

Floating Car Data Augmentation Based on Infrastructure Sensors and Neural Networks

José E. Naranjo, Felipe Jiménez, Francisco J. Serradilla, and José G. Zato

Abstract—The development of new-generation intelligent vehicle technologies will lead to a better level of road safety and CO₂ emission reductions. However, the weak point of all these systems is their need for comprehensive and reliable data. For traffic data acquisition, two sources are currently available: 1) infrastructure sensors and 2) floating vehicles. The former consists of a set of fixed point detectors installed in the roads, and the latter consists of the use of mobile probe vehicles as mobile sensors. However, both systems still have some deficiencies. The infrastructure sensors retrieve information from static points of the road, which are spaced, in some cases, kilometers apart. This means that the picture of the actual traffic situation is not a real one. This deficiency is corrected by floating cars, which retrieve dynamic information on the traffic situation. Unfortunately, the number of floating data vehicles currently available is too small and insufficient to give a complete picture of the road traffic. In this paper, we present a floating car data (FCD) augmentation system that combines information from floating data vehicles and infrastructure sensors, and that, by using neural networks, is capable of incrementing the amount of FCD with virtual information. This system has been implemented and tested on actual roads, and the results show little difference between the data supplied by the floating vehicles and the virtual vehicles.

Index Terms—Floating car data, neural networks, traffic flow.

I. INTRODUCTION

THE USE of variable message signs lets infrastructure managers and authorities control the traffic and provide road users with information, which prevents dangerous situations, reduces accidents, improves traffic flow, and reduces energy consumption and CO₂ emissions. The main advantage of variable message signs over conventional traffic signs resides in the fact that the information shown by the latter is permanent and cannot be adapted to different traffic conditions (tail-backs, accidents, fog, slippery road surface, etc).

In [1], a list of different studies on the effectiveness of these panels is shown, with a very wide range of results having been obtained. However, all of these studies coincide in confirming that the application of these panels leads to a considerable reduction in accidents with injuries, particularly when the information provided is related to accident warnings or weather conditions that imply speed reduction [2].

However, to provide panel information, it is necessary to previously detect the different conditions to be displayed. To extend the driver's field of vision, we can define an "electronic horizon" much more extensive than the "visual horizon." By so doing, we can take advantage of information from the following sections of road that are not visible to the driver so that the speed can be adapted [3], [4]. The electronic horizon can be obtained from sensors (radar, laser, image processing, etc.) or telecommunications and positioning by means of satellites and navigation systems.

There are several methods for developing this task. It is possible to use traffic video surveillance, whose images are viewed by operators in a control center where the warnings are emitted. In recent research, this viewing has been automated through computer vision for congestion detection. To analyze traffic intensity and speed, the most usual way is to obtain information from sensors placed in the infrastructure, mainly magnetic loop detectors under the road [5]–[10]. These kinds of sensors have the following two main limitations.

- 1) Their performance is based on magnetic principles that make them capable of detecting the passing of metallic masses (vehicles) over them; the activation of two consecutive spires enables an estimation of the speed to be calculated. However, the resolution of this speed calculation is low, and there are problems with the different configurations and sizes of the road vehicles.
- 2) The information provided by these sensors is related to concrete road positions. As a result, if the sensors are not placed with a small separation between them, the resolution for incident detection is very low.

On the other hand, a different solution for obtaining traffic information is to use "floating vehicles" [11]–[17]. Huber *et al.* [11] raised the possibility of having "sensorized floating vehicles" circulating that could gather information on the condition of the roadway, meteorological conditions (rain, fog, temperature, etc.), and traffic density. This information would be sent to a control center that would analyze and order the appropriate signs for other road users. Although position and speed are the typical variables managed by these vehicles, it is possible to add any other information, such as external

temperature, light sensor activation, or rain sensor activation. If this system achieves a sufficient level of penetration [11], the information sent to the traffic control centers would substitute that obtained through infrastructure sensors because it would allow estimating average speeds, traffic intensity, and other relevant conditions used to detect incidents, accidents, and traffic jams with better accuracy and not only in some specific instrumented road areas.

However, the main difficulty in a real floating car data (FCD) application is linked to the realistic possibility of having the “numbers” of vehicles adapted to ensure that the set of traffic is sufficiently accurate. In other words, the penetration rate of FCD currently available is too low to use this system as a real alternative to conventional infrastructure sensors [18].

Various studies indicate that there are two main aspects that must be taken into account to determine the reliability of the information supplied by a set of floating vehicles. On the one hand, the minimum number of probe cars should be larger than 2% of the traffic flow [19]–[21], and on the other hand, the minimum information rate should be between 10 min [18] and 3 min [17]. With these features, it is possible to detect abnormal events in traffic as well as calculate the travel time [22].

Although the information provided by the infrastructure sensors and floating vehicle data should be redundant, the limitations and uncertainty of both methods mean that this does not happen, and therefore, it is necessary to analyze the coherence of the information retrieved [23]. A combination of both sources of information is shown in [24] and [25].

This paper focuses on the development of a system that augments the main lack of the use of FCD and the minimum number of vehicles required to obtain a clear picture of the traffic situation in the circulation area. To this end, we combine information from infrastructure sensors and one piece of FCD and process it using a neural network (NN)-based system capable of learning traffic behavior and response. Once the system has been trained, it is able to use the real-time infrastructure sensor information to augment the number of FCDs on the road, calculating the circulation speed of virtual probe cars, each distributed every 100 m on the road at a rate of one per minute, keeping error rates down to less than 8%. This system can be used as a source for complex systems, ranging from micro-/macrosimulators to urban traffic planning, intelligent route calculation systems, or travel time estimation systems [26]. The system presented has been designed implemented and tested on actual roads using a 22-km instrumented segment of Madrid’s M40 highway.

II. EXPERIMENTAL EQUIPMENT

A. Intelligent Vehicle Instrumentation

The testbed vehicle used as floating vehicle is an instrumented Peugeot 307 equipped with an acquisition computer with controller area network (CAN) bus access, one noncontact speed sensor, a gyroscopic platform, several Global Positioning System (GPS) receivers, Universal Mobile Telecommunications System Internet access, and a touch panel accessible to the driver in order for them to be able to manage the acquisition system (see Fig. 1).

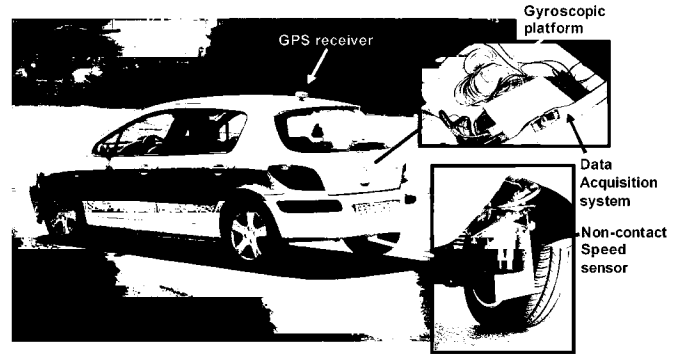


Fig. 1. Outline of where the sensors are placed in the testbed vehicle.

The variables of the floating vehicle that are measured to perform the comparison with infrastructure signals are position and speed. The position is obtained by three GPS receivers with different accuracies and functionalities: Garmin eTrex H and Astech G-12 in autonomous mode and Topcon GB-300 with real-time kinematics NTRIP differential correction via Internet, connected to the computer through the RS232 ports. This way, it is also possible to evaluate the differences among the receiver models, since in normal conditions, the mass-production vehicles equip low-performance receivers, and consequently, the uncertainty of this measurement has to be taken into account when integrating the results.

It is possible to acquire speed using GPS receivers. However, we acquire it from the vehicle CAN bus to perform experiments from standard equipment. Additionally, the instrumented vehicle equips a dedicated Correvit L-CE noncontact speed sensor used as a high precision sensor to guarantee the accuracy of the measurement. As a result, the accuracy of the speed measurement of the floating vehicle is about 0.24 km/h.

A record of the sensor information is made using a laptop and a National Instruments DAQCard-6062E acquisition card.

B. Infrastructure Sensor Data Retrieving

At present, there are several web-based information sources that provide public and accessible real-time traffic information. In Spain, the most important source is the Spanish Directorate General of Traffic (DGT), which provides information on road incidents and traffic as well as data from the sensors placed in these infrastructures, mainly in reference to highways. Although this information is public, it is disseminated through web pages, with a special authorization being necessary to directly query the system database.

In our case, we have chosen to access the database using the web interface through the implementation of an intelligent agent capable of retrieving information from web pages and reusing it in other applications, allowing the creation of mashups (web applications that use or combine data or functionalities from two or more external sources to create a new service). In this case, the problem consists of extracting information from semi-structured information. To obtain this traffic information, we have developed an information agent (software robot) that is able to do the following.

- 1) Realize a query of traffic incidents in the DGT web page.

TABLE I
EXAMPLE OF THE DATA CAPTURED BY THE SOFTWARE ROBOT

Road	KP	Direction	Intensity (Veh./hour)	Average Speed (km/h)
M-40	Pk 10.3	I	1380	125
M-40	Pk 12.7	D	5760	98
M-40	Pk 16.7	D	3780	73
M-40	Pk 12.7	I	3000	100
M-40	Pk 20.2	D	3300	99
M-40	Pk 20.2	I	5040	91
M-40	Pk 17.1	I	5100	98
M-40	Pk 17.1	D	5460	82
M-40	Pk 12.2	I	5700	121
M-40	Pk 12	D	4500	107
M-40	Pk 10.4	D	4380	116
M-40	Pk. 21.8	I	1920	100
M-40	Pk. 21.8	D	2280	97
M-40	Pk. 25.1	D	3840	98
M-40	Pk. 23.8	I	5340	90
M-40	Pk. 23.8	D	4620	86
M-40	Pk. 6.35	D	2760	91
M-40	Pk. 7.65	I	4800	126

- 2) Extract relevant data related to traffic incidents in the city of Madrid, Spain, from the information obtained in the previous query.
- 3) Realize an information query from the DGT's traffic sensors for the field operational test area (M40 highway).
- 4) Navigate to the web information from each of the sensors.
- 5) Extract the information from each sensor.
- 6) Finally, every piece of information will be retrieved every 40 s and will be logged in a file while the floating vehicles are performing tests so that the information captured can be compared with that obtained from the testbed vehicle.

The software robot has been implemented in Python programming language used for regular expression extraction.

An example of the information captured by the robot is shown in Table I. In this table, the first column represents the identification of the road. The second column indicates the marker post (MP) kilometer where the sensors are placed. The third column indicates the direction (increasing and diminishing). The fourth column indicates the traffic intensity expressed in vehicles per hour, and the fifth column is the average speed in kilometers per hour. In our application, this table updates every 40 s since it is the update time of the web information available from the DGT.

C. Field Operational Test Area

Field operational tests were performed on a segment of 22 km of Madrid's M40 highway, instrumented with 26 traffic sensors (see Fig. 2), whose information is available in real time through the public database of the DGT. The information supplied by these sensors was previously analyzed showing that

only 18 of them are operative 24 h/day, and eight of them are disabled. This means that these 18 loop sensors are used as input in the FCD augmentation system.

In this database, the identification from each of the infrastructure sensors is expressed with a road number plus marker post (MP) kilometer plus direction (increasing, diminishing). However, to compare this information with that supplied by the instrumented vehicle equipped with a GPS receiver, it is necessary to geolocate all of these MPs. Table II shows the GPS Universal Transverse Mercator position of every infrastructure sensor used in the field operational tests, all of which are located on the M40 highway around the South Campus of the Technical University of Madrid.

With this information, it is now possible to compare the results of the infrastructure information and the onboard vehicle information.

III. FLOATING CAR DATA AUGMENTATION SYSTEM

With this equipment, we have designed and implemented a system that is capable of solving one of the most important problems of FCD retrieval: the small number of probe cars circulating in the infrastructure, which considerably limits the efficiency and possible applications of this instrument. In this paper, we propose that virtual vehicles be included in the FCD system to increase the amount of available information, and that the use of FCD be extended to a wide range of traffic applications that would otherwise be unable to use this method because of the nonavailability of a sufficient number of probe cars.

The first step to develop this system is to select a methodology to virtualize floating cars in real time and with real traffic information. At present, the only available source is the infrastructure sensor, whose information is available from the DGT. The problem with these sensors is that they offer occasional information from individual locations in the road. This means that it is very difficult to estimate crisp traffic information for the entire road, and there are multiple road events not detected by the sensors [24]. To solve this deficiency, we have designed and implemented an intelligent system based on NNs that is able to virtualize FCD from the information from infrastructure sensors.

An NN is an intelligent system that can emulate the computational capabilities of a living being by mimicking the brain cells (neurons). To do this, NNs use a mathematical neuron model that performs the single processing of its inputs [27]. This processing consists of the weighted sum of the N neuron inputs (net) whose result is applied to a nonlinear function, among which the most widely used is the sigmoid function S

$$net = \sum_{n=0}^N i_n w_n$$

$$S_j = \frac{1}{1 + e^{net_j}} \quad (1)$$

where i and w represent the normalized value and weight for the input n , and S is the output j of the NN. In this case, the value W_0 represents the neuron bias, defined as the default neuron behavior when the other inputs are 0. Consequently, to incorporate the bias as input, the value of i_0 is always 1.

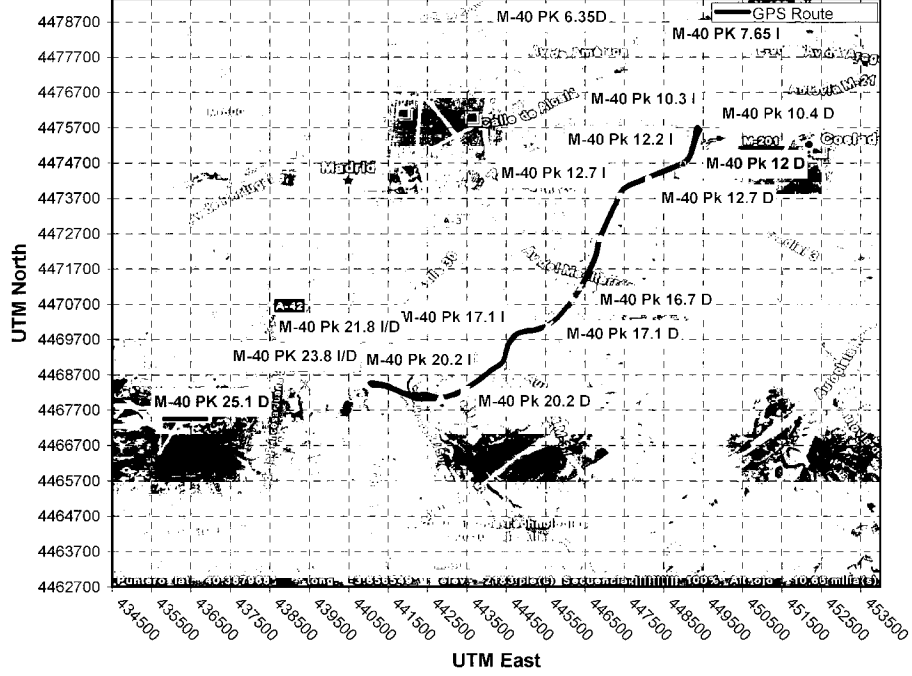


Fig. 2. Field operational test area and position of the infrastructure loop sensors.

TABLE II
GEOLOCATION OF THE KILOMETRIC POINTS OF THE M40 HIGHWAY

Road	MP	UTM North (m)	UTM East (m)
M-40	PK 6.35D	4479066.21	447310.87
M-40	PK 7.65I	4478353.35	448445.89
M-40	PK 12.2I	4474671.17	448987.95
M-40	PK 16.7D	4470833.63	446184.63
M-40	PK 17.1I/D	4470085.26	445567.53
M-40	PK 20.2I/D	4468243.68	443460.42
M-40	PK 21.8I/D	4468401.34	441273.48
M-40	PK 12D	4474671.17	448987.95
M-40	PK 12.7I/D	4474260.32	448112.99
M-40	PK 10.4I/D	4476429.97	449670.50
M-40	PK 25.1D	4468469.91	438610.96
M-40	PK 23.8I/D	4468435.95	439401.61

These NNs can be connected to one another in different ways and can be configured to use various learning methods to determine the best weights to be applied to the neuron connections to obtain an NN fit to a concrete problem. One of the most widely used architectures is the multilayer feedforward network with backpropagation learning rule, where the different layers of networks are completely interconnected, that is, each neuron of layer 1 is connected to every neuron of layer 2 and successively (see Fig. 3).

The fundamental problem of an NN is determining the value of the weights of the connections among neurons (w) for the output of the network to be the targeted one. This problem was solved in [28], and since then, this soft computing technique has been widely used in engineering, in the classification [29], [30], and in the approximation of unknown functions [31], [32]. It is

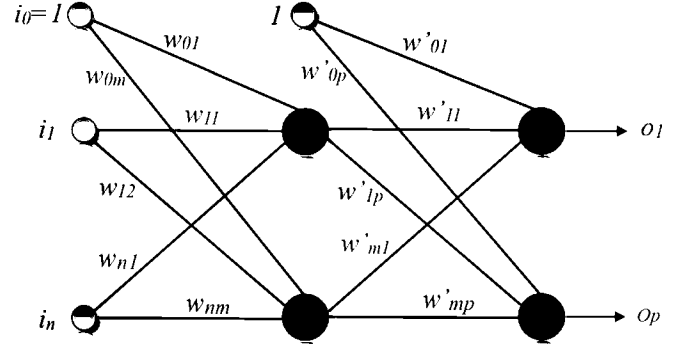


Fig. 3. Example of a multilayer feedforward network. Input i_0 to i_n , output o_1 to o_p , link weight w_{xy} , where x is the number of the neuron origin of the link, and y is the number of the neuron destination of the link. In this example, a single hidden layer is shown whose neurons are numbered from 0 to m .

also demonstrated that NNs can behave as universal function approximators [33], [34], depending on the adequate selection of the network architecture and the number of neurons in each layer. Unfortunately, there is no analytic procedure for making a prior estimate of these two parameters, so this selection depends on the experience of the designer.

Normalized root mean square error (NRMS) is used to calculate the measurement of the network error. This is a measurement of the standard deviation of the outputs of the network from the target values supplied by the training set of the network. NRMS is calculated through cross evaluation, that is, calculating the error with the data sets not used in the training of the network to obtain a realistic measurement of the error when the system is operating. NRMS is represented in

$$\text{NRMS} = \frac{1}{T} \sqrt{\frac{\sum_{i=1}^N (d_i - o_i)^2}{N}} \quad (2)$$

TABLE III
FCD ROUTE RECORDING DESCRIPTION

Hour	Routes recorded	Routes with congestion
9.00 – 11.00	6	2
14.00 – 16.00	6	1
20.00 – 22.00	6	2

where d and o are the desired corresponding values and the network output for the neuron i , respectively. T is the number of training examples.

In our case, we consider an NN as a soft computing method that is able to learn the behavior of a nonlinear system using real data to train its response. This training phase consists of the execution of a set of learning algorithms whose input will be the information from the infrastructure sensors (speed) of a segment of road and whose output is the information of a real floating car that has performed several routes on this road segment. In an overall description, the learning algorithms will deal with each input and will compare the corresponding output with the FCD output, calculating the related error. On the basis of this error, the training algorithm will modify the internal parameters of the NN and adapt them so as to reduce their value to a minimum. Once the NN is trained, it is able to generalize and generate the equivalent values of FCD with this minimum error when new infrastructure sensor information is introduced into the system.

With this philosophy, we set up our NN-based system to augment the amount of FCD on the road using the information from infrastructure sensors. Speed data obtained using infrastructure sensors or floating vehicles are different (in the first case, data are collected at one specific location, and in the second case, data are collected over time while driving). However, it should be taken into account that the algorithm only considers speed data from floating vehicles at certain locations and not the continuous signal over time. Then, the input values of the speed from the infrastructure (one sensor for each 1 km) were captured by the previously mentioned software robot at a rate of one measurement each 40 s. The values of speed from the probe car were taken every 100 m; this means that the system generates ten pieces of virtual car data for each loop sensor.

This FCD has been obtained from 18 routes across a 22.3-km segment of Madrid's M40 highway, recorded at several peak and low traffic times from 9.00 to 22.00 h, including free-flow traffic (13 routes) and congested traffic (five routes), as shown in Table III. In total, the source information from the FCD for training is 223 records per 18 routes. In a first step, we needed to organize the input and output information to develop an efficient training set. As a solution to organize these data, and due to our having more input than output information available, we developed a training set where each set of input values (loop sensors every 40 s) corresponded to the entire set of FCD values for this route (one track approximately every 20 min). To include a relative degree of freedom in the network, we also added an additional input value called the "route index," which represents the temporal sequence of the input data to allow the network to identify how the input varies over time for the same output. Then, the input/output patterns are obtained as

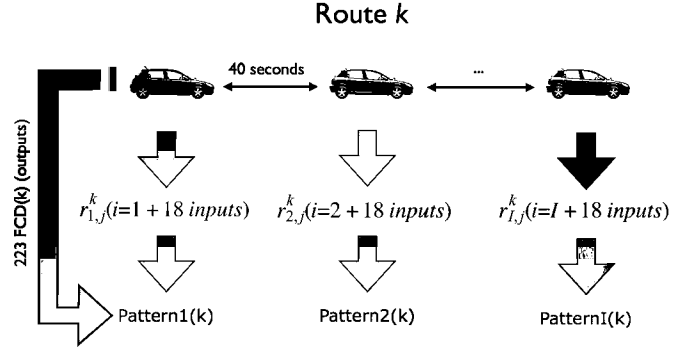


Fig. 4. Graphical representation of one pattern structure.

described next (see Fig. 4), where each value of the input of the network is represented as $r_{i,j}^{(k)}$ and defined as

$r_{i,j}^{(k)} \equiv$ is the measurement of the sensor j

in the data sample i th of the route k (3)

where $i \in \{1..I\}$, $j \in \{1..J\}$, and $k \in \{1..K\}$. In our case, J is the total number of loop infrastructure sensors captured (18), I is a variable number depending on the duration of the route k , and K is the total number of routes recorded (18). Additionally, we also add to the input set the value of the i th data sample (route index), as mentioned previously. Then, for each sample i of a route k , the input set is defined as $(i, r_{i,j}^{(k)})$, $j \in \{1 \dots 18\}$.

Similarly, each value of the output of the system can be defined as

$O_p^{(k)} \equiv$ is the measurement of the FCD

speed each 100 m in the route k (4)

where p indicates the sequence of the measurement, defined as $p \in \{1 \dots P\}$, with P being 223 in our case.

Each i th data sample defines an input vector for the same $O_p^{(k)}$ output vector for each route k (see Fig. 4), as shown in the example of (5), shown below, in this case, taking $k = 2$, i.e.,

$$\begin{pmatrix} 1 & r_{1,1}^{(2)} & r_{1,2}^{(2)} & \dots & r_{1,J}^{(2)} \\ 2 & r_{2,1}^{(2)} & r_{2,2}^{(2)} & \dots & r_{2,J}^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I & r_{I,1}^{(2)} & r_{I,2}^{(2)} & \dots & r_{I,J}^{(2)} \end{pmatrix} \begin{matrix} \xrightarrow{\text{Input1}} \\ \xrightarrow{\text{Input2}} \\ \vdots \\ \xrightarrow{\text{InputI}} \end{matrix} \times \left(O_1^{(2)} O_2^{(2)} \dots O_{223}^{(2)} \right) \begin{matrix} \xrightarrow{\text{Pattern1}} \\ \xrightarrow{\text{Pattern2}} \\ \vdots \\ \xrightarrow{\text{PatternI}} \end{matrix} \quad (5)$$

where each file contains the data set for each input i of route 2.

Due to the different durations of the K routes, the number of patterns I generated is not the same in every track k . This way, the temporal relation between inputs and outputs is explicit.

The total number of patterns is 196. In this case, we used cross validation of this total pattern set to estimate the network error in the exploitation phase. To do that, we reserved about 30% of the patterns as a test set. The final training set with this configuration comprised 133 (68% of the total pattern set) patterns. Each pattern comprised 19 inputs (18 loop sensors + route index) and 223 outputs (FCD). Additionally, we reserved 63 patterns (32% of the total pattern set), randomly chosen to be used as a test set. NRMS was calculated with this test set to assure the quality of the developed systems and that the output of the system could be equivalent to real FCD. The input and output values were normalized in $[-1, 1]$, using (6a), shown below and $[0, 1]$, using the (6b), shown below, range, respectively, i.e.,

$$\begin{aligned} a) \quad x'_i &= 2 \cdot ((x_i - \text{Max}(X)) / (\text{Max}(X) - \text{Min}(X)) - 1 \\ b) \quad x'_i &= (x_i - \text{Min}(X)) / (\text{Max}(X) - \text{Min}(X)) \end{aligned} \quad (6)$$

where x'_i represents the normalized value for a crisp input, and x_i is a member of one pattern of the total set of patterns (X). In our case, for the 18 loop sensors, $\text{Min}(X)$ is 0 km/h, the situation where every car is stopped, and $\text{Max}(X)$ is 100 km/h, the maximum speed authorized on this road. For the route index, $\text{Min}(X)$ is 1, and $\text{Max}(X)$ is K . The reason for selecting the normalization b) for the output is that the activation function of the NN is a sigmoidal one, whose value range is always between 0 and 1. On the other hand, the input values of a multilayer perceptron must be normalized to avoid saturation problems in the sigmoidal function. In our group, it is usual to normalize these values in the range $[-1, 1]$. In fact, this range can be extended or reduced to adapt to the author's requirements [35].

We also selected a feedforward NN with sigmoidal activation architecture and standard back propagation learning rule with a learning rate of 0.1. To implement the System for Automated Learning and Modeling of Operative Network experiment system, an NN operating environment developed by our group was used.

IV. RELATED EXPERIMENTS

A large set of experiments has been performed retrieving information from infrastructure sensors and floating vehicle data. In this paper, we present three of these experiments on a route along the M40 highway in increasing and diminishing directions, with a total length of 22 km.

The first experiment was used to analyze the linearity of the training data as well as to be used as a control trial. To do this, we implemented a simple perceptron (no hidden layer) with 19 input neurons and 223 output neurons. In this case, the learning was stopped after 150 epochs (training iterations) with an RMS error of 0.156 (the average error between the NN output and the FCD measured output was 15.6%). Similarly, the RMS error when compared with the test set was 0.187. Comparing these two results, it is clear that the RMS is too high for it to be considered as a proper behavior of the network. This means that there is an important nonlinear component in the system to solve that must be considered to obtain good results

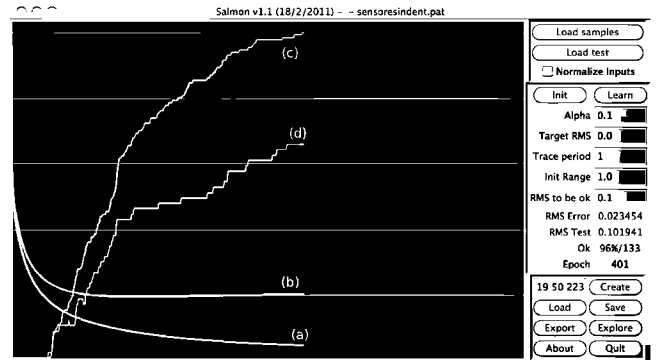


Fig. 5. Error evolution during the learning in experiment 2. (a) RMS error in training set. (b) RMS error in test set. (c) Optimal cases in training set. (d) Optimal cases in test set.

since it is well known that the simple perceptron can only deal with linear systems.

To adapt the network to the inherent nonlinear behavior of the FCD augmentation system proposed, the NN architecture was modified by adding one hidden layer with 50 neurons between the input and output layers. This output layer would allow the system to approximate nonlinear systems like this one. With this configuration, the NN was trained for 401 epochs, with an RMS error of 0.0234 being obtained (the average error between the NN output and the FCD measured output was 2.34%). Similarly, during the execution of the NN, using the training set as input, the RMS error was 0.1019 with a standard deviation of 0.00198, better than with the previous experiment. This is due to the multilayer NN being nonlinear function approximators that allow a better fit to the input data.

Tests with a larger number of hidden units did not improve the results.

The evolution of the algorithm is shown in Fig. 5. Lines (a) and (b) indicate the evolution of the normalized RMS error in each iteration of the algorithm (epoch). Lines (c) and (d) show the percentage of examples with an error of less than 0.1 in each of the 223 outputs. On the y -axis, (a) and (b) are represented on a scale normalized between 0 and 0.52, and in (c) and (d), the scale is defined from 0% to 100%. The x -axis represents the number of epochs executed $[0-700]$.

A typical effect in the evolution of the error in the training is that there is a point (around epoch 150) where the error in tests starts rising while the training error continues to drop. In theory, this is the optimum point to stop the training, since here we can find the best generalization capacity, that is, a network configuration that is optimal for operation.

In the third experiment, an optimization over the second experiment was developed. In this experiment, a second hidden layer was added, with the 19-30-30-223 architecture and trained for 234 epochs. In this case, the RMS error was 0.02345 (the average error between the NN output and the FCD measured output was 2.345%). Similarly, during the execution of the NN using the training set as input, the RMS error was 0.0852 with a standard deviation of 0.0015, better than with the previous experiment, as Fig. 6 shows.

These RMS values are obtained around epoch 150. In this figure, it is shown that this NN configuration improves the

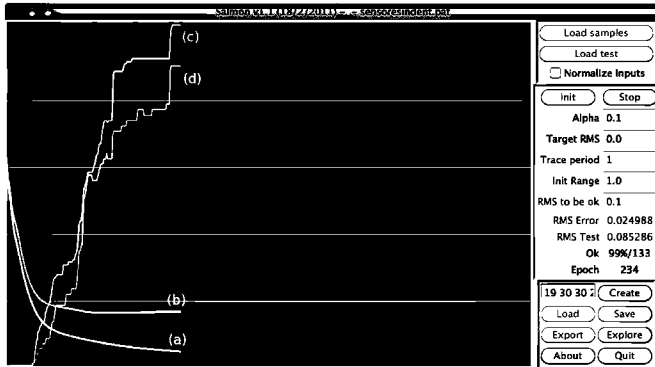


Fig. 6. Error evolution during learning in experiment 3. (a) RMS error in training set. (b) RMS error in test set. (c) Optimal cases in training set. (d) Optimal cases in test set.

TABLE IV
VIRTUAL FCD NN ARCHITECTURE RESULTS

Input Neurons	Hidden layer Neurons	Output Neurons	RMS Learning error	RMS Test error	RMS Test error unbiased variance
19	0	223	15.6 km/h (15.6%)	18.7 km/h (18.7%)	0.529 km/h
19	50	223	2.34 km/h (2.34%)	10.19 km/h (10.19%)	0.198 km/h
19	30-30	223	2.345 km/h (2.345%)	8.52 km/h (8.52%)	0.150 km/h

results of experiment 2 due to a larger degree of freedom of the network to fit with the output function of the training data. Up to 40 experiments with several hidden layers and architectures were developed without obtaining better results.

An RMS in-test error of 0.0852 (8.5%) can be interpreted as the average standard deviation that we can find between the speed estimated by the NN for the virtual FCD (see Table IV) and the actual speed of the floating car, measured by the vehicle sensor for an unknown system data set. Taking into account that normalization has been calculated in the data to establish it in the range [0, 1], we can estimate that the real range of the variables, once denormalized, corresponds to a standard deviation of 2.13 km/h.

V. CONCLUSION

In this paper, we have designed, implemented, and tested a system to augment the FCD available in real time using information generated by the infrastructure loop sensors as source and NNs as processing method. This system has been trained and validated using information from an instrumented probe car circulating along a 22-km segment of Madrid's M40 highway, where 26 infrastructure loop sensors are also installed. Eighteen of these 26 sensors are continuously enabled, and these were the ones used to feed the neural system. However, we still keep open the possibility of fixing some of the disabled sensors and including their information as an additional source in the system, only adding the corresponding neuron in the input layer retraining the network.

Several NN configurations have been tested and trained to obtain an optimal configuration that represents a set of virtual FCDs with a minimum error from the real FCD. This is clearly a nonlinear problem demonstrated by the poor results obtained in the perceptron experiment. Additionally, acceptable results were obtained by adding a hidden layer of 50 neurons, and these results were even improved by using two hidden layers with 30 and 30 neurons. Additionally, it is well known that NNs have a good capacity for generalizing. This means that the system could have a high potential to behave correctly in nonrecurrent congestions due to incidents not included in the training set. However, this case will be tested in future works.

This system can be used as an information source for any kind of application that requires FCD but that does not have sufficient resources available to instrument the necessary number of vehicles, which is one of the main drawbacks for obtaining FCD and even for correcting systematic error in infrastructure sensors. Micro- and macroscale traffic simulations and traffic planning can also benefit from the system presented in this paper by using it as an information source. Finally, the presented system is anticausal because the network inputs are the measurements of the infrastructure sensors used to calculate the output in the same instant. In future works, it should be interesting to study the possibility of using as inputs the measurements taken in previous instants to elaborate the status of the virtual FCD in the future, building, in consequence, a causal system.

REFERENCES

- [1] R. Elvik and T. Vaa, *The Handbook of Road Safety Measures*. Amsterdam, The Netherlands: Elsevier, 2004.
- [2] P. Lertworawanich, P. Mongkonrattanawong, and L. Elefteriadou, "Effects of variable message signs on traffic operations," in *Proc. Eur. Congr. Intell. Transp. Syst.*, Jun. 2007, p. 2355.
- [3] G. Reichart, S. Friedmann, C. Dorrer, H. Rieker, E. Drechsel, and G. Wermuth, "Potentials of BMW driver assistance to improve fuel economy," in *Proc. FISITA World Automotive Congr.*, Paris, France, Sep. 27–Oct. 1, 1998.
- [4] P. J. T. Venhovens, J. H. Bernasth, J. P. Löwenau, H. G. Rieker, and M. Schraut, "The application of advanced vehicle navigation in BMW driver assistance systems," presented at the Int. Congr. Expo., Detroit, MI, 1999, SAE Paper 1999-01-0490.
- [5] K. F. Petty, P. Bickel, M. Ostland, J. Rice, F. Schoenberg, J. Jiang, and Y. Ritov, "Accurate estimation of travel times from single-loop detectors," *Transp. Res. A*, vol. 32, no. 1, pp. 1–17, Jan. 1998.
- [6] B. Coifman, "Estimating travel times and vehicle trajectories on freeways using dual loop detectors," *Transp. Res. A*, vol. 36, no. 4, pp. 351–364, May 2002.
- [7] A. M. El-Geneidy and R. L. Bertini, "Toward validation of freeway loop detector speed measurements using transit probe data," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Washington, DC, 2004, pp. 779–784.

- [8] B. Auffray, K. A. Tufte, Z. Horowitz, S. Matthews, and R. L. Bertini, "Evaluation of single-loop detector vehicle classification algorithms using an archived data user service system," in *Proc. ITE District 6 Annu. Meeting*, Honolulu, HI, Jun. 2006.
- [9] H.-E. Lin, M. A. P. Taylor, and R. Zito, "A review of travel-time prediction in transport and logistics," in *Proc. Eastern Asia Soc. Transp. Stud.*, 2005, vol. 5, pp. 1433–1448.
- [10] C.-H. Wu, J.-M. Ho, and D. Lee, "Travel time prediction with support vector regression," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 276–281, Dec. 2004.
- [11] W. Huber, M. Lädke, and R. Ogger, "Extended floating-car data for the acquisition of traffic information," in *Proc. 6th World Congr. Intell. Transp. Syst.*, 1999, pp. 1–9.
- [12] C. Demir, B. S. Kerner, R. G. Herrtwich, S. L. Klenov, H. Rehborn, M. Aleksic, T. Reigber, M. Schwab, and A. Haug, "FCD for urban areas: Method and analysis of practical realizations," in *Proc. World Congr. Intell. Transp. Syst.*, Madrid, Spain, Nov. 2003.
- [13] S. Breitenberger, K. Bogenberger, M. Hauschild, and K. Laffkas, "Extended floating car data—An overview," in *Proc. World Congr. Intell. Transp. Syst.*, Madrid, Spain, Nov. 2003.
- [14] L. Lin, T. Osafune, and M. Lenardi, "Floating car data system enforcement through vehicle to vehicle communications," in *Proc. 6th Int. Conf. ITS Telecommun.*, Jun. 2006, pp. 122–126.
- [15] H. Lahrmann, "Floating car data for traffic monitoring," in *Proc. i2TERN Conf.*, Aalborg, Denmark, Jun. 2007, pp. 20–21.
- [16] S. Maerivoet and S. Logghe, "Validation of travel times based on cellular floating vehicle data," in *Proc. Eur. Congr. Intelligent Transport Systems*, Aalborg, Denmark, Jun. 2007.
- [17] S. Messelodi, M. Modena, M. Zanin, F. G. B. De Natale, F. Granelli, E. Betterle, and A. Guarise, "Intelligent extended floating car data collection," *Expert Syst. Appl. Int. J. Archive*, vol. 36, no. 3, pp. 4213–4227, Apr. 2009.
- [18] E. Brockfeld, P. Wagner, S. Lorkowski, and P. Mieth, "Benefits and limits of recent floating car data technology—An evaluation study," in *Proc. 11th World Conf. Transp. Res.*, Berkeley, CA, pp. 2007-06-24–2007-06-28.
- [19] R. L. Cheu, C. Xie, and D. Lee, "Probe vehicles population and sample size for arterial speed estimation," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 17, no. 1, pp. 53–60, 2002.
- [20] J. Hong, X. Zhang, Z. Wei, L. Li, and Y. Ren, "Spatial and temporal analysis of probe vehicle-based sampling for real-time traffic information system," in *Proc. IEEE Intell. Veh. Symp.*, Istanbul, Turkey, Jun. 13–15, 2007, pp. 1234–1239.
- [21] Srinivasan, Karthik, and Jovanis, "Determination of number of probe vehicles required for reliable travel time measurement in urban network," *Transp. Res. Rec.*, vol. 1537, pp. 15–22, 1996.
- [22] A. Simroth and H. Zähler, "Travel time prediction using floating car data applied to logistics planning," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 1, pp. 243–253, Mar. 2010.
- [23] J. E. Naranjo, F. Jiménez, F. J. Serradilla, and J. G. Zato, "Comparison between floating car data and infrastructure sensors for traffic speed estimation," in *Proc. 13th Int. IEEE Conf. Intell. Transp. Syst., Workshop Emergent Cooperative Technol. Intell. Transp. Syst.*, 2010.
- [24] R. Wang and H. Nakamura, "Expressway travel time estimation by using the combination of detector and probe vehicle data," in *Proc. World Congr. Intell. Transp. Syst.*, Madrid, Spain, Nov. 2003.
- [25] A. Bhaskar, "Fusing loop detector and probe vehicle data to estimate travel time statistics on signalized urban networks," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 26, no. 6, pp. 433–450, Aug. 2011.
- [26] F. Soriguera and F. Robusté, "Requiem for freeway travel time estimation methods based on blind speed interpolations between point measurements," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 1, pp. 291–297, Mar. 2011.
- [27] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Washington, DC: Spartan, 1961.
- [28] D. E. Rumelhart, E. H. Geoffrey, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel distributed processing: Explorations in the microstructure of cognition*, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA: MIT Press, 1986.
- [29] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Oxford Univ. Press, 1995.
- [30] A. A. Salah, E. Alpaydin, and L. Akarun, "A selective attention-based method for visual pattern recognition with application to handwritten digit recognition and face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 420–425, Mar. 2002.
- [31] M. W. Gardner and S. R. Dorling, "Artificial neural networks (the multi-layer perceptron)—A review of applications in the atmospheric sciences," *Atmosp. Environ.*, vol. 32, no. 14/15, pp. 2627–2636, Aug. 1, 1998.
- [32] K. Xu, M. Xie, L. C. Tang, and S. L. Ho, "Application of neural networks in forecasting engine systems reliability," *Appl. Soft Comput.*, vol. 2, no. 4, pp. 255–268, Feb. 2003.
- [33] G. Cybenko, "Approximations by superpositions of sigmoidal functions," *Math. Control, Signals, Syst.*, vol. 2, no. 4, pp. 303–314, 1989.
- [34] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Netw.*, vol. 4, no. 2, pp. 251–257, 1991.
- [35] W. J. Jansen, M. Diepenhorst, J. A. G. Nijhuis, and L. Spaanburg, "Assembling engineering knowledge in a modular multi-layer perceptron neural network," in *Proc. Int. Conf. Neural Netw.*, Houston, TX, 1997, pp. 232–237.



José E. Naranjo was born in Orense, Spain, in 1975. He received the B.E., M.E., and Ph.D. degrees in computer science engineering from the Technical University of Madrid (UPM), Madrid, Spain, in 1998, 2001, and 2005, respectively.

From 1999 to 2007, he was with the Instituto de Automática Industrial, Madrid. In 2007, he was an Associate Professor with the School of Computer Science, UPM, and became part of the Region of Madrid Excellence Group Safety in Automobile Vehicles with special interest in Reduced Mobility Persons. He is currently the Vice-Director of the Intelligent Systems Department, UPM. His research interests include intelligent systems, fuzzy logic control, intelligent transport systems, usability, and accessibility.



Felipe Jiménez received the Master's degree in industrial engineering from the Polytechnic University of Madrid (UPM), Madrid, Spain, in 2001, the Master's degree in physical science from the National University of Distance Education of Spain, Madrid, in 2005, and the Ph.D. degree in mechanical engineering from UPM in 2006.

He is currently an Associate Professor with UPM and the Head of the Intelligent Systems Unit, University Institute for Automobile Research, UPM. He has been involved in several research projects and has developed engineering studies for relevant national firms. His fields of interest are the automotive industry, vehicle safety, mechanical design, driver-assistance systems, and intelligent transport systems. He is the author of several papers in relevant scientific journals and has participated in several national and international congresses.



Francisco J. Serradilla was born in Sevilla, Spain, in 1965. He received the B.S. and Ph.D. degrees in computer science from the Universidad Politécnica de Madrid (UPM), Madrid, Spain.

He is currently a Lecturer with the Department of Applied Intelligent Systems, Universidad Politécnica de Madrid, and the Director of the UPM research group in Intelligent Agents and Ubiquitous Computing. His main research fields are intelligent information agents, including software robots, search engines, and collaborative recommender systems. He is working on the collaborative filtering kernel of web site filmaffinity.com and is developing ubiquitous tools to interact with this site using iPhone devices.



José G. Zato was born in La Coruña, Spain, in 1944. He received the B.E. and M.E. degrees in physics engineering in 1973 and the Ph.D. degree in physics in 1982 from the Universidad Complutense de Madrid, Madrid, Spain.

Since 1989, he has been a Professor with the School of Computer Science, Technical University of Madrid. He is the Head of the Intelligent Systems for Accessible Mobility and Communication Group, with a wide experience in leading regional, national, and international projects, supported by public as well as private funds. His research interest includes intelligent systems, usability and accessibility, and rehabilitation technologies.