IoT

Article

Predictive Maintenance of Bus Fleet by Intelligent Smart Electronic Board Implementing Artificial Intelligence

Alessandro Massaro *, Sergio Selicato and Angelo Galiano

Dyrecta Lab srl, Research Institute, Via Vescovo Simplicio 45, 70014 Conversano, Italy;
sergio.selicato@dyrecta.com (S.S.); maurizio.galiano@dyrecta.com (A.G.)
* Correspondence: alessandro.massaro@dyrecta.com; Tel.: +39-080-495-8477

Received: 9 September 2020; Accepted: 29 September 2020; Published: 1 October 2020

Abstract: This paper is focused on the design and development of a smart and compact electronic control unit (ECU) for the monitoring of a bus fleet. The ECU system is able to extract all vehicle data by the on-board diagnostics-(ODB)-II and SAE J1939 standards. The integrated system Internet of Things (IoT) system, is interconnected in the cloud by an artificial intelligence engine implementing multilayer perceptron artificial neural network (MLP-ANN) and is able to predict maintenance of each vehicle by classifying the driver behavior. The key performance indicator (KPI) of the driver behavior has been estimated by data mining k-means algorithm. The MLP-ANN model has been tested by means of a dataset found in literature by allowing the correct choice of the calculus parameters. A low means square error (MSE) of the order of 10^{-3} is checked thus proving the correct use of MLP-ANN. Based on the analysis of the results, are defined methodologies of key performance indicators (KPIs), correlating driver behavior with the engine stress defining the bus maintenance plan criteria. All the results are joined into a cloud platform showing fleet efficiency dashboards. The proposed topic has been developed within the framework of an industry research project collaborating with a company managing bus fleet.

Keywords: ECU; predictive maintenance; artificial neural networks; driver behavior

1. Introduction

The controller area network (CAN) bus and on-board diagnostics (ODB) communication interfaces II are standards typically used to extract information from a vehicle [1–3], to control the vehicle conditions, and to deduce any anomalies by accessing the electronic control unit (ECU). The CAN system is characterized by a relatively low cost per node when compared with other information systems in the automotive bus systems [4]. The CAN standard can be integrated with mobile information systems [5] and data mining algorithms such as artificial neural networks (ANN) [6]. In [7], some researchers have provided important indications about the procedures being performed for the diagnosis and prognosis of the vehicle starting from the analysis of the ECU data, thus suggesting adopting vehicle data for maintenance plan. In [8], the integrability of the sensors in a complete diagnostic network has been demonstrated, thus suggesting ideating a smart and compact unit adaptable to different types of vehicles by using specific connectors. In this scenario, ODB II connector has been adopted in [9,10] to acquire data for the fuel consumption trend, which could therefore indirectly provide indications on the analysis of drivers behavior and in a certain way is able to predict the vehicle wear (the fuel consumption is a parameter that is a function of the acceleration and of the vehicle speed). The analysis, mainly derived from the speed and consumption indications of a bus, can also be carried out by analyzing the data acquired by global positioning systems (GPS) [11–19], able to track in real time the vehicle route and to estimate vehicle speed. GPS technology can be integrated with global system for mobile (GSM) [11,12], providing additional support for data transmission, for anti-theft systems [13], and for signaling emergency situations or engine failures [14]. GPS tracking is strategic also for logistics [16]. Machine learning (ML) algorithms are good candidates to support data analysis, especially for engine failures prediction [20]. On the other side, data mining could provide decision-making in fleet management, estimating economical maintenance by using k-means algorithm [21]. Other unsupervised algorithms have been adopted for predictive maintenance in the automotive sector [22]. Moreover, bus surveillance systems by camera could provide further information about violators and traffic [23], thus supporting drivers. Recently, the use of Internet of Things (IoT) and microcontroller technologies enabled automatic systems predicting fleet health and maintenance [24]. A full architecture able to implement all the facilities fleet management, including the use of external sensors and microcontrollers, has been analyzed in [25]. Different frameworks have been proposed for decision-making processes in bus fleet maintenance [26]. Concerning driver behavior, clustering techniques are suitable to classify characteristics such as safe and ecological driving [27]. Driver behavior features can also be extracted by deep learning (DL) approaches analyzing road complexity [28]. Following the state-of-the-art, are formulated the main specifications of an industry project, concerning the development of a software and hardware platform oriented on predictive maintenance and driver behavior estimation. Specifically, the research project concerns the design and development of an engineered system for the acquisition of bus fleet data and for the management of their maintenance, using predictive analysis. The project also provides the monitoring of each individual bus vehicle by means of the GPS system and the integration of surveillance system. Specifically, the main specifications of project include:

- the use of data acquisition interfaces (electronic interfaces) implementing on-board diagnostics II (OBD-II) communication standard diagnostics II (OBD-II) communication standard (the data are extracted from the control units by means of scheduled procedures);
- the use of a central database (MySQL) for the collection of data, which are processed by data mining and artificial intelligence algorithms (e.g., clustering, artificial neural networks), supporting the formulation of the fleet maintenance plan, based on wear prediction deriving from the analysis of vehicle data such as revolutions per minute (RPM), accelerations (throttle position), stops, refueling, fuel consumed, inconsistencies between loaded values and actually consumed volumes, etc.;
- the creation of dashboards indicating the wear levels of the single bus, and the predictive scheduling of the maintenance plan based on the outputs of the artificial intelligence algorithms;
- a camera module: two night vision cameras are mounted on each vehicle (front and rear view) for video streaming and for recording. The cameras, in addition to the linking of the GPS signal, can be also used to verify the driving style of the driver, and to analyze the status of the itineraries that could influence the vehicle wear for a long time period (intense traffic conditions, roads with potholes, etc.);
- a GPS monitoring module allowing the tracking of all the movements and activities of each vehicle and to monitor any inefficiencies (for example excessive consumption due to an inappropriate driving style, risky driving styles due to speed limits not respected, etc.). GPS data can be processed by the data mining engine for the definition of the driver's reliability and efficiency indices, for the mapping of the activities of each individual vehicle, and for the support in predicting maintenance procedures.

Figure 1 illustrates the electronic architecture related to the data on acquisition on board: the ECU is connected to the ODB II port transferring data to a raspberry Pi board; the boards trough an internet key, transmit data and images to the cloud. Data and images are then processed by a server where an artificial intelligence engine is implemented. The artificial intelligence algorithms provide as output the driver key performance indicators (KPIs) and predictive maintenance procedures (see Figure 2).

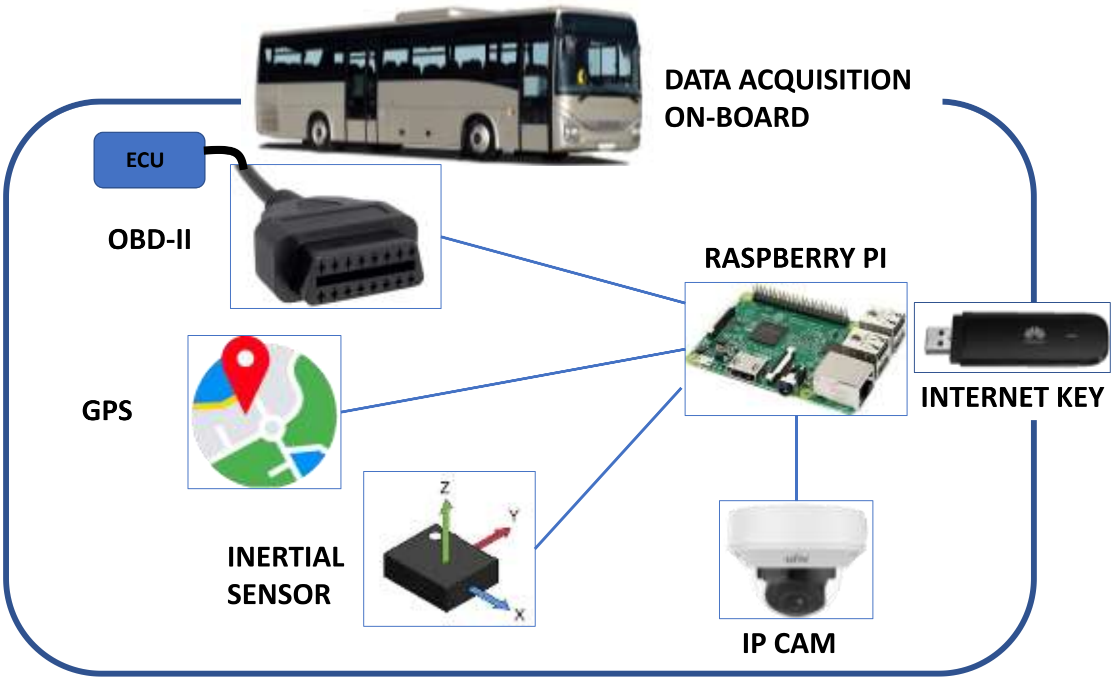


Figure 1. Architecture of the data acquisition on board system.

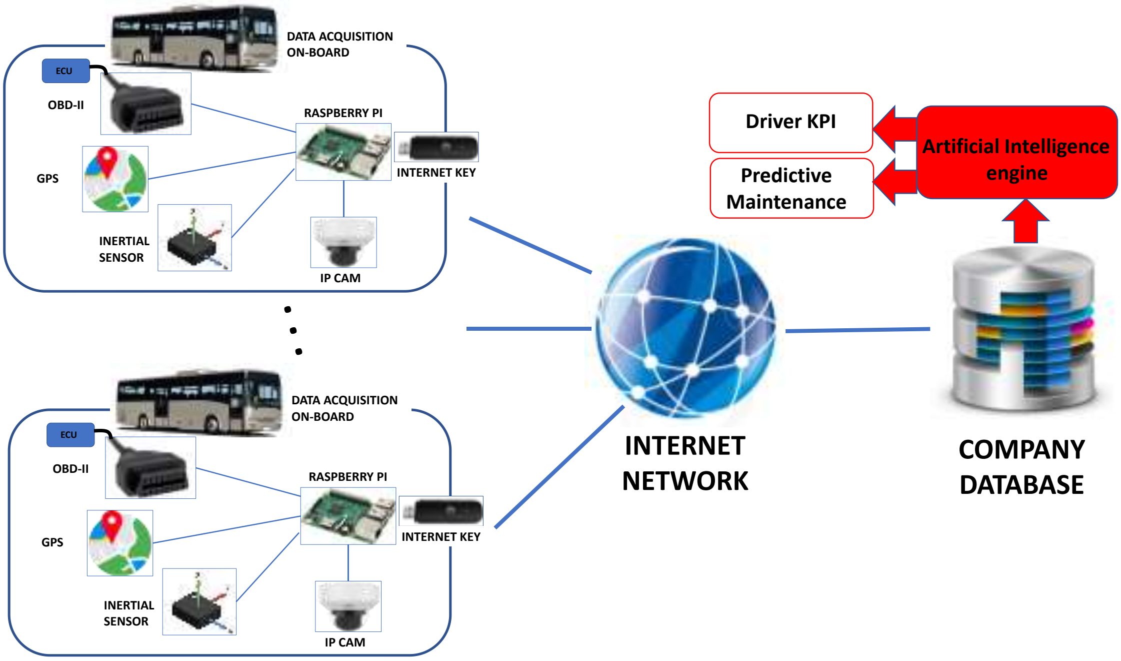


Figure 2. Platform architecture enabling driver key performance indicators (KPI) and predictive maintenance by artificial intelligence.

The software design (artificial intelligence engine) integrates the following modules:

- Multilayer perceptron (MLP) artificial neural network (ANN) model providing prediction about vehicle wear;
- k-means algorithm able to provide driver clusters indicating the correct and inappropriate behaviors.

The outputs of both the algorithms are combined to update the predictive maintenance procedure. The paper is structured by the following discussions:

- Bus protocol description;

- Software design and data modelling;
- Preliminary testing of electronic components;
- Testing of the artificial neural network (estimation of the calculus error);
- Testing of the MLP-ANN algorithm;
- Testing of the k-means algorithm;
- Key performance indicator (KPI) evaluation of drivers and vehicles;
- Correlation analysis of main vehicle parameters;
- Dashboards of the implemented platform.

2. Methodological Approaches and Experiments

The ODB II communication standard is applied to the BUS Iveco Crossway. The used standards for bus communication are SAE J1962 (HW connector) and SAE J1939 (PGN) standards, useful for the design and development of the IOT system recovering the parameters from the ECU. The OBD port has been designed to communicate with different transmission protocols, and therefore with different ECU models, which over time have been replaced with CAN. The use of OBD-II technology allows direct access to the data of the engine control unit (ECU), by means of the SAE J1939 standard.

The type of frame used for the SAE J1939 standard is the extended one, which provides the 29-bit identification (ID) field given by the sum of two sub-fields:

- 11-bit ID A
- 18-bit ID B

The maximum payload size is 8 bytes. The protocol is shown in Table 1

Table 1. Frame protocol adopted for data transmission (SAE J1939).

SOF	ID A	SRR	IDE	ID B	RTR	Res	DLC	DF	CRC	Del CRC	ACK	Del ACK	EOF
1 bit	11 bit	1 bit	1 bit	18 bit	1 bit	2 bit	4 bit	8 byte	15 bit	1 bit	1 bit	1 bit	7 bit

The following fields are identified:

- Start of frame (SOF) (1bit): indicates the beginning of the transmission sequence;
- Identification A-ID A (11bit): first part of the identifier inherent to the recipient and sender component;
- Substitute remote request (SRR) (1 bit): it is a recessive bit;
- Additional identification bit (IDE) (1 bit): it is a recessive bit;
- Identification B-ID B (18bit): second part of the identifier inherent to the recipient and sender component;
- Remote transmission request (RTR) (1 bit): it is a dominant bit;
- Reserved (Res) (1 bit): reserved according to the type of standard used;
- Data length code (DLC) (4 bits): indicates the number of bytes of the frame payload;
- Data field (DF) (0–8 bytes): represents the payload of the frame whose length is indicated by the DLC field;
- Redundancy parity check (CRC) (15 bit): field to verify that the frame during transmission has not been corrupted;
- CRC delimiter (Del CRC) (1 bit): it is a recessive bit;
- Receipt confirmation (ACK) (1 bit): the recessive bit indicates receipt;
- ACK delimiter (Del ACK) (1 bit): it is a recessive bit;
- End of Frame (EOF) (7 bits): they are recessive bits.

Through the SAE J1939 standard, the messages are identified as "parameter group" or PG (group of parameters) corresponding to a set of quantities belonging to the same topic or subsystem. To each PG is assigned a "parameter group number", or PGN ("Parameter Group Number"), which identifies the same dataset of information.

Two types of PGN are possible:

- PG global PGN: it identifies a group of PG parameters, which are sent to all devices or broadcast. Here the Protocol Data Unit (PDU) format, PDU specific, data page and extended data page are used for the identification of the corresponding PG. Global PGNs occur when the PDU format value is greater than or equal to 240. In fact, the PDU specific corresponds to the group extension. The format of the PDU used for this data is the second.
- Specific PGN: are parameter group PG transmitted to particular devices (peer-to-peer). Here the PDU format, data page and the extended page are used for the identification of the corresponding PG. As for the PDU Format, it assumes a value less than 240 and the specific PDU is set to zero. The format of the PDU used for this data is therefore the first.

More details about SAE J1939 protocol are provided in Appendix A.

Predictive maintenance algorithm has been implemented by multilayer perceptron neural networks (MLP) implemented by Konstanz Miner (KNIME) open source tool based on the use of graphical user interfaces (GUIs) as blocks enable data processing [29,30]. The MLP is a feed-forward artificial neural network (ANN) model that maps sets of input vehicle data by providing output wear prediction. The MLP network is constituted by multiple nodes linked in different layers. Each node behaves as a calculus cell named neuron able to process data by means of a properly defined activation function. The MLP approach implements the backpropagation training computing the gradient of the loss function with respect to the weights of the network for a single input–output sample.

The approach used for the predictive maintenance is the definition of the training and testing dataset following the scheme of Figure 3: a first data partition is used for the model training and the last sample is adopted for the model testing. The detected data samples are stored into a MySQL table where each record contains the attributes to process.

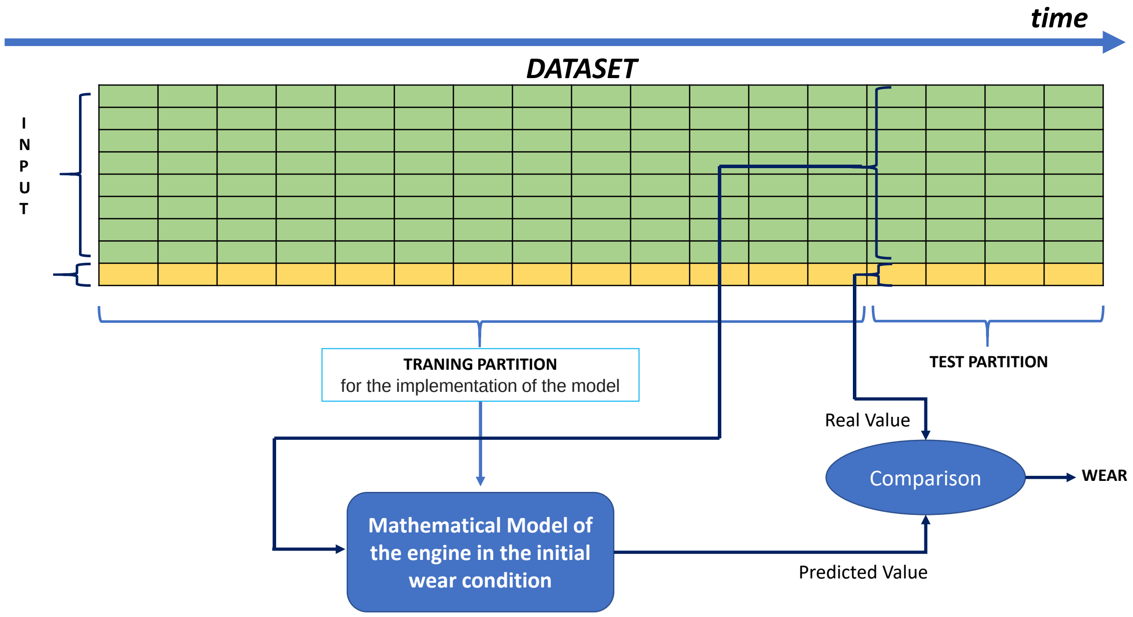

time
DATASET
INPUT
TRANING PARTITION
for the implementation of the model
TEST PARTITION
Real Value
Comparison
WEAR
Mathematical Model of the engine in the initial wear condition
Predicted Value

Figure 3. Training and testing dataset of the multilayer perceptron neural (MLP) network predicting maintenance.

The dataset partition used for the MLP-ANN calculus is: 80% of training and 20% of testing. This proportion provides the best model for the analyzed dataset (low calculus error value).

The MLP model has been tested using the dataset found in [31], where the following attributes were identified:

- GPS time (time acquired by the GPS module);
- Device time (internal clock time of the device);
- Longitude (longitude of the GPS coordinate);
- Latitude (latitude of the GPS coordinate);
- GPS speed (measured in meters/second representing the speed of the vehicle);
- Horizontal dilution of precision (horizontal error on the GPS position);
- Altitude (altitude acquired by the GPS module);
- Bearing (the horizontal angle between the direction and the north);
- G(x) (angular velocity (degree per second) on the X axis acquired with gyroscope);
- G(y) (angular velocity (degree per second) on the Y axis acquired with gyroscope);
- G(z) (angular velocity (degree per second) on the Z axis acquired with gyroscope);
- G calibrated (gyro calibration error);
- Engine coolant temperature (temperature in °C of coolant engine liquid);
- Engine RPM (angular speed of the motor shaft expressed in rpm);
- Intake air temperature (temperature of the air entering the combustion chamber expressed in °C);
- Engine load % (percentage of the maximum power supplied by the engine);
- Mass air flow rate (flow rate of the air flow entering the combustion chamber of the engine expressed in g/s);
- Throttle position manifold % (percentage of the accelerator position pressed).

Figure 4 shows the statistical plots of the dataset adopted for the performance check of the MLP algorithm.

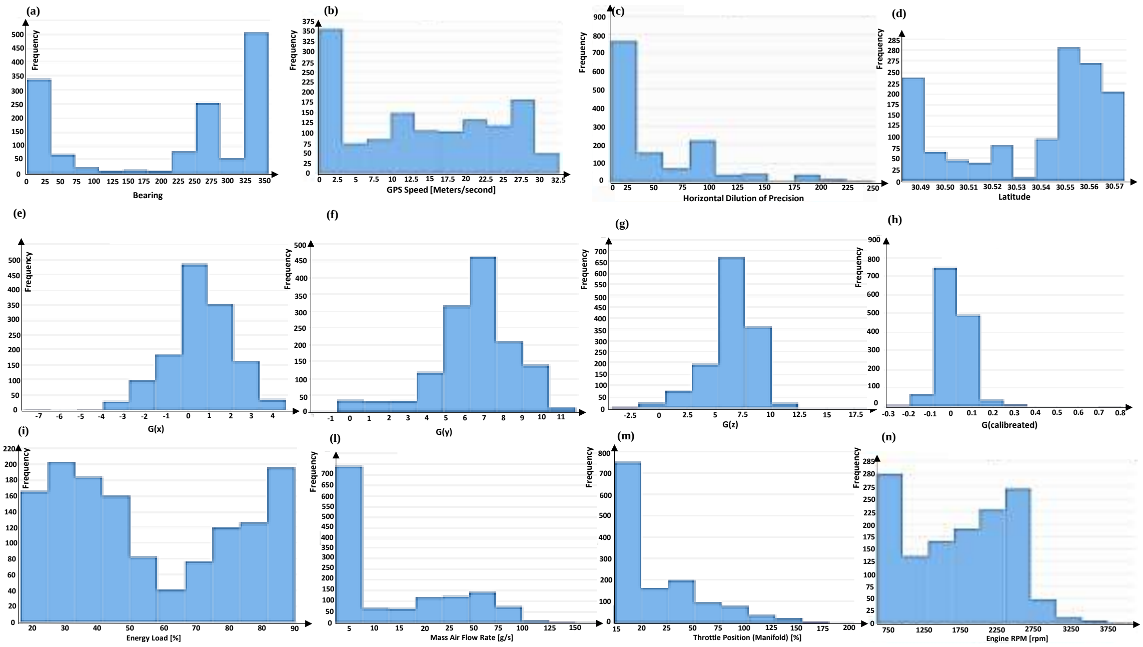


Figure 4. Statistical plots of parameter frequency of dataset used for the test of MLP algorithm: (**a**) bearing, (**b**) global positioning systems (GPS) speed, (**c**) horizontal dilution of precision, (**d**) latitude, (**e**) G(x), (**f**) G(y), (**g**) G(z), (**h**) G calibrated, (**i**) engine load %, (**l**) mass air flow rate (g/s), (**m**) throttle position manifold %, (**n**) engine revolutions per minute (RPM).

The MLP network has been implemented by the KNIME workflow of Figure 5 structured as follows:

- A data source “CSV Reader” block loading the bus data into a local repository (data extracted from the MySQL database);
- A data pre-processing filtering attributes to select for the data processing (“Column Filter” block);
- A data pre-processing block normalizing all numerical data of the filtered dataset (predictive model markup language (PMML) normalizer “Normalizer (PMML)”);
- A data pre-processing partitioning data for the training and testing processing;
- For the training of the MLP is adopted the efficient RProp algorithm [32,33] (“RProp MLP Learner” block constituting the training dataflow);
- The “MultiLayerPerceptronPredictor” block model the MLP neural network merging the training workflow with the testing one;
- The numeric score provides the mean squared error (MSE) defined as:

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{n} \tag{1}$$

where y_i is the measured value and $\bar{y}$ is the predicted one;

- the “Excel Writer (XLS)” block writes the scoring results in an excel file.

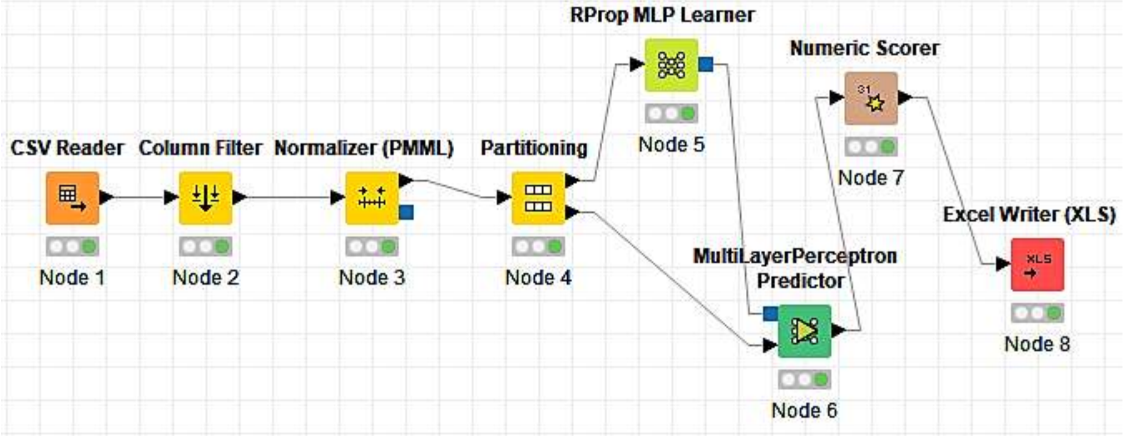


Figure 5. Konstanz Miner (KNIME) workflow implementing MLP for predictive maintenance.

For clustering results, indicating driver behavior has been applied to the k-means algorithm [34,35] using RapidMiner tool (see the related workflow in Figure 6). For the analysis of correlations between the variables, the correlation matrix algorithm of RapidMiner tool has been adopted [36] (see the related workflow in Figure 7).

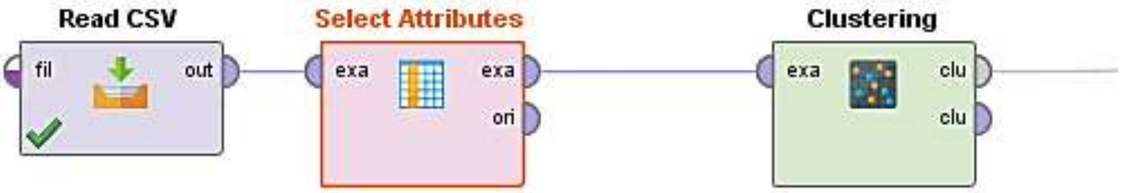


Figure 6. RapidMiner workflow implementing k-means algorithm.

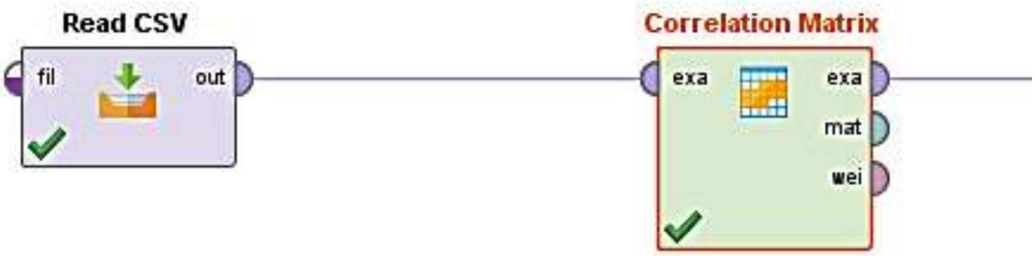


Figure 7. RapidMiner workflow implementing correlation matrix algorithm.

3. Testing and Results

The system represented in Figures 1 and 2 has been implemented by performing the following preliminary tests:

- Verification of the correct functioning of the accelerometer and the GPS module (this check also implies the correct connection with the Raspberry input pins);
- Checking of the auto-starting operation with .desktop files;
- Firmware testing;
- Verification of solution validity using the 4G WiFi router;
- Verification of server data receipt.

Figure 8a shows the testing circuit system assembling the components of the architecture of Figure 1, additionally Figure 8b illustrates the photo concerning server linking. The Raspberry is powered by the cigarette lighter socket, thanks to the car adapter, which has two USB 5V sockets at the output. The preliminary tests for the accelerometer firmware are performed with the engine off, while the preliminary tests of the Bluetooth OBD reader are executed with the engine running. When the Raspberry board receiver is turned on, it connects the testing laptop with the WiFi router. The Raspberry board is remotely controlled through the remote desktop control application. The OBD-related test of the reading script proved the detection of the vehicle data. The data flow is enabled through the Bluetooth OBD, by timing and synchronizing the Raspberry acquisition every ten seconds. The command *r = requests.post (URL_ODB_BUS, json = payload)* provides the following server connection response checking the correct data flow into the json package file:

- <response [200]>: json received successfully;
- <response [400]>: json not arrived at destination.

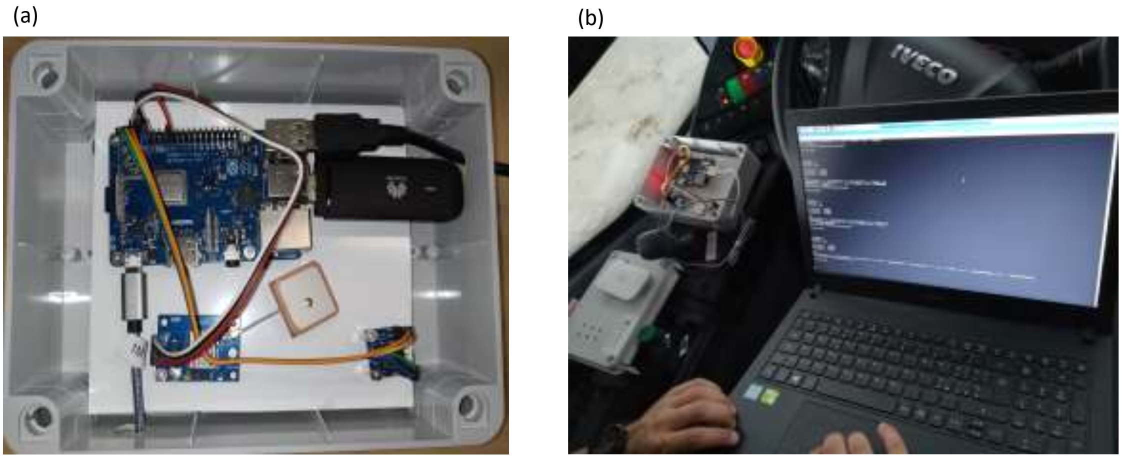


Figure 8. Preliminary assembly and online linking tests: (**a**) circuits of the architecture of Figure 1; (**b**) photo concerning remote server linking test.

In Figure 9 is illustrated a preliminary test of acceleration data acquisition.

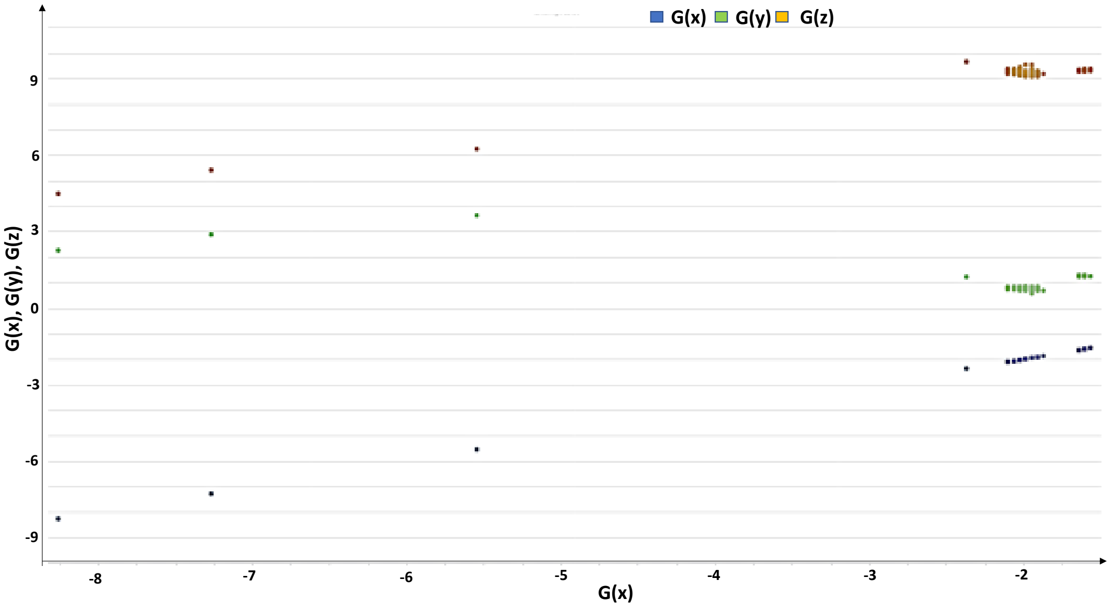


Figure 9. First check of data transmission concerning acceleration along the tree direction (G(x), G(y), and G(z)).

The MLP model has been checked by obtaining the parameters listed in Table 2, indicating the number of the hidden layers, the neuron number for the hidden layers and the MSE: by considering the testing dataset, very low MSE values are obtained, in the order of 10^{-2}. The MSE results delineate a good error trend versus the variation of the parameters, such as the number of hidden layers and number of neurons for the hidden layers thus proving the correct choice of the algorithm used for the prediction.

Table 2. Means square error (MSE) values and MLP parameter setting.

Number of Hidden Layers	Neuron Number for Hidden Layer	MSE
5	5	0.0149
5	10	0.0129
5	15	0.0098
5	20	0.085
10	5	0.0252
10	10	0.0122
10	15	0.0094
10	20	0.0082
15	5	0.0143
15	10	0.0031
15	15	0.0224
15	20	0.0323

An example of application of the KNIME MLP network is illustrated in Figure 7, where it is possible to observe that the predicted engine power (engine load) is higher than the measured engine load values thus predicting an accentuate engine wear. Moreover, as expected, Figure 10 illustrates a close correlation between engine RPM values and engine power.

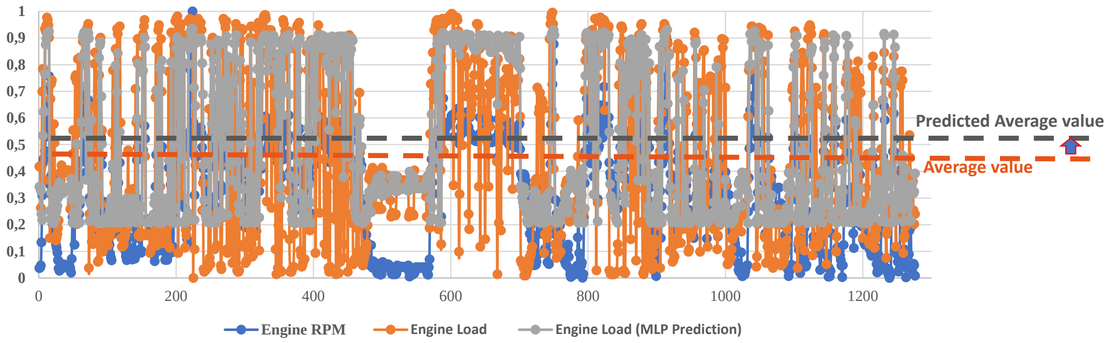


Figure 10. Comparison between engine RPM, engine load and predicted engine load. On the y-axis are indicated the normalized values, and on x-axis are indicated the records number function of the acquisition time.

In order to provide information about driver behavior, we executed the k-means algorithm fixing as K = three the number of clusters (three main driver behavior). Figure 11 illustrates the clusters by grouping the GPS speed and engine RPM parameters: the cluster indicated by the orange color (cluster 0) is representative of drivers tending to travel at low velocities and by accelerating slowly (prudent driving behavior). The drivers of the cluster indicated by the green color (cluster 1) travel with low speed but forcing the engine (high average RPM engine values), thus denoting an inefficient driving style, which could accelerate the engine wear. The cluster represented by the blue color (cluster 2) denotes drivers that mainly contribute to the vehicle wear.

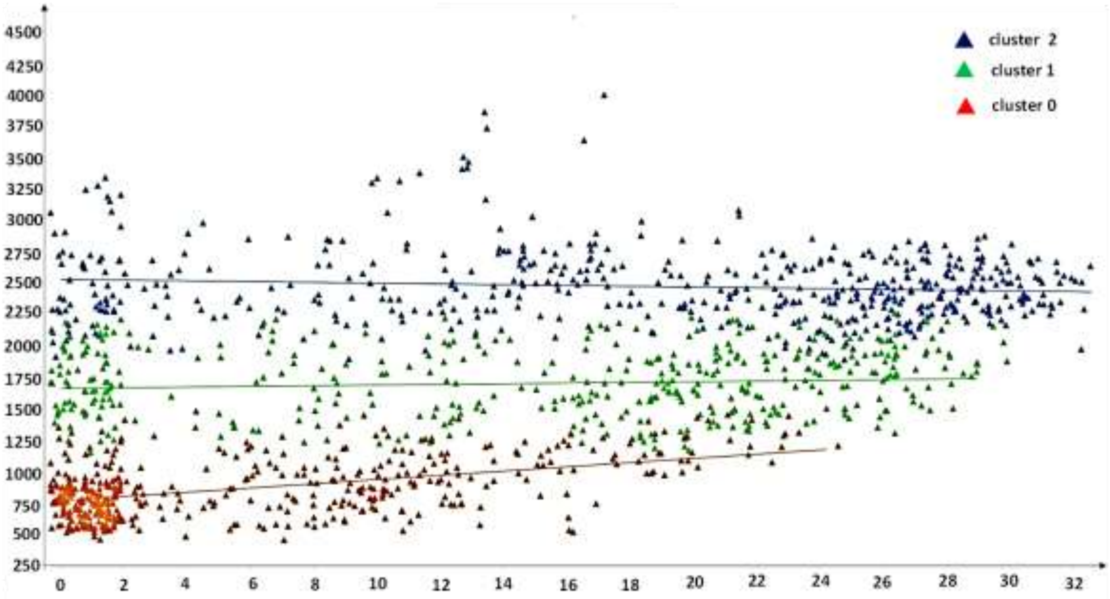


Figure 11. k-means analysis: clusters grouped by GPS speed and engine RPM parameters, and linear regression trend.

Figure 12 represents the same clusters and the linear regression trends, denoting that the cluster 0 is characterized by a low percentage of the accelerator pressed (throttle position), and a low engine power (engine load). The cluster 1 do not press excessively on the accelerator but the engine is forced, denoting that the gears of the vehicle are not often changed. The cluster 2 indicate the engine stress due to a strong pression of the accelerator.

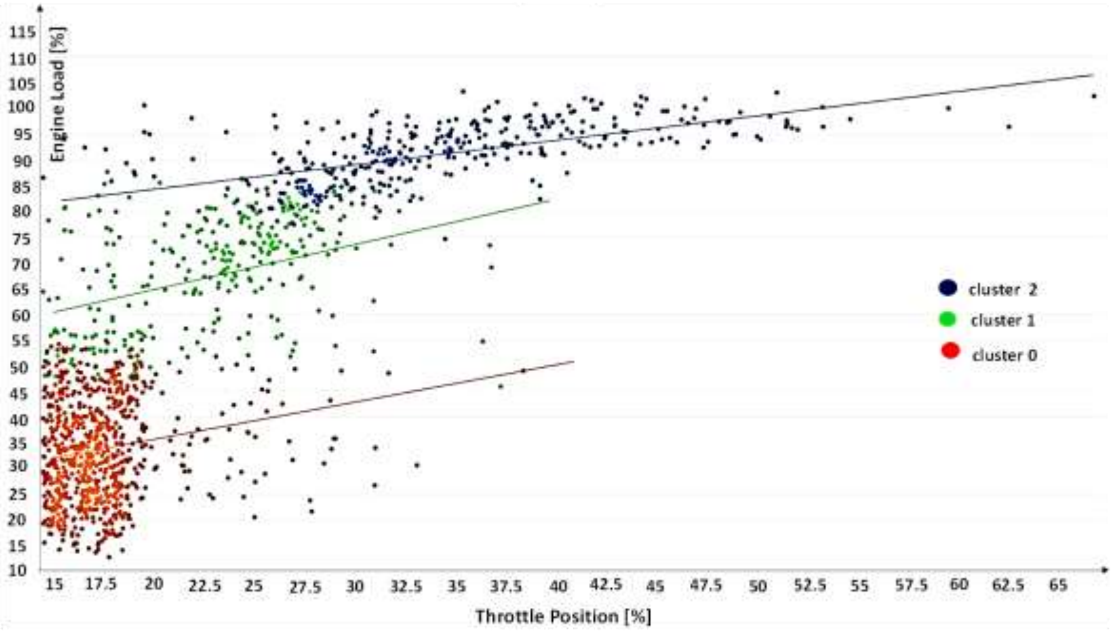


Figure 12. k-means analysis: clusters grouped by throttle position and engine load parameters, and linear regression trend.

The plot of Figure 13 confirms the correlation between engine load and engine RPM as deduced by the MLP analysis: a similar trend is observed in Figure 8.

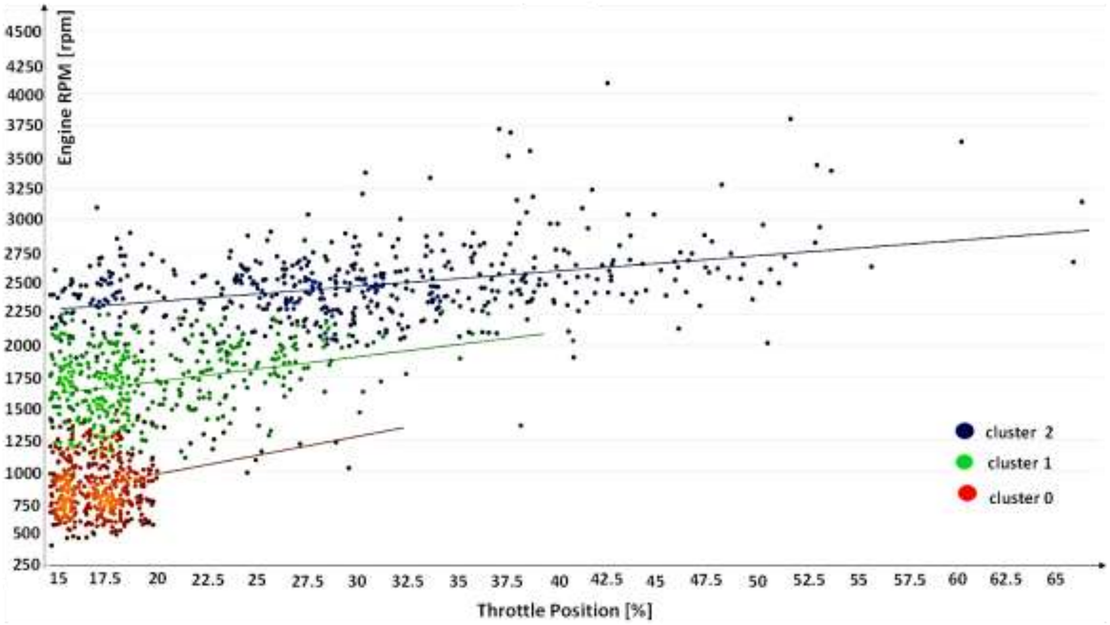


Figure 13. k-means analysis: clusters grouped by throttle position and engine RPM parameters, and linear regression trend.

4. Discussion

Starting with the MLP-ANN prediction of the engine stress, it is possible to re-plan the maintenance schedule of each vehicle. The standard predictive maintenance plan could change based on the MLP-ANN prediction of the engine stress, which is a function of the driver behavior: the planned period to perform bus maintenance can be anticipated by predicting a high engine wear. Figure 14 illustrates a theoretical plot merging this concept.

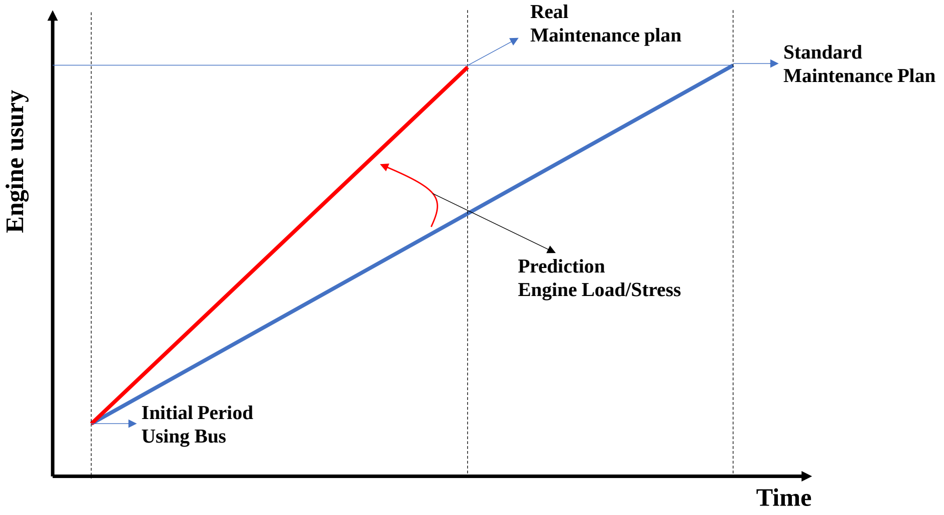


Figure 14. Bus maintenance plan variation based on prediction results.

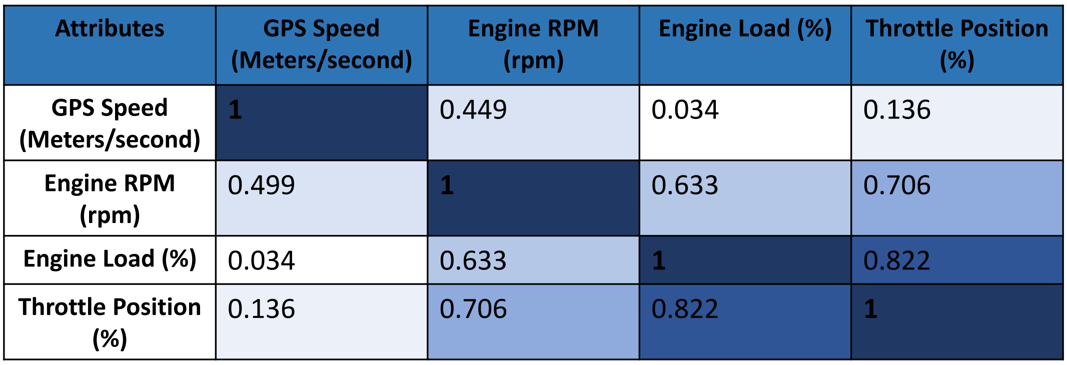
The driver style and behavior are represented in three cluster typologies by analyzing the more significant parameters such as the GPS speed, the engine RPM, the engine load, and the throttle position. Each cluster can be associated with a score (average, low and high) as KPI concerning the driver velocity, the engine stress and the driver caution (see Table 3). The driver velocity and the engine stress are also representative of the fuel consumption:

Table 3. Driver KPI: clusters and driver behavior.

Cluster Number	Color	Velocity	Engine Stress	Caution	Fuel Consumption	Total Efficiency Rate
Cluster 0	Orange	Average/Low	Low	High	Low	High
Cluster 1	Green	Average/Low	Average/Low	Average	Average	Average
Cluster 2	Blue	High	High	Low	High	Low

The low, average and high scores are denoted by red, orange and green color, respectively.

The same scoring of Table 3 is deduced mainly by the regression line slopes of Figure 7, Figure 8, and Figure 9. We observe that the predicted results and the KPI can be normalized to the unit, thus achieving an estimation scale (the results can be expressed in percentage). A full scenario is provided by the correlation matrix analysis, providing possible correlations between the most significant variables. Figure 15 reports the correlation matrix calculus enhancing the high correlation between throttle position and engine load and between throttle position and engine RPM, additionally a moderate correlation is observed between engine RPM and GPS speed, indicating the correct gear use of the drivers.



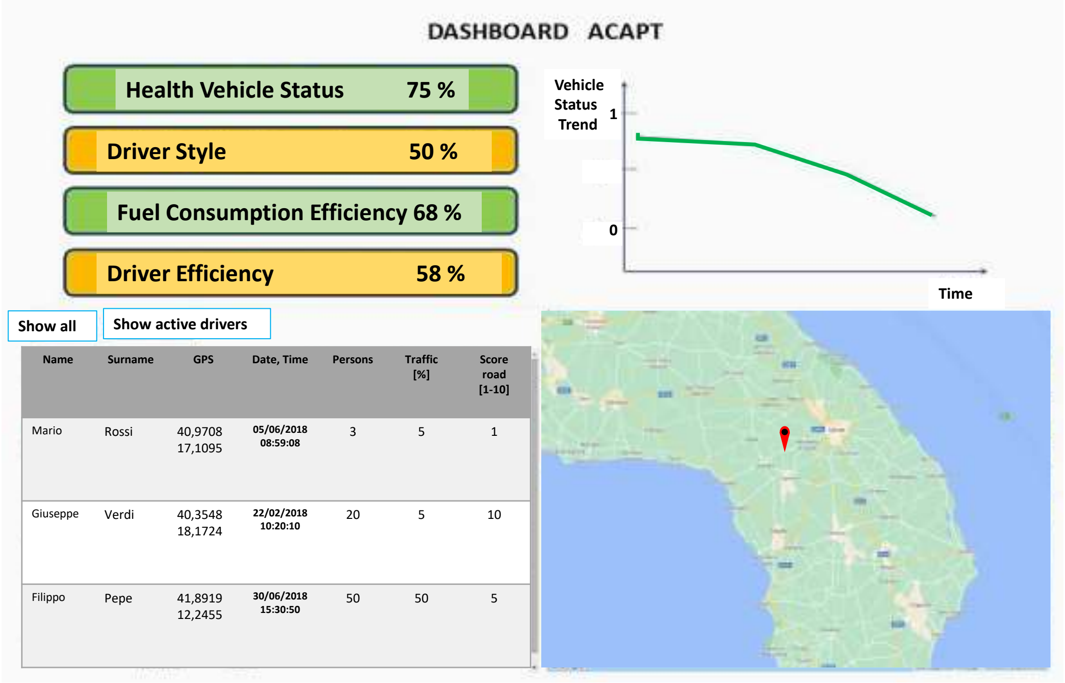
Attributes	GPS Speed (Meters/second)	Engine RPM (rpm)	Engine Load (%)	Throttle Position (%)
GPS Speed (Meters/second)	1	0.449	0.034	0.136
Engine RPM (rpm)	0.499	1	0.633	0.706
Engine Load (%)	0.034	0.633	1	0.822
Throttle Position (%)	0.136	0.706	0.822	1

Figure 15. Correlation matrix results.

The data are sampled every second. All data are grouped for a basic daily analysis. The k-means and MLP-ANN algorithms are also able to process all the collected data for monthly and yearly estimation, thus providing criteria for predictive maintenance.

The graphical dashboards can be automatized by inserting in the main workflow the delay blocks, timing the reading [37], and by simply substituting the reading CSV blocks with Python-based script, enabling the automated reading from the MySQL database trough web services [38]. All the results are collected into database system linked to a cloud platform with dashboards.

The platform provides online monitoring of the KPI, vehicle health status, fuel consumption efficiency, and driver efficiency. Figure 16 illustrates the implemented dashboard. The thresholds expressed in percentage defining the low, high and average KPI are low: 0% ÷ 40%, average: 41% ÷ 60%, and high: 61% ÷ 100%.



Health Vehicle Status 75 %

Driver Style 50 %

Fuel Consumption Efficiency 68 %

Driver Efficiency 58 %

Name	Surname	GPS	Date, Time	Persons	Traffic [%]	Score road [1-10]
Mario	Rossi	40,9708 17,1095	05/06/2018 08:59:08	3	5	1
Giuseppe	Verdi	40,3548 18,1724	22/02/2018 10:20:10	20	5	10
Filippo	Pepe	41,8919 12,2455	30/06/2018 15:30:50	50	50	5

Figure 16. Mock-up of the monitoring dashboard.

In Appendix A and in Appendix B are reported more details about SAE J1939 protocols and IVECO vehicle parameters, respectively.

Recent studies oriented the research in maintenance procedures by considering the programming approach [39], by classifying the state of the vehicles [40], or by applying artificial intelligence algorithms for the predictive maintenance of the only engine part [41]. The proposed research is oriented on a new concept of predictive maintenance merging procedures, artificial intelligence prediction and KPI driver efficiencies, thus providing a methodology that takes into account multiple weighted factors potentially influencing vehicle maintenance.

5. Conclusions

The proposed work shows how it is possible to combine IoT devices detecting bus status with the data mining algorithms simultaneously estimating engine status prediction and driver behavior. The compact and implemented electronic architecture can be applied for each vehicle characterized by ODB-II and SAE J1939 standards. Data of vehicles are transmitted in the cloud to a data mining engine performing driver KPI by defining a score using k-means clustering analysis, and by predicting engine stress through MLP-ANN algorithms. The proposed data mining models have been tested mainly with a stable dataset by providing a low MSE error, thus confirming the model accuracy. The output of the data mining algorithms allowed the establishment of criteria for the predictive maintenance, thus anticipating the maintenance in cases of predicted engine stress due also to incorrect driver behavior. The bus fleet efficiency has been estimated by considering the engine stress prediction and the driver KPI. The efficiency parameters are stored into a database system and remotely visualized by dashboards. The perspectives of the proposed research are mainly oriented on the automatic management of the maintenance of a large number of vehicles, and on the possibility to choose dynamically the drivers according to the KPI evaluation. The followed scientific approach is able to combine the predictive maintenance procedure updated by wear prediction with the driver efficiency, balancing the assignment of vehicles and drivers. The adopted self-learning MLP-ANN network is stable and can be improved if a large number of vehicles and drivers will be assigned according to the project perspectives. The proposed electronic components are adaptable to different types of vehicles. The limitations of the on-board IoT solution are mainly due to few connections to dedicate to other sensors or measuring devices. In particular, the Raspberry board has an I2C port (to which the 3D accelerometer is connected), a serial universal asynchronous receiver-transmitter (UART) port (to which the GPS module is connected) and a serial peripheral interface (SPI) port. In order to overcome this limit, have been added the four USB ports, two of which are used for the OBD device and for the internet key. To have other connection points, it is necessary to insert other boards such as Raspberry and Arduino, which also have analog inputs.

Author Contributions: Conceptualization, A.M.; methodology, A.M. and S.S.; software, A.M., S.S.; validation, A.M., and S.S.; formal analysis, A.M.; investigation, A.G. and S.S.; resources, A.G.; data curation, A.M., and S.S.; writing—original draft preparation, A.M.; supervision, A.G.; project administration, A.G. and A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The proposed work has been developed within the framework of the industry project titled "Gestione Predittiva della Manutenzione della flotta Autobus/Bus mediante Acquisizione Dati da Centralina e Sistemi Integrati di Monitoring Camera/GPS: "Bus Predictive Maintenance".

Conflicts of Interest: There are no conflicts of Interest.

Appendix A

As has already been said, with the SAE J1939 standard, data frames have an extended structure that has an extended 29-bit identification ID field. The latter is essential since it provides information on the type of PGN, on the priority of the same message, on the address of the device to which it is sending, and the intended recipient. In particular, we have the following structure:

Table A1. The 29-bit identifier (SAE J1939).

IDENTIFIER A (11 bit)				**SRS**	**IDE**	**IDENTIFIER B (18 bit)**		
PR	EDP	DP	PDU Format			PF Cont.	PDU Specific	Source Address
3 bit	1 bit	1 bit	6 bit (MSB)			2 bit	8 bit	8 bit

where:

- Priority (PR) (3 bit): defined during the arbitration phase.
- Extended data page (DP) (1 bit): generally set to zero.
- Data page (DP) (1 bit): if it is equal to one then the data frame is coded in compliance with the SAE J1939 standard.
- PDU format (PF) (8 bit): indicates the format of the data frame since its value varies the structure of the same frame.
- PF < 239: the data frame refers to specific devices with their 8-bit address. If it is encoded as (0xFF) then it is transmitted to all connected devices, i.e., broadcast.
- PF ≥ 240: the data frame corresponds to a broadcast message.

- PDU specific (PS) (8 bit): this field is specific to the data frame to be transmitted, as it varies in meaning according to the value encoded in the PF.
- PF < 239: the PDU specific field corresponds to the address of the device to which you want to send the data frame.
- PF ≥ 240: the PDU specific field becomes "Group Extension" to form the PGN of the transmitted PG.
- Source address (SD) (8 bit): it is the address of the device that is transmitting the CAN data frame.

The following example shows how CAN data frames are communicated and implemented with the SAE J1939 standard. Suppose we want to know the thermal parameters of the engine of the vehicle in question, which can be a bus or a road tractor. The group of parameters to be referred to corresponds to the PGN (0x00FEEE) entered in document J1939–71 relating to the same standard.

First of all, it is appropriate to request this PG by sending the request data frame to which the PGN is associated with broadcast. The structure of the ID is as follows:

Table A2. Identifier request (SAE J1939).

Priority	EDP	DP	PDU Format	PDU Specific	Source Address
110 (6 dec.)	0	0	234 (0xEA)	Destination Address-Global (255) o Specific	xxxx xxxx

The other fields of the data frame:

- DLC: set to a value of three.
- Data filed: the first three bytes correspond to the PGN you want to request, which in the case in question is (0x00FEEE).

Consequently, the device used, or the heavy vehicle engine ECU, sends a response to the request made with the related group of parameters (0x00FEEE). The structure of the ID is as follows:

Table A3. Identifier response (SAE J1939).

Priority	EDP	DP	PDU Format	PDU Specific	Source Address
110 (6 dec.)	0	0	254 (0xFE)-PGN	239 (0xEE)-PGN	xxxx xxxx

Appendix B

Tables A4 and A5 explain the variables of the IVECO vehicle.

Table A4. Parameter detected during the experimentation (BUS Iveco Crossway).

Variable	Variable Type	PGN hex	SPN	Number of Bits	Resolution	Offset
Total_Fuel_Cons	float	FEE9	250	32	0.5 L/Bit gain	0 L
Fuel_Rate	float	FEF2	183	16	0.05 L/h per bit	0 L/h
Inst_Fuel_Eco	float	FEF2	184	16	1/512 km/L per bit	0 km/L
Pos_Valve	float	FEF2	51	8	0.4%/bit	0%
Fuel_Level	float	FEFC	96	8	0.4%/bit	0%
Engine_Hours	float	FEE5	247	32	0.05 ora/bit	0
Total_RPM	float	FEE5	249	32	1000 giri/bit	0
Total_Distance	float	FEC1	917	32	5m/bit	0 m
Speed	float	FEF1	84	16	1/256 km/h Bit gain	0 km/h
RPM	float	F004	190	16	0.125 rpm/Bit gain	0 rpm
Percent_Torque	int	F004	513	8	1%/Bit	−125%
Percent_Engine	int	F003	91	8	1%/Bit	0%
Temp_Cool	int	FEEE	110	8	1 °C/bit	−40 °C
Temp_fuel	int	FEEE	174	8	2 °C/bit	−40 °C
Temp_Oil	float	FEEE	175	16	0.03125 °C/bit	−273 °C
Volt_Batt	float	//(AT RV)	//	//	//	//
SPN	int	ATMPFECA 1	//	19	//	//

Table A5. Iveco parameter meaning.

Variable	Description
Total_Fuel_Cons	By running a count for each service delivery, it is possible to monitor the total fuel consumption.
Fuel_Rate	It takes into account the driving style of the driver and any losses due to idling.
Inst_Fuel_Eco	High consumption indicates an incorrect driving style or the occurrence of a fault.
Pos_Valve	It allows to monitor the status of the engine fuel system.
Fuel_Level	It is closely linked to the autonomy of travel or to a possible loss of fuel.
Engine_Hours	It is useful for monitoring ordinary maintenance actions based on a time scale.
Total_RPM	It is useful for monitoring ordinary maintenance actions based on the RPM scale.
Total_Distance	It verifies the ordinary maintenance actions based on kilometric scale.
Speed	It is closely related to the driver's driving style.
RPM	A high value means more waste of fuel and an incorrect driving style.
Percent_Torque	It takes into account the performance of the engine with a view to safety and failures.
Percent_Engine	It decreases due to engine deterioration.
Temp_Cool	Index of correct engine operation for road safety and guide efficiency.
Temp_fuel	It is useful for fire risk estimation.
Temp_Oil	It is adopted to control the correct functioning of the motion transmission parts.
Volt_Batt	It is used to check possible outages following a continuous engine shutdown.
SPN	Index of any anomalies that may require roadside assistance.

References

1. Koscher, K.; Czeskis, A.; Roesner, F.; Patel, S.; Kohno, T.; Checkoway, S.; McCoy, D.; Kantor, B.; Anderson, D.; Shacham, H.; et al. Experimental security analysis of a modern automobile. In Proceedings of the 2010 IEEE Symposium on Security and Privacy, Berkeley/Oakland, CA, USA, 16–19 May 2010.
2. Wolf, M.; Weimerskirch, A.; Paar, C. Secure in-vehicle communication. In *Embedded Security in Cars Securing Current and Future Automotive IT Applications*; Lemke, K., Paar, C., Wolf, M., Eds.; Springer: Berlin/Heidelberg, Germany; GmbH: Berlin, Germany, 2006; pp. 95–109.
3. Klinedinst, D.; King, C. *On Board Diagnostics: Risks and Vulnerabilities of the Connected Vehicle*; Software Engineering Institute, Carnegie Mellon University: Pittsburgh, PA, USA, 2016.
4. Wolf, M.; Weimerskirch, A.; Paar, C. Security in automotive bus systems. In Proceedings of the Workshop on Embedded Security in Cars (ESCAR), Bochum, Germany, 10–11 November 2004; pp. 1–13.
5. Prakash, C.B.; Sirisha, K. Design and implementation of a vehicle theft control unit using GSM and CAN technology. *Int. J. Innov. Res. Electron.* **2014**, *1*, 46–53.
6. Kang, M.-J.; Kang, J.-W. Intrusion detection system using deep neural network for in-vehicle network security. *PLoS ONE* **2016**, *11*, e0155781. [CrossRef] [PubMed]
7. Suwatthikul, J. Fault detection and diagnosis for in-vehicle networks. In *Fault Detection*; Zhang, W., Ed.; InTech: Rijeka, Croatia, 2010; pp. 283–306.
8. Jyothi, M.K.; Ravi, S.T. Vehicle health monitoring system. *Int. J. Eng. Res. Appl.* **2012**, *2*, 1162–1167.
9. Lee, M.G.; Park, Y.K.; Jung, K.K.; Yoo, J.J. Estimation of fuel consumption using in-vehicle parameters. *Int. J. u e Serv. Sci. Technol.* **2011**, *4*, 37–46.
10. Ericsson, E. Independent driving pattern factors and their influence on fuel-use and exhaust emission factors. *Transp. Res. Part D Transp. Environ.* **2001**, *6*, 325–345. [CrossRef]
11. Pethakar, S.S.; Srivastava, N.; Suryawanshi, S.D. GPS and GSM based vehicle tracing and employee security system. *Int. J. Comput. Appl.* **2013**, *62*, 37–42. [CrossRef]
12. Pooja, S. Vehicle tracking system using GPS. *Int. J. Sci. Res.* **2013**, *2*, 128–130.
13. Singh, P.; Sethi, T.; Biswal, B.B.; Pattanayak, S.K. A smart anti-theft system for vehicle security. *Int. J. Mater. Mech. Manuf.* **2015**, *3*, 249–254. [CrossRef]
14. Rana, T.; Shah, A.; Rana, P.; Chandak, S. Smart vehicle security. *Int. J. Eng. Sci. Comput.* **2017**, *7*, 1–3.
15. Dukare, S.S.; Patil, D.A.; Rane, K.P. Vehicle tracking, monitoring and alerting system: A review. *Int. J. Comput. Appl.* **2015**, *119*, 39–44. [CrossRef]
16. Kim, J.-S.; Lee, H.-J.; Oh, R.-D. Smart integrated multiple tracking system development for IOT based target-oriented logistics location and resource service. *Int. J. Smart Home* **2015**, *9*, 195–204. [CrossRef]

17. Tarapiah, S.; Atalla, S. Public transportation management system based on GPSWiFi and open street maps. *Int. J. Adv. Comput. Sci. Appl.* **2015**, *6*, 189–194. [CrossRef]
18. Omar, H. Intelligent traffic information system based on integration of internet of things and agent technology. *Int. J. Adv. Comput. Sci. Appl.* **2015**, *6*, 37–43. [CrossRef]
19. Menon, A.; Sinha, R.; Ediga, D.; Iyer, S. Implementation of internet of things in bus transport system of Singapore. *Asian J. Eng. Res.* **2013**, *1*, 8–17.
20. Nowaczyk, S.; Prytz, R.; Rögnvaldsson, T.; Byttner, S. Towards a machine learning algorithm for predicting truck compressor failures using logged vehicle data. In Proceedings of the Twelfth Scandinavian Conference on Artificial Intelligence, Aalborg, Denmark, 20–22 November 2013; Jaeger, M., Nielsen, T.D., Viappiani, P., Eds.; IOS Press Frontiers in Artificial Intelligence and Applications: Amsterdam, The Netherlands, 2013; pp. 205–214.
21. Arora, S.; Kaur, P.; Arora, P. Economical maintenance and replacement decision making in fleet management using data mining. *SIJ Trans. Comput. Sci. Eng. Appl.* **2013**, *1*, 9–20. [CrossRef]
22. Fan, Y.; Nowaczyk, S.; Rögnvaldsson, T. Incorporating expert knowledge into a self-organized approach for predicting compressor faults in a city bus fleet. *Front. Artif. Intell. Appl.* **2015**, *278*, 58–67.
23. Li, X.; Yu, X.; He, K. Towards effective bus lane monitoring using camera sensors. *Wirel. Sens. Netw.* **2011**, *3*, 174–182. [CrossRef]
24. Hussain, S.; Mahmud, U.; Yang, S. Car e-talk: An IoT-enabled cloud-assisted smart fleet maintenance system. *IEEE Internet Things J.* **2020**, 1. [CrossRef]
25. Killeen, P.; Ding, B.; Kiringa, I.; Yeap, T. IoT-based predictive maintenance for fleet management. *Procedia Comput. Sci.* **2019**, *151*, 607–613. [CrossRef]
26. Sanches, A.M.; Loures, E.D.F.R.; De Lima, E.P. Use of PROMETHEE method for decision making in bus fleet maintenance proposal of framework. *Procedia Manuf.* **2019**, *39*, 1913–1920. [CrossRef]
27. Yao, Y.; Zhao, X.; Wu, Y.; Zhang, Y.; Rong, J. Clustering driver behavior using dynamic time warping and hidden Markov model. *J. Intell. Transp. Syst.* **2019**, *1*, 1–14. [CrossRef]
28. Bichicchi, A.; Belaroussi, R.; Simone, A.; Vignali, V.; Lantieri, C.; Li, X. Analysis of road-user interaction by extraction of driver behavior features using deep learning. *IEEE Access* **2020**, *8*, 19638–19645. [CrossRef]
29. Berthold, M.R.; Cebron, N.; Dill, F.; Gabriel, T.R.; Kötter, T.; Meinl, T.; Ohl, P.; Sieb, C.; Thiel, K.; Wiswedel, B. KNIME: The konstanz information miner. In *Data Analysis, Machine Learning and Applications. Studies in Classification, Data Analysis, and Knowledge Organization*; Preisach, C., Burkhardt, H., Schmidt-Thieme, L., Decker, R., Eds.; Springer: Berlin/Heidelberg, Germany, 2008.
30. Massaro, A.; Maritati, V.; Savino, N.; Galiano, A.M.; Convertini, D.; De Fonte, E.; Di Muro, M. A study of a health resources management platform integrating neural networks and DSS telemedicine for homecare assistance. *Information* **2018**, *9*, 176. [CrossRef]
31. Data from OBD (On Board Diagnostics). Available online: https://www.kaggle.com/vbandaru/data-from-obd-on-board-diagnostics (accessed on 28 August 2020).
32. Riedmiller, M.; Braun, H. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In Proceedings of the IEEE International Conference on Neural Networks, San Francisco, CA, USA, 28 March–1 April 1993.
33. Igel, C.; Toussaint, M.; Weishui, W. Rprop using the natural gradient. In *Trends and Applications in Constructive Approximation*; Mache, D.H., Szabados, J., de Bruin, M.G., Eds.; Birkhäuser: Basel, Switzerland, 2005; Volume 151, pp. 259–272. [CrossRef]
34. Massaro, A.; Dipierro, G.; Cannella, E.; Galiano, A.M. Comparative analysis among discrete fourier transform, K-means and artificial neural networks image processing techniques oriented on quality control of assembled tires. *Information* **2020**, *11*, 257. [CrossRef]
35. Massaro, A.; Leogrande, A.; Lisco, P.; Galiano, A.; Savino, N. Innovative Bi approaches and methodologies implementing a multilevel analytics platform based on data mining and analytical models: A case of study in roadside assistance services. *Int. J. Soft Comput. Artif. Intell. Appl.* **2019**, *8*, 17–36. [CrossRef]
36. Massaro, A.; Vitti, V.; Lisco, P.; Galiano, A.; Savino, N. A business intelligence platform implemented in a big data system embedding data mining: A case of study. *Int. J. Data Min. Knowl. Manag. Process.* **2019**, *9*, 1–20. [CrossRef]

37. Massaro, A.; Mastandrea, G.; D'Oriano, L.; Rana, G.R.; Savino, N.; Galiano, A. Systems for an intelligent application of automated processes in industry: A case study from "PMI IoT Industry 4.0" project. In Proceedings of the IEEE MetroInd4.0&IoT 2020, Rome, Italy, 3–5 June 2020; pp. 21–26. [CrossRef]
38. Massaro, A.; Maritati, V.; Galiano, A.; Birardi, V.; Pellicani, L. ESB platform integrating knime data mining tool oriented on industry 4.0 based on artificial neural network predictive maintenance. *Int. J. Artif. Intell. Appl.* **2018**, *9*, 1–17. [CrossRef]
39. Sanchez, D.T.; Boyacı, B.; Zografos, K.G. An optimisation framework for airline fleet maintenance scheduling with tail assignment considerations. *Transp. Res. Part B Methodol.* **2020**, *133*, 142–164. [CrossRef]
40. Shafi, U.; Safi, A.; Shadid, A.R.; Ziauddin, S.; Saleem, M.Q. Vehicle remote health monitoring and prognostic maintenance system. *J. Adv. Transp* **2018**, 1–10. [CrossRef]
41. Mashhadi, P.S.; Nowaczyk, S.; Pashami, S. Stacked ensemble of recurrennt neural networks for predicting turbocharger remaining useful life. *Appl. Sci.* **2019**, *10*, 69. [CrossRef]