that frequency range either by postprocessing data or configuring built-in sensor filters;
- Vehicle acceleration: As previously mentioned, a vehicle's moving acceleration, together with the road grade and cargo weight, can be a promising indicator of the engine's need (and consequently fuel consumption). Movement acceleration can be

obtained by sampling acceleration data with a low pass filter to exclude vibration, road roughness and irregularities. Additionally, as an experimental project, having higher-frequency data also allows perceiving the engine's rotational speed by carrying through a frequency analysis of the acceleration signal within 13–83 Hz, which corresponds to 800 to 5000 rotations per minute. According to Nyquist's theorem, the signal must be sampled at, at least, twice the frequency of the original signal, thus at 166 Hz. Moreover, to classify road quality and roughness, which have a substantial impact on fuel consumption, a sampling frequency of around 250 Hz [19,20] is required to measure the vehicle's frequency vibration. This way, three *datasets* are defined (vehicle moving speed, road quality/roughness, and motor speed), which can be retrieved by applying three different filters;

- Vehicle global position: Gathering position data allows calculating the total distance traveled, as well as the average speed if the data are timestamped. Distance, together with time, is what the simpler tools often use to estimate fuel consumption, being able to provide a rough estimate of consumption on light vehicles, so one could expect these data to increase the accuracy of the prediction model. Given that the maximum speed allowed on highways in Portugal is 120 km/h and considering the desired 5 m positioning updates, the sampling frequency of this parameter is required to be at least 6.67 Hz. Moreover, with awareness of the road being used together with real-time road quality assessment, one could create a map of roads annotated with the corresponding pavement surface regularity to provide the prediction algorithm with a more accurate and case-specific fuel estimation;
- Cargo weight: As any moving body, the vehicle's weight influences its inertia and momentum, which in turn dictate the amount of mechanical energy the engine is forced to use to increase or maintain the vehicle's speed, thus having a strong influence on fuel consumption. Therefore, monitoring the load weight is very important. However, as the load does not vary continuously, it is only required to weigh the load when the truck is loaded or unloaded, allowing these data to be input by a user when a sensor is not present. Measuring the cargo weight, apart from providing data to the fuel prediction model, is also an opportunity to digitize and automate cargo weighing, which is very useful for operations (e.g., material stocking and productivity evaluation), which, at the time, is still mostly a manual process.

2.2. Data Storage and Communication

Since the accuracy of prediction models is strongly linked to the size of the *dataset* used to train them, it is key that within the training data a vast set of distinct scenarios is covered and these span across, at least, several months. Considering the possibility of bugs or unforeseen situations in the sensor acquisition system, which should be fixed as quickly as possible to avoid loosing data, sensor data must be uploaded to a server that one could access in the lab and easily evaluate the data's integrity. To cope with extended periods without Internet connectivity, typical of the remote nature of infrastructure works, a large local data buffer is required to be in place so the data can be opportunistically uploaded to the server. Furthermore, it is important to have raw sensor data available in the lab as soon as they are available so that the training process of the prediction model can be started and continuously evolve. To this end, raw sensor data were locally stored on the acquisition device's memory, divided into a *dataset* per *run*, which consists of each one-way trip the truck makes in which the load and the consumed fuel are known. Then, when an Internet connection becomes available, the data are uploaded to the cloud, the implementation details of which are further discussed in Section 3.

As concerns cloud connectivity, since the data were only required to be downloaded once per day and the data size could vary based on the remote reconfigurations of the target device resulting from the implementation iterations, a WiFi connection available at the plant was used. This approach allowed dramatically reducing the implementation time to rapidly test our proposal. However in the near future, in order to increase the

scalability of the training phase of the machine learning model, which requires a vast and diversified *dataset*, a sensor-data-optimized communication protocol, for example NB-IoT or LoRaWAN, would be more suitable for this task by reducing the hardware costs and providing real-time features that can expand this system's possibilities.

2.3. Machine Learning

Random Forests (RFs) [21], Artificial Neural Networks (ANNs) [22], and Support Vector Machines (SVMs) [23] are examples of well-known and widely used machine learning algorithms, capable of scrutinizing extensive databases in view of extracting patterns and tendencies in the data, resulting in a deeper understanding of the latter and potentially the generation of new knowledge for the user. Guided by domain knowledge and under a semi-automated process, ML is an iterative and interactive process, in which the extracted knowledge can be used to understand the connections and influence of the independent variables on the dependent variable, ultimately being able to predict the behavior of the latter. The process is often framed in methodologies such as the Cross-Industry Standard Process for Data Mining (CRISP-DM) [24,25], a widely known, tool-neutral methodology that facilitates understanding, implementation, and analysis (Figure 2).

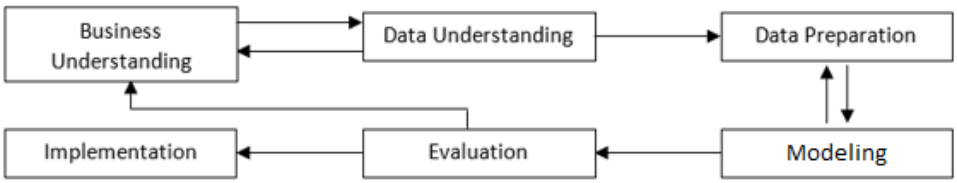


Figure 2. CRISP-DM methodology [24].

The different algorithms can be used for different tasks, such as data classification or regression. Whereas the former focuses on analyzing the behavior of the data in order to classify the target variable into classes or discrete values, the latter aims to predict continuous values. While the predictive evaluation of classification models usually revolves around the quantification of its accuracy, regression model assessment focuses on the calculation of errors and its capability to fit to the data. In this work, since the implemented techniques focus on regression models, their evaluation was primarily based on three main metrics, namely the Mean Absolute Error (*MAE*), depicting the error associated with the degree of learning of a given model, the square root of the average of the squared errors, *RMSE*, which penalizes higher error values, and the correlation coefficient, R^2, comprising the correlation between the observed and the predicted values [26]:

$$MAE = \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)}{N} \quad (1)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}} \quad (2)$$

$$R^2 = \left(\frac{\sum_{i=1}^{N}((y_i - \hat{y}_i) \times (\hat{y}_i - \bar{\hat{y}}_i))}{\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2 \times \sum_{i=1}^{N}(\hat{y} - \bar{\hat{y}})^2}} \right)^2 \quad (3)$$

where y is the computed network output vector, $\hat{y}$ is the target output vector, and N is the number of samples in the database.

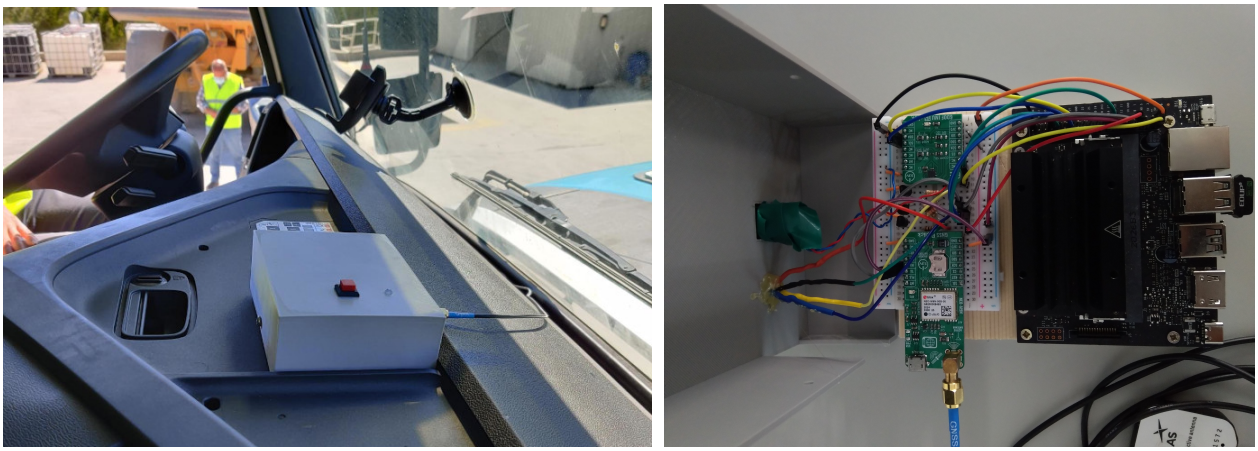
Additionally, the Regression Error Characteristic (REC) curve [27] was also adopted as a measure of the cumulative distribution function of the error of different regression algorithms, allowing for a comparative analysis among the latter.

3. Experimental Project

The purpose of the developed IoT framework is to gather data from sensors and store them for later use as the foundation of the aforementioned ML algorithms. To this end, the first step was to define the sensor acquisition system, as well as the data storage and communications system.

The system was designed with transportation equipment in view. The latter consisted of a truck that transported material from a quarry and an asphalt plant to several different work fronts. This diversity in the truck's routes, as well as in the carried loads adds significant variability to the available *dataset*, which is a relevant contribution towards achieving a better predictive accuracy by the ML algorithms.

The acquisition system was installed in the driver's cabin (see Figure 3a), facilitating the interface by the operator through a simple start/stop button and a status LED. When activated, the system gathers sensor data from IMU and GNSS sensors and records the information locally. When the driver finishes a *run*, the button is pressed to interrupt the gathering process, and the associated data are saved and ready to be sent to the server. Since, most of the time, the truck does not have Internet access, data were stored and later sent to a server when it was in range of the WiFi network present at the quarry near the weight scale. Figure 3b depicts the electronics inside the prototype's housing, which include an Nvidia Jetson Nano 2GB and a breadboard, which has the GNSS and IMU sensors mounted, as well as some discrete electronics to interface with the user button and the RGB LED. All the electronics were powered from the vehicle's lighter port using a general purpose 12 V to 5 V USB adapter.



(**a**) Final installation (**b**) Electronics inside

Figure 3. Sensor acquisition prototype.

3.1. Cyber-Physical Systems

As presented in the previous section, the sampling rate of each sensor was configured to match the information desired while balancing it with the data size. For global positioning, the ublox NEO-M9N sensor was selected and for acceleration and inclination, the STMicroelectronics ASM330LHH.

According to the considerations in Section 2.1, with the exception of the acceleration sensor, the sensors were configured to the nearest available settings as presented in Tables 1 and 2. The acceleration sensor was configured to 1 Hz due to reliability issues in the recording of IMU sensor data, which was only used for data validation purposes and not used for the prediction model. Instead, GNSS data were used to obtain the traveled distance by integrating speed over time, while inclination was obtained from the derivative of the altitude over time. Resorting to the GNSS resulted in decreased accuracy and, due to its data sampling frequency, prevented measuring road roughness.

Table 1. IMU sensor configuration.

Parameter	Value
Accelerometer data rate	1 Hz
Accelerometer, full-scale	2 g
Gyroscope data rate	12.5 Hz
Gyroscope, full-scale	250 dps

Table 2. GNSS sensor configuration.

Parameter	Value
Data rate	10 Hz
GNSS constellations	GPS, Galileo, GLONASS, and Beidou

The Nvidia Jetson Nano 2GB Development Kit was used as the application processor. This embedded computer includes an Nvidia GPU with 128 CUDA cores, which is a good platform to run small-to-medium-complexity computational tasks. The drive behind this choice was two-fold: in the short-run, the processor is able to provide an effective sensor interface; while in the long-run, it may facilitate the processing of data through an edge computing framework (local computation). We used a WiFi USB dongle for Internet connectivity during the experimental project, and we noticed that Nvidia sells this PC under two different part numbers and one of them does not include the WiFi dongle.

Upon selecting the required hardware, a service daemon, GPSd, which is capable of parsing National Marine Electronics Association (NMEA) sentences and controls a multitude of Global Navigation Satellite System (GNSS) receivers, was configured to gather data from the GNSS receiver and expose a TCP API. Thus, it became possible to quickly implement the acquisition of global positioning and clock information. Acceleration and slope data were retrieved using the I2C communication protocol with the configurations listed in Table 1.

Sensor acquisition was implemented using Python, where the use of threads allowed code and execution modularity. By assigning a thread to each task, upon sensor's failure to emit a response or a crash of a task, the other tasks can continue running with small degradation, resulting in a fault-tolerant system. In order to run the code automatically upon the system's boot, *systemd* services were designed, which allowed starting each process in the desired order.

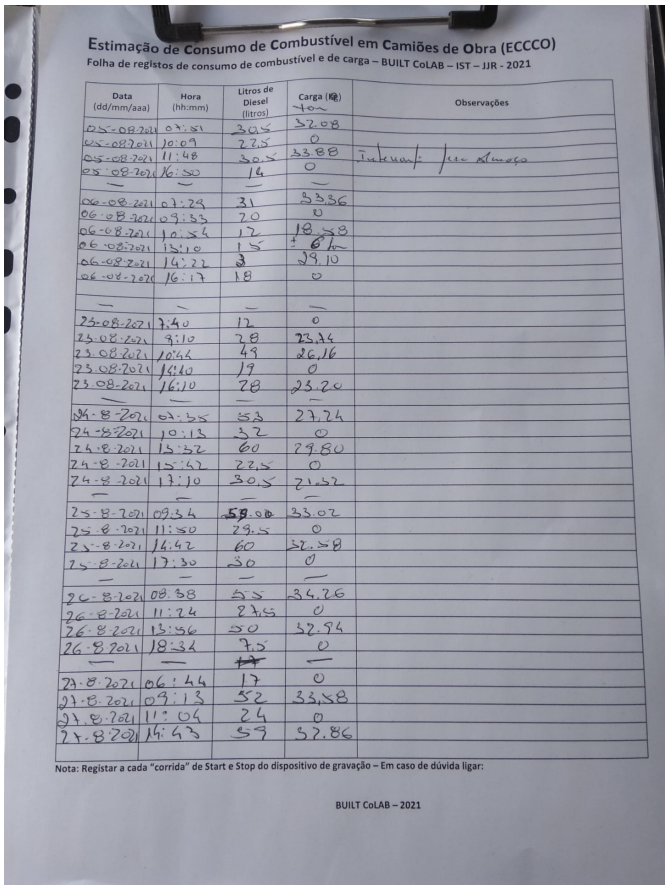
As a parallel experiment, the installation of a distance sensor that could infer the load weight of the truck was considered. Coupling this sensor to the truck's suspension system or equivalent would allow for the measurement of the suspension's contraction. With proper calibration, it would then be possible to infer the real cargo's weight in real time, without the need to weigh the truck itself. However, this method turned out to be unsuccessful since the reliable installation of the sensor proved infeasible on the truck used in this experiment. Yet, from the observations gathered in the field, a wire-wound encoder-based sensor could be an alternative solution, as it was observed that some truck manufacturers have been adopting it in brake systems with a similar purpose.

3.2. Data Storage and Communications

Other than the low-pass filters that the sensors already include (which were configured according to Tables 1 and 2), there was no data processing performed by the acquisition device installed on the truck. This was an intentional choice to avoid further compressing of data and to facilitate the thorough experimentation of the data, since the amount of generated data did not impose high-bandwidth or high-data-volume requirements on the server communication. Data were stored inn Comma-Separated Value (CSV) files and sent to an on-premises server when a preconfigured WiFi network was in range. The process is automatic and uses the Secure Copy Protocol (SCP), a protocol that uses Secure Shell (SSH) as a network protocol to transfer files in which the authentication was configured to

use symmetric key authentication. This authentication method is a strong cybersecurity measure against exterior attacks.

Since the truck's load is already weighed before each trip and fuel consumption could only be obtained by accessing the vehicle's CAN bus, which usually is restricted or only available on modern vehicles, both of these parameters were recorded manually by the driver during the experimental project in which phase a ground truth was needed to train the machine learning algorithms. This information was recorded in a form sheet (see Figure 4) in order to avoid issues with the mobile devices' Internet connectivity and the potential data losses that a digital recording method would incur, if not addressed correctly. Regarding fuel consumption, the precision and reliability of this measurement have a strong impact on the prediction model's performance. Thus, third party systems were avoided, so as to prevent measurement errors or malfunctions. For this reason, the driver accepted the responsibility to start every trip with a full fuel reservoir, filling it back up at the end of each trip, while registering the amount of fuel used in each trip.



Estimação de Consumo de Combustível em Camiões de Obra (ECCCO)

Folha de registos de consumo de combustível e de carga – BUILT CoLAB – IST – JJR - 2021

Data (dd/mm/aaa)	Hora (hh:mm)	Litros de Diesel (litros)	Carga (t)	Observações
05-08-2021	07:51	30.5	32.08	
05-08-2021	10:09	27.5	0	
05-08-2021	11:48	30.5	33.88	Intervalo: Jantar almoço
05-08-2021	16:50	14	0	
06-08-2021	07:29	31	33.56	
06-08-2021	09:33	20	0	
06-08-2021	10:54	17	18.38	
06-08-2021	13:10	15	+ 6 L	
06-08-2021	14:22	3	19.10	
06-08-2021	16:17	18	0	
23-08-2021	7:40	12	0	
23-08-2021	9:10	28	23.74	
23-08-2021	10:44	49	26.16	
23-08-2021	14:40	19	0	
23-08-2021	16:10	28	23.20	
24-8-2021	07:35	53	27.24	
24-8-2021	10:13	32	0	
24-8-2021	13:32	60	29.80	
24-8-2021	15:42	22.5	0	
24-8-2021	17:10	30.5	21.52	
25-8-2021	09:34	58.00	33.02	
25-8-2021	11:50	29.5	0	
25-8-2021	14:42	60	32.58	
25-8-2021	17:30	30	0	
26-8-2021	08:38	55	34.26	
26-8-2021	11:24	27.5	0	
26-8-2021	13:56	50	32.94	
26-8-2021	18:34	7.5	0	
27-8-2021	06:44	17	0	
27-8-2021	09:13	52	33.58	
27-8-2021	11:04	24	0	
27-8-2021	14:43	59	32.86	

Nota: Registar a cada "corrida" de Start e Stop do dispositivo de gravação – Em caso de dúvida ligar:

BUILT CoLAB – 2021

Figure 4. One of the form sheets.

After having implemented the software for the prototype, the system was thoroughly tested in the lab, stressing that the prototype has several vectors. However, to further prevent any possible issues, an Over-The-Air (OTA) mechanism was developed to allow remote software reprogramming of the acquisition device. This mechanism relies on the least possible amount of subsystems and is subsequently able to work even when the majority of subsystems are failing. Since the code is hosted on a GitHub repository, the system periodically pulls the repository and, upon changes to a specific git branch, downloads the latest version, updates itself (including the updater script), and restarts the application. This process is completely seamless for the operator since this is only allowed to occur when the device is idle. The unavailability of the system during the update, which is usually short (within a second), does not the compromise usability because, during load

weighing, the vehicle has a strong WiFi signal for which it needs to remain still for around half a minute.

3.3. Experimental Setup

Following the stabilization of the hardware of the prototype, the latter was mounted in a specific box produced in a 3D printer. Because it was installed in the vehicle's cabin, the housing was designed to protect the embedded computer, as well as the sensors included from damage from other objects. This placement (see the previous Figure 3a) allowed the operator to interact with the prototype by pressing a button to signal the beginning and end of each *run* and check the system status via an RGB LED. With the prototype in the cabin, special attention was given to the GNSS receiver's antenna, which was routed to the exterior of the vehicle and attached magnetically to the outer side of the vehicle (see Figure 5) to improve reception.

(**a**) Overview (**b**) GNSS antenna installation

Figure 5. Volvo truck used in the field experiments.

After installing the prototype on the vehicle, a dry run was performed to ensure the instructions gave to the driver were clear and to establish a brief ground truth for early data validation. At the same time, the complete data path was tested by carrying out a full trip and accessing the server to check the presence of a CSV file. At this point, the acceleration sensor was only capable of being read at a maximum frequency of 1 Hz. This issue was tracked back to a bottleneck related to the I2C communication protocol, as well as a bug in the implementation of the data synchronization mechanism. The latter affected the GNSS receiver and gyroscope data as well, although with smaller effective data loss, as these were able to maintain the average sample rates of 10Hz for the GNSS sensor data and 12.5 Hz for the gyroscope. Nevertheless, these sample rate values were deemed acceptable for the goals of the experiment at issue at this stage.

4. Results and Discussion

4.1. Raw Data and Data Preparation

As previously mentioned, data are the foundation of ML models. Yet, the fact that these predictive algorithms depend so strongly on the availability of data simultaneously represents their greatest potential and their greatest limitation. Indeed, the predictive capability of an ML model depends both on the amount and the quality of the available data. In other words, to achieve a perfect prediction of the behavior of a target variable, the model must not only be fed enough data, but these data must also encompass all the independent variables that may have some level of influence on said behavior. Moreover,

the variability of the data must be consistent enough to allow the ML algorithms to gather knowledge and insight on how much each independent variable affects the target variable, as well as how the independent variables connect to and affect each other.

In this context, one can easily find several preprocessed *datasets* associated with different fields, which can result in near-perfect models (i.e., with prediction accuracy close to 100% in the case of classification models or very low error margins in the case of regression models), depending on the adopted ML technique, as is the case of the well-known VGG-16 [28] or the ResNet50 [29] databases in the image classification field. However, this level of accuracy/error is not always possible when dealing with real-world data, since the latter tend to be noisy and afflicted by higher levels of unreliability or missing data. While there are a number of actions that can be taken to deal with these issues during the data cleaning and processing phase (which will be discussed in the ensuing subsection), typical real-world-based models can only very rarely achieve perfect fits to the data, and subsequently perfect predictive capabilities.

Bearing these considerations in mind, the data used in this work stemmed from a real-world setting consisting of a road construction site in Portugal. The raw data concerned the activities of the sensorized construction material transportation truck, featuring around 25 trips through different routes and pavement surfaces (i.e., construction site, national road, and highway), while transporting different materials, among which the most common was a bituminous mix from an asphalt plant to a road construction site. Besides cargo and pavement surfaces, the trips also present some variation regarding total distance, ranging from 20–70 km. The main measured parameters included time, location/GPS data, altitude, speed, and the three-axis inclination of the truck, as exemplified in Table 3.

Table 3. Example of values extracted from the raw database.

Inclination X (Degrees)	Inclination Y (Degrees)	Inclination Z (Degrees)	Latitude (Degrees)	Longitude (Degrees)	Altitude (m)	Speed (m/s)	Clock (yyyy-MM-ddTHH:mm:ssZ in UTC)
0.778198	−0.29755	−1.31226	39.4447	−7.47812	447	0.043	2021-07-22T12:42:36.100Z
0.778198	−0.29755	−1.31226	39.4447	−7.47812	447	0.038	2021-07-22T12:42:36.400Z
0.839233	−0.30518	−0.03052	39.4447	−7.47812	447	0.038	2021-07-22T12:42:36.700Z
0.839233	−0.30518	−0.03052	39.4447	−7.47812	447	0.013	2021-07-22T12:42:36.800Z
0.839233	−0.30518	−0.03052	39.4447	−7.47812	447	0.014	2021-07-22T12:42:37.100Z
0.839233	−0.30518	−0.03052	39.4447	−7.47812	447	0.044	2021-07-22T12:42:37.500Z
0.923157	−0.28992	−1.95313	39.4447	−7.47812	447	0.004	2021-07-22T12:42:37.900Z
0.923157	−0.28992	−1.95313	39.4447	−7.47812	447	0.036	2021-07-22T12:42:38.200Z

As one can easily infer, the refresh rate associated with the IoT framework collects data several times per second, ultimately generating very large databases in the form of CSV files, each one corresponding to one trip of the truck. Since, as mentioned, these trips can be as long as nearly 70 km, the associated data files also increase proportionally to around 12,500 entries in those cases. Figure 6 depicts one of the most typical altimetric profiles through which the truck traveled. These were extrapolated by integrating the speed data over time and validated by resorting to the inclinometer data. From this point on, a 10 m sliding window methodology was adopted to adjust the fit lines throughout the altimetric profile, allowing for the determination of the slope of each 10 m section. Through this methodology, each trip was translated into the accumulated distance that the truck spent in each type of slope, according to the considerations described in Table 4:

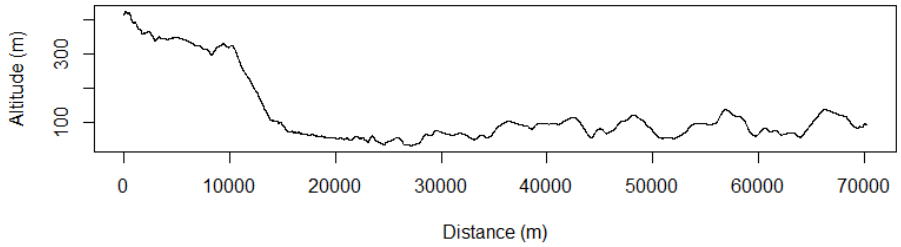


Figure 6. Example of the altimetric profile of a trip as measured by the sensorized truck.

Table 4. Considered slope ranges and description.

Slope Description	Range	Feature Designation
Flat surface	$-1\% < \text{Slope} \leq +1\%$	AD_0.01n_0.01
Light upwards slope	$+1\% < \text{Slope} \leq +5\%$	AD_0.01_0.05
Moderate upwards slope	$+5\% < \text{Slope} \leq +10\%$	AD_0.05_0.1
Steep upwards slope	$\text{Slope} \geq +10\%$	AD_0.1
Light downwards slope	$-5\% < \text{Slope} \leq -1\%$	AD_0.01_0.05n
Moderate and steep downwards slope	$\text{Slope} \leq -5\%$	AD_0.05n

Moderate and steep downwards slopes were grouped since we obtained more accurate prediction results while doing this, which seems reasonable given the fact that trucks can rapidly develop speed without any throttle, resulting in no fuel consumption on any of these road slopes.

Ultimately, this conversion consisted of determining the percentage of the total route distance that the truck spent on each type of inclination throughout each trip, resulting in the bulk of the training and testing database. The latter were complemented with the data on total trip distance (TDistance, meters), average speed (AvSp, meters per second), cargo weight (Cargo, tons), and fuel consumption (FConsumption, liters, target variable). The latter two features were manually inserted into the database, as they originated from the manual records taken by the truck driver in between trips and during each refueling action. Table 5 depicts the processed database that supported the final machine learning algorithms.

Table 5. Example of the values extracted from the training and testing database.

AD_0.01n_0.01 (% TDistance)	AD_0.01_0.05 (% TDistance)	AD_0.05_0.1 (% TDistance)	AD_0.1 (% TDistance)	AD_0.01_0.05n (% TDistance)	AD_0.05n (% TDistance)	TDistance (m)	AvSp (m/s)	Cargo (ton)	FConsumption (L)
0.32	0.25	0.01	0	0.36	0.06	66,341.91	18.48	29.48	30
0.4	0.33	0.05	0	0.21	0.02	65,704.09	20.52	0	23
0.2	0.33	0.13	0.01	0.23	0.11	52,992.28	12.39	0	20.5
0.2	0.25	0.08	0.01	0.28	0.17	35,887.31	9.15	33	25.5
0.19	0.25	0.1	0.01	0.28	0.17	36,563.17	8.52	32.68	25
0.37	0.23	0.01	0	0.35	0.05	62,786.77	17.4	33.76	26.5
0.43	0.3	0.05	0	0.21	0.01	62,282.68	20.79	0	23.5
0.38	0.22	0.01	0	0.33	0.05	62,786.77	18.91	32.46	26
0.41	0.31	0.05	0	0.22	0.01	62,606.6	19.16	0	23

4.2. Estimation Performance

The results were obtained by resorting to the package rminer [30] for R [31]. Using this tool, several models were trained and tested on the processed database, namely Random Forests (RFs), Artificial Neural Networks (ANNs), and Support Vector Machines (SVMs). In addition, a simple statistical model in the form of a Multiple Regression (MR) was also provided for comparison purposes. Since the generalization capacity is a key concern for future implementation, as well as for the assessment of the models, a 10-run cross-validation approach was adopted. A k-fold value of three was used to account for the

relatively small size of the data. This means that the data were evaluated across the entire training set by dividing the latter into three folds. The model was then trained three times while reserving a different fold as a testing *dataset* each time, thus using the available data to their full potential [26]. Figure 7 shows a comparison between the REC curves of the three models. The analysis of this figure suggests that the random forests seemed to slightly outperform the other two models, which is consistent with the findings obtained in the literature concerning the prediction of heavy vehicle fuel consumption [17].

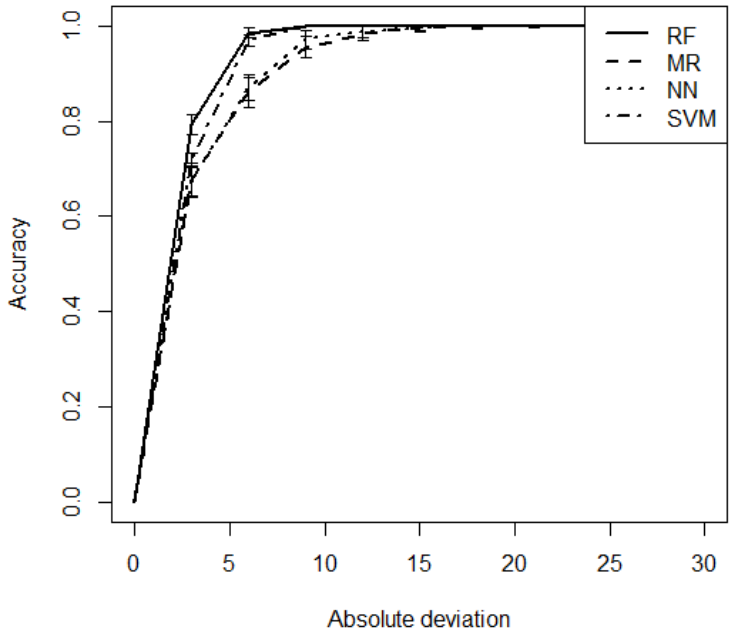


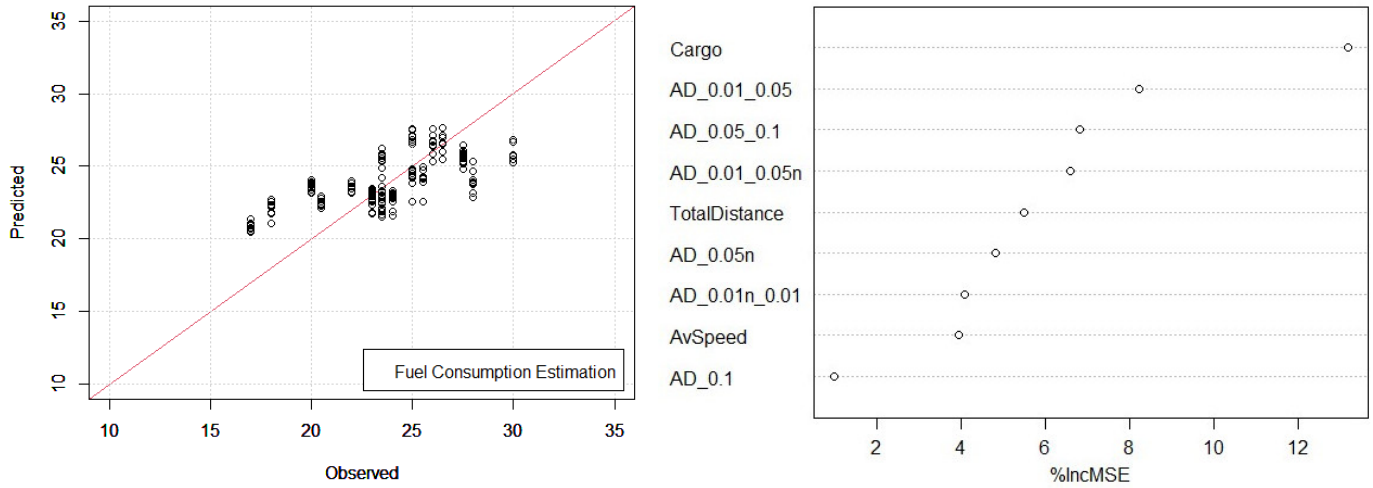
Figure 7. REC curve comparison of different models.

In fact, the RF model exhibited a slightly lower error, with an MAE of 1.723915 (corresponding to a percentual error of around 7.6%), an RMSE of 2.127431, and a slightly better R2 fit to the data, with a value of 61.4%. Although this value is expected to improve as more data become available, it was still deemed a reasonable value when taking into account that the model was based entirely on real-world data. This hypothesis is supported by the value of the Pearson's correlation coefficient, R, which was at 78.4% for the RF model, denoting a very good interdependence of the independent variables and the target variable.

Figure 8a shows how the predicted values fit the observed values in the testing *dataset* for all 10 runs, each corresponding to a k-fold training and testing methodology with a k value of three. As one can easily infer, the closer the points are to the diagonal line, the better the fit and, subsequently, the better the value of R2. Analyzing the figure reveals that the model seemed to be able to reproduce the behavior of the target variable especially well in the values in the middle range (i.e., fuel consumption values from 20–27 L), though the values closer to the lower and upper extremity were slightly overestimated and underestimated, respectively. As can be expected, this was due to the fact that there was a much lower amount of records in the database that fell within these extremity ranges, which is also expected to improve with the expansion of the database over time.

Another noteworthy aspect for analysis is the relative importance of the variables for the RF model, depicted in Figure 8b. The figure conveys the Increase in the Mean Squared Error (%IncMSE) as a result of the corresponding variable being permuted. In other words, the higher the value of %IncMSE, the more important the corresponding variable for the predictive capability of the model is. In this context, it is interesting to observe that the weight of the cargo transported by the truck is considered by the model as its most important variable, followed by the predominance of light and moderate upwards slopes, as well as moderate downward slopes. Intuitively, it is easy to infer how these aspects are

valid, as the higher the weight and the upward slopes, the higher the consumption of the fuel, while downward slopes imply a nearly null value of fuel consumption. Conversely, the variable corresponding to steep upwards slopes is considered the least relevant for the model. This makes sense, as there are very few sections throughout the routes of the truck that display this kind of inclination, and as such, the model identifies the associated variable as having very low importance. However, the removal of this variable from the training *dataset* still had a negative impact on the predictive capability of the final model, even if to a low degree, and as such, it was kept in the training *dataset* at this stage.



(**a**) Observed vs. predicted values for the RF model (**b**) Feature relative importance

Figure 8. Prediction model results.

5. Conclusions and Future Work

This work proposed a scenario-based fuel consumption prediction model that can be used within a tool for project planning and budget analysis. The novelty of the project is the integration of an IoT framework for data gathering and transmission into the database that comprises the training and testing data for the predictive models. The results showed that fuel consumption has a strong correlation with cargo, route inclination, and total distance, thus proving to be key input parameters to achieve accurate and reliable fuel consumption predictions. These results are particularly interesting for engineering planners and designers, since this information is easily accessible to them from existing GIS systems (e.g., route inclination and total distance) and the construction project plans or BIM models themselves (e.g., cargo).

Although the output machine learning models obviously lack the necessary amount of data at this stage to be considered generalizable, and thus to be implemented in practice, the project's premise has potential, and the results show promise. Above all, the methodology is a relevant contribution to the state of knowledge in the sense that it provides an initial step towards a real-world implementation of a digital twin, as well as of a self-learning machine learning system in an Internet of Things framework, thus following the current trends in automation, digitalization, and Industry/Construction 4.0.

One of the limitations of the current model is that the analyst is required to estimate average speed over the entire route, which can comprise a significant obstacle. However, this issue can potentially be mitigated by the introduction of the data streaming from the accelerometers. As a matter of fact, leveraging the vertical axis of the accelerometers to infer a rough classification of each type of surface through which the truck circulates (e.g., compacted dirt road, regular road, highway) can provide insight into the behavior of the truck in different environments (e.g., average speed, average number of full stops, traffic conditions, among others). Subsequently, this type of information may even be valuable enough to the model for it to eventually even replace the need for the user to estimate the speed, who instead may only

have to estimate the percentage of each type of surface in relation to the trip's total distance, similar to the road inclination features already present in the model.

In addition to this, other future work directions should naturally include expanding the study to encompass a higher amount of vehicles, routes, and carried loads, so as to produce a robust and generalizable prediction model. Then, as previously mentioned, one of the outputs of the project will be translated into the development of a web API, which will be made available online to support decision-making or any third-party software tools that may benefit from an accurate and parametric fuel estimation. Furthermore, the achieved results motivate the development of a real-time sensing acquisition system capable of dealing with the current sensor sampling frequency bottlenecks, thus supporting the continuous and automatic training and testing process of the prediction models, ultimately improving their accuracy and reliability by increasing the amount of information retrieved from the sensors. Concurrently, this development should be accompanied by a more robust data-processing workflow, which should be capable of automatically addressing common issues found in real-world data, such as missing or partial data. This would be a relevant step to achieve a truly automatic, self-learning, and self-feeding prediction system, capable of gathering data from several simultaneous heavy machines working at different work fronts and sites, processing it as additions to the previous database, and automatically updating the predictive models to constantly improve their effectiveness, robustness, and efficiency, as they constantly learn and accumulate experience from ongoing construction sites.

Author Contributions: G.P.: IoT hardware, software development and communication system, validation, formal analysis, investigation, and writing–original draft preparation. M.P.: machine learning, conceptualization, investigation, methodology, validation, writing—original draft preparation, supervision, and formal analysis. J.M.: IoT architectures and communication systems, investigation, conceptualization, methodology, validation, resources, writing—original draft preparation, writing—review and editing, visualization, and supervision. M.S.: IoT hardware and software development and writing—original draft preparation. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by European Structural and Investment Funds in the FEDER component, through the Operational Competitiveness and Internationalization Programme (COMPETE 2020)-Highly Qualified Human Resources programmes: NORTE-06-3559-FSE-000176 and LISBOA-05-3559-FSE-000014.

Acknowledgments: We would like to acknowledge the support and cooperation of the JJR Construction Company, which has cooperated with the project and provided a truck and a driver to conduct the experimental procedure, supported hardware installation, and allowed the several data acquisitions for the experiments.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ANN	Artificial Neural network
API	Application User Interface
CAN	Controller Area Network
CSV	Comma-Separated Values
GNSS	Global Navigation Satellite System
ML	Machine Learning
MR	Multiple Regression
OBD	On-Board Diagnostic
OCR	Optical Character Recognition
OTA	Over-The-Air
RF	Random Forest
RPM	Rotations Per Minute

SCP	Secure Copy Protocol
SSH	Secure Shell
SVM	Support Vector Machine

References

1. Caterpillar. *Caterpillar Performance Handbook*; Caterpillar: Peoria, IL, USA, 2018.
2. Correia, A.; Cortez, P.; Tinoco, J.; Marques, R. Artificial Intelligence Applications in Transportation Geotechnics. *Geotech. Geol. Eng.* **2013**, *31*, 861–879. [CrossRef]
3. Parente, M.; Correia, A.; Cortez, P. Use of DM techniques in earthworks management: A case study. In Proceedings of the Geo-Hubei 2014 International Conference on Sustainable Civil Infrastructure, Yichang, China, 20–22 July 2014. [CrossRef]
4. Hola, B.; Schabowicz, K. Estimation of earthworks execution time cost by means of artificial neural networks. *Autom. Constr.* **2010**, *19*, 570–579. [CrossRef]
5. Tinoco, J.; Gomes Correia, A.; Cortez, P.; Toll, D.G. Stability condition identification of rock and soil cutting slopes based on soft computing. *J. Comput. Civ. Eng.* **2018**, *32*, 04017088. [CrossRef]
6. Tinoco, J.; Correia, A.; Cortez, P. Application of data mining techniques in the estimation of the uniaxial compressive strength of jet grouting columns over time. *Constr. Build. Mater.* **2011**, *25*, 1257–1262. [CrossRef]
7. Ma, X.; Dong, Z.; Chen, F.; Xiang, H.; Cao, C.; Sun, J. Airport asphalt pavement health monitoring system for mechanical model updating and distress evaluation under realistic random aircraft loads. *Constr. Build. Mater.* **2019**, *226*, 227–237. [CrossRef]
8. Zhao, H.; Wu, D.; Zeng, M.; Tian, Y.; Ling, J. Assessment of concrete pavement support conditions using distributed optical vibration sensing fiber and a neural network. *Constr. Build. Mater.* **2019**, *216*, 214–226. [CrossRef]
9. Marques, R.; Correia, A.G.; Cortez, P. Data mining applied to compaction of geomaterials. In Proceedings of the 8th International Conference on the Bearing Capacity of Roads, Railways and Airfields, Champaign, IL, USA, 29 June–2 July 2009; pp. 597–605.
10. Edwards, D.; Griffiths, I. Artificial intelligence approach to calculation of hydraulic excavator cycle time and output. *Min. Technol. Trans. Institutions Min. Metall.* **2000**, *109*, 23–29. [CrossRef]
11. Tam, C.; Tong, T.K.L.; Tse, S.L. Artificial neural networks model for predicting excavator productivity. *Eng. Constr. Archit. Manag.* **2002**, *9*, 446–452. [CrossRef]
12. Parente, M.; Correia, A.G.; Cortez, P. Metaheuristics, data mining and geographic information systems for earthworks equipment allocation. *Procedia Eng.* **2016**, *143*, 506–513. [CrossRef]
13. Alves de Souza, V.; Giusti, R.; Batista, A. Asfault: A low-cost system to evaluate pavement conditions in real-time using smartphones and machine learning. *Pervasive Mob. Comput.* **2018**, *51*, 27–42. [CrossRef]
14. Nunes, D.; Mota, V. A participatory sensing framework to classify road surface quality. *J. Internet Serv. Appl.* **2019**, *10*, 1–16. [CrossRef]
15. Majidifard, H.; Adu-Gyamfi, Y.; Buttlar, W.G. Deep machine learning approach to develop a new asphalt pavement condition index. *Constr. Build. Mater.* **2020**, *247*, 118513. [CrossRef]
16. Svärd, C. *Predictive Modelling of Fuel Consumption Using Machine Learning Techniques*; Technical Report; Scania CV AB: Stockholm, Sweden, 2014.
17. Perrotta, F.; Parry, T.; Neves, L.C. Application of machine learning for fuel consumption modelling of trucks. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 3810–3815. [CrossRef]
18. Schoen, A.; Byerly, A.; Hendrix, B.; Bagwe, R.; Santos, E.; Ben-Miled, Z. A Machine Learning Model for Average Fuel Consumption in Heavy Vehicles. *IEEE Trans. Veh. Technol.* **2019**, *68*, 6343–6351. [CrossRef]
19. Opara, K.R.; Brzezinski, K.; Bukowicki, M.; Kaczmarek-Majer, K. Road roughness estimation through smartphone-measured acceleration. *IEEE Intell. Transp. Syst. Mag.* **2021**, in press. [CrossRef]
20. Sattar, S.; Li, S.; Chapman, M. Road Surface Monitoring Using Smartphone Sensors: A Review. *Sensors* **2018**, *18*, 3845. [CrossRef] [PubMed]
21. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
22. McCulloch, W.S.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **1943**, *5*, 115–133. [CrossRef]
23. Boser, B.E.; Guyon, I.M.; Vapnik, V.N. A training algorithm for optimal margin classifiers. In Proceedings of the 5th Annual Workshop on Computational Learning Theory, Pittsburgh, PA, USA, 27–29 July 1992; pp. 144–152.
24. Chapman, P.; Clinton, J.; Kerber, R.; Khabaza, T.; Reinartz, T.; Shearer, C.; Wirth, R. CRISP-DM 1.0: Step-by-Step Data Mining Guide. The CRISP-DM consortium. *SPSS INC* **2020**, *9*, 13.
25. Fayyad, U.; Piatetsky-Shapiro, G.; Smyth, P. From data mining to knowledge discovery in databases. *AI Mag.* **1996**, *17*, 37.
26. Friedman, J.; Hastie, T.; Tibshirani, R. *The Elements of Statistical Learning*; Springer Series in Statistics; Springer: New York, NY, USA, 2001; Volume 1.
27. Bi, J.; Bennett, K.P. Regression error characteristic curves. In Proceedings of the 20th International Conference on Machine Learning (ICML-03), Washington, DC, USA, 21–24 August 2003; pp. 43–50.
28. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
29. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

30. Cortez, P. Data mining with neural networks and support vector machines using the R/rminer tool. In Proceedings of the Industrial Conference on Data Mining, Berlin, Germany, 12–14 July 2010; pp. 572–583.
31. R Core Team. *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, Austria, 2020. Available online: https://www.R-project.org/ (accessed on 8 February 2021).