Traffic Data Analysis for Modelling and Prediction of Traffic Scenario

Oddo Girolamo[1]

April 2022

Abstract: With the development of self-driving vehicles models for operating in traffic and complex situations are becoming increasingly necessary. In this paper, several ML and DL models have been implemented in order to obtain predictions of vehicle traffic behaviour. An application method of the obtained models based on V2V (Vehicle-to-Vehicle) connectivity has been proposed.

Keywords: Big Data, Autonomous Vehicles, Machine Learning, Deep Learning, Automotive

1. Introduction

With the development of autonomous driving the interest in the development of models capable of predicting the behaviour of vehicles in traffic is growing in order to increase the reliability of driving and the understanding that the vehicle has of the surrounding context. In this work we will try to propose a solution by exploiting Machine Learning and Deep Learning.
This work started as a project for the MathWorks Excellence in Innovation initiative, where the aim is to predict the behaviour of cars in traffic. The dataset "Dataset of Annotated Car Trajectories (DACT)-SRICT ANNOTATION"[1] will be used as a reference from which the classification models will be developed. All models will be developed entirely in MATLAB R2021a using the Deep Learning Toolbox™ and Statistics and Machine Learning Toolbox™ packages. Two solving approaches will be followed, one according to models based on Decision Tree and one according to models based on Neural Networks of different architectures and dimensions. In particular, solutions with Ensamble learning techniques and Fully Connected network architectures with more layers and with the presence of SOM layers and NARX Network will be evaluated. After obtaining the reference model, an application method will be proposed, exploiting the Vehicle-to-Vehicle connectivity in order to predict the behaviour of neighbouring vehicles and therefore decide the behaviour to follow.

2. Dataset presentation

The dataset used for this work is the "Dataset of Annotated Car Trajectories (DACT)-SRICT ANNOTATION", which has a dimension of 47847x11. For the purposes of this work, the dataset will be divided into two macro areas, one for the Training phase and one for the Testing phase, with a 70:30 ratio.
The following will be used as predictors:

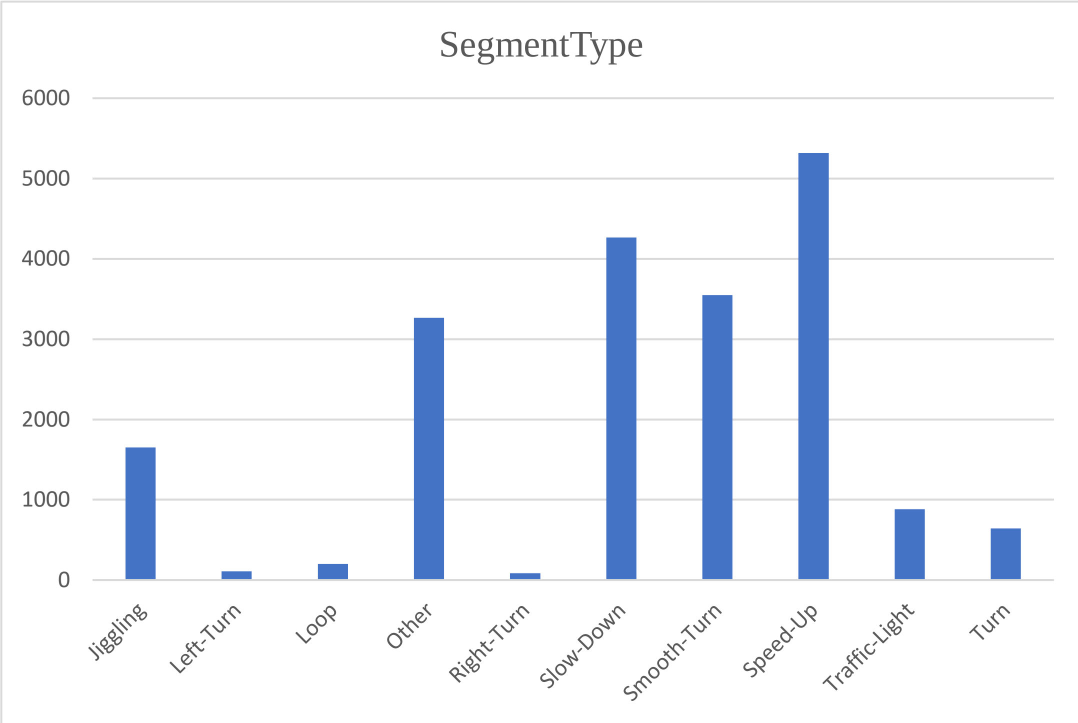
Speed	Acceleration	Heading	HeadingChange

As target classes you will have:

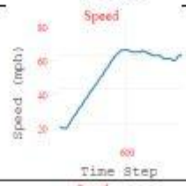
Jiggling	Turn	Smooth-Turn	Left-Turn	Right-Turn	Speed-Up	Slow-Down	Traffic-Light	Loop	Other

[1] E-mail address:
Oddo Girolamo, *Vehicle Engineering Master's Student, University of Modena and Reggio Emilia*
girolamo.oddo.98@gmail.com

Exit; Merge; Exit-Merge have been removed from the target classes due to their scarcity. The distribution of the data for the various segments is shown below. There is a strong imbalance in the distribution. In addition, the "NULL" classes in the dataset have been replaced with the segment to which they belong. The backward-fill method was chosen because the manoeuvre tends to evolve like the next one.
This method makes it possible to increase the number of readings and the continuity between conditions.



Loop	Car takes the ramp and makes a complete loop to merge into a highway/road.	
Turn	Car makes a turn which has significant effect in change of direction of moving vehicle.	
Smooth-Turn	Car makes a smooth turn which has some effects in change of direction of moving vehicle.	

Left-Turn	Car makes a left turn in an intersection.	
Right-Turn	Car makes a right turn in an intersection.	
Jiggling	Car keeps moving to left and right which has continues effect in change of direction of vehicle, as it is observable from heading change values.	
Speed-Up	Car speeds up to merge a highway, after traffic light, after stop sign, etc.	
Slow-Down	Car slows down because of traffic congestion, traffic light, stop sign, intersection, etc.	
Traffic-Light	Car stops behind a traffic light for a while, in such case, we can see the speed is zero or has some minor changes.	

[1]

3. Modalities and Analysis

In this section, forecasting models will be developed according to an approach based on Decision Tree and one based on Neural Networks. After that a possible application method of the obtained models will be presented. The models that will be obtained using the above mentioned predictors will operate only with the running conditions of the vehicle, neighbouring vehicles will not be considered as we are interested in predicting the start of a given manoeuvre as a function of the running conditions.

It has been decided not to take into account the surrounding vehicles because many manoeuvres in driving practice follow recurrent patterns regardless of the presence of other vehicles. e.g. turning manoeuvres are generally preceded by a Slow-Down and Smooth-Turn phase or even acceleration phases are generally preceded by a condition of zero steering angle. This introduces a strong simplification which, however, allows the problem to be treated and initial conclusions to be drawn in order to complicate the solution approach later.

3.1 Implementation of the Single Car Classification Model

In order to predict the behaviour of individual vehicles in different driving situations, different classification algorithms will be considered and trained and tested in order to obtain a reliable model. All models will be validated using the cross-validation technique with 7 folds. For the evaluation of the models, the results of the Accuracy in Test and the Test Confusion Matrix will be compared. Among the two, the Confusion Matrix will have more weight in the evaluation phase due to the unbalanced distribution of the data. In particular, models based on Decision Tree and models based on Neural Networks will be evaluated. We choose not to work with SVM models as they are more burdensome in terms of resources required than Ensamble models based on Decision Tree.

3.1.2 Decision Tree Models

We proceed with a search process using Bayesian optimisation of Ensamble models based on Decision Tree. This approach is used in order to obtain a solid starting point, on which further actions to improve the model can be evaluated. Ensamble techniques are chosen as they tend to perform better with unbalanced datasets.

The characteristics of the model and the results obtained from the research are presented below:

Training Results

Accuracy (Validation)	58.5%
Total cost (Validation)	8288
Prediction speed	~990 obs/sec
Training time	2054.6 sec

Test Results

Accuracy (Test)	43.9%
Total cost (Test)	3375

Model
TypePreset: Optimizable Ensemble
Learner type: Decision tree

Optimized Hyperparameters
Ensemble method: Bag
Maximum number of splits: 5923
Number of learners: 491

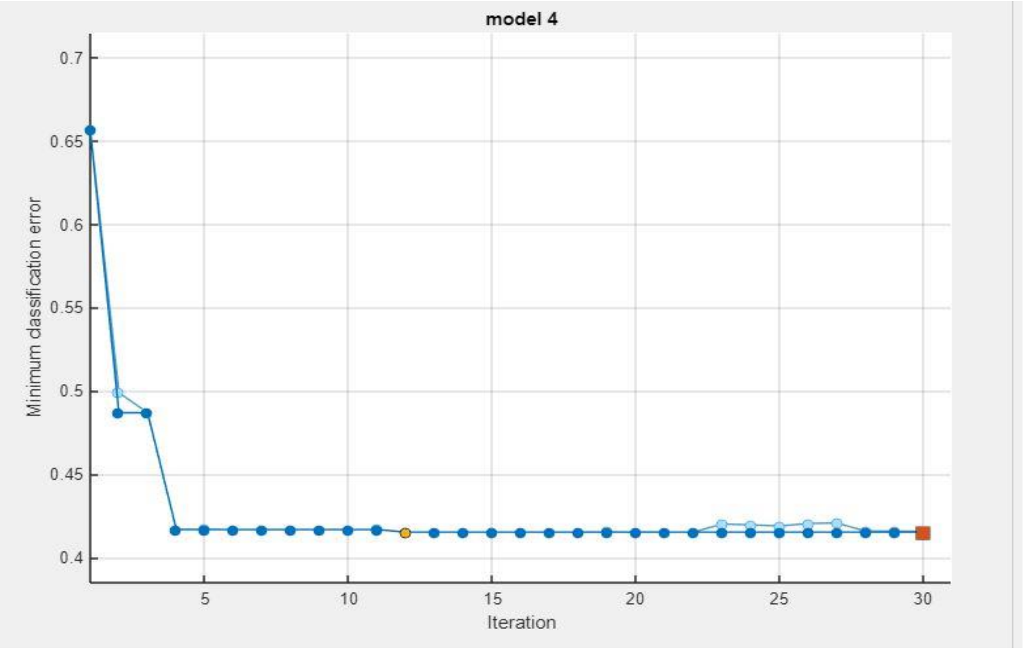
Hyperparameter Search Range
Ensemble method: Bag, AdaBoost, RUSBoost
Number of learners: 10-500
Learning rate: 0.001-1
Maximum number of splits: 1-19975

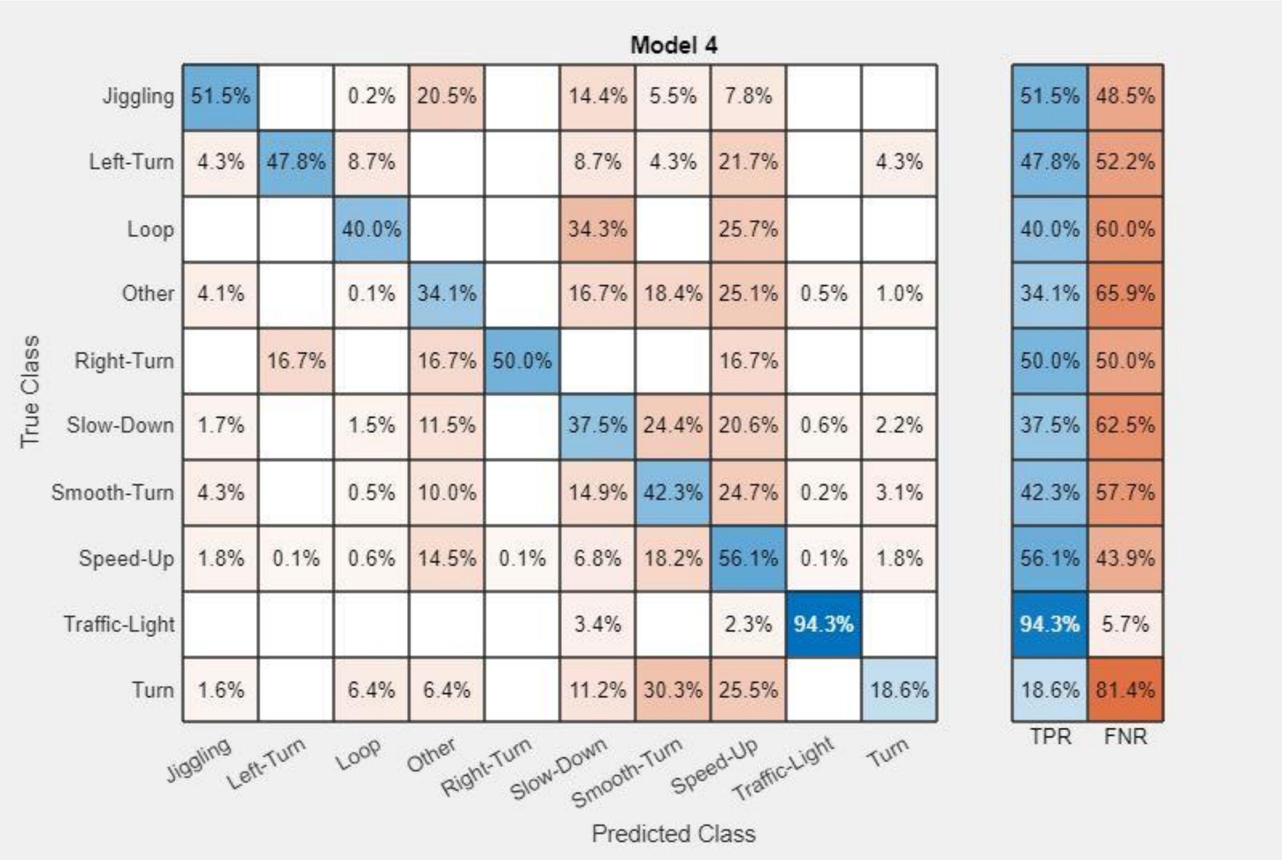
Optimizer Options
Optimizer: Bayesian optimization
Acquisition function: Expected improvement per second plus
Iterations: 30
Training time limit: false

Feature SelectionAll
features used in the model, before PCA

PCA
disabled

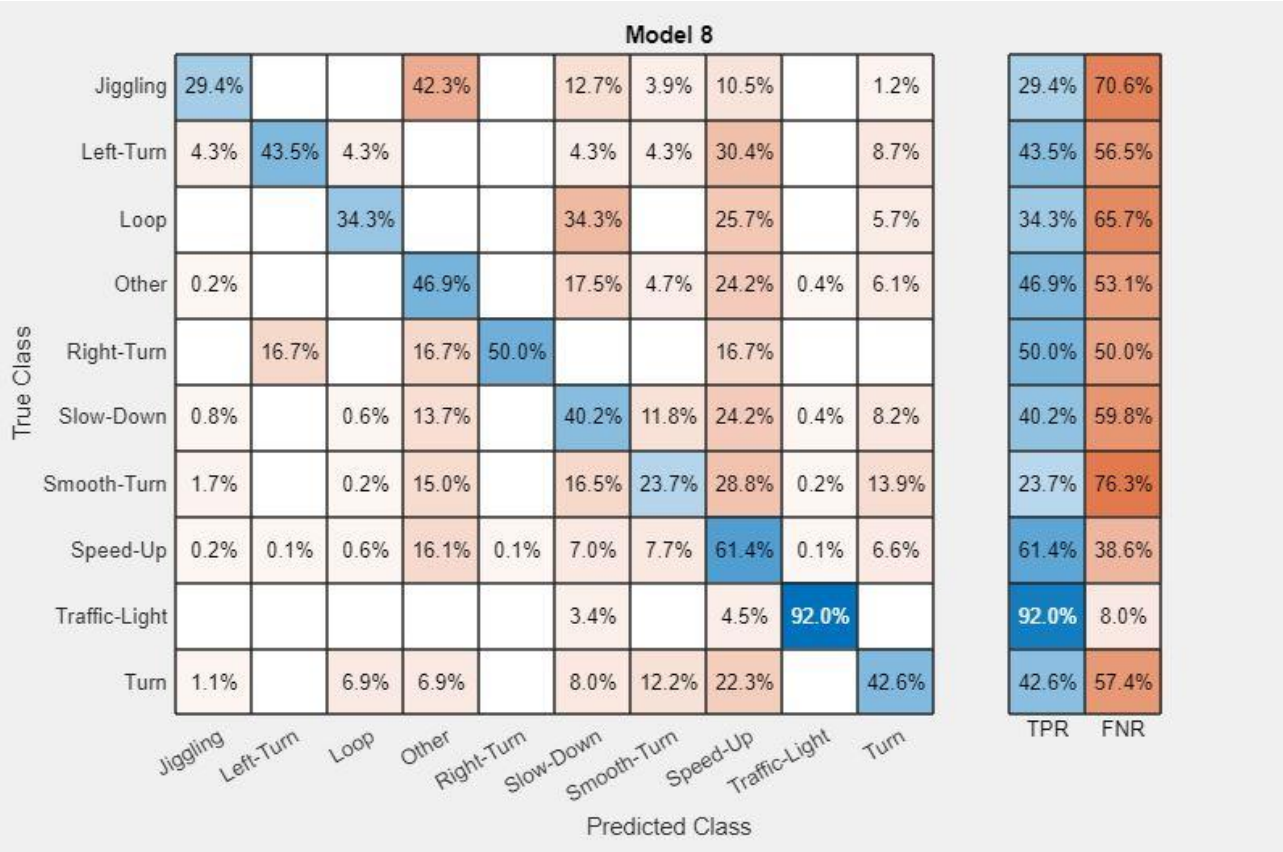
Misclassification Costs
Cost matrix: default





It is noted that following the optimisation the model manages to obtain results that are generally acceptable as a starting point, we proceed by operating on the Cost Matrix to correct the error in classifying "Turn" and minor errors reported by the model. we apply the following Cost Matrix to the model obtained previously.

		Predicted Class									
		Jiggling	Left-Turn	Loop	Other	Right-Turn	Slow-Down	Smooth-Turn	Speed-Up	Traffic-Light	Turn
True Class	Jiggling	0	1	1	1	1	2	1	2	1	1
	Left-Turn	1	0	1	1	1	2	1	5	1	1
	Loop	1	1	0	1	1	2	1	2	1	1
	Other	1	1	1	0	1	5	1	5	1	1
	Right-Turn	1	10	1	1	0	2	1	2	1	1
	Slow-Down	1	1	1	1	1	0	1	10	1	1
	Smooth-Turn	1	1	1	1	1	2	0	2	1	1
	Speed-Up	1	1	1	1	1	10	1	0	1	1
	Traffic-Light	1	1	1	1	1	2	1	2	0	1
	Turn	10	10	10	10	10	10	10	10	10	0



The model with the modified costs is stronger in basic manoeuvres such as Speed-Up, Slow-Down and Turn, but loses performance in more complex manoeuvres. This behaviour is assessed as favourable to safety. In fact, with a reduction in accuracy (1.7 %), the model fails in less critical classes while driving.

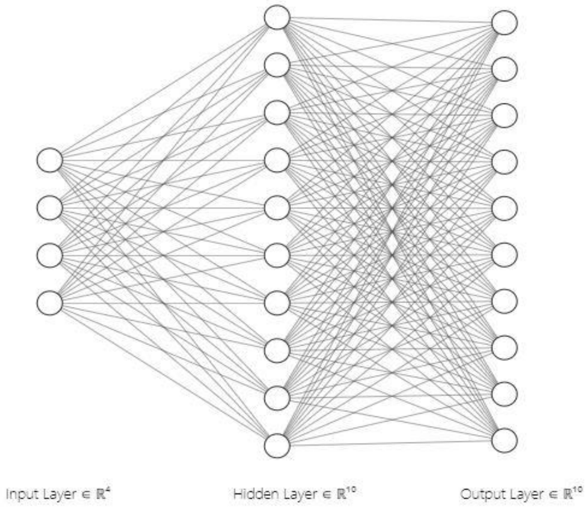
The latter is then taken as the reference model for the Decision Tree class.

It should also be noted that RUSBoost techniques, generally indicated for unbalanced datasets, have led to very good results in classifying manoeuvres with fewer samples in the dataset, while significantly penalising more numerous manoeuvres. Operating on the cost matrices as was done for the Bag model did not yield satisfactory results for the other manoeuvres.

3.1.3 Models based on Neural Networks

We continue by evaluating models based on Neural Networks, in particular we will evaluate networks with fully connected layers by acting on the number of layers and their size. To begin to perform evaluations, we use a network with a single layer, here are the details of the network and the performance obtained:

Model
TypePreset: Narrow Neural Network
Number of fully connected layers: 1
First layer size: 10
Activation: ReLU
Iteration limit: 1000
Regularization strength (Lambda): 0
Standardize data: Yes

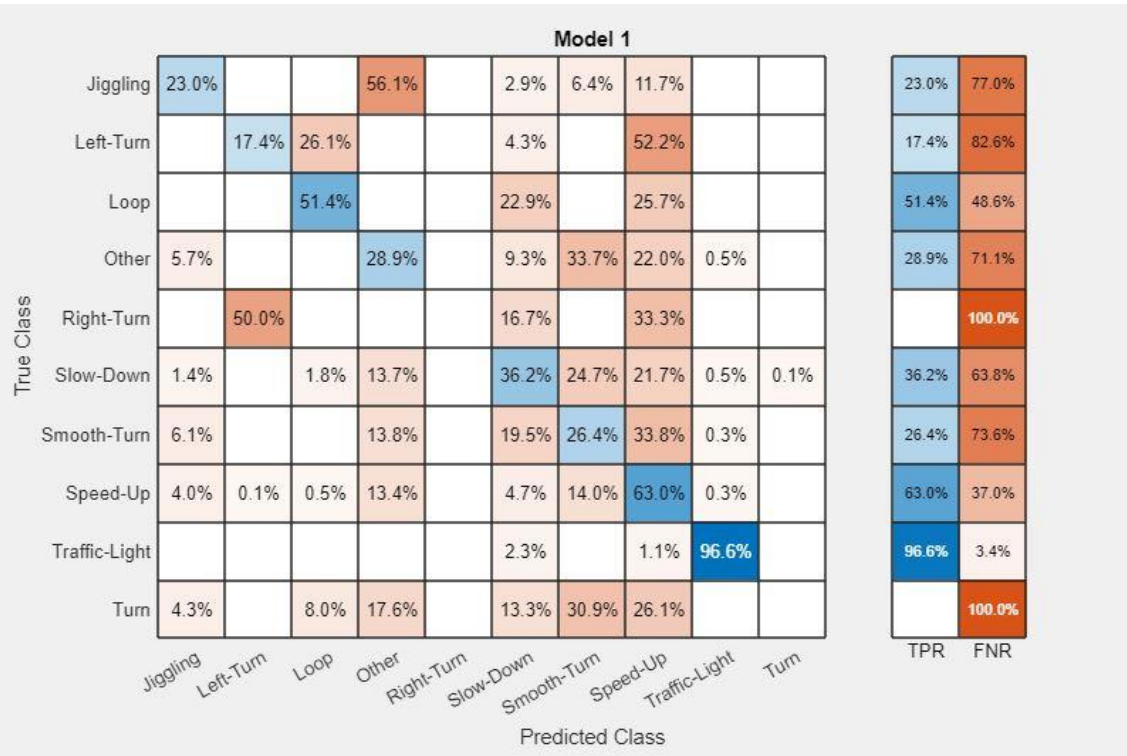


Training Results

Accuracy (Validation)	48.0%
Total cost (Validation)	Not applicable
Prediction speed	~180000 obs/sec
Training time	180.84 sec

Test Results

Accuracy (Test)	37.5%
Total cost (Test)	Not applicable

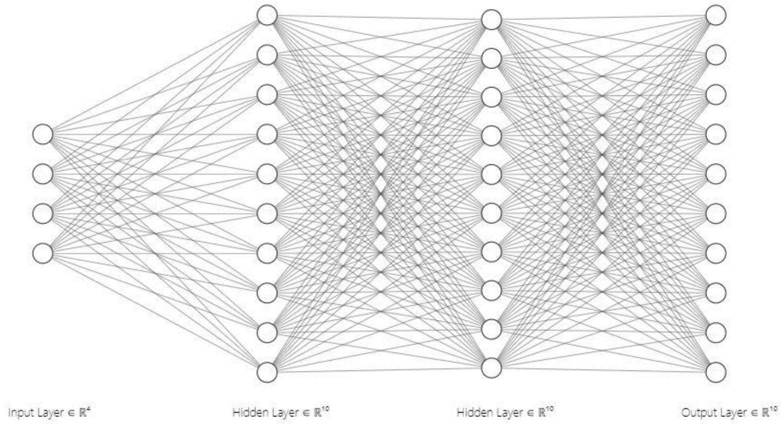


The model obtained shows serious shortcomings, we will proceed with a systematic approach by complicating the architecture step-by-step.

In order to obtain a sufficiently elastic model, a two-layer model is used:

The specific characteristics and results obtained are given below

Model
TypePreset:
Bilayered Neural Network
Number of fully connected layers: 2
First layer size: 10
Second layer size: 10
Activation: ReLU
Iteration limit: 1000
Regularization strength (Lambda): 0
Standardize data: Yes

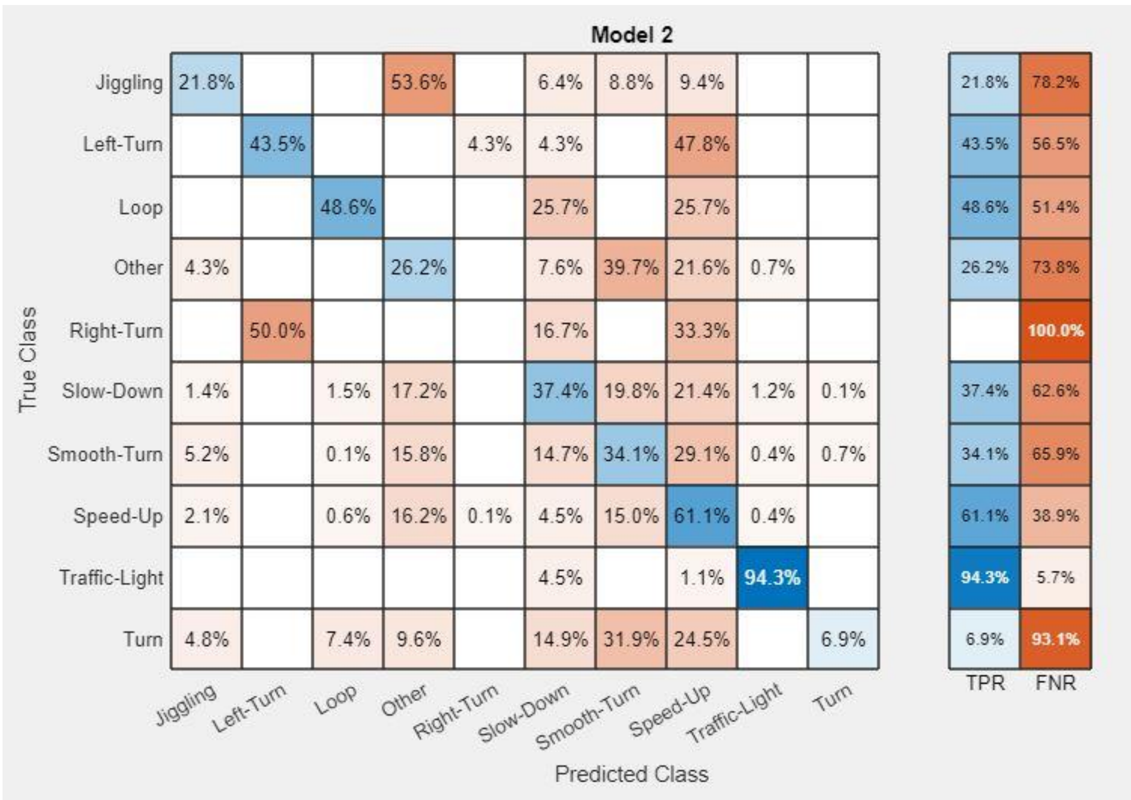


Training Results

Accuracy (Validation)	48.7%
Total cost (Validation)	Not applicable
Prediction speed	~370000 obs/sec
Training time	198.03 sec

Test Results

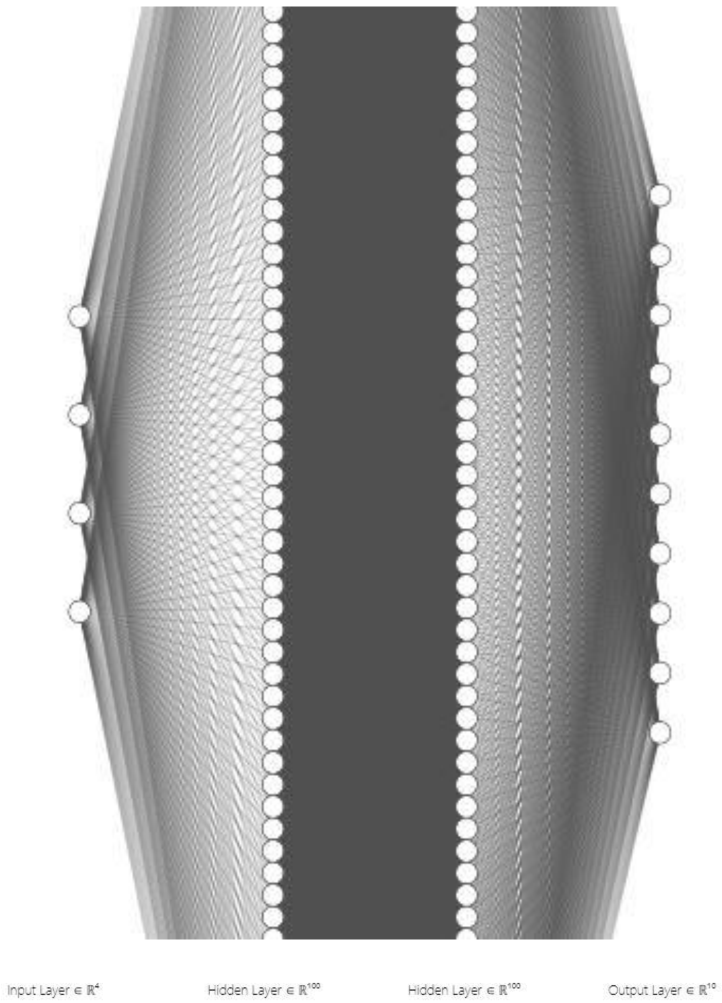
Accuracy (Test)	38.9%
Total cost (Test)	Not applicable



The model obtained does not show any substantial increase in performance, but there are still considerable errors in the classification of "Right-Turn" and "Turn". We therefore proceed to increase the size of the layers by an order of magnitude, from 10 to 100 nodes per layer.

The resulting network shows the following specifications and performance:

Model
TypePreset: Bilayered Neural Network
Number of fully connected layers: 2
First layer size: 100
Second layer size: 100
Activation: ReLU
Iteration limit:1000
Regularization strength (Lambda): 0
Standardize data: Yes

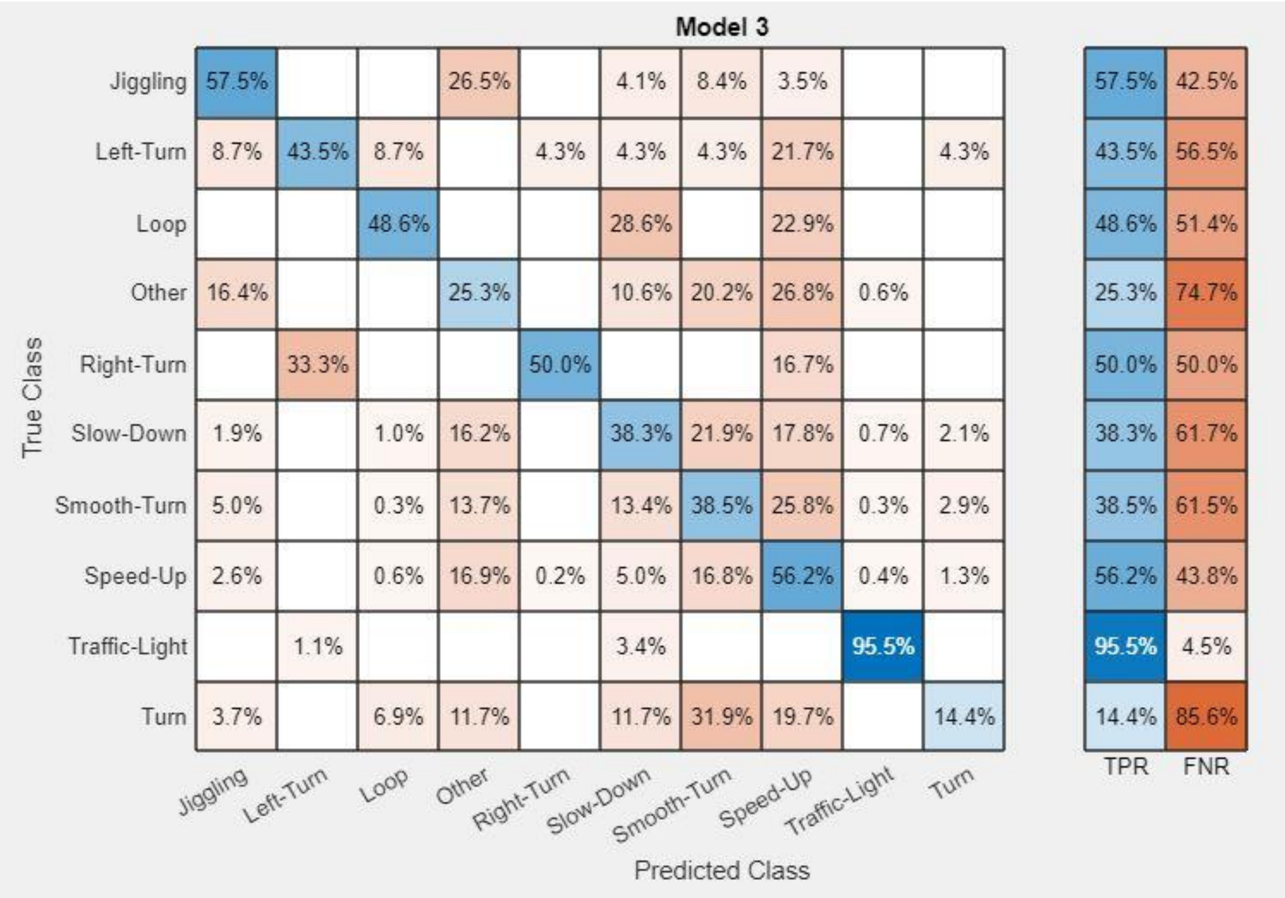


Training Results

Accuracy (Validation)	53.0%
Total cost (Validation)	Not applicable
Prediction speed	~210000 obs/sec
Training time	830.39 sec

Test Results

Accuracy (Test)	42.1%
Total cost (Test)	Not applicable



The model shows a significant improvement, but the performance in some basic manoeuvres such as "Turn" remains critical. This therefore requires further development of the architecture.

We then proceed to increase the size of the layers further.
The resulting model shows the following specifications and performance:

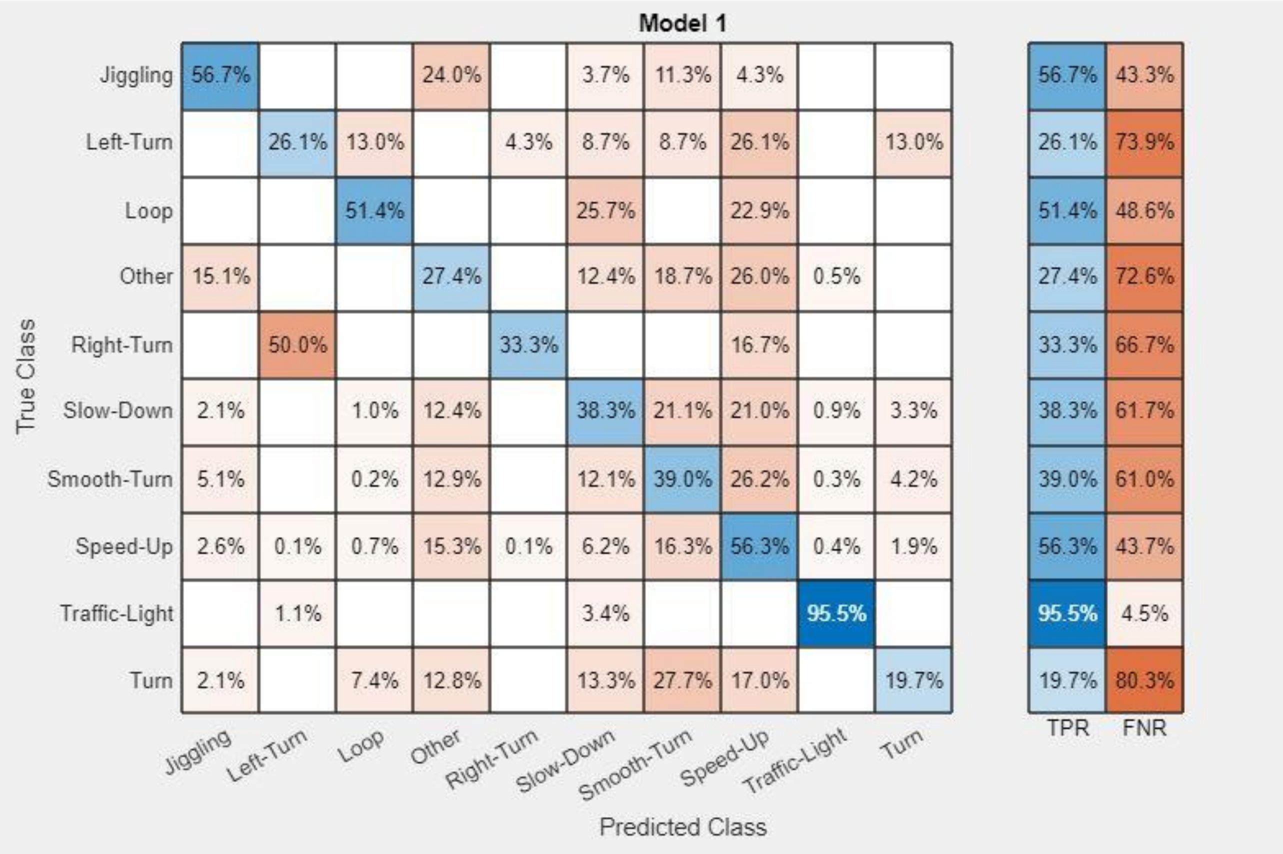
Model
TypePreset: Bilayered Neural Network
Number of fully connected layers: 2
First layer size: 125
Second layer size: 125
Activation: ReLU
Iteration limit: 1000
Regularization strength (Lambda): 0
Standardize data: Yes

Training Results

Accuracy (Validation)	53.4%
Total cost (Validation)	Not applicable
Prediction speed	~110000 obs/sec
Training time	1131.7 sec

Test Results

Accuracy (Test)	42.6%
Total cost (Test)	Not applicable

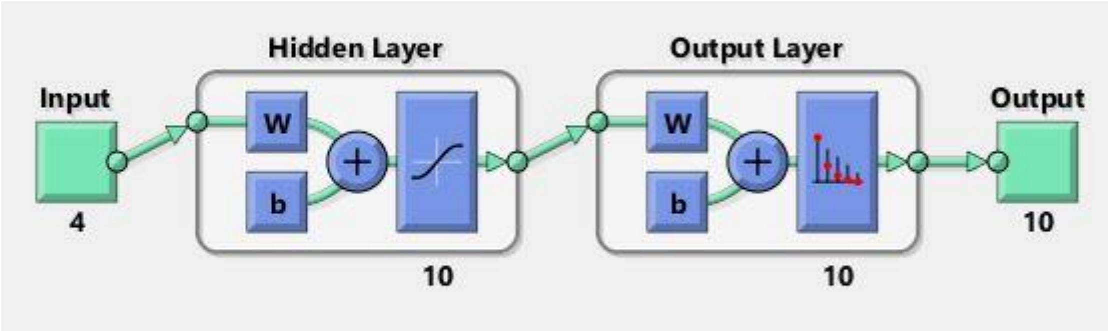


The model obtained is slightly improved with respect to the previous one, but again shows a critical behaviour with respect to the "Right-Turn" manoeuvre. For this reason, we will stop this approach and proceed with architectures consisting of two layers, one of which with neurons organised in a map.

In order to be able to use this architecture in Matlab, it was necessary to reformulate the target variables by representing them as a matrix of 0 and 1 as follows:

Slow-Down	1	1,0,0,0,0,0,0,0,0,0
Speed-Up	2	0,1,0,0,0,0,0,0,0,0
Turn	3	0,0,1,0,0,0,0,0,0,0
Smooth-Turn	4	0,0,0,1,0,0,0,0,0,0
Left-Turn	5	0,0,0,0,1,0,0,0,0,0
Right-Turn	6	0,0,0,0,0,1,0,0,0,0
Traffic-Light	7	0,0,0,0,0,0,1,0,0,0
Loop	8	0,0,0,0,0,0,0,1,0,0
Other	9	0,0,0,0,0,0,0,0,1,0
Jiggling	10	0,0,0,0,0,0,0,0,0,1

To begin evaluating the performance of this architecture, a network with 10 neurons is used, as was done for the previous network architecture.



The results of a training session using the Scaled Conjugate Gradient Backpropagation technique are shown below:

The model finds its Best-Point in Cross-Entropy after 169 Epochs with an EC Test: 5.36

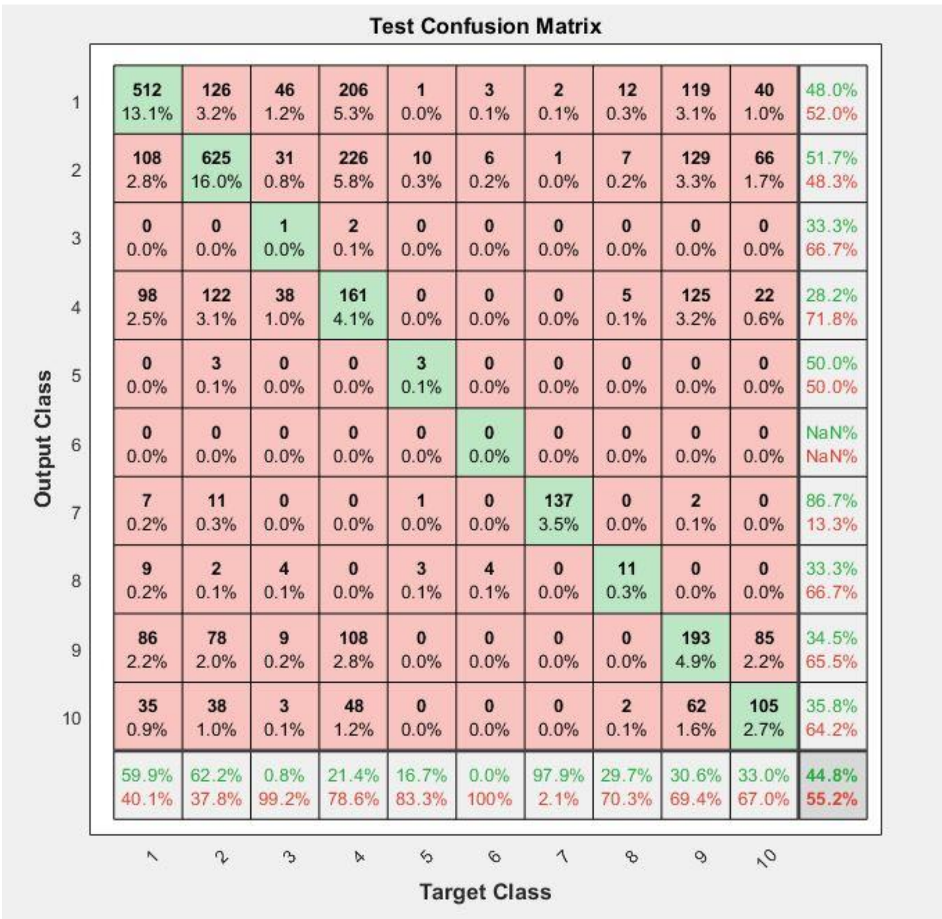
Training Results

Accuracy (Validation) 45.7%

Training time 48 sec

Test Results

Accuracy (Test) 44.8%



It should be noted that the Confusion Matrix is related to the Test phase, in this case a sub-set is not explicitly selected but is randomly extracted by Matlab, it is therefore possible to obtain NaNs in manoeuvres with fewer samples.

In view of the discrete performance of the model, it was decided to expand the layers to 100 neurons.

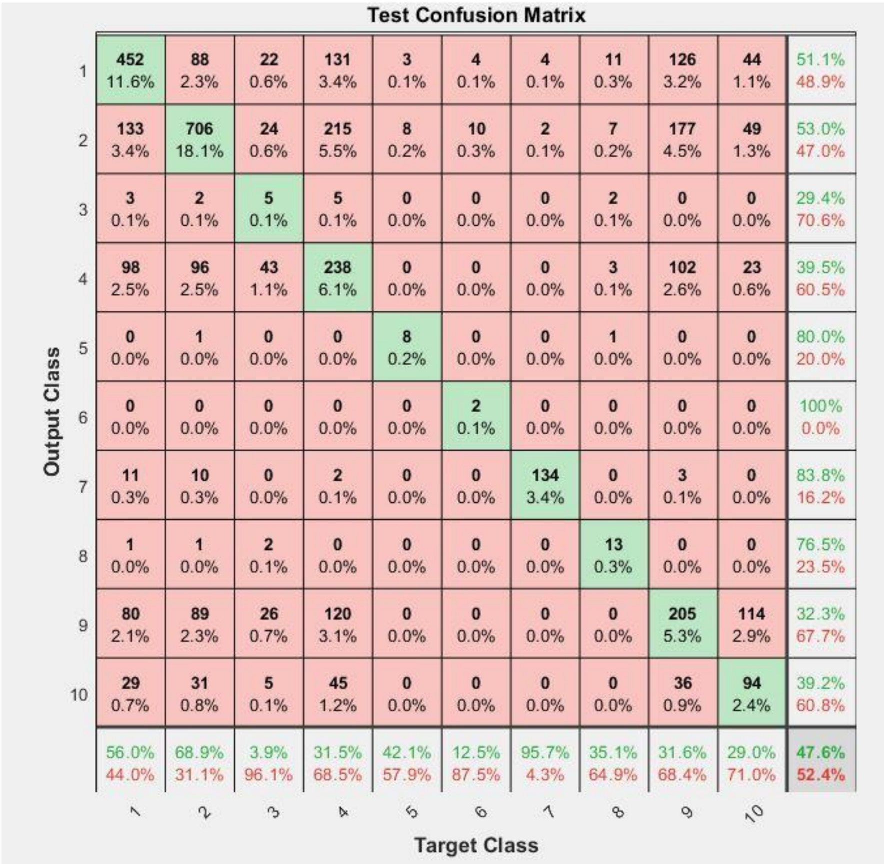
The model finds its Best-Point in Cross-Entropy after 171 Epochs with an EC Test: 6.33

Training Results

Accuracy (Validation)	47.9%
Training time	21 sec

Test Results

Accuracy (Test)	47.6%



The model obtained shows an increase in accuracy and a general improvement in the understanding of behaviour. The classification of classes linked to the turning manoeuvre is still critical.

We will iterate one last time increasing the size of the network again, bringing the number of neurons to 250.

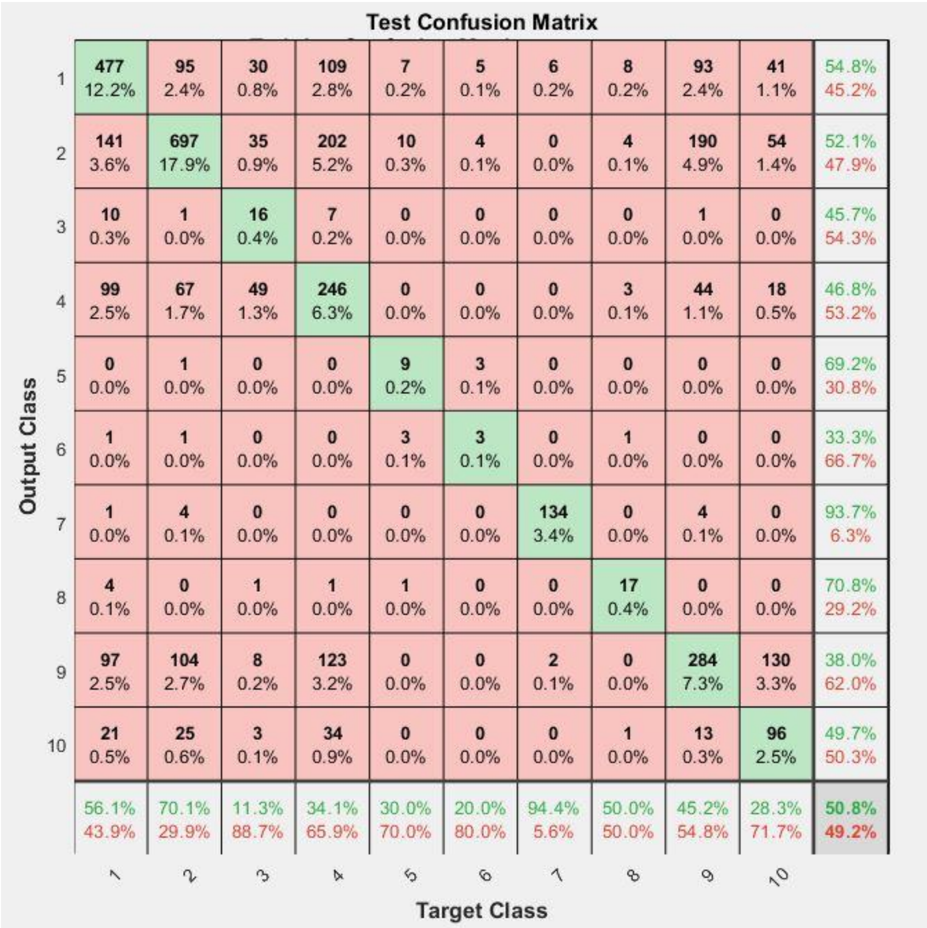
The results obtained are reported:

Training Results

Accuracy (Validation)	51.1%
Training time	111 sec

Test Results

Accuracy (Test)	49.2%



The model with 250 neurons is the best performing model obtained in the study group.

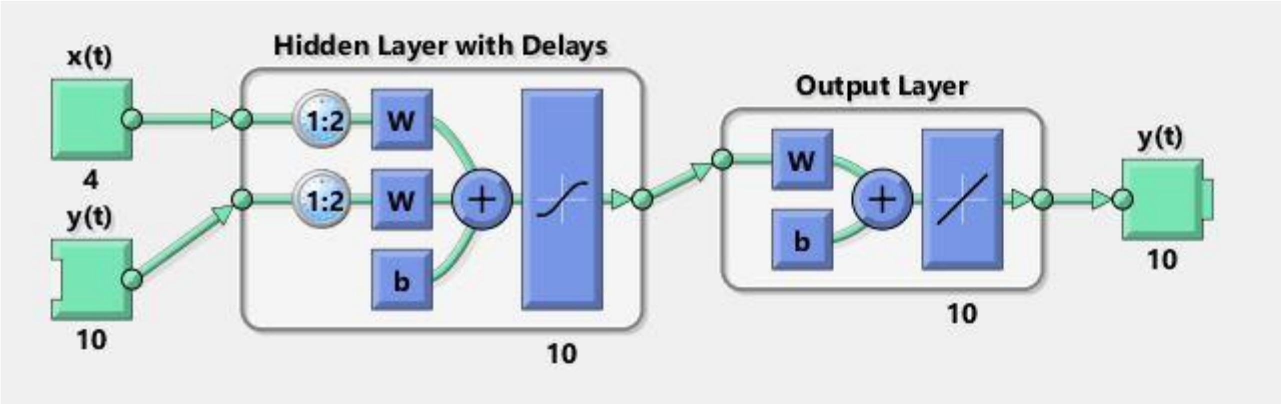
After the evaluation of these models, we will proceed with neural networks capable of taking into account the temporal evolution of the phenomena involved.

In fact, the previously proposed networks did not take into account the previous states of the problem, but only evaluated point solutions given by how the four predictors presented themselves without evaluating the previous states. To evaluate the work of these architectures we will use the indicators MSE (Mean Squared Error) and the index R^2 . The networks will be trained using the Scaled Conjugate Gradient technique.

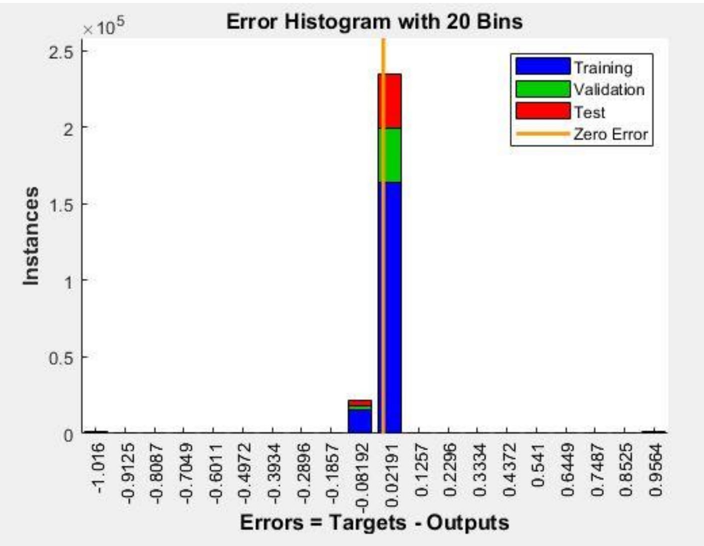
We will now work with NARX-type networks (nonlinear autoregressive exogenous model) evaluating their initial performance with a network of 10 neurons 2 delay obtaining a problem in the form:

$$y(t) = f(\text{x (t-1),...,x (t-d), y (t-1),...,y (t-d)})$$

Obtaining:



The proposed network has the following performance in Test:



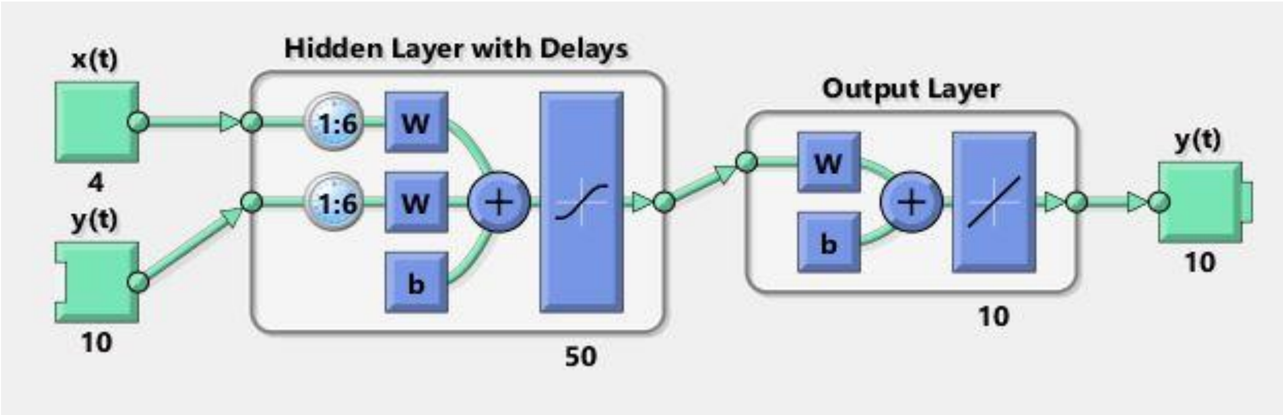
R^2 0.90%

MSE 9.47 e-3

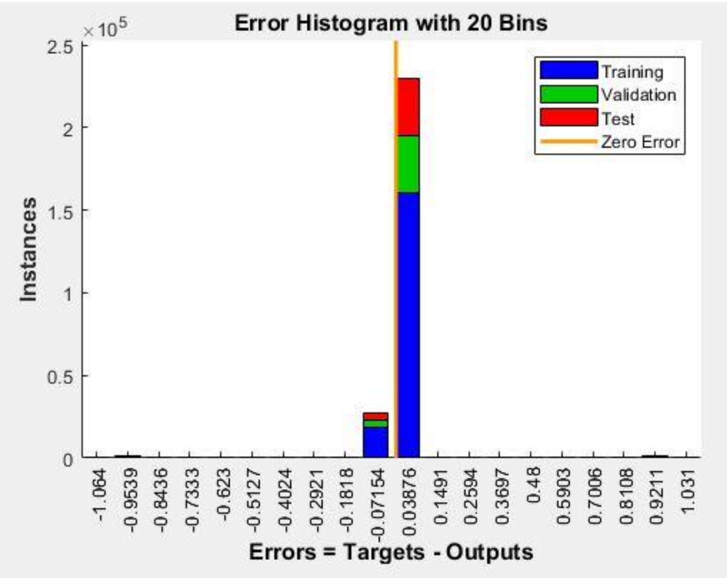
The results show that this type of architecture is better able to capture trends at low complexity than previous architectures.

However, a further iteration will be carried out increasing the number of delays from 2 to 6 in order to almost completely capture the interval between segments. In addition to this, the number of neurons will be increased to 50.

The following network is obtained:



Achieving the following performance in Test:



R^2	0.91%
MSE	8.54 e-3

The results show a slight increase in performance in the chosen indicators, which does not justify the increase in complexity.

The reference model for future evaluations will therefore be the architecture with 10 neurons and 2 delays. It is considered that the phase of realisation of the predictive model has been completed, despite the limits that this presents due to the strong unbalance in the dataset. The latter is taken as the reference model for the applications that will be proposed later.

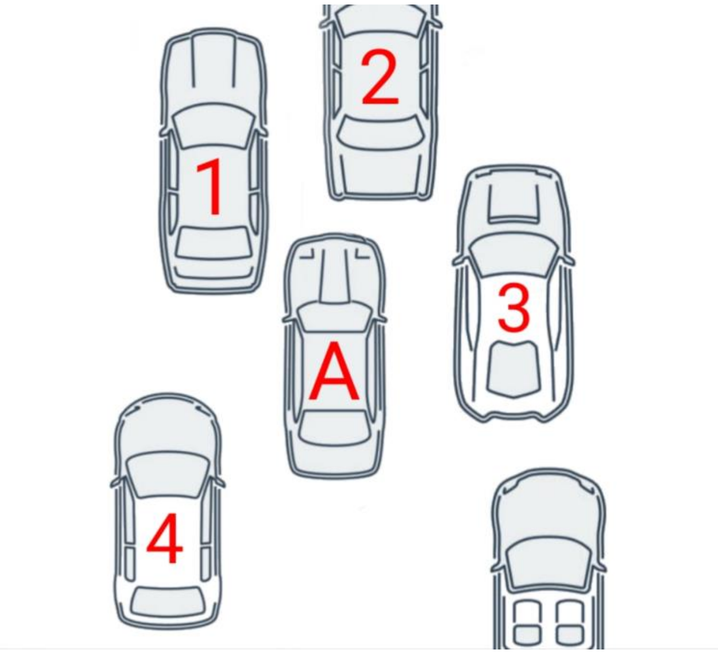
3.2 Model Applications

For the application of the model, an approach based on V2V connectivity is proposed, where each vehicle communicates to its neighbours the information used in the models developed for individual cars; i.e:
Speed; Acceleration; Heading; HeadingChange. From these values it is possible to obtain an array as follows:

Distance	Speed	Acceleration	Heading	Hading Change
D1	S1	A1	H1	C1
D2	S2	A2	H2	C2
D3	S3	A3	H3	C3
D4	S4	A4	H4	C4

It should be noted that the distance between vehicles has been introduced in order to have information about the most critical vehicle for driving safety. This distance can be obtained as the difference in GPS coordinates between A and n_Vehicle.

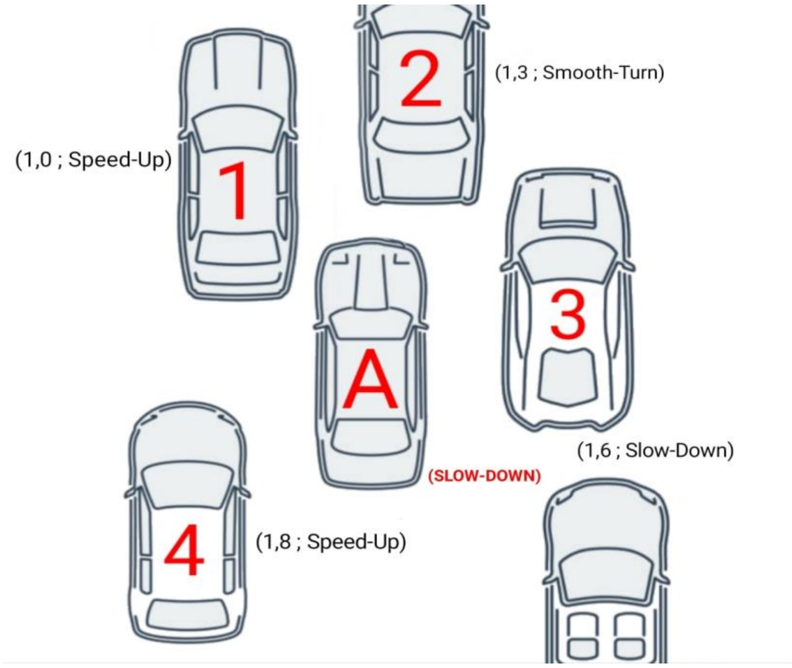
Car A, once it has obtained the information, will be asked to predict the behaviour of each neighbouring car, applying the models in 3.1. With this approach it is possible to construct a behaviour map, where each predicted external condition is associated with a behaviour that car A must follow.



For a case like the one chosen where only the four closest vehicles will communicate with A, a behaviour map is obtained as follows:

Distance	Predicted Behaviour
D1	P1
D2	P2
D3	P3
D4	P4

To clarify the function of the proposed method, an example is given where A sees the distance in metres and has already made the prediction for the four neighbouring vehicles. In this case A will associate a Slow-Down manoeuvre with these external conditions because the vehicle in front (2) is in the incipient steering phase. This choice has been assigned for the sole purpose of making the proposed operating logic understood.



In this paper we will not go into the choice of manoeuvre to be performed by A, but only present a possible use of the models obtained in 3.1. For the subsequent evaluations, the behaviour of A will be attributed to the surrounding context only in order to continue the analysis.

One could therefore proceed by performing a direct attribution of the behaviour of A for each combination of the 10 behaviours of the 4 vehicles, but this would result in a map that is too complex to make initial assessments of the proposed approach ($P(n,r) = n^r = 10000$ cases). However, it is possible to use a simplified version where only the previous and the following vehicles are considered and only 3 basic behaviours such as: Speed-up; Slow-Down; Turn and if the distance of at least one vehicle to A is within a safe range. ($P(n,r) = n^r = 9$ permutations · 2 conditions on distance = 18 cases)

Previus Vehicle - Pred. Behaviour	A Behaviour
Following Vehicle - Pred. Behaviour	

For Distance > Security Distance

Speed-Up	Speed-Up
Speed-Up	
Speed-Up	Hold
Slow-Down	
Speed-Up	Hold
Turn	

Slow-Down	Slow-Down
Speed-Up	
Slow-Down	Slow-Down
Slow-Down	
Slow-Down	Slow-Down
Turn	

Turn	Hold
Speed-Up	
Turn	Hold
Slow-Down	
Turn	Hold
Turn	

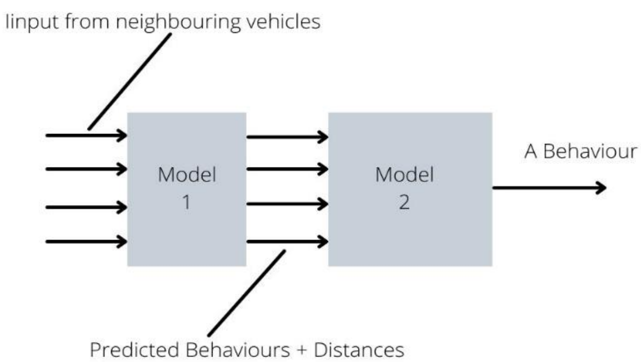
For Distance < Security Distance

Speed-Up	Speed-Up
Speed-Up	
Speed-Up	Hold
Slow-Down	
Speed-Up	Hold
Turn	

Slow-Down	?
Speed-Up	
Slow-Down	Slow-Down
Slow-Down	
Slow-Down	Slow-Down
Turn	

Turn	Hold
Speed-Up	
Turn	Hold
Slow-Down	
Turn	Hold
Turn	

This approach, although simplified, shows in the Slow-Down / Speed-Up case a possible case of incipient rear-end collision, where the behaviour of the vehicle can be evaluated ad-hoc. In addition, the term Hold is used to indicate the maintenance of the running condition.
Acting in this way allows one to have full control over all the cases, but implies a rapid increase in the number of conditions to be examined as the number of communicating vehicles and behaviours to be predicted increases ($n^r$for a case where all classes of the initial dataset could be effectively evaluated, i.e. twelve behaviours for four vehicles, 20736 cases would be obtained). It would be more functional, following a targeted data acquisition campaign, to use again an approach through ML\DL, thus building a model capable of attributing a behaviour to A according to the predictions.
An application approach is then obtained as follows:



4. Conclusions

In the present work we tried to model the traffic phenomenon according to the DACT - Strict Annotation dataset, we evaluated the behaviour of a single vehicle in traffic in order to obtain a predictive model as a function of Speed; Acceleration; Heading; HeadingChange. Two modelling approaches were evaluated, one based on Decision Tree and one based on Neural Networks. It emerged that the models based on NARX Neural Networks are the most performing for the dataset under examination. A method was then proposed to make the models obtained for each single car operate in a traffic context, where a plurality of cars are present. The proposed method aims to provide a first application approach of the forecasting model obtained by exploiting V2V connectivity.

Further Proposal

In order to develop the proposed method it is necessary to carry out a data acquisition campaign as shown in 3.2. In order to make the model more performant it can be considered to include among the predictors the speed limit in order to provide the model with an additional parameter to predict the Speed-Up and Slow-Down phase.

Conflict of Interest

The author declares that the work, which began as a project for the MathWorks Excellence in Innovation initiative, was carried out without any conflicts of interest.

References

[1] Sobhan Moosavi, Behrooz Omidvar-Tehrani, R. Bruce Craig, Rajiv Ramnath "Annotation of Car Trajectories based on Driving Patterns".

DATASET: https://figshare.com/articles/dataset/DACT_Dataset_of_Annotated_Car_Trajectories/5005289