COMPUTER SCIENCE, COMPUTATIONAL METHODS, ALGORITHMS AND ARTIFICIAL INTELLIGENCE SESSION

PREDICTION OF CORRECT READINGS OF CAR ENGINE MASS AIR FLOW SENSORS

Dejan Čugalj*,
Marina Marjanović Jakovljević,
Miodrag Živković

Singidunum University,
Belgrade, Serbia

Abstract:

Let us suppose a situation in which it is necessary to check the car before going on a journey, a situation in which breakdowns appear but also mysteriously disappear, a situation where solving problems with the vehicle is only possible by looking at the data obtained by reading sensors, routine vehicle checking (insight into data irregularity), "real-time" monitoring of car condition…

The main goal of this paper is the early detection of erroneous readings of vehicle airflow systems in real time and constant monitoring of the subject vehicle, by coupling and using trained models based on machine learning tools of artificial intelligence and using "Infinity" device.

Keywords:

Can-*bus*, Automotive, Mass air flow sensor, Linear regression, Random forest

INTRODUCTION

As a result of climate change and the greenhouse effect, the Environmental Protection Agency (EPA) [1] and California Air Resources Board (CARB) [2] were authorized by the government to apply protocols that monitor the exhaust gas emissions from car manufacturers. During a car’s lifetime, the owner is obliged to maintain the emission of exhaust gases in the prescribed parameters, while the manufacturers are obliged to provide the necessary infrastructure which gives an insight into the emission of harmful gases into the atmosphere.

Electronic monitoring and diagnostic operation of internal combustion engines, cars and light trucks was introduced in the late 1970s, and already in the early 1980s the OBD (On-Board Diagnostics) system designed to inspect the compliance with EPA and CARB emission control standards started to be applied. Over the following years, the diagnostic systems became increasingly sophisticated, so in the mid-1990s, the OBD standard received its upgrade called the OBDII standard. This new standard provides almost complete insight of car exhaust emissions, but also monitors other systems such as chassis parts, accessories, electronic vehicle slip system, etc.

Correspondence:
Dejan Čugalj

e-mail:
dejan.cugalj.10@singimail.ac.rs

DOI: 10.15308/Sinteza-2021-3-8

By implementing the OBDII standard, and in order to control the operation of all car systems and subsystems, the automotive industry had the obligation to introduce standardization, protocols according to which computers communicate with each other, without the need to introduce a master host as a control point in the exchange of information.

The auto industry that was obliged to implement the OBDII standard, inspired "Robert Bosch GmbH" to begin developing CAN protocol in 1983. It was officially introduced in 1986 at the conference of the Society of Automotive Engineers (SAE) [3] and as early as 1991, CAN protocol started to be implemented in Mercedes cars.

Shortly afterward, the acceptance of this protocol became wide spread, in other words, all car manufacturers were obliged to implement CAN protocol with part of OBDII standard [4] which are associated with control of car exhaust emission.

The CAN protocol is specific in that every unit sends data with a single header to the network, and when an end node computer needs it, it uses it by intercept the information from the CAN-*bus* network.

This research uses the data collected from the CAN-bus car network for the purpose of predicting incorrect readings of mass air flow sensors, by coupling and using trained models, machine learning tools of artificial intelligence, with the help of the "Infinity" device.

2. OVERVIEW OF TEST CYCLE TOPOLOGY

The proposed cycle as we can see at Figure 1, is based on the idea of training the data obtained from the vehicle in a relatively short interval while the vehicle engine is idling, by gradually increasing and decreasing the engine speed. A piece of hardware was developed for this research which, connected to the diagnostic port of the car, performs the basic purpose of collecting and sending data to a remote server, as well as predicting, by regression model of real-time machine learning, the values of mass air flow sensor which is described in further bellow in this paper.

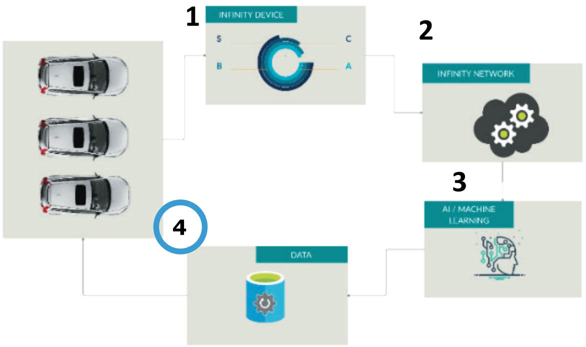


Figure 1. Topology of MAF sensor test cycle

3. NATURE OF DATA STRUCTURE

The CAN protocol [5], although standardized, did not offer a significant opportunity for an enthusiastic researcher to gain an insight into it, because each manufacturer interpreted and implemented the CAN protocol in its own way. This created an absurd situation that the standardized protocol was not standard. The problem was that the headers in sending messages through the car network were not unique to each make, type and model of a particular vehicle manufacturer.

This situation presented an opportunity to the world regulatory body for pollution control to bring under control air pollution caused by cars, and the epilogue of the introduction of standards is an obligation for all manufacturers in the automotive industry to implement standardized protocol headers for exhaust gas purification.

The standard information request headers, On-Board Diagnostics Parameter IDs (OBDII PIDs), which provide insight into a car's air purification system, have opened up the possibility of standardizing at least a small segment the CAN protocol and enable the development of various applications that are not related to exhaust gas control but use standard request headers (hereinafter: PID) [Figure 2].

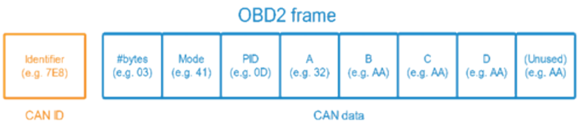


Figure 2. CAN-bus message header structure (PID)

The data analyzed in this paper were obtained from the CAN-*bus* network, utilizing several standardized PIDs that were used to predict faulty, incorrect readings of the vehicle engine Mass Air Flow sensor (MAF) [Figure 3].

4. DATASET STRUCTURE

The CAN-*bus* network is "noisy", while the maximum flow is 500 Kb/s, and all car sensors, actuators and computers constantly send information about their status, condition and values. Sensor values that are by their nature linear are discretized and as such sent to the network. The end nodes (computers) in a vehicle have access to all the data and only those that are essential and necessary for operation are prioritized and "extracted" as information.

The data status that the Engine Control Unit (hereinafter: ECU) "extracted" from the CAN-*bus* network was used to generate a dataset [6], in order to predict the correct operation of the MAF sensor. The generated dataset has the information structured of three independent variables obtained from the ECU, specifically from the sensors:

- *RPM - Revolutions Per Minute sensor*
- *MAP - Manifold Absolute Pressure sensor*
- *FGP - Fuel Gauge Pressure sensor*

The values of the selected prediction attribute, the dependent variable, are obtained from the Mass Air Flow sensor [Figure 3], and the measured values are mapped into the amount of air "sucked" into the car engine. In short, this sensor is of prime importance for the proper operation of an engine, since the data obtained from this sensor are used by the central ECU to correct the amount of injected fuel and thus enable coordinated engine operation.

Figure 3. Mass Air Flow sensor (MAF)

The combination of independent attributes (RPM, MAP, FGP) and dependent variable (MAF), can be clearly seen in the correlation matrix in Figure 4. and Figure 5.

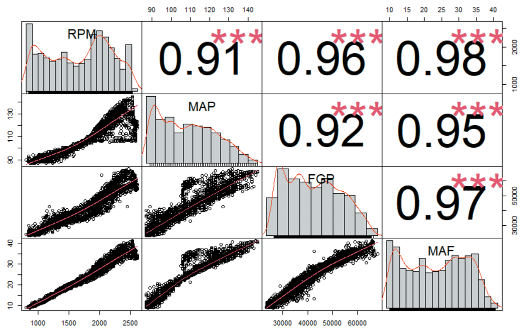


Figure 4. Correlation matrix of dependent variables (RPM, MAP, FGP) and independent variable (MAF)

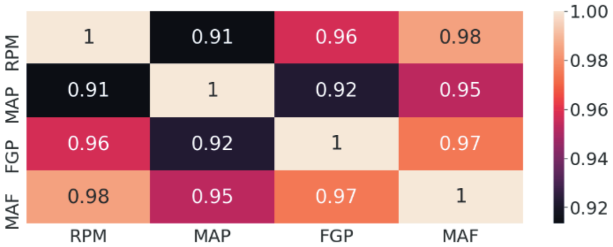


Figure 5. Correlation matrix of dependent variables (RPM, MAP, FRGP) and independent variable (MAF)

5. CHOICE OF MACHINE LEARNING METHOD

The problem of machine learning is of regression type with data that are linearly related. It should be noted that the training data set is not large since a small number of samples is sufficient for the proposed method. For that reason, a ten-minute training on a good condition vehicle can be considered sufficient for future prediction of incorrect MAF sensor readings. The data on which the training and evaluation were performed in this paper contain 2.759 samples obtained from the CAN-*bus* network. Machine learning of time series regression rules can be viewed as an interpretation of recognizing the dependence of independent attributes in the collected data. Inductive learning based on the obtained examples, after the application of machine learning methods, maps the time sequences into a prediction attribute which further predicts and checks the future input values of the MAF sensor.

Considering the linear connection of dependent attributes, this paper uses and evaluates the following regression methods of machine learning, which are:

- *Simple Linear Regression (SLR)*
- *Multiple Linear Regression (MLR)*
- *Random Forest (RF)*

6. SIMPLE LINEAR REGRESSION (*SLR*)

The general form of SLR model, according to [7], can be represented as:

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \qquad i = 1,2,...,N$$

where Y_i is the dependent variable, x_i is the value of the independent attribute, β_0, β_1 is the unknown constant (regression parameters), N is the size of the basic set of attributes.

The best candidate of the independent attribute of SLR was obtained by looking at the correlation matrix in Table 1. The strongest correlation of MAF are RPM sensor readings.

	RPM	MAP	FGP	MAF
RPM	1.000000	0.913430	0.957484	**0.984641**
MAP	0.913430	1.000000	0.922442	0.953109
FGP	0.957484	0.922442	1.000000	0.974711
MAF	**0.984641**	0.953109	0.974711	1.000000

Table 1. Correlation matrix RPM, MAP, FGP, MAF dependence of SLR attributes

A graphical representation of the linearity of the RPM and MAF sensors can be seen in Figure 6 and Figure 7.

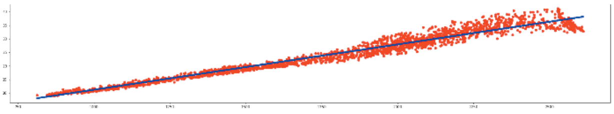


Figure 6. RPM and MAF attribute linearity (training set)

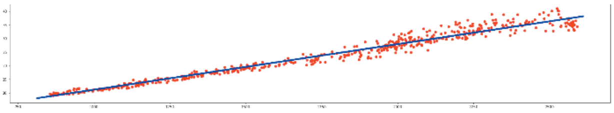
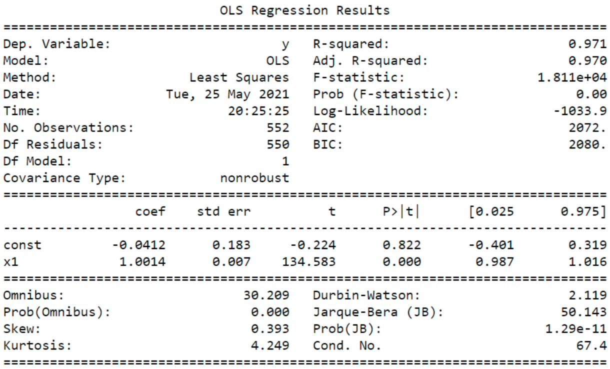


Figure 7. RPM and MAF attribute linearity (test set)

6.1. SIMPLE LINEAR REGRESSION - EVALUATION OF THE OBTAINED RESULTS

After applying OLS and statistical methods of error measurement, the following results were obtained on the test set [Figure 8]:



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.971
Model:                            OLS   Adj. R-squared:                  0.970
Method:                 Least Squares   F-statistic:                 1.811e+04
Date:                Tue, 25 May 2021   Prob (F-statistic):               0.00
Time:                        20:25:25   Log-Likelihood:                -1033.9
No. Observations:                 552   AIC:                             2072.
Df Residuals:                     550   BIC:                             2080.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.0412      0.183     -0.224      0.822      -0.401       0.319
x1             1.0014      0.007    134.583      0.000       0.987       1.016
==============================================================================
Omnibus:                       30.209   Durbin-Watson:                   2.119
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               50.143
Skew:                           0.393   Prob(JB):                     1.29e-11
Kurtosis:                       4.249   Cond. No.                         67.4
==============================================================================
```

Figure 8. OLS evaluation of learned Linear Regression model (test set)

- Mean Squared Error = **2.48**
- Root Mean Squared Error = **1.57**
- Mean Absolute Error = **1.15**
- R-squared (R^2) = **0.971**

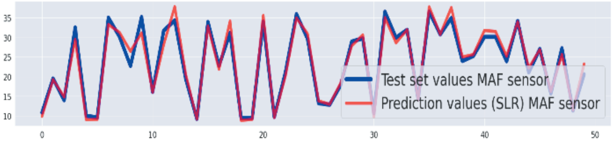


Figure 9. Comparative representation of predictive and test results (SLR)

7. MULTIPLE LINEAR REGRESSION (MLR)

The general form of the MLR predictor, , according to [8], can be represented as:

$$Y_i = \beta_0 + \beta_{1\ xi,1} + \ldots + \beta_1 x_{p-1} x_{i,p-1} + \varepsilon_i$$
$$i = 1,2,\ldots,N$$

where $x_{i,1}$ represents the i -th value of the k-th predictor for $i = 1, \ldots, n$.

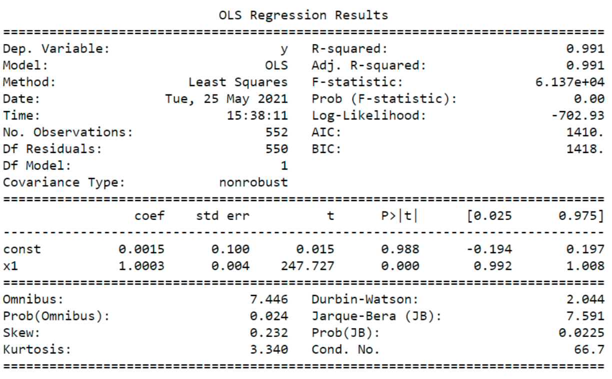
MLR takes several independent attributes, which can be seen in the correlation matrix done over the dataset in Table 2:

	RPM	MAP	FGP	MAF
RPM	1.000000	0.913430	0.957484	**0.984641**
MAP	0.913430	1.000000	0.922442	**0.953109**
FGP	0.957484	0.922442	1.000000	**0.974711**
MAF	**0.984641**	**0.953109**	**0.974711**	1.000000

Table 2. Correlation matrix RPM, MAP, FRGP, MAF

7.1. MULTIPLE LINEAR REGRESSION - EVALUATION OF THE OBTAINED RESULTS

After applying OLS and statistical methods of error measurement, the following results were obtained [Figure 10]:



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.991
Model:                            OLS   Adj. R-squared:                  0.991
Method:                 Least Squares   F-statistic:                 6.137e+04
Date:                Tue, 25 May 2021   Prob (F-statistic):               0.00
Time:                        15:38:11   Log-Likelihood:                -702.93
No. Observations:                 552   AIC:                             1410.
Df Residuals:                     550   BIC:                             1418.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0015      0.100      0.015      0.988      -0.194       0.197
x1             1.0003      0.004    247.727      0.000       0.992       1.008
==============================================================================
Omnibus:                        7.446   Durbin-Watson:                   2.044
Prob(Omnibus):                  0.024   Jarque-Bera (JB):                7.591
Skew:                           0.232   Prob(JB):                       0.0225
Kurtosis:                       3.340   Cond. No.                         66.7
==============================================================================
```

Figure 10. OLS evaluation of learned multiple linear regression model (test set)

- Mean Squared Error = **0.75**
- Root Mean Squared Error = **0.86**
- Mean Absolute Error = **0.67**
- R-squared (R^2) = **0.991**

A graphical representation comparing predictive and test values can be seen in Figure 11.

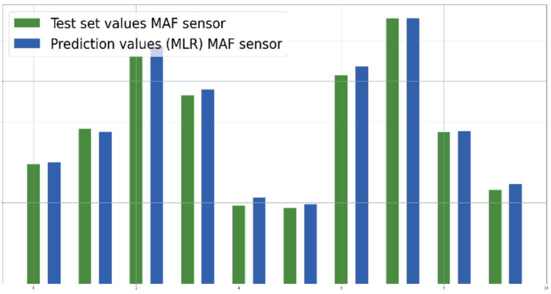


Figure 11. Comparative representation of predictive and test results (MLR)

8. RANDOM FOREST ENSEMBLES (RF)

The RF method of machine learning [9] is implemented in this paper from the Scikit-learn library, which in its implementation uses Gini Importance, which can be presented in its basic form of a binary tree as:

$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)}$$

Where ni_j is the importance of node *j*, w_j is the weight number of samples that reached node *j*, C_j is the noise value of point *j*, *left(j)* is the left point of the child in the branches of the tree, *right(j)* is the right point of the child in the branches of the tree.

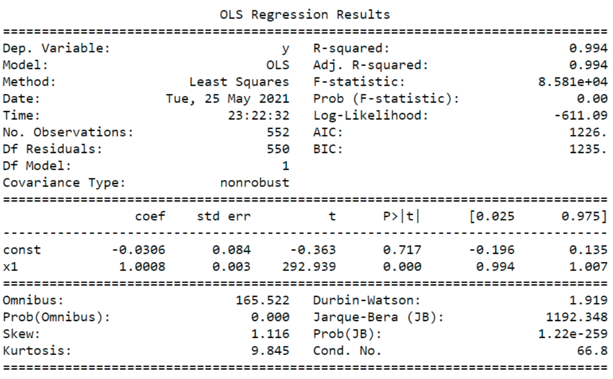
As with the MLR method of machine learning, RPM, MAP, FGP independent attributes were taken into account as input learning parameters of the RF model. The correlation matrix of the coupling can be seen in Table 2.

The parameters of the RF regressor that were taken into account in the training set are:

- *bootstrap = True*
- *n_estimators = 100*

8.1. RANDOM FOREST ENSEMBLE – EVALUATION OF THE OBTAINED RESULTS

After applying OLS and statistical methods of error measurement, the following results were obtained [Figure 12]:



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.994
Model:                            OLS   Adj. R-squared:                  0.994
Method:                 Least Squares   F-statistic:                 8.581e+04
Date:                Tue, 25 May 2021   Prob (F-statistic):               0.00
Time:                        23:22:32   Log-Likelihood:                -611.09
No. Observations:                 552   AIC:                             1226.
Df Residuals:                     550   BIC:                             1235.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.0306      0.084     -0.363      0.717      -0.196       0.135
x1             1.0008      0.003    292.939      0.000       0.994       1.007
==============================================================================
Omnibus:                      165.522   Durbin-Watson:                   1.919
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1192.348
Skew:                           1.116   Prob(JB):                    1.22e-259
Kurtosis:                       9.845   Cond. No.                         66.8
==============================================================================
```

Figure 12. OLS evaluation of learned Random Forest model (test set)

- Mean Squared Error (test set) = **0.53**
- Root Mean Squared Error (test set) = **0.73**
- Mean Absolute Error (test set) = **0.52**
- R-squared (R^2) = **0.994**

A graphical representation comparing predictive and test values can be seen in Figure 13.

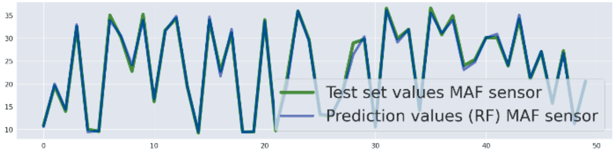


Figure 13. Comparative representation of RF model predictive and test results

9. CONCLUDING REMARKS

The main goal of the paper is to check the correctness of the MAF sensor reading in a car by applying machine learning tools over time samples obtained from the CAN-*bus* network of vehicles.

The paper presents a situation when the methods of simple linear regression take into account only one correlation attribute, namely the reading of the engine speed sensor (RPM), while other evaluations of the learned models of machine learning show the results obtained over test sets.

Since the nature of the data is linearly dependent, the application of multiple linear regression usually gives better results than simple linear regression while the best results are obtained by Random Forest ensembles learning methods. The simplicity and speed of implementation of the methods presented in the paper, as well as the size of the training set over which the evaluation models were performed, are acceptable for performing operations on minimal computing resources.

REFERENCES

[1] "U.S. Environmental Protection Agency," [Online]. Available at: https://www.epa.gov. [Accessed 20 5 2021].

[2] "California Air Resources Board," [Online]. Available at: https://ww2.arb.ca.gov. [Accessed 20 5 2021].

[3] "Society of Automotive Engineer," [Online]. Available at: https://www.sae.org. [Accessed 21 5 2021].

[4] "OBDII "SAE J1979"," [Online]. Available at: sae.org/standards/content/j1979_201702. [Accessed 23 5 2021].

[5] "CAN „ISO 11898“," [Online]. Available at: https://www.iso.org/standard/63648.html. [Accessed 24 5 2021].

[6] "CAN-bus dataset," [Online]. Available at: https://github.com/Dejan-Cugalj/CANBUS_dataset. [Accessed 30 5 2021].

[7] Altman and Krzywinski, "Simple linear regression. Nat Methods 12," 2015, p. 999–1000.

[8] "Višestruka Linearna Regresija," [Online]. Available at: http://www.matf.bg.ac.rs/p/files/69-Visestruka1.html. [Accessed 27 5 2021].

[9] B. L, "Random Forests. Machine Learning. vol 45," https://doi.org/10.1023/A:1010933404324, 2001, p. 5–32.