Proceedings of the 11th International Conference on Applied Informatics
Eger, Hungary, January 29–31, 2020, published at http://ceur-ws.org

# Data Mining and Analysis for Data From Vehicles Based on the OBDII Standard

**Balázs Bánhelyi, Tamás Szabó**

University of Szeged, Hungary
banhelyi@inf.u-szeged.hu

**Abstract**

Today every new car has an OBDII (On Board Diagnostic II) port that can be used to retrieve vehicle diagnostic data using an ELM327 or STN1110 chip. This microcontroller can be used to determine the currently measured parameters of the vehicle, such as speed, engine and water temperature, battery charge level, and error codes for fault detection. Our research aimed at developing an application and an algorithm for limited HW resources that performs the relevant analysis of the collected data and produces statistics on whether the currently measured value is within the suitable range. Since the algorithm is executed and the data is stored on a mobile phone, it is impossible to store and analyze all measured values. By examining the different readings (if they are alarming several times in a row) it would be possible to warn the user that there may be a problem with the vehicle. By monitoring the data, it would be possible to reduce the probability of major faults occurring and provide information about the occurrence of the fault.

*Keywords:* OBD II, Fault Detection, Confidence Interval

*MSC:* 97K80 Applied statistics



## Introduction

On-board diagnostics (OBD) is an automotive term that refers to the self-diagnosis and reporting capability[11]. In the development of OBD, the problems caused by periodic emission monitoring played an important role. CARB (California Air Resources Board) has recognized this and made continuous monitoring compulsory for manufacturers.


Figure 1: Interface scanner with ELM327 chip

The Onboard Diagnostic System (called OBDI) became obligatory in the USA in 1988. The technical specifications were defined by SAE (Society of Automobile Engineers) standards and recommendations. In 1994 OBDI was replaced by OBDII, and from 1996 it was also mandatory for diesel vehicles.

EOBD is the European equivalent of OBDII, which had to be introduced in the member states of the European Union by Directive 98/69/ EC.

A large amount of diagnostic information about the car can be obtained via OBD. To analyze this information, a simple and cheap ELM327 chip was used based on OBD interfaces (Figure 1), which is powered by the Microchip Technology PIC18F2480 Micro Controller. Newer devices use the STN1110 chip, which is fully compatible with ELM. ELM is a preprogrammed microcontroller, and the ELM327 Command Protocol is one of the most popular PC -OBD interfaces.

Many applications process OBD signals and have used Android mobile phones for that[2, 7]. Many applications display basic information for the car owner, and there are many experiments to process this data for its original purpose. For example, some developers try to retrieve information that is specific to the driver[4–6], or the vehicle state[9].

Some researchers are also working on automatic error detection. Most of them look at a single data series. In particular, cases when data that are outside the expected values are reported to the user as fail[3, 13]. Another new feature is that changes in the linear relationship between certain parameters are detected. If these values change, that indicates a message. This can also be used for special features based on preliminary tests where they have had good results[1].

In contrast, our research aimed at the development of a fully automatic fail detection system. The detection was based on rare data. The good vehicle works with typical values for a long time. During this time the correct working values were recorded, but a failed vehicle usually showed values that were not usual. Our algorithm tries to find automatically the rare data and the parameters indicating the failure.

# Android Application

An Android application was written that communicates with the ELM interface via Bluetooth. On the main screen, you can see the data of the real-time measurement


Figure 2: Main screen

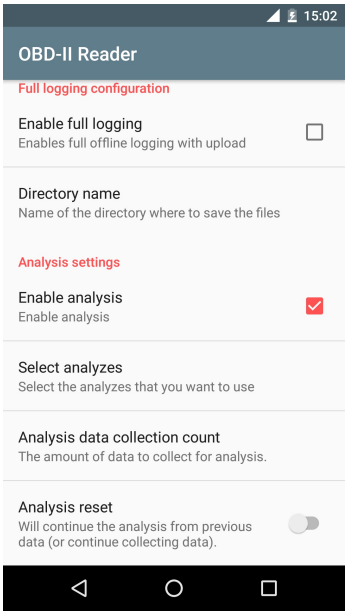


Figure 3: Setting

(see Figure 2). Benchmarks can also be performed like Engine Load, Throttle Position, Engine Coolant Temperature, Air Intake Temperature, etc.

Under settings (Figure 3) further details can be set, e.g. the parameters to be measured. You can do this under OBD commands. Under OBD protocol one can select the most frequently used protocols. Supported protocols:

- SAE_J1850_PWM
- SAE_J1850_VPW
- ISO_9141_2
- ISO_14230_4_KWP
- ISO_14230_4_KWP_FAST
- ISO_15765_4_CAN(_B, _C,_D)
- SAE_J1939_CAN

The ratio of successful communication of interactions between interface and Android phone was investigated. As it can be seen in Table 1, that below 800ms too much data are lost. The conclusion is that we can communicate enough data with a density of 1 second.

Data collection on mobile phones is fast enough, but still it is impossible to process this amount of data in the long run on the resources of the phones. However,

| communication interval | 1 data | 5 data | 20 data |
|---|---|---|---|
| 100 ms | - | - | - |
| 300 ms | 6/10 | 4/10 | 3/10 |
| 800 ms | 10/10 | 9/10 | 8/10 |
| 1000 ms | 10/10 | 10/10 | 9/10 |

Table 1: Number of useful data from 10 communication.

there is a growing need for data that does not leave the owner's devices. In the following section, we recommend a procedure that does not require data to be recorded but uses more recent data to detect errors more accurately.

# Statistical fault detection

The data processing is performed with a confidence interval analysis. The confidence ellipsoid is calculated for the data with a normal distribution [8]. Previously in our publications, this was used for conditions of optimization problems for non-independent variables. In the conditions, a better estimate was given of the co-occurrence probability of non-independent variables. Predicting rare events in the case of a related variable is useful[10].

To calculate the Confidence Ellipsoid, the sum of the measured data ($\sum y_1$), a sum of the square of the data ($\sum y_1^2$) and a sum of the product ($\sum y_1 y_2$) must be collected. This information can be calculated with the following methods, where the new data is $y_1^{(n+1)}$ and $y_2^{(n+1)}$:

$$\sum_{i=1\ldots n+1} y_1^{(i)} = \sum_{i=1\ldots n} y_1^{(i)} + y_1^{(n+1)}$$

$$\sum_{i=1\ldots n+1} \left(y_1^{(i)}\right)^2 = \sum_{i=1\ldots n} \left(y_1^{(i)}\right)^2 + \left(y_1^{(n+1)}\right)^2$$

$$\sum_{i=1\ldots n+1} \left(y_1^{(i)} y_2^{(i)}\right) = \sum_{i=1\ldots n} \left(y_1^{(i)} y_2^{(i)}\right) + y_1^{(n+1)} y_2^{(n+1)}.$$

This information can be calculated without storing the data. In our application, this information is stored in high-precision variables to process a long series of data (annual data). Our application used the Java classes BigInteger and BigDecimal classes[12]. We always count against the previous data.

From this information, the covariance matrix can be calculated for all data pairs:

$$cov_{y_1 y_2} = \begin{pmatrix} \frac{\sum y_1^2 - 2\frac{\sum y_1}{n}\sum y_1 + n\frac{\sum y_1}{n}\frac{\sum y_1}{n}}{n-1} & \frac{\sum y_1 y_2 - \frac{\sum y_1}{n}\sum y_2 - \frac{\sum y_2}{n}\sum y_1 + n\frac{\sum y_1}{n}\frac{\sum y_2}{n}}{n-1} \\ \frac{\sum y_1 y_2 - \frac{\sum y_1}{n}\sum y_2 - \frac{\sum y_2}{n}\sum y_1 + n\frac{\sum y_1}{n}\frac{\sum y_2}{n}}{n-1} & \frac{\sum y_2^2 - 2\frac{\sum y_2}{n}\sum y_2 + n\frac{\sum y_2}{n}\frac{\sum y_2}{n}}{n-1}. \end{pmatrix}$$

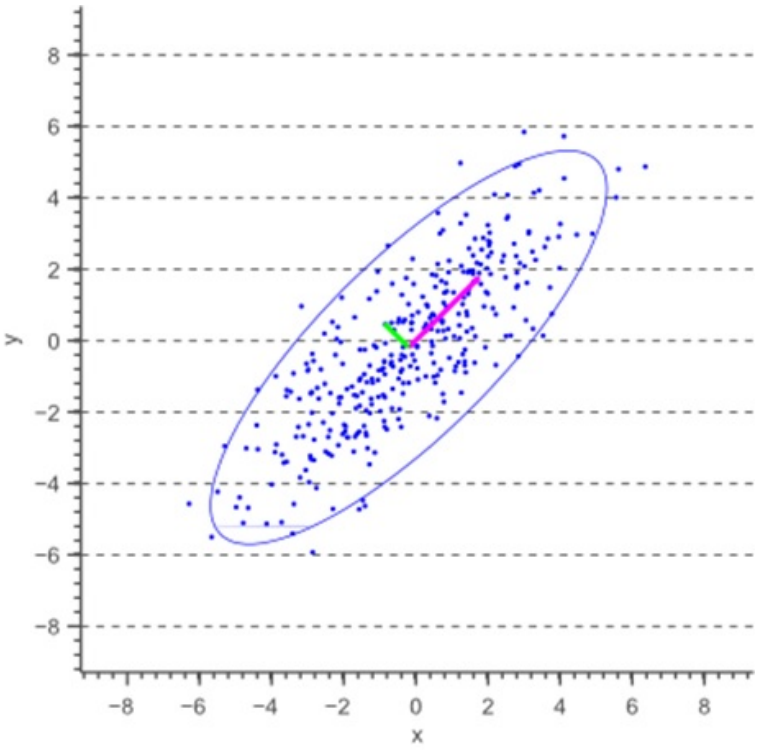


Figure 4: Measured data with confidence ellipsoid and eigenvectors

The eigenvalues and eigenvectors determine the confidence ellipsoids for the normal values with a different confidence level. To calculate the confidence interval of the data, the data was transformed into a normal distribution with expected values of zero. The transformation whitening matrix is

$$W = V\sqrt{D},$$

where $D$ is a diagonal matrix of eigenvalues and the $V$ matrix is the one whose columns are the corresponding right eigenvectors of the covariance matrix. This was illustrated for two measured data sets in Figure 4.

This matrix can be calculated from previous information. If the following condition applies to the new pair of values: $\left(y_1^{(n)}, y_2^{(n)}\right)$, then the measured data is considered to be not rare.

$$\left\| W^{-1} \left( \left(y_1^{(n)}, y_2^{(n)}\right) - \left( \frac{\sum y_1}{n}, \frac{\sum y_2}{n} \right) \right) \right\| > c,$$

where $c$ is the confidence level.

If the new values are rare, the users can be warned. After a certain large number of warnings, the application displays errors. In other cases, the new values

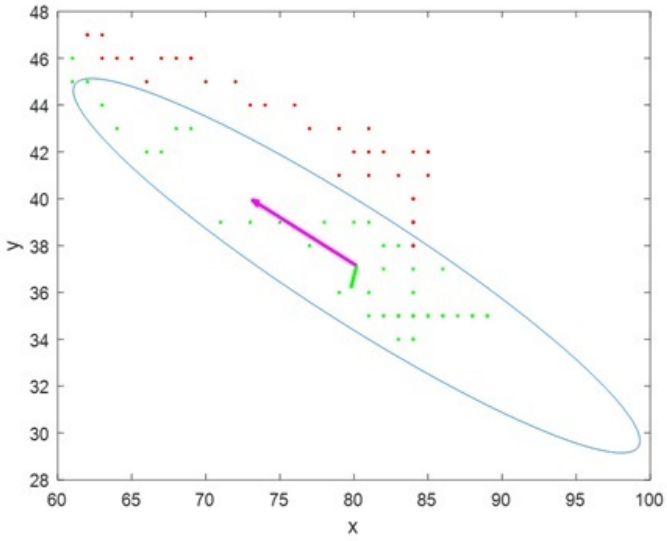


Figure 5: Illustration of measured data with cooling error

are inserted into the previously calculated information to improve the statistics of the rare values.

# Results

We have completed further measurements with data such as Vehicle Speed, Engine RPM, Engine Coolant Temperature, Fuel Pressure, Engine Load, Throttle Position, and Air/Fuel Ratio.

We have generated some faulty operation, for example, a bad cooling system, an error in the power source or a poor fuel supply.

In these measurements we found that although the data showed mean values, when we looked at 2 data series we could already find pairs of variables where the data were outside the expected range.

In one case the power source for the cooling system was switched off. The measured data of the cooling water and engine compartment temperatures are shown in Figure 5. The cooling water temperature is on the horizontal axis, while the engine compartment temperatures are on the vertical axis. The green dots illustrate the proper operation and the red dots are measured under the cooling failure system. The values for both sets are in the normal range, but in the worst cases they will be shifted to the warmer range. It can be seen that the majority of the values measured for a defective vehicle were outside the 95% confidence interval, while in the good case they are mostly inside.

The other faulty cases had similar results. Unfortunately, it is not trivial which

of the two data sets are useful for monitoring. It is a good idea to keep an eye on all of them, especially when other untested bugs are likely to be helpful.

# Conclusions

We have developed a system that is capable to detect rare data. If it appears, the users can be alerted. In the future, we would like to expand the application. We also plan to generate more errors in cars and investigate whether higher dimensions lead to better results. The long-term goal is to create a uniform database for comparing vehicles based on these results.

**Acknowledgements.** This research was supported by the projects "Extending the activities of the HU-MATHS-IN Hungarian Industrial and Innovation Mathematical Service Network" EFOP-3.6.2-16-2017-00015, the János Bolyai Research Scholarship of the Hungarian Academy of Sciences, and the Unkp-19-4-Bolyai+ New National Excellence Program of the Ministry of Human Capacities. Special thanks to Tamás Radványi for the implementation and running the related computer programs, who was supported by the "Integrated program for training new generation of scientists in the fields of computer science", EFOP-3.6.3-VEKOP-16-2017-0002.

# References

[1] Bandara, D., Amarasinghe, M., Kottegoda, S., et al.: *Cloud-Based Driver Monitoring and Vehicle Diagnostic with OBD2 Telematics*, in: vol. 6, Aug. 2015, doi: 10.4018/IJHCR.2015100104.

[2] Čabala, M., Gamec, J.: *Wireless Real-Time Vehicle Monitoring Based on Android Mobile Device*, Acta Electrotechnica et Informatica 12 (Jan. 2012), doi: 10.2478/v10198-012-0039-x.

[3] Grimaldi, C., Mariani, F.: *OBD Engine Fault Detection Using a Neural Approach*, in: Mar. 2001, doi: 10.4271/2001-01-0559.

[4] Hwang, C.-p., Chen, M.-S., Shih, C.-M., Chen, H.-Y., Liu, W.: *Apply Scikit-Learn in Python to Analyze Driver Behavior Based on OBD Data*, in: May 2018, pp. 636–639, doi: 10.1109/WAINA.2018.00159.

[5] Jang, J.-W., Yu, Y.: *A Study on In-Vehicle Diagnosis System using OBD-II with Navigation* (Jan. 2020).

[6] Jang, W., Jong, D., Lee, D.: *Methodology to improve driving habits by optimizing the in-vehicle data extracted from OBDII using genetic algorithm*, in: Jan. 2016, pp. 313–316, doi: 10.1109/BIGCOMP.2016.7425936.

[7] Kalmeshwar, M., Prasad, K.: *Development of On-Board Diagnostics for Car and it's Integration with Android Mobile*, in: Dec. 2017, pp. 1–6, doi: 10.1109/CSITSS.2017.8447540.

[8] Lefever, D.: *Measuring Geographic Concentration by Means of the Standard Deviational Ellipse*, American Journal of Sociology 32.1 (1926), pp. 88–94.

[9] Moniaga, J. V., Manalu, S. R., Hadipurnawan, D. A., Sahidi, F.: *Diagnostics vehicle's condition using obd-ii and raspberry pi technology: study literature*, Journal of Physics: Conference Series 978 (Mar. 2018), p. 012011, DOI: 10.1088/1742-6596/978/1/012011, URL: https://doi.org/10.1088%2F1742-6596%2F978%2F1%2F012011.

[10] Neumaier, A., Fuchs, M., Dolejsi, E., et al.: *Application of clouds for modeling uncertainties in robust space system design*, tech. rep., ARIADNA Study 05/5201, European Space Agency (ESA), 2007.

[11] *OBD II - On-Board Diagnostic System*, URL: http://www.obdii.com/.

[12] *Oracle java documentation*, URL: https://docs.oracle.com/javase/7/docs/api/java/lang/Number.html.

[13] *PCMSCAN-Palmer Performance Engineering*, URL: https://www.palmerperformance.com.