Joint Feature Selection and Parameter Tuning for Short-term Traffic Flow Forecasting based on Heuristically Optimized Multi-layer Neural Networks

Ibai Laña[1], Javier Del Ser[1,2,3], Manuel Vélez[2], and Izaskun Oregi[1]

[1] TECNALIA, E-48160 Derio, Spain,
{ibai.lana,javier.delser,izaskun.oregui}@tecnalia.com
[2] University of the Basque Country UPV/EHU, 48013 Bilbao, Spain,
{javier.delser, manuel.velez}@ehu.eus
[3] Basque Center for Applied Mathematics (BCAM), 48009 Bilbao, Spain

Abstract. Short-term traffic flow forecasting is a vibrant research topic that has been growing in interest since the late 70's. In the last decade this vibrant field has shifted its focus towards machine learning methods. These techniques often require fine-grained parameter tuning to obtain satisfactory performance scores, a process that usually relies on manual trial-and-error adjustment. This paper explores the use of Harmony Search optimization for tuning the parameters of neural network jointly with the selection of the input features from the dataset at hand. Results are discussed and compared to other tuning methods, from which it is concluded that neural predictors optimized via the proposed heuristic wrapper outperform those tuned by means of naïve parametrized algorithms, thus allowing for longer-term predictions. These promising results unfold potential applications of this technique in multi-location neighbor-aware traffic prediction.

Keywords: Traffic forecasting; Neural networks; Bioinspired heuristics

1 Introduction

Forecasting traffic conditions is a key element in the development of Intelligent Transport Systems (ITS), providing the means to implement management (ATMS) and information (ATIS) systems for both road managers and users. Anticipating future traffic can aid the first to regulate signals, lanes and to cope with congestion, and the latter to plan travels and select the best routes to their destinations. For decades, researchers have built traffic models to predict volume, occupancy, speed, travel time or level of service, consisting of elements from time-series analysis in the beginning, and later evolving to non-parametric and machine learning models such as kNN, Artificial Neural Networks (ANN), Support Vector Machines (SVM), Bayesian Networks and Fuzzy Logic models, among others [1]. In the last decade an upsurge of traffic-related data has become available, which has lead, along with advances in computational technologies and machine learning techniques, to a noticeable research drift towards data-driven approaches, with more diverse and abundant data sources that conform large databases with hidden knowledge to be discovered by pattern recognition algorithms. Changes are also observable in predicted variables, which tend to become more user-friendly (travel time versus volume) and in the scope of predictions which are increasingly urban and network-wide [2].

In this context, ANNs and their combination with other methods have been extensively used with relative success over naïve (historic average and last measurement predictions) and time-series models [3–7], fueled by prior literature evincing that ANNs are more responsive to changes in data [8]. However, neural networks behave in a black-box manner that hinders their understanding. Furthermore, their internal structure and training process is known to be a slow and inefficient trial-and-error procedure. In this regard, seeking the optimal neural network structure started to be automated with bio-inspired heuristics [9]. In particular for the transport sector, Genetic Algorithms (GA) were first explored for refining the calibration parameters that enhanced neural network behavior without linking them to particular traffic characteristics, thus increasing generalization capacity of the model [10]. Researchers have used GA optimization [10–12] and other bio-inspired methods for configuring neural networks, such as Particle Swarm Optimization (PSO) in recent short-term traffic prediction literature [13, 14], with significant results.

This research work will delve into the use of an specific meta-heuristic algorithms as a hyperparameter tuning wrapper for complex neural networks. To be concise the so-called Harmony Search (HS) algorithm [15] will be used to calibrate a multi-layer perceptron (MLP), which to the knowledge of the authors has not been so far used to this end. The proposed scheme will yield a set of neural network configurations for which the "best" one will balance the trade-off between accuracy and MLP training time. The meta-optimized MLP model will be used to predict traffic flow in an urban center location in Madrid, where recent reports have revealed a high seasonal dependence and a stable behavior that complicates outperforming naïve approaches [16]. 15-minute resolution traffic flow data of one entire year will be explored with different time windows and prediction horizons to show the promising performance of the models optimized by means of our proposed wrapper.

2 Materials and Methods

Flow is one of the most predicted traffic features [2] and embodies the data substrate on which most traffic models are built. Road traffic can be quantified by the amount of vehicles per hour crossing a certain link of the road network and it is measured, among others, with Automatic Traffic Recorders (ATR) sensors, magnetic loops embedded in roads that are able to count how many vehicles pass over them. Around 3700 of these ATRs are at the disposal of the Madrid City Council, with readings taken every 5 minutes. Traffic flow and other metric measurements obtained from traffic counts are published in a live feed in the Madrid Open Data portal [17]. The portal also provides historically aggregated traffic flow data with 15 minute granularity. From the latter data collection, a year worth of traffic flow data has been extracted and processed so as to compose the target dataset tackled in this work. These data correspond to the year 2015 for model building (training and validation), whereas the first three months of 2016 are left out to test the generalization performance of the optimized predictive model.

Table 1. Description of the loops under consideration

Loop	Location	Details
A	C. Alcalá and C. O'Donnell	City centre, intersection in a 4-lane road
B	Av. Monforte de Lemos and Av. Betanzos	Urban residential area

This research work focuses on two loops placed in a residential and a center area of Madrid (Table 1). Thus prediction models are built for locations with very dissimilar traffic and optimization results can be compared and assessed disregarding any particular set of traffic conditions. As noted in Figure 1, there are substantial differences between both locations, introducing diversity to the study. Besides, a completeness criterion has been used to select the sensor locations: for all the loops deployed in this city, the available data are incomplete, and in some cases invalid, but some loops are considerable more complete than others. Both loops provide ca. 30000 valid flow readings corresponding to the whole year which, along with timestamps, are used to build the datasets based on three specifications: 1) step, i.e. the time between two readings (in this case 15 minutes); 2) depth or window size, namely, the time span of past observations that are used in each instance; and 3) prediction horizon, i.e. the number of steps into the future for which the prediction is made.

For a certain timestamp these readings configure a dataset with n features, which are the n observations prior to the timestamp, being n the window size. The target variable is defined by the observation y, corresponding to y slots after the timestamp, as explained in Figure 3. Initially, a dataset for each loop has been created with default parameters. To begin with, the forecast horizon is set to 4 and 8 time steps (i.e. 1 and 2 hours into the future), which is in accordance with the literature on short-term traffic prediction with horizons normally shorter than one hour [1, 2]. The window size is fixed to 8 steps, so the predictions are based in the previous 2 hours of observations. Indeed tailoring the window size is crucial for the predictive model, but unfortunately its value is strongly determined by the scenario at hand as the information provided by features in the window can be very divergent in different traffic areas [19]. Figure 2 shows the evolution of the average traffic over the slots of a day; in consonance with the stability of the observations, it is expected that for a residential loop this parameter becomes less relevant. Adjusting this window is a vibrant research topic [20] for which optimization strategies as the one presented in this paper

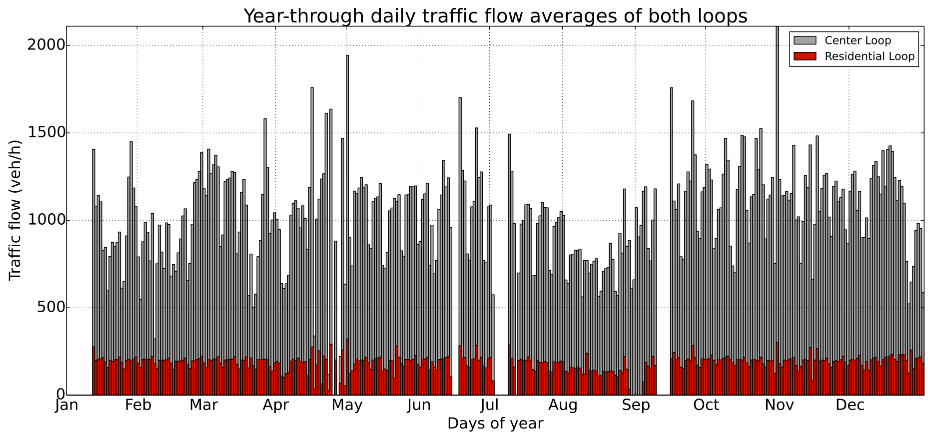

Fig. 1. Comparison of the day-average traffic flow registered by the two considered loops along the year. The x axis represents each one of the days in the sample. The loop in the city center has well-defined periods, while the loop in a residential area is more stable through the year. Empty days represent no available data.

take a key role in finding the best window sizes. In this research work window sizes of each location will be jointly optimized (along with regression model parameters) by the HS heuristic.

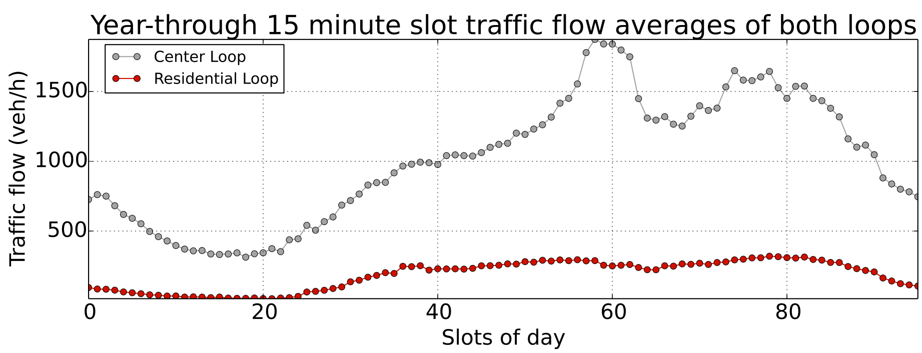

Fig. 2. Year-averaged traffic flow on each slot of the day for both loops.

Once both datasets are created with above defined terms, they are splitted in 10 random parts, leaving 9 for training and 1 for test. This process is repeated 10 times, feeding each of the training subsets to the regression algorithm, and testing it against the test subset. A R^2 performance metric is obtained in each execution, and they are averaged to obtain the model performance, as depicted in Figure 3.

2.1 Regression and Optimization

Our regression algorithm is a multilayer perceptron. Artificial Neural Networks (ANNs) have been extensively used for traffic forecasting applications, being the most extended non-parametric method used [1]. Multilayer perceptron falls under the feed-forward networks category [21], and is able to model highly non-linear patterns. An input vector is mapped to output through layers of weighted neurons with activation functions. In this research, logistic activation is used, whereas the rest of the MLP architecture is left undefined for its optimization through the proposed scheme. In the previous literature it is usual that the MLP configuration hinges on a trial-and-error process that does not always yield a configured MLP that outperforms other approaches such as ARIMA models. For this reason a default parameter setting often used in other related contributions has been used as an initial baseline for comparison. Then, the regressor model is wrapped by means of an optimization algorithm that iteratively refines such a baseline configuration. In the last decade, optimization techniques have allowed a considerable improvement in ANNs performance [20]. However, to the best of the authors' knowledge there is no previous evidence of Harmony Search applied to the optimization of neural networks, nor has this heuristic been used to select features jointly with the model configuration.

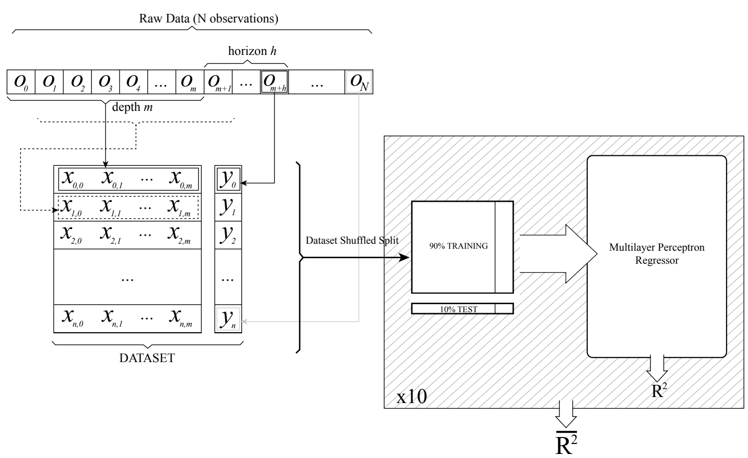

Fig. 3. Definition of the core predictive model.

Evolutionary algorithms are heuristic search techniques, and their operation grounds on the evolution of a group of candidate solutions towards progressively better individuals, with their *quality* defined by a fitness function. This evolutionary process finds its inspiration in the concept of natural evolution, encompassing selection, crossover and mutation of individuals [18]. Harmony Search (HS) can be thought of as being a specific yet differently motivated evolutionary solver, which imitates the seek of harmonies in a musical improvisation process to optimize a set of variables under a measure of quality. To do so, a vector of parameters and their boundaries are defined. Besides, HS wrapper is arranged to seek the optimal depth of each dataset. Figure 4 shows a harmony search implementation that jointly optimizes the feature selection and neural network parameter settings.

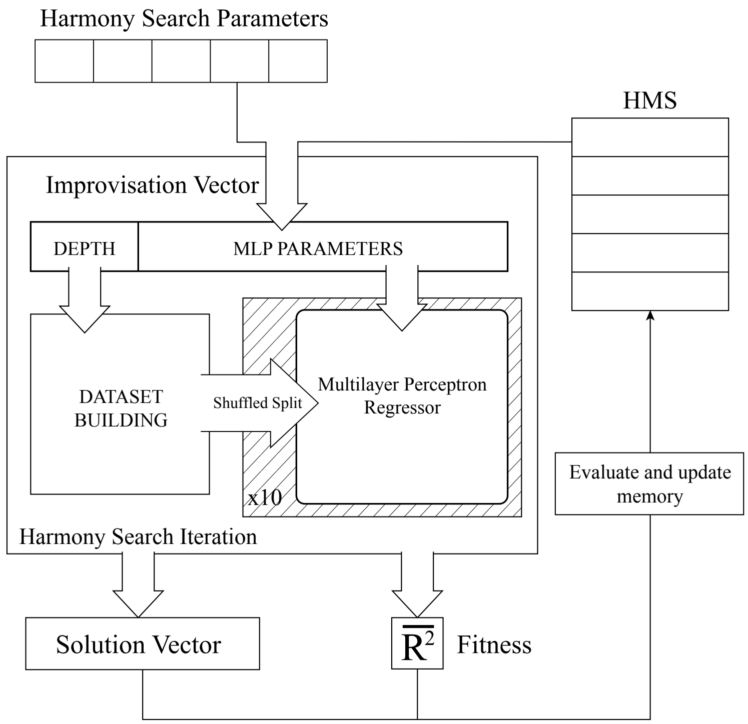

Fig. 4. Definition and operation of the HS wrapper.

The HS algorithm requires input parameters that define its memory (hms), number of improvisations (I), and fitness function. At each iteration all raw data are used to build a new different dataset depending on the window size selected by the solution vector provided by HS, and evaluates the performance of the regression model by averaging over the scores produced by 10-fold cross-validation of the built dataset. The average R^2 metric obtained from the MLP over the 10 folds is the fitness function to maximize. In terms of computation a complete iteration takes around 2 minutes in an Intel i7 processor, which amounts up to an average of 720 iterations per day on a dedicated machine. We have set $hms = 25$ harmonies in the pool of candidate solutions, and a total of $I = 2000$ iterations. Besides these parameters we have set $hmcr = 0.75$, $par = 0.2$, $mpai = 3$ (maximum pitch adjustment for discrete variables) and $mpap = 0.25$ (maximum pitch adjustment for continuous variables).

Table 2 shows the MLP parameters chosen to optimize with their default values and maximum and minimum boundaries set for the experiment. Depth value has been set to take values between 4 and 48 steps, or 1 and 12 hours of previous observations. For the neural network, default values are taken as a reference and for continuous variables, with maxima and minima one order of magnitude higher and lower, respectively. For discrete variables, boundaries are defined considering their purpose and operation.

Table 2. Optimized set of variables default values and boundaries.

Parameter	Type	Default	Min.	Max.
Depth	Discrete	8	4	48
Hidden Layer Sizes	Discrete	100	1	300
Alpha	Continuous	0.001	0.01	0.0001
Max. Iterations	Discrete	200	10	1000
Tolerance	Continuous	0.0001	0.00001	0.0001
Learning Rate Init.	Continuous	0.001	0.0001	0.01
Epsilon	Continuous	1e-08	1e-9	1e-07

After the whole process depicted in Figure 4 is executed for both loops and horizons (4 and 8 steps), the obtained models are trained with the selected parameters and a dataset formed by the entire year 2015, and tested against the dataset corresponding to the first trimester of 2016.

3 Experimental Results

One of the main advantages of using HS to tune a predictive model is the possibility of adjusting its continuous parameters without discretizing them. In our model an automated exploratory search would require a discretization of four of the parameters that are to be estimated by the HS algorithm. With our set of parameters an exhaustive search would need a number of iterations given by

$$I = \left(\prod_{m=1}^{M_D} DPR_m\right) \cdot \left(\prod_{m'=1}^{M_C} CPR_{m'}\right), \tag{1}$$

where DPR_m is the number of possible values for the discrete parameter $m \in \{1, \ldots, M_D\}$, and $CPR_{m'}$ is the number of discretization steps for the continuous parameter $m' \in \{1, \ldots, M_C\}$, with M_D and M_C denoting the number of discrete and continuous parameters, respectively. In this case, only evaluating the combinations of discrete variables would require more than 26 million of iterations, each requiring an average of 2 minute to be executed. HS, instead, has produced a result in less than three days.

Beyond the above gains in terms of computational effort, the tuning achieved by HS has provided improved performance results over the baseline configuration. Figure 5 shows scatter plots of both loops with both temporal horizons, and R^2 metric obtained for each dataset by using default parameters for the MLP and window size. In both loops, two hour horizon produces less accurate predictions (less fit to the line), and also, loop B, in a residential area, produces better overall performance, as a result of traffic stability there (figures 1 and 2).

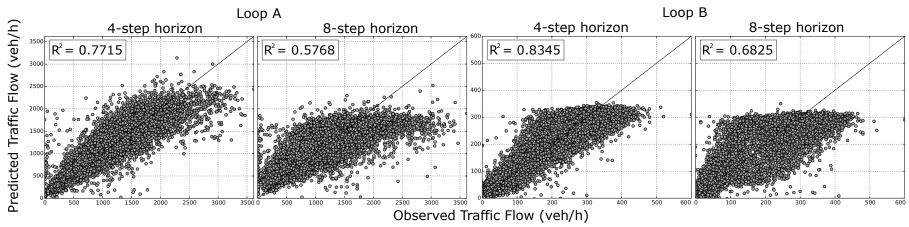

Fig. 5. Performance of the default MLP for both loops and temporal horizons.

Table 3. Optimized solution vectors for each loop and horizon

Loop	Horizon	Depth	H. L. Size	Alpha	Max. Iter.	Tolerance	L.R. Init.	Epsilon
A	4	35	285	2.96e-4	647	5.46e-5	4.94e-4	5.01e-9
A	8	42	160	9.25e-4	719	4.59e-5	5.26e-4	8.4e-8
B	4	40	262	3.67e-4	828	3.24e-5	8.54e-4	2.07e-8
B	8	45	81	6.84e-5	641	3.4e-5	3.4e-3	4.84e-8

After evolutionary search for optimal parameters, the algorithm has produced solution vectors presented in Table 3, which give rise to the predictive performance plots depicted in Figure 6. All cases achieve an increment in R^2 metric, more noticeable in the 8-step prediction with almost 27% and 25% of R^2 performance relative gains. Due to their better performance with the default configuration, the improvement experienced by the MLP models is less significant – yet still notorious – in the case with 4-step horizon prediction. Window size is considerably larger than initially estimated, with up to 42 steps – 10.5 hours – for the 8-step prediction in location A. This means that a forecast for a certain timestamp is based in values up to 12.5 hours before, which according to Figure 1 will always include values from both top and bottom ends of the curve. This result unveils that data are stable along days and that, taken the same range, the 42 steps that predict a value on e.g. a Saturday are similarly good to forecast the same timestamp value on a Monday.

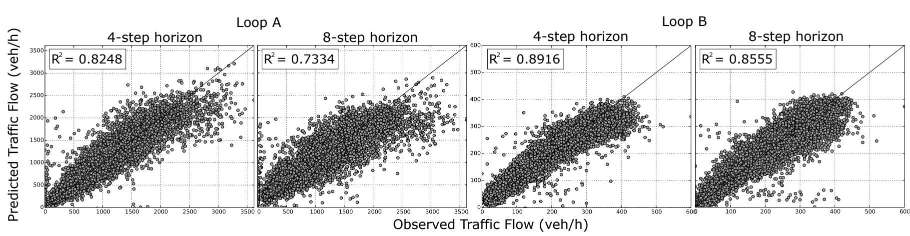

Fig. 6. Performance of optimized multilayer perceptron on both loops with both temporal horizons.

Scatter plots are also helpful to visualize how predictions approach their exact values, specially for location B. Loop A, being placed in a very active zone of Madrid, produces more outliers which are difficult to predict, especially in the right part of the graphs where high traffic flow observed values are predicted lower. Indeed the regressor is unable to forecast a traffic peak flow of 3500 vehicles per hour, probably on account of an incident in the road at hand. All in all, these results buttress the need for an optimization wrapper when building a traffic forecasting model so as to configure it optimally.

4 Conclusions and Future Work

Prediction horizons greater than 1 hour are rarely explored in literature, being usually inefficient in urban areas as there are multiple immediate factors that influence traffic, ultimately making predictions useless. Nonetheless, this experiment has shown that in the absence of extraordinary circumstances, good prediction scores are attainable for longer-term predictions when the window sizes and algorithms are optimized. In this manuscript real traffic data in two different locations of Madrid (Spain) have been used to prove that the HS solver can efficiently optimize the parameters of a neural network and simultaneously select the depth of the input data to the model. This method allows for the automation of the tuning process of the algorithm, improving its overall performance in terms of computational complexity and predictive accuracy.

This work paves the way towards multi-location forecast. Network-wide traffic forecasting methods are gaining momentum, specially in urban environments. Part of the vehicles passing over one loop are likely to have passed through surrounding loops, thus the measurements taken in one loop might influence their neighboring loops. Without detailed origin-destination matrices, estimating this influence is an arduous task. The experiment presented in this paper can be extended to a multi-loop problem, where instances of the dataset are formed by observations of different loops, and the windows sizes of each loop are optimized. This extended implementation is expected to bring better performance levels, but also to unveil influence areas of each magnetic loop over the whole city.

Acknowledgments

This work has been supported by the Basque Government through the ELKARTEK program (ref. KK-2015/0000080 and the BID3ABI project).

References

1. Van Hinsbergen CP, Van Lint JW, Sanders FM (2007) Short term traffic prediction models. In: Proceedings of the 14th World Congress on Intelligent Transport Systems (ITS). Beijing, pp 22–41.
2. Vlahogianni EI, Karlaftis MG, Golias JC (2014) Short-term traffic forecasting: Where we are and where we're going. Transportation Research Part C: Emerging Technologies 43:3–19.

3. Dunne S, Ghosh B (2011) Regime-based short-term multivariate traffic condition forecasting algorithm. Journal of Transportation Engineering 138(4):455–466.
4. Yang H, Dillon TS, Chen YPP (2015) Evaluation of Recent Computational Approaches in Short-Term Traffic Forecasting. In: IFIP International Conference on Artificial Intelligence in Theory and Practice. Springer International Publishing, pp 108–116.
5. Zhang W, Xiao R, Deng J (2015) Research of Traffic Flow Forecasting Based on the Information Fusion of BP Network Sequence. In: International Conference on Intelligent Science and Big Data Engineering. Springer International Publishing, pp 548–558.
6. Meng M, Shao CF, Wong YD, Wang BB, Li HX (2015) A two-stage short-term traffic flow prediction method based on AVL and AKNN techniques. Journal of Central South University 22:779-786.
7. Tang J, Zou Y, Ash J, Zhang S, Liu F, Wang Y (2016) Travel Time Estimation Using Freeway Point Detector Data Based on Evolving Fuzzy Neural Inference System. PloS one 11(2)
8. Smith BL, Demetsky MJ (1994) Short-term traffic flow prediction: neural network approach. Transportation Research Record 1453:98–101.
9. Abdulhai B, Porwal H, Recker W (1999) Short term freeway traffic flow prediction using genetically-optimized time-delay-based neural networks. California Partners for Advanced Transit and Highways (PATH)
10. Vlahogianni EI, Karlaftis MG, Golias JC (2005) Optimized and meta-optimized neural networks for short-term traffic flow prediction: a genetic approach. Transportation Research Part C: Emerging Technologies 13(3):211-234.
11. Ishak S, Kotha P, Alecsandru C (2003) Optimization of dynamic neural network performance for short-term traffic prediction. Transportation Research Record: Journal of the Transportation Research Board 1836:45–56.
12. Zhong M, Sharma S, Lingras P (2005) Refining genetically designed models for improved traffic prediction on rural roads. Transportation planning and technology 28(3):213-236.
13. Nagare A, Bhatia S (2012) Traffic flow control using neural network. Traffic 1(2):50–52.
14. Liu SY, Li DW, Xi YG, Tang, QF (2015) A short-term traffic flow forecasting method and its applications. Journal of Shanghai Jiaotong University 20:156-163.
15. Geem ZW, Kim JH, Loganathan GV (2001) A New Heuristic Optimization Algorithm: Harmony Search. Simulation 76(2):60–68.
16. Laña I, Del Ser J, Olabarrieta I (2016) Understanding Daily Mobility Patterns in Urban Road Networks using Traffic Flow Analytics. In: Network Operations and Management (NOMS). IEEE, Istanbul, pp 1157–1162.
17. Madrid Open Data portal, `http://datos.madrid.es`, retrieved on 11-18-2016.
18. Bäck T, Schwefel H (1993) An overview of evolutionary algorithms for parameter optimization. Evolutionary computation 1(1):1–23.
19. Hu W, Liu Y, Li L, Xin S (2010) The Short-Term Traffic Flow Prediction Based on Neural Network. In: International Conference on Future Computer and Communication. IEEE, Wuhan, pp 293–296.
20. Vlahogianni EI, Karlaftis MG, Golias JC (2007) Spatio-Temporal Short-Term Urban Traffic Volume Forecasting Using Genetically Optimized Modular Networks. Computer-Aided Civil and Infrastructure Engineering 22:317–325.
21. Jain AK, Mao J, Mohiuddin KM (1996) Artificial neural networks – a tutorial. Computer: 31–44