Evaluation of OBDII data contribution in Tiny Machine Learning based Driving Behaviour Monitoring

Massimo Merenda[a,b], Vincenzo Mazzullo[a], Marco Princi[a]

[a]Department of Information Engineering, Infrastructures and Sustainable Energy (DIIES), University Mediterranea of Reggio Calabria

[b]HWA s.r.l.-Spin Off of the University Mediterranea of Reggio Calabria

Motivations

More than 90% of light vehicle accidents are caused by **human driver misbehavior** only in the United States.
The context awareness of the driver's behavior could alert about accidents, infrastructures emergency or malfunctioning.
The accident rate can be reduced by 10% to 20% with a precise driver behavior monitoring system.

Applications

Some applications of driver behavior monitoring system:

- **Smart Cities** (Figure 1)
- **Insurance** (Figure 2)
- **Car rental providers**(Figure 3)

Figure 1: Smart Cities.

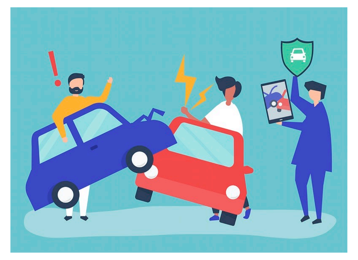

Figure 2: Insurance.

Figure 3: Car rental providers.

Tiny Machine Learning

In our work, we use data from accelerometers, gyroscopes and OBDII port to feed a **Convolutional Neural Network (CNN)** able to infer behavior classification among 4 classes:

1. Stationary vehicle
2. Safe driving behavior
3. Normal driving behavior
4. Dangerous driving behavior

Development of a **Machine Learning (ML)** application that is able to run on **tiny hardware** with limited resource.

Hardware

The first dataset creation (without OBDII data) from real sensors data and real-time inference of machine learning model have been obtained using the development board **B-L475E-IOT01A2** by STMicroelectronics. It features an ultra-low power STM32L4 Series MCU based on Arm® Cortex®-M4 core, with 80 MHz clock, 1 MByte flash memory and 128 kByte SRAM. See Figure 4.
The second dataset (with OBDII data) was created using data from motion sensors and data gathered from the vehicle's on-board diagnostic.
The application acquired data from the OBDII port using an **HH OBD Advanced** device. See Figure 5.

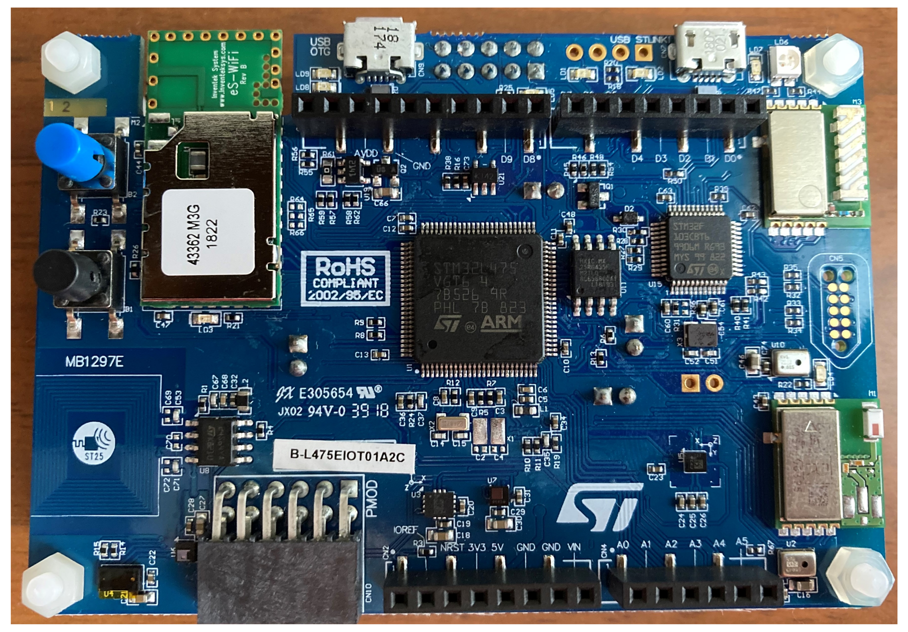

Figure 4: B-L475E-IOT01A2 by ST Microelectronics used to develop this project.

Figure 5: HH OBD Advanced.

JMS Dataset and OBDII Dataset

Two datasets were created, JMS (Just Motion Sensors) dataset and OBDII dataset.
The dataset consisted of 4 subsets of data captured separately and finally merged into a single csv file. The dataset was preprocessed to have 6 columns (3D Gyroscope and 3D Accelerometer) and a number of rows multiple of 20, after which it is scaled, permuted and split.
Figure 6 shows the flowchart of the dataset creation process.

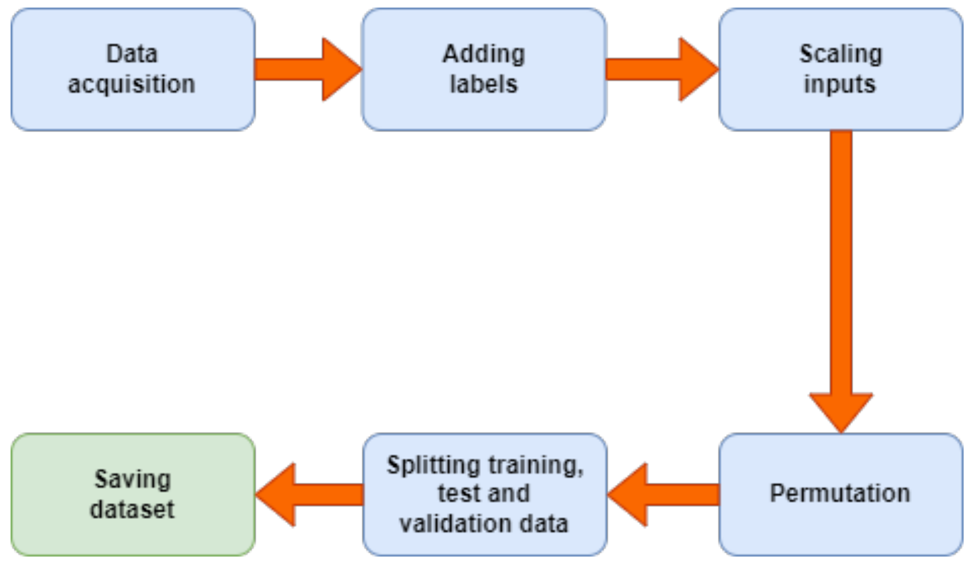

Figure 6: Flowchart of the dataset creation process.

An **JMS dataset** was created using data from accelerometer and gyroscope, A **OBDII dataset** was created with data captured in vehicle using motion sensors and OBDII port.
Both the datasets were obtained testing 4 different driving styles to which the labels were assigned.
Figure 7 shows the setup used for onboard data acquisition to create datasets.

Figure 7: Setup used for onboard data acquisition to create datasets.

CNN

CNN is a Deep Neural Network and has proven to be one of the most reliable solutions to Time Series Classification problems.
To classify the various driving styles, the solution that proved to be the most effective and efficient was CNN.
With the use of convolution, it is possible to highlight important characteristics of the input data. Filters with an adequate kernel size are used to perform the convolution operation.
The structure of the CNN model has one-dimensional convolutional layers that alternate with "Average Pooling 1D" layers and with "Dropout" layers, subsequently, a "Global Average Pooling 1D" layer is present and ends with three "Dense" layers in cascade. The filters used in convolutional layers are 18, 30, 62, respectively. Convolution layer filters have a kernel size of 5. Filters used in Dense layers are 64, 32, 16.
Figure 8 shows the structure of the CNN model.

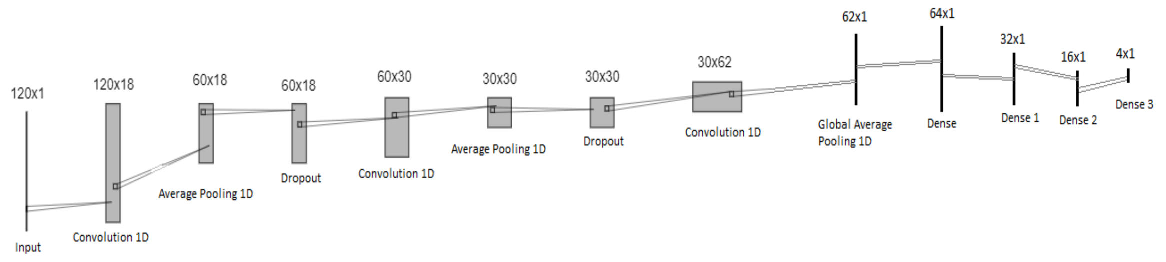

Figure 8: Structure of the CNN model.

Implementation using MCU

For the implementation of the model, the software STM32CubeIDE by STMicroelectronics was used, with the Software Pack for Artificial Intelligence "X-CUBE-AI" version 6.0.0.
An effective algorithm was developed allowing to obtain an evaluation of the driver's driving style based on output values acquired every 2 seconds. See Figure 9.

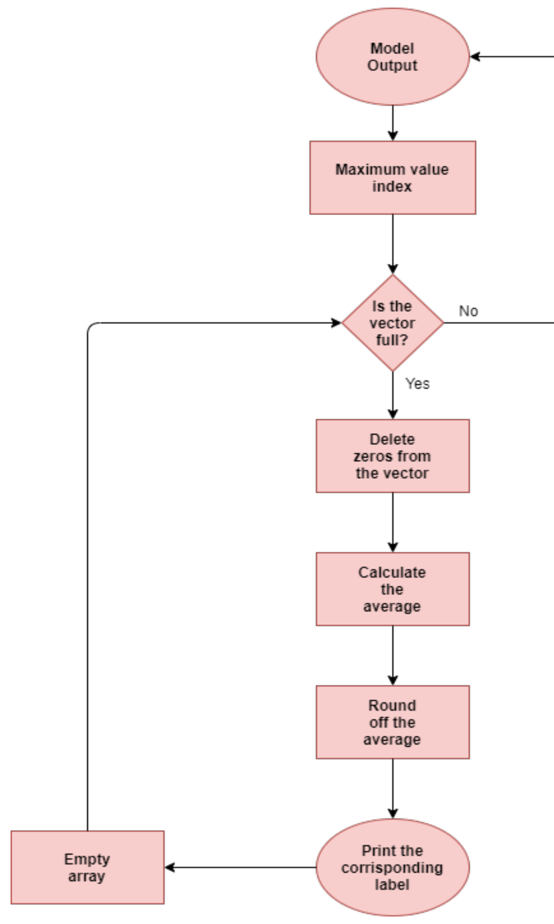

Figure 9: Flowchart of the algorithm.

Testing and Results

A first phase of testing was performed on the model using Python on Desktop environment.
Results using JMS dataset: 86% of accuracy. (Figures 10, 11)

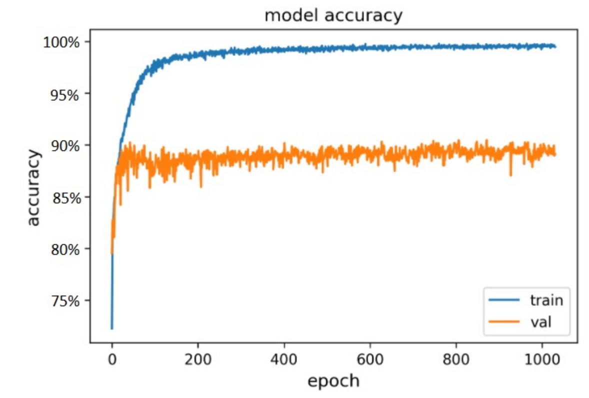

Figure 10: Accuracy trend (JMS dataset)

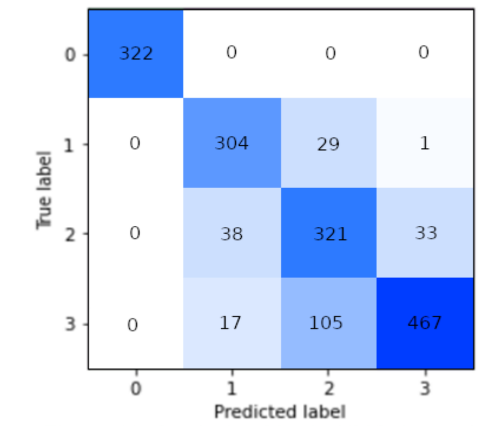

Figure 11: Confusion Matrix (JMS dataset).

Results using OBDII dataset: 96% of accuracy. (Figures 12, 13)

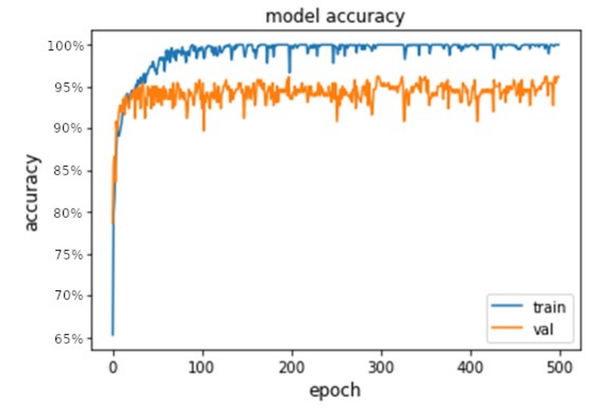

Figure 12: Accuracy trend (OBDII dataset).

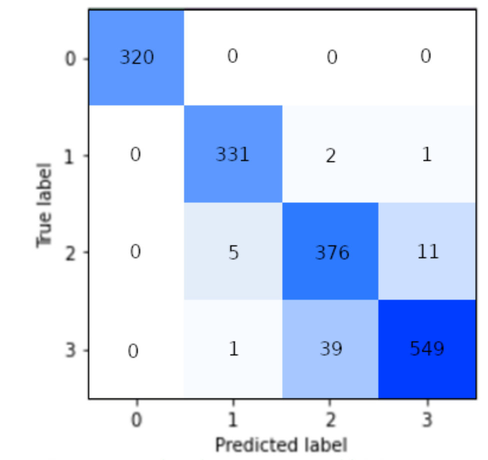

Figure 13: Confusion Matrix (OBDII dataset).

Conclusion

A system based on the use of an MCU and artificial intelligence was designed.
This system is able to classify the driver behavior using the sensors installed on the board (accelerometer and gyroscope) and OBDII port.
The system can be useful to reduce the number of road accidents and improve the driving experience by making it safer and more comfortable.
Many applications and services within smart cities, insurances and car rental providers can benefit of the presented system.