

Document downloaded from:

<http://hdl.handle.net/10251/67206>

This paper must be cited as:

Meseguer Anastasio, JE.; Tavares De Araujo Cesariny Calafate, CM.; Cano Escribá, JC.; Manzoni, P. (2013). DrivingStyles: a smartphone application to assess driver behavior. 18th IEEE symposium on Computers and Communications (ISCC 2013). IEEE.
doi:10.1109/ISCC.2013.6755001.



The final publication is available at

<http://dx.doi.org/10.1109/ISCC.2013.6755001>

Copyright IEEE

Additional Information

©2013 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

DrivingStyles: a smartphone application to assess driver behavior

Javier E. Meseguer, Carlos T. Calafate, Juan Carlos Cano, Pietro Manzoni
Department of Computer Engineering
Universitat Politècnica de València
Camino de Vera S/N, 46022, Spain
jmesegue@upvnet.upv.es, {calafate, jucano, pmanzoni}@disca.upv.es

Abstract—The DrivingStyles architecture integrates both *data mining* techniques and *neural networks* to generate a classification of driving styles by analyzing the driver behavior along each route. In particular, based on parameters such as speed, acceleration, and revolutions per minute of the engine (rpm), we have implemented a neural network based algorithm that is able to characterize the type of road on which the vehicle is moving, as well as the degree of aggressiveness of each driver. The final goal is to assist drivers at correcting the bad habits in their driving behavior, while offering helpful tips to improve fuel economy.

In this work we take advantage of two key-points: the evolution of mobile terminals and the availability of a standard interface to access car data. Our DrivingStyles platform to achieve a symbiosis between smartphones and vehicles able to make the former operate as an onboard unit. Results show that neural networks were able to achieve a high degree of exactitude at classifying both road and driver types based on user traces. DrivingStyles is currently available on the Google Play Store platform for free download, and has achieved more than 1550 downloads from different countries in just a few months.

Index Terms—Driving styles; Android smartphone; OBD-II; neural networks; eco-driving.

I. INTRODUCTION

Mobile devices have experienced a technological breakthrough in recent years, evolving towards high performance terminals with multi-core microprocessors, being smartphones a clear representative exponent of this trend. In addition, the On Board Diagnostics (OBD-II) [1] standard, available since 1994, has recently become an enabling technology for in-vehicle applications due to the available of Bluetooth OBD-II connectors [2]. These connectors enable a transparent connectivity between the mobile device and the vehicle's Electronic Control Unit (ECU).

The range of possibilities that arise when combining cars and smartphones is endless, allowing, for example, diagnosing the car via mobile devices which assume the tasks that are typically performed by the On Board Unit (OBU) of the vehicle, or sending the collected data to a platform where diagnosis and vehicle maintenance can be done, detecting possible failures automatically.

Based on obtained data about the speed, acceleration, and revolutions per minute of the engine, we have implemented our DrivingStyles platform, which is based on neural networks and permits characterizing the driving style of each user, as well as

the type of road on which the vehicle is circulating. Currently, this information is being collected and used in applications aimed at improving road safety and promoting eco-driving [3], thus reducing fuel consumption and greenhouse gas emissions. The given recommendations addressing the driving style of each user can save up to 20% of fuel while improving driving safety.

The rest of this paper is organized as follow: Section II briefly reviews previous related works. Then, Section III provides an overall overview of the DrivingStyles architecture. Both client and server side applications are described in more detail in sections IV and V, respectively. The neural network we have developed, along with the achieved accuracy results, are presented in section VI. Finally, section VII presents the conclusions of our work.

II. RELATED WORK

Technological achievements in the field of mobile are making smartphones very powerful devices. In the research world, this high computing power opens new and attractive research areas where many of them are able to generate economic profit and significant advantages for our society.

One of the main problems of eco-driving systems [3] is identifying the factors that affect energy consumption. Ericsson [4] suggests that, in order to save fuel, sudden changes in acceleration and high speed driving should be avoided. Johansson et al. [5] suggest maintaining low levels of deceleration, minimizing the use of the first and second gears, and putting every effort into using the 5th and 6th gears, while avoiding continuous gear changes.

There are several proposals that analyze which variables affect fuel consumption. Kuhler [6] presented a set of ten variables. These variables are used in laboratories that work with the consumption of fuel and gas emissions. Other authors, like André [7] and Fomunung [8], increased the number of parameters, or replaced some of them in order to improve the obtained results. The problem of all the previous proposals is that the environment in which the vehicle is circulating is not considered, despite it has a very significant influence on the energy consumption.

Regarding other works related to the proposal discussed in this paper, it is worth highlighting the work by Chen et al.

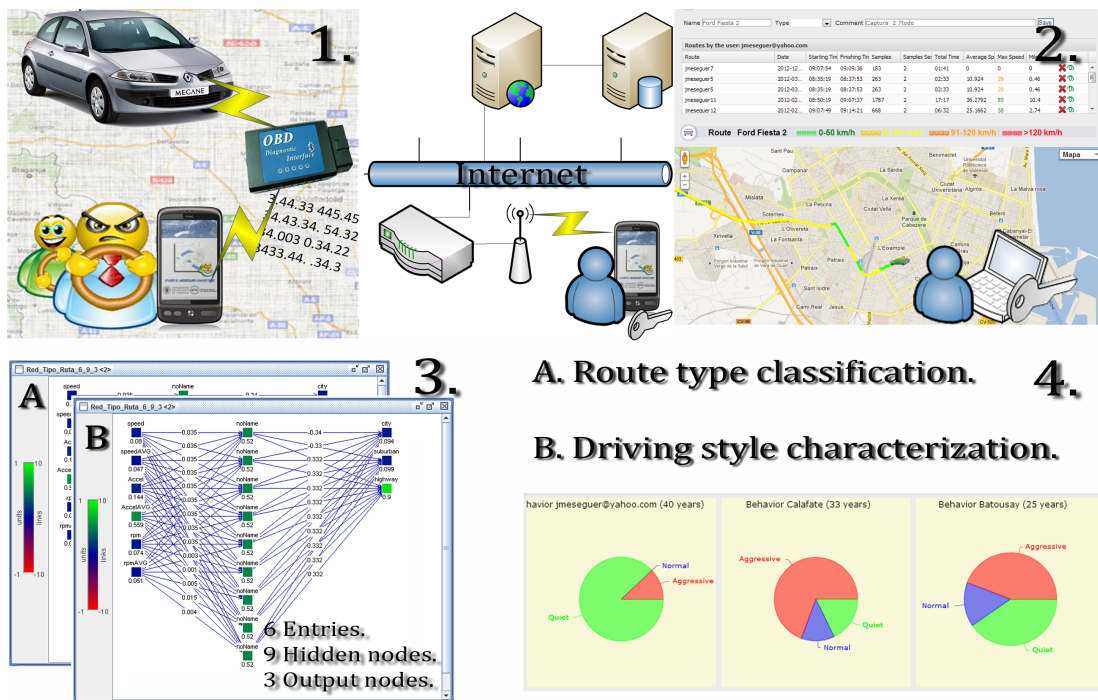


Figure 1: DrivingStyles: architecture overview.

[9], who proposed an Android/OSGi vehicular platform able to diagnose and manage the status of a vehicle remotely, using visual intelligence to continuously update their application services based on context, and without user intervention. Experiments conducted on a test vehicle showed that Android/OSGi applications are able to achieve higher performance compared to a pure Android development, especially when complex operations must be performed.

Focusing on road safety, systems such as e-NOTIFY [10] allow a fast detection of traffic accidents, improving the assistance to injured passengers by reducing the response time of emergency services. The proposed system requires installing OBUs on board vehicles that are in charge of detecting accidents, as well as notifying them to external control units (CU); the latter estimates the severity of the accident and automatically informs the emergency services. Zaldivar et al. [11] propose an Android-based application that monitors the vehicle through the On Board Diagnostics (OBD-II) interface, being able to detect accidents and sending details about the accident to pre-defined destinations through either e-mail or SMS; these tasks are immediately followed by an automatic phone call to the emergency services.

Our solution differs from the former ones by providing an analysis of the driving style of each user in the scope of urban, suburban, or highway scenarios, which is based on neural network techniques, and promotes a more responsible and eco-driving behavior.

III. DRIVINGSTYLES ARCHITECTURE

The proposed architecture applies data mining techniques to generate a classification of the driving styles of users based

on the analysis of their mobility traces. Such classification is generated taking into consideration the characteristics of each route, such as whether it is urban, suburban or highway.

To achieve the overall objective, the system is structured around the following four elements, see Figure 1:

- 1) An application for Android based smartphones. Using an OBD-II Bluetooth interface, the application collects information such as speed, acceleration, engine revolutions per minute, throttle position, and the vehicle's geographic position. After gathering this information in real time, the user uploads the route data to the remote data center for analysis.
- 2) A data center with a web interface able to collect large data sets sent by different users concurrently, and to graphically display a summary of the most relevant results. Our solution is based on open source software tools such as Apache, PHP and Joomla.
- 3) A neural network, which must be designed and trained using the most representative route traces in order to correctly identify, for each path segment, the driving style of the user, as well as identify the segment profile: urban, suburban or highway. To implement our neural network we adopted the backpropagation algorithm [16], which has been proven to provide good results in classification problems such as the one associated to this project.
- 4) Integration of the neural networks in the data center platform. The goal is to dynamically and automatically analyze user data, allowing users to find out their profiles as a driver, thus promoting a less aggressive and more ecological driving.

Figure 1 graphically shows the steps followed by users of the DrivingStyles platform to assess their driving styles. The first step is registering the user at <http://www.drivingstyles.info>, and downloading the free Android application.

After installing the Android application in the smartphone, and after connecting to the bluetooth ELM327 [2] interface inside the car (this connector is mandatory on all vehicles since 2001), the data acquisition process can start. Once the route is completed and stored, the user can send it to the server, which checks the username and password before accepting the incoming XML file. Through the web interface, the user can view all the routes sent to the server, and have access to the different statistics, the map of the route, the characterization of the driving style, and the recommendations on eco-driving.

IV. ANDROID APPLICATION

The Android application is a key element of the system, proving connectivity to the vehicle and to the DrivingStyles web platform. Currently, it can be downloaded for free from the DrivingStyles website <http://www.drivingstyles.info>, or from Google Play <https://play.google.com/store/apps/details?id=com.driving.styles>.

A. Configuration options

In order to adjust the functionality of our Android application to the user requirements, several configuration options must be defined related to user creation, connection options, GPS activation, and sensor sampling.

The available functionalities are the following:

- 1) **User creation:** A username and password matching those registered in the website must be introduced, allowing the server to validate the user credentials before storing any data in the database.
- 2) **OBD Connection:** Before starting the sampling process, the device must be paired with the OBD-II interface. Once the device is paired with the smartphone, it can be selected from the list of available devices within the application.
- 3) **GPS Activation:** although GPS data is not mandatory for data capturing purposes, it is necessary in the broader scope where the study is being done, since the variables retrieved from the vehicle are recorded along with the car's geographical position for later analysis.
- 4) **Sensor sampling:** The user can individually select the vehicle sensors used by the application for monitoring and storage among the following ones: acceleration, engine revolutions per minute (rpm), throttle position, and instantaneous speed. Additionally, the user can define the sampling rate (number of samples per sensor, and per second).

B. Application Modules

The main module of our application launches the background processes responsible for capturing data sent by the OBD-II and the GPS interfaces, as well as the phone's accelerometer.



Figure 2: Snapshots of the acceleration, speed, rpm parameters and map module.

Besides showing the sensors we are monitoring, we can perform several parallel actions without affecting the data capture. Possible options are:

- Start and stop data capture of the Electronic Control Unit (ECU), the mobile's accelerometer, and the GPS.
- List the routes captured by the app.
- Driving style display.
- Show the current position of the vehicle on the map, as well as the detailed route followed whenever an Internet connection is available.
- Real time visualization of the speed, rpm, and acceleration in a time window of 10 seconds.

The route upload module is in charge of sending the users' traces to the website data center for further analysis. This module can be accessed either from the historic stored routes, or immediately after stopping the data capture. The information screen displays the header information of the selected route such as: date of the captured data, start time, finish time, and maximum speed.

This module includes a graphical interface in charge of showing routes on a map, as well as the collected statistics. Additionally, it also includes communication facilities for uploading the collected routes to the data center.

Finally, the file received in the DrivingStyles data center is

stored in the corresponding user directory, and also generates a record in the database for every sample submitted for analysis.

The Map Module and Graphical Information are in charge of displaying the relevant information to the user in the most convenient manner. The graphs can be displayed in real-time or by selecting data from previously stored paths. Depending on the device model, the user can also zoom in and out to display all or part of the graph using the device's touch screen.

The charts that appear on the screen are the acceleration, the speed, and the revolutions per minute (rpm). We have chosen these three parameters since they are the most relevant ones, and because they are the ones we selected for training our neural network.

The map module allows displaying the GPS position on the map. GPS coordinates are drawn using the Google Maps APIs. A green car icon indicates the beginning of the route, and a red icon shows the current position of the vehicle. The path is shown by using different colors depending on the vehicle's speed (see Figure 2).

V. DRIVINGSTYLES WEB INTERFACE

The second main component of our architecture corresponds to the data center and its web interface. For this endeavor we have selected open source software such as Apache HTTP, and Joomla as the content management system (CMS). The URL of this module is <http://www.drivingstyles.info>.

Once the user is logged in, he is asked to record a number of important data, especially for future data mining studies. The most relevant items are sex, age, and other details concerning the vehicle used: car manufacturer, model, fuel type, and the theoretical 0-100 acceleration (important to normalize the user behavior in our study). Finally, a third block allows drivers to indicate what they feel about their own behavior behind the wheel, i.e., whether they perceive themselves as aggressive, moderate, or quiet drivers.

In the Routes' section, the users can access all the routes they have uploaded. The first grid shows routes that are in the database, including the name, date, starting time, ending time, samples sent, total time, average speed, and kilometers travelled. Below the grid, the selected route is shown in a map. The path varies its color depending on the speed of the car (see Figure 3).

In addition, the last two graphs show the results returned by the neural network, including the driving style and the route characteristics (see Figures 6b, and 6a).

Next, in section VI we provide detailed information about the neural network we proposed for characterizing routes and drivers.

VI. NEURAL NETWORKS BASED DATA ANALYSIS

In this project we face a classification problem: starting from some input data, which in our case are the speed, the acceleration, and the revolutions per minute of the engine (rpm), we obtain as output the type of road and the driving style.

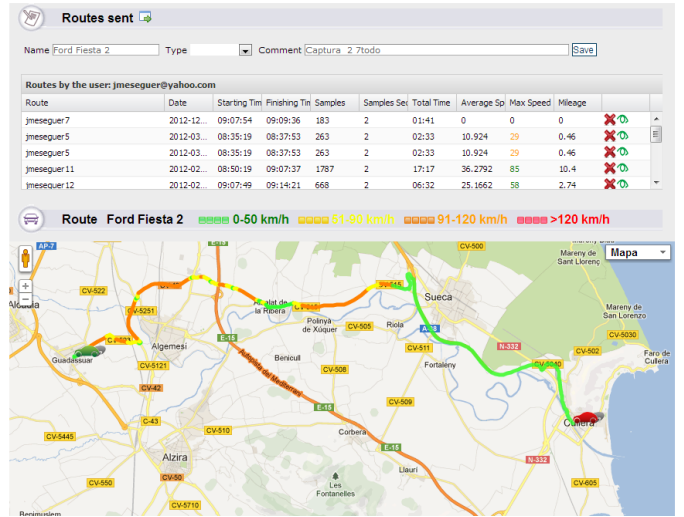


Figure 3: Snapshot of a route map.

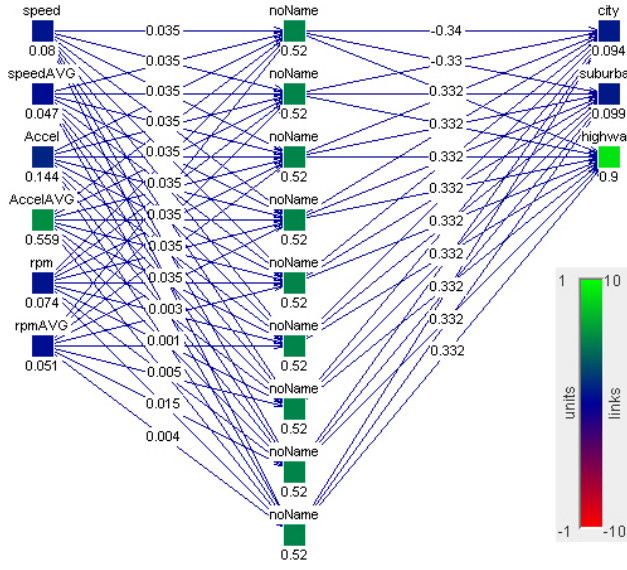
We train Artificial Neural Networks to classify the driving style of each user and the type of route based on a well-defined set of rules and the ECU input variables. There are many different learning algorithms, such as backprop_momentum, Hebbian, or delta-rule, each one having its own advantages and disadvantages depending on the type of problem to solve [14]. In the scope of this work, we decided to choose backpropagation [16] since this kind of algorithm provides very good results in classification problems.

After considering the many variables that can be obtained from the Electronic Control Unit (ECU), we have chosen to train the neural network using the mean and the standard deviation values of: a) the speed, b) the vehicle acceleration, and c) the rpm. In all vehicles used for testing, these variables were easily obtained, while other variables, such as the position of the throttle, and despite being able to provide important information for the neural network training, finally had to be rejected because not all ECU manufacturers provide such information. The data input of each parameter is normalized between 0 and 1; this normalization should take into consideration the whole range of possible values. The equation used to normalize the input parameters is the following:

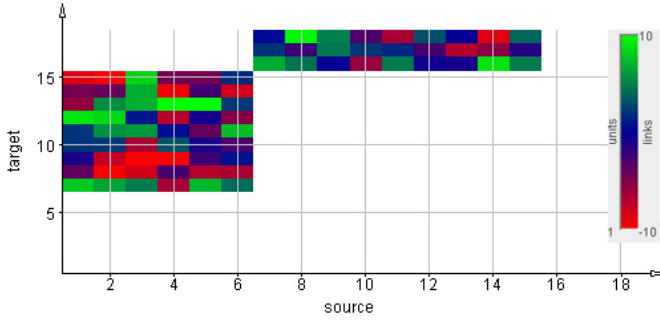
$$x^{p'} = \frac{x^p - \min(x)}{\max(x) - \min(x)}$$

Notice that $x^{p'}$ is the normalized value of input variable x for pattern p , x^p is the original value of variable x for that pattern, and $\min(x)$ and $\max(x)$ represent the minimum and maximum values for the input variable, i.e., the minimum and maximum values of the column corresponding to variable x .

The application used for the creation and training of the neural networks required by this project was JavaNNS [12], which is a Java version of the SNNs software developed at the University of Tübingen.



(a) Trained neural network.



(b) Weights of the links of the neural network.

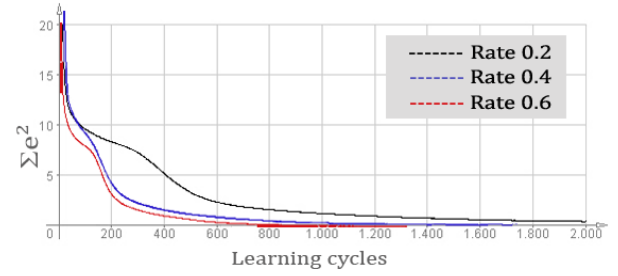
Figure 4: Trained neural network and associated weights.

A. Neural Network description

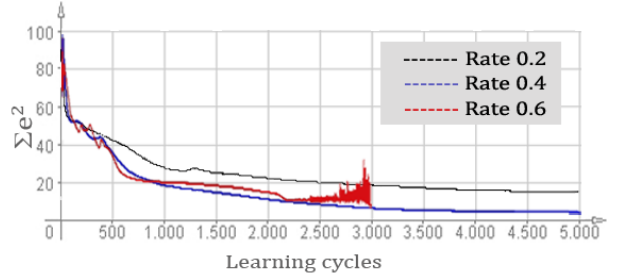
An empty neural network was created by defining the number of entries mentioned previously, along with the number of hidden nodes (nine in our case). A larger number of hidden nodes can improve the success rate, but it has the negative effect of increasing the response time. Two similar neural networks were created, each one with three output nodes; the first neural network (see Figure 4a) allows to classify the type of road (urban, suburban or highway), and the second permits to characterize the user's driving style (quiet, normal or aggressive). Random weights between -1 and 1 are assigned to the links of both neural networks during initialization. Figure 4b shows the equivalence color map with their weight values for the neural network identifying the road type (first one).

B. Training the Neural Network

After normalizing the input variables, the training set was generated. This data set is used to adjust the weights of the neural network. After the neural network is trained, we generate validation files. This data set is used to minimize



(a) Convergence when training the road type with 2000 cycles.



(b) Convergence when training the driving style with 5000 cycles.

Figure 5: Neural networks convergence behavior.

overfitting, verifying that any increase in accuracy over the training data set actually yields an increase in accuracy over a broader data set that has not been used for training.

In order to do so, we rely on the average error for the test set: $E = \frac{\sum_{p=1}^m |d^p - y^p|}{m}$ where $|d^p - y^p|$ is the absolute error for pattern p , i.e., the absolute value of the difference between the desired output and the output obtained, m is the number of patterns, and the sum corresponds to the sum of absolute errors for all patterns.

We started training the neural network by adjusting the learning rate to 0.2 and observing how the error affects our neural network. The higher the learning rate, the higher the modification of the weights for each iteration, so the learning process becomes fast. However, this can cause unwanted oscillations in the network. The convergence results of the two neural networks are the following:

- 1) Road type characterization: By iterating with a learning rate of 0.2, we see how the learning process slows down for a local minimum MSE with an error of 0.021. The network starts converging quickly again beyond 300 cycles, as shown is the mean square error shown in Figure 5a.
- 2) Driving style characterization: This characterization is more complex than the road type, and no local minimum is reached as quickly as in the previous case; also, the error of the neural network is much higher (see Figure 5b). At the end of these tests we chose to use the trained neural network with a learning rate of 0.4, which achieves the lowest error among all the tested values, with a mean square error of 0.43 and a sum of squared

errors of 6.11 (SSE).

Once the neural network is successfully trained (see Figure 4a), the knowledge obtained must be converted into usable code. We use the snns2.exe application [15], to convert the trained neural network into C code, and this code was then integrated in to the web platform for data classification purposes.

C. Obtained results

With the neural network already implemented, every time a route or route segment is selected, the system automatically returns the type of road, and the associated driving style. The overall behavior of each user can also be obtained by evaluating all the routes sent by the user.

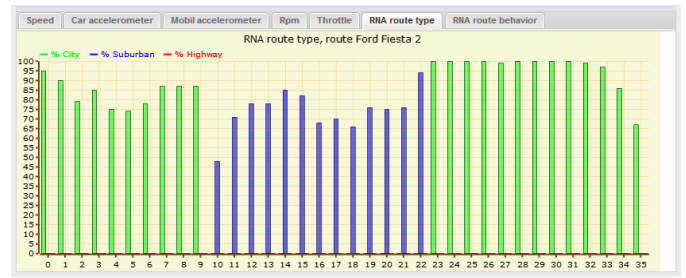
The results obtained from a real set of data can be seen in Figure 6. Figure 6a graphically shows the result returned by our neural network when determining the route type. The abscissa axis corresponds to the timeline, and the vertical axis to the degree of matching for the dominant route type. We can see that, in this example, the user starts and ends in an urban environment, going through a suburban route in between; such results represent an accurate classification of the actual route followed by the driver. Figure 6b refers to the neural network results that determine the behavior on that route; as in the previous graph, the abscissa axis corresponds to the temporal line, and the ordinate axis is a behavior correlation percentage, where each color identifies a different type of driver behavior. We can observe that most of the time the driver is showing an aggressive behavior, which alternates with periods of quiet behavior. When analyzing the actual route with the driver, we find that periods of quiet behavior were associated with slow transit areas where vehicle queueing and semaphores impeded a more aggressive driving style.

Overall, and after analyzing a wide set of routes with different drivers, the system has shown to provide a correct classification of the different route traces registered in the database, both in terms of route types and driving styles, validating the proposed solution.

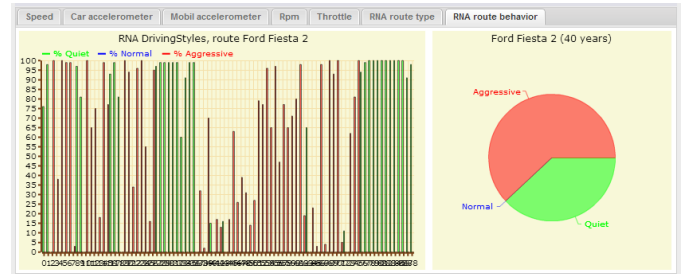
VII. CONCLUSIONS AND FUTURE WORK

In this paper we presented the DrivingStyles platform, which integrates mobile devices with data obtained from the Electronic Control Unit (ECU) to determine the type of road where the driver is circulating, as well as his driving habits. Its main goal is to help promoting a safer and more ecological driving style by making drivers more conscious about their behavior on the road. Our platform offers helpful tips to reduce fuel consumption with the consequent impact on the environment. We implemented this platform using real devices, and the results we obtained based on real user traces are quite encouraging, showing that the classification of both routes and driving styles using neural networks presents a high correlation with the actual routes and driver behavior.

The application, which is available for free download in the DrivingStyle's website and in the Google Play Store, has achieved more than 1550 downloads from different countries



(a) Route type.



(b) Route behavior.

Figure 6: Snapshots of route type and behavior.

in just a few months. This shows the great interest for applications integrating smartphones with vehicles.

As future work we plan to monitor the fuel consumption in order to correlate it with the driver aggressiveness so making the driver more aware of their bad habits while driving. We are also extending the platform to provide traffic recommendations based on real-time feedback about the congestion of different routes.

ACKNOWLEDGMENTS

This work was partially supported by the Ministerio de Ciencia e Innovación, Spain, under Grant TIN2011-27543-C03-01.

REFERENCES

- [1] International Organization for Standardization, "1999: Road vehicles, Diagnostic systems, Keyword Protocol 2000", 1999.
- [2] ELM327DS. OBD to RS232 Interpreter. Elm Electronics Circuits for the Hobbyist.
- [3] Co Eco-Driving: Pilot Evaluation of Driving Behavior Changes among U.S. Drivers.
- [4] E.Eriksson. Independent driving pattern factors and their influence on fuel-use and exhaust emission factors Transportation Research Part D: Transport, 2001 Elsevier, 325-345J.
- [5] Johansson, H., Gustafsson, P., Henke, M., Rosengren, M., 2003. Impact of EcoDriving on emissions. International Scientific Symposium on Transport and Air Pollution, Avignon, France.
- [6] Kuhler, M., Kartens, D., Improved driving cycle for testing automotive exhaust emissions. SAE Technical Paper Series 780650. 1978.
- [7] André, M., 1996. Driving cycles development: Characterization of the methods. SAE Technical Papers Series 961112.
- [8] Fomunung, I., Washington, S., Guensler, R. A statistical model for estimating oxides emissions from light duty motor vehicles. Transportation Research Part D, pp.333-352. 1999
- [9] M.-C. Chen, J.-L. Chen, and T.-W. Chang, Android/OSGi-based vehicular network management system, Elsevier Computer Communications, vol. 34, no. 2, pp. 169 183, 2011

- [10] Fogue, M. Garrido, P., Martinez, F.J., Cano, J., Calafate, C.T., Manzoni, P. Automatic Accident Detection: Assistance Through Communication Technologies and Vehicles. Vehicular Technology Magazine, IEEE, Sept. 2012.
- [11] Zaldivar, J., Calafate, C.T., Cano, J.C., Manzoni, P., Providing accident detection in vehicular networks through OBD-II devices and Android-based smartphones. Local Computer Networks (LCN), 2011 IEEE 36th Conference 4-7 Oct. 2011.
- [12] JavaNNS, Java Neural Network Simulator, User Manual, Version 1.1 Igor Fischer, Fabian Hennecke, Christian Bannes, Andreas Zell.
- [13] Haykin, S. (1994), Neural Networks: A Comprehensive Foundation, NY: Macmillan, p. 2.
- [14] Efficient Classification for Multiclass Problems Using Modular Neural Networks Rangachari Anand, Kishan Mehrotra, Chilukuri K. Mohan, and Sanjay Ranka, Member, IEEE
- [15] <http://www-ra.informatik.uni-tuebingen.de/SNNS/>
- [16] Hecht-Nielsen, R., Theory of the backpropagation neural network - Neural Networks, 1989. IJCNN., International Joint Conference.