

Multilayer Perceptron and Particle Swarm Optimization Applied to Traffic Flow Prediction on Smart Cities

Lucas Rodrigues Frank[1], Yan Mendes Ferreira[1], Eduardo Pagani Julio[1], Francisco Henrique C. Ferreira[1], Bruno José Dembogurski[2], and Edelberto Franco Silva[1(✉)]

[1] Federal University of Juiz de Fora, Juiz de Fora, Brazil
{lucasrodrigues,yanmendes,eduardo.pagani,francisco.henrique, edelberto}@ice.ufjf.br
[2] Federal Rural University of Rio de Janeiro, Seropédica, Rio de Janeiro, Brazil
brunodembogurski@ufrrj.br

Abstract. Smart cities can increase the live quality of its citizens and Intelligent Transportation Systems is a key topic in this area. When the population density living in the same region increases more and more, the cities suffer from problems as constant traffic jams. Thinking this way, in this paper are present the uses of computational intelligence techniques and analyses to aid in traffic dimensioning solutions. To do this, prediction models and heuristics are the best way to create a more autonomous and intelligent environment. In this work, an application is introduced applying machine learning and an optimization technique to empower a smart ecosystem. To validate it, an evaluation using Multi-Layer Perceptron together with Particle Swarm Optimization was performed, comparing it with the state-of-the-art. All evaluations were done using real data traffic with a free traffic flow scenario. Applying the Particle Swarm Optimization to optimize the activation functions' parameters, we obtained 3.1% average MAPE for Logistic activation function and a MAPE of 3.4% for ReLU activation function.

Keywords: Smart cities · Machine learning · Traffic flow prediction · Intelligent Transportation Systems · Particle Swarm Optimization

1 Introduction

New urban development policies are increasingly directing its focus towards the Internet and broadband network as enablers of a variety of electronic services (e-services). These new concepts and solutions are turning cities into centers for innovation in many areas such as mobility, health, security, socio-environmental, economic and many others [2].

Therefore new problems and challenges are presented when considering how cities and their surroundings can evolve to include smart objects and smart applications, thus accelerating Future Internet research which includes the aforementioned innovations.

© Springer Nature Switzerland AG 2019
S. Misra et al. (Eds.): ICCSA 2019, LNCS 11622, pp. 35–47, 2019.
https://doi.org/10.1007/978-3-030-24305-0_4

Here, the main focus is traffic management, a segment of Intelligent Transportation Systems (ITS) [5,8,12] that tackles traffic dimensioning and orientation.

Considering this topic, the major objective is to provide accurate traffic predictions. This can be achieved through different models aiming for an autonomous and intelligent environment [8]. Usually, these prediction models are used to create simulations of an entire city, but they are not limited to this single purpose. Smaller environments, e.g a university campus, can be modeled as well as introducing new challenges, such as heterogeneous traffic [2,3,16].

The SmartCampus UFJF project aims to create an intelligent campus environment for its users. In order to achieve this goal, many campus' services are being integrated, updated and new ones are being developed. One of these new initiatives seeks a better understanding of vehicle traffic in the campus area, generating new data that can be used to improve the quality of life expectations of all campus users and to support decision making.

To provide a clear understanding of the problem description of the campus environment is necessary. The campus interconnects two key neighborhoods of the Juiz de Fora - Minas Gerais, Brazil - city, allowing free citizen and vehicle traffic through its premises. Thus, traffic volume is not limited to its students, professors, and staff, but also by those who use the campus as a place to practice leisure or exercising activities and those who just pass by. This creates a heterogeneous field, where drivers have different objectives within the campus - mainly academic, leisure and passerby. These contrasting profiles make the prediction task a real challenge when considering the huge traffic volume variation that occurs during the day. This variance is intrinsically linked to the drivers' profile and their schedules - *e.g.* beginning of academic activities and morning commuters going to work.

The main contribution of this work is to use a computational technique to improve the configuration of a neural network model. Specifically, to find the best values of activation functions parameters. After to evaluate and validate the improvements, the model is used in conjunct with predictive techniques in the context of traffic volume estimation, culminating in an application called $SmartTraffic^{pso}$. The evaluations are done using real data traffic in a real free traffic flow scenario.

The rest of this paper is organized as follows. Initially, the related work in Sect. 2 is discussed. Section 3 presents this work architectural proposal and describes its workflow. Moreover, in Sect. 4, the experimental results are presented. Finally, Sect. 5 summarizes the paper and outlines future research directions.

2 Related Work

In this section, ITS and PSO (Particle Swarm Optimization) related papers will be discussed, focusing on traffic prediction and parameters optimization. For this reason, each one will be discussed separately and then a comparison will be made between this works proposal and the ones in the literature.

To complete our study about the state-of-the-art and initiatives applied to ITS two works are present using PSO benefits. The first one is [17] is our best-known application of a metaheuristic to improve a neural network model. In this paper were propose two mechanisms to increase the models' performance. Firstly, the use of Deep Learning which is a recent technique that implies using a neural network with multiple layers. Based on this, the Deep Learning help to solve a problem of predicting the number of occupants in a given location at 15, 30 and 60 min from the current time via wifi data collected. In order to reach the goal, a computational intelligent method, or metaheuristic, called Particle Swarm Optimization (PSO) were proposed to be used to improve the prediction model. PSO as a good iterative population-based method to create particles that represent solutions, in order to achieve the best possible solution a fitness function is used. These particles use mathematical formulations to move in the search space, this one being the number of layers and the number of neurons they seek to optimize. After the results, they concluded that the PSO was successful in reducing model training time from 77% to 85%.

In [7], a combination of SVR and PSO is proposed to improve the results of a vehicle traffic prediction problem. The authors conducted a comparison among its performance with other methods, such as multiple linear regressions and neural networks. In this case, the PSO seeks to optimize the input parameters of the SVR, which are: penalty C, radius and kernel function. The evaluation metric used as fitness function was the Root Mean Square Error (RMSE). At the end of, was possible to conclude that the combination of PSO and SVR had a positive impact, doing a better performance in almost 10% compared to the other models. It is necessary to be clear the limitation of scenario and models compared by the authors, where the use of traffic flow was not based on an open database, and the models compared was not the main on the state-of-the-art today. Because of it, our proposal will be compared with the main work from [8].

The work [8], mentioned from now on by Huang *et al.*, poses as one of the state-of-the-art solutions for traffic flow prediction. The authors propose a Deep Learning Architecture which is achieved by training a Deep Belief Network (DBN) - that can be seen as a stack of Restricted Boltzmann Machines (RBMs) -. The RBM is an unsupervised pattern learning Artificial Neural Network (ANN) that learns through a greedy layerwise training. To fine tune the model's parameters, a supervised regression layer is added on top of the DBN. Using the premise that traffic is a correlated network, the idea of multi-task regression was exploited in the supervised learning process.

The authors implement different traffic flow forecast approaches and compare to their own architecture, outperforming all the others. Despite utilizing a sigmoid regression layer, the authors point to the possibility of replacing it with other regression algorithms such as Support Vector Regression (SVR).

In [13], the authors also applied a deep learning strategy and reinforced the use of historical data to predict traffic flow. Its model is based on Stacked Autoencoders (SAE): A deep neural network that aims to reproduce the input layer into the output layer by adjusting the weights of the synapses. They train the network in a greedy fashion. The unsupervised learning algorithm parameters are then fine-tuned with a sigmoid regression layer.

The model presented in [6] propose the use of the Gated Recurrent Units (GRU). That is a modification of the Long Short Term Memory (LSTM) neural network, whose main characteristic is the storage of information. Thus, allowing the consideration of previous entries when processing new data, enabling it to work better with time series.

While the LSTM has been used as a predictive model for traffic flow in previous works such as [15,19], GRU has not been used for this purpose until then. These two models are evaluated, comparing the results with an AutoRegressive Integrated Moving Average (ARIMA) model, concluding that both models outperformed the ARIMA model and that in 84% of the tests the model GRU outperformed the LSTM.

It is valid to mention that the authors do not assess the scalability of the proposed model and all tests are conducted utilizing a very small amount of data (4 weeks). Since the work proposed in this paper aims to empower a smart ecosystem, scalability is key, therefore this solution was not implemented and evaluated.

The previously mentioned papers evaluate their architectures with free-flowing highways, which is a less challenging task. Differently, [4] propose an architecture composed by a spin-off implementation of a SVR, suited for online applications called On-line SVR (Ol-SVR) [14]. In this investigation, the authors propose an architecture that works both in typical and atypical scenarios (e.g. vehicular crashes, work zone, holidays, etc). In their work, other solutions from the literature were implemented and compared, showing that their proposal presents a solid option for both scenarios, even though it was outperformed in some evaluations.

All mentioned papers utilize the dataset PeMS - introduced in Subsect. 4.1 - to evaluate their proposal. The first three utilize data collected on highways, both recent and historical - *i.e.* traffic volume of past weeks on the same weekday -, while the latter utilizes both typical and atypical traffic scenarios for their evaluation, but only recent data is inputted in the prediction process. The approaches found in the literature are summarized in Table 1 and compared to the one presented in this paper.

3 Proposal

The SmartCampus project is an initiative to create an intelligent and autonomous university campus. Its results are not restricted to campus environments and can be used on other scenarios, like a big smart city. To achieve that, a series of smaller initiatives are being proposed and implemented as services and applications that will be available in an integrated API.

Table 1. Related work comparison.

Work	Architecture	Optimization	Historical	Atypical traffic
[8]	DBN + SVR		**X**	
[13]	SAE + SR		**X**	
[6]	GRU		**X**	
[14]	Ol-SVR			**X**
[7]	SVR	X (PSO)	**X**	
SmartTraffic^{pso}	MLP	X (PSO)	**X**	**X**

SmartTraffic and its evolutions, *SmartTraffic*^{pso}, is the initiatives' pilot, whose main objective is to provide live traffic volume estimation, serving as a support for future applications - *e.g.* estimating parking spots availability - and supporting the administration in the decision-making process, such as estimating traffic volume in a public event on the campus.

The data can come from different sources, *e.g.* security software and cameras for security environment, traffic cameras to traffic flow monitor. On the traffic flow monitor side, is possible to register all vehicles license plates that pass through its cameras. It is worth mentioning that, among other transformations, the plates are hashed in the Extract-Transform-Load (ETL) process to the database and only quantitative analysis are conducted. Other sources of data flow can be simulated too. As the case of our evaluation, when a real traffic flow data will be used.

Figure 1 synthesizes the proposal. The SmartCampus project provides encapsulated services as endpoints of an API and the workflow adopted in the implementation of *SmartTraffic*^{pso} application is presented. Foremost, it exposes the utilized predictor and then discusses the steps in this pipeline in the following subsections.

3.1 Model Specification

Time series models are well-known problem solvers for traffic flow prediction. From simpler and traditional models, as the ARIMA [20], to more complex and non-linear [9], the time series approach is well-taken in the literature. As opposed to that, another noteworthy line of research utilizes data-driven algorithms such as ANNs [11] or Local Weighted Learning [18] to solve traffic flow prediction problems.

This work proposal, beyond to optimize its activation functions' parameters, an application of neural network on traffic flow prediction. Our application is part of an smart city ecossystem, and is called *SmartTraffic*^{pso}. In our experiments was created a model composed by triple layers Multilayer Perceptron (MLP) ANN.

The MLP is an ANN where all neurons in a layer are connected to all other neurons in adjacent layers. It contains an input layer, a number of hidden layers

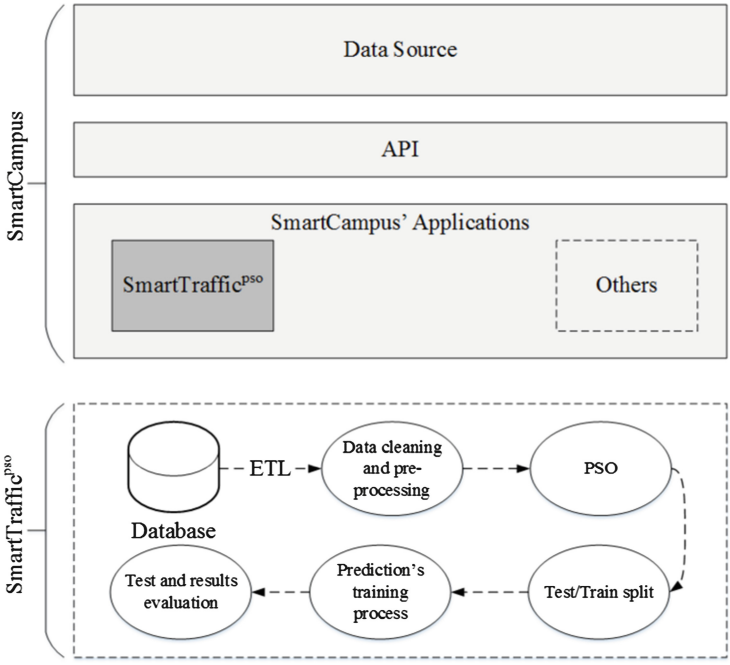


Fig. 1. Global view of SmartCampus project and *SmartTraffic*^{pso}'s pipeline.

- three in this particular architecture - and an output layer. Since the result is a one dimension answer, *i.e.* the traffic flow forecast, the output layer contains only one neuron. The model is described in Eq. 1.

$$f(t+1) = \sum_{i=2}^{4} W_i \cdot g(W_{i-1}^T \cdot x_{i-1} + b_{i-1}) + b_i \tag{1}$$

Where: W_i and W_{i-1} is the set of weights of the i^{th} and $i-1^{th}$ layer, respectively. b_i and b_{i-1} are the bias added to the $i+1^{th}$ and i^{th} layer, respectively. Furthermore, $g(x)$ is the activation function and, finally, x_{i-1} is the set of neurons of the $i-1^{th}$ layer.

The input layer fed to the network, defined in Eq. 2, are past observations, *i.e.* the traffic volume. Moreover, since a cyclic pattern was observed, the model does not use only recent observations but also historic observations, *i.e.* the traffic volume of the same weekday and time of day in previous weeks. The use of historical data shows an improvement in results as seen in the work of [1,8,13].

$$\begin{aligned} x_1 = &\{\#V_t,\ \#V_{t-1},\ ...,\ \#V_{t-P}\} \cup \\ &\{\#V_{t+1-(1\ week)},\ \#V_{t+1-(2\ weeks)},\ ...,\ \#V_{t+1-(Q\ weeks)}\} \end{aligned} \tag{2}$$

Where V_t is the traffic volume at the t^{th} time stamp, P is the number of recent observations taken into account and Q is the number of historical observations considered.

The network learns through backpropagation [21]. In this paper, two distinct neuron activation functions are evaluated: logistic function (Logi), defined in Eq. 3, and REctified Linear Unit (ReLU), defined in Eq. 4. Those were chosen due to their popularity in the literature and their very different behavior. The solver for the weight optimization process of choice was the stochastic gradient descent [10].

$$g(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

$$g(x) = max(0, x) \tag{4}$$

In order to guarantee reproducibility and to compare this works proposal with [8], the *scikit-learn*[1] implementations of the RBM and SVR was chosen and *keras*[2] to implement a MLP regressor.

As were mentioned, the PSO a metaheuristic and was applied to improve the implementation of the neural network model using MLP. With PSO is possible to the best combination of parameter values for the activation functions, *i.e.* in our case, ReLU and Logi. As can be seen on Algorithm 1, the PSO algorithm works by having a population/swarm of candidate solutions, called particles. To find the best fitness, these particles are moved around in the search space. Its movements are guided by their own best-known position in the search space as well as the entire swarm's best-known position. The process is repeated iteratively to find the best/satisfactory solution. Seen the algorithm can be clear how the population size would find the best values of parameters.

For particle movement, we need to define an initial velocity and position for each particle randomly. And so, from this initial velocity, we can calculate new positions and velocities for the particle [17]. The formulas for new position and velocity can be found in Eqs. 5 and 6, respectively. In Eq. 6 *pbest* is the particle best position and *gbest* is the particle best position in the population.

$${x_i}^{t+1} = {x_i}^t + {v_i}^{t+1} \tag{5}$$

$${v_i}^{t+1} = w.{v_i}^t + c_1.rand.(pbest - {x_i}^t) + c_2.rand.(gbest - {x_i}^t). \tag{6}$$

In our proposal, we want to optimize the number of neurons, a learning rate and the alpha parameter (for the logistic function). For the PSO to work correctly, we had to limit the search space to minimize training time and neural network convergence. Therefore, the limit values for each of the parameters are: n_MAX = 140, n_MIN = 60, learning_rate_MAX = 0.0009, learning_rate_MIN = 0.0001, alpha_MAX = 52, alpha_MIN = 2. It is also necessary to define the

[1] http://scikit-learn.org/.
[2] https://keras.io/.

acceleration coefficients c_1 and c_2, and the coefficient of inertia w, both aim to assist in the movement of the particles and were assigned the value 2 for the first two and 0.7 for the latter.

Finally, we define the size of the population and the maximum number of iterations over the population, being 20 for both.

Algorithm 1. PSO

```
   Input  : numberIteration, populationSize
   Output: numberNeurons, learningRate, alpha
 1 particlePop ← list of particles;
 2 for i ← 0 up to i < populationSize do
 3 |   create a particle p;
 4 |   initialize p with random parameters within their correspondet intervalls;
 5 |   p.pBest ← fitness function calculated after neural network test for particle
   |   p;
 6 |   add particle p in particlePop;
 7 end for
 8 gBest ← best particle in particlePop;
 9 while numberIteration do
10 |   for each particle p in particlePop do
11 |   |   p ← calculate new velocity;
12 |   |   p ← calculate new position;
13 |   |   error ← fitness function calculated after running new parameters in
   |   |   neural network for particle p;
14 |   |   if error < p.pBest then
15 |   |   |   p.pBest ← error
16 |   |   |   p.positionBest ← p.position
17 |   |   |   if error < gBest.pBest then
18 |   |   |   |   gBest ← p;
19 |   |   |   end if
20 |   |   end if
21 |   end for
22 |   numberIteration ← numberIteration − 1;
23 end while
```

3.2 Data Cleaning and Pre-processing

During peaks in traffic, the software would register multiple entries for a single vehicle. Specifically, it would move slow enough for the system to trigger another snapshot. All observations of the same vehicle within a 1-min range were removed and only the first entry was maintained.

Some entries would not have a license plate tied to it. Since the provided dataset only had raw data and no study of the vehicle detection software was conducted, two options were available on how to deal with this limitation. It could be a fault in the image processing algorithm, where it would not read

the plate, but it was indeed a vehicle and should be kept as a piece of valid information or it was a capture of something other than a vehicle - *e.g.* bike or pedestrian - and should be removed. The latter was the chosen option.

After cleaning these sets, the traffic volume was aggregated in a time window and normalized. The normalization is applied to both datasets whereas the other processes are applied only to the university dataset.

3.3 Test/Train Split

For both scenarios, a train-test split of 2/3 and 1/3, respectively, was conducted. It is noteworthy that in Huang et al. [8] original paper, the authors used 10 months of the dataset to train their model and only 2 months to test it. Even though the test set represents 17% of the dataset size, the experiments presented here utilize 33% to minimize overfitting effects and, therefore, there is a marginal difference between the results obtained in this work and [8].

3.4 Test and Evaluation Results

Along with P and Q introduced in Sect. 3.1, the number of neurons N in the hidden layer compose the parameters of the proposed architecture. The same parameters are used by Huang *et al.* [8]. However, N represents the number of components in the DBN's stack.

To focus only on the parameters we seek to optimize, we did several initial tests where the parameters followed the following ranges: P ranging from 10 to 15 with a gap of 1, Q ranging from 0 to 5 and N ranging from 80 to 120 with 10 as a gap. After filtering the obtained results, we separated the best parameters P and Q to be fixed, to save time and to improve the results already found.

The train-test split used was 70% and 30%, respectively.

4 Evaluation

In this section, the model previously presented is validated and compared with the results presented in [8] in a scenario using the PeMS benchmark dataset. It is valid to mention that this work focus on comparing the results with Huang et al. [8] approach since it was the technique that obtained the best results in the literature, considering the state-of-the-art review done during this work. The metric used to compare the models' performance is the Mean Absolute Percentage Error (MAPE), defined by Eq. 7.

$$MAPE = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{\hat{y}_t - y_t}{\hat{y}_t} \right| \tag{7}$$

Finally, the performance of the proposed model is evaluated utilizing two different activation functions, as described in Subsect. 3.1, comparing the results obtained.

All tests were run in a machine which has the following configuration: Intel Core i5-7400 CPU @ 3.00GHz × 4 processor, 8GB of RAM and on a 64-bit Windows 10 operational system. The programming language of choice was Python 3.6.5.

4.1 PeMS

Scenario Description. The PeMS[3] is a well-known benchmark dataset for traffic-related problems. It consists of an open repository of data collected by a network of sensors. Its sensors are spread across the state of California that register traffic-related information such as vehicle volume in a highway.

In order to reproduce the test scenario from [8], a dataset with the same characteristics used by them was extracted, *i.e.* 12 months from the fifty most busy roads in the year of 2011.

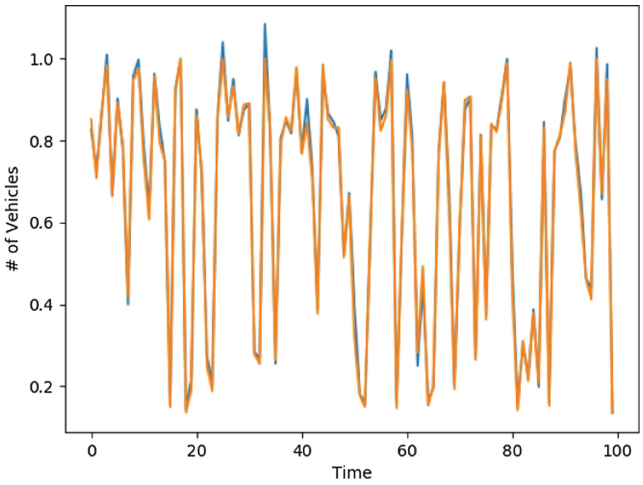


Fig. 2. Prediction curve of the *SmartTraffic*^{pso} for Logistic activation function. The blue line represents the true value and the orange one the model's forecast.

Results. In comparison, the Huang *et al.* approach obtained an average MAPE of 15.81% and its best performance resulted in a 13.18% MAPE. Even though there are implementation and parametrization differences, the results obtained on the same dataset are comparable by, approximately, 1% for the best model.

[3] http://pems.dot.ca.gov.

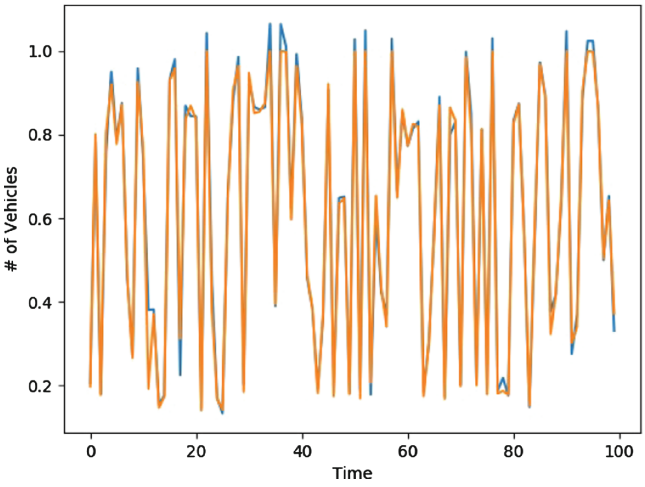


Fig. 3. Prediction curve of the *SmartTraffic*^{pso} for ReLU activation function. The blue line represents the true value and the orange one the model's forecast.

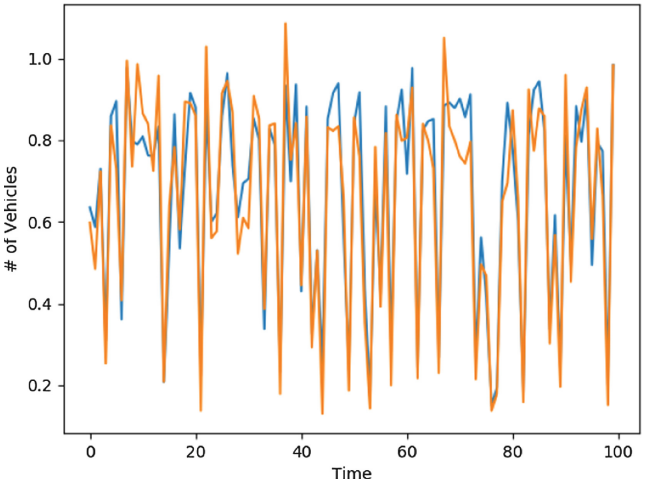


Fig. 4. Prediction curve of the Huang et al. approach. The blue line represents the true value and the orange one the model's forecast.

The MAPE metric evaluation effort the *SmartTraffic*^{pso} consistently outperforms Huang *et al.*'s approach. This can be verified by the best performing model, which had the following configuration: $P = 11$, $Q = 3$, $N = 90$ and it scored a 3.1% average MAPE for Logistic activation function and a MAPE of 3.4%. for ReLU activation function. The plots of said model is shown in Figs. 2 and 3. To conclude our analysis of the approaches performance, Fig. 4 is present.

Our improvements can be seen more clearly in Table 2, where is possible to see 3 points (first, middle and last) for real values of traffic flow on the first row. On the next rows are show the distance in percentage gets by the two main approaches evaluated in this research and the real value. To conclude, the last table line shows how much our approach is better than Huang et al.

Table 2. *SmartTraffic*^{pso} gain over Huang et al. in relation to the real value and the predicted value.

	First	Middle	Last
Huang et al.	4.5%	8.9%	1%
SmartTraffic^{pso}	1%	2.2%	0.5%
SmartTraffic^{pso} **gain over Huang et al.**	**3.5%**	**6.7%**	**0.5%**

5 Conclusion

This paper presented the SmartTraffic and one of its application, one of the pilot initiatives of the SmartCampus project, an effort to support traffic-related applications in order to create a more intelligent and autonomous campus. Moreover, it presents an extensive evaluation of different state-of-the-art solutions using real data traffic in a free traffic flow scenario.

Also, the results of the statistical analysis conducted ratified the evidences pointed by [1,8,13] that the usage of historical data helps the prediction accuracy, guiding future efforts. Finally, the benefits of PSO between two activation functions is done.

As for future work, the experiments will be re-run enabling more parameters optimization such as number of epochs and number of layers. The last interesting investigation to conduct is a study with other activation functions and other scenarios, different traffic flow prediction.

References

1. Abadi, A., Rajabioun, T., Ioannou, P.A.: Traffic flow prediction for road transportation networks with limited traffic data. IEEE Trans. Intell. Transp. Syst. **16**(2), 653–662 (2015)
2. Abuarqoub, A., et al.: A survey on internet of thing enabled smart campus applications. In: Proceedings of the International Conference on Future Networks and Distributed Systems, p. 38. ACM (2017)
3. Alghamdi, A., Shetty, S.: Survey toward a smart campus using the Internet of Things. In: 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud), pp. 235–239. IEEE (2016)
4. Castro-Neto, M., Jeong, Y.S., Jeong, M.K., Han, L.D.: Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. Expert Syst. Appl. **36**(3), 6164–6173 (2009)

5. Dalal, K., Dahiya, P.: State-of-the-art in VANETs: the core of intelligent transportation system. IUP J. Electr. Electron. Eng. **10**(1), 27 (2017)
6. Fu, R., Zhang, Z., Li, L.: Using LSTM and GRU neural network methods for traffic flow prediction. In: 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), pp. 324–328, November 2016. https://doi.org/10.1109/YAC.2016.7804912
7. Hu, J., Gao, P., Yao, Y., Xie, X.: Traffic flow forecasting with particle swarm optimization and support vector regression. In: 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 2267–2268. IEEE (2014)
8. Huang, W., Song, G., Hong, H., Xie, K.: Deep architecture for traffic flow prediction: deep belief networks with multitask learning. IEEE Trans. Intell. Transp. Syst. **15**(5), 2191–2201 (2014)
9. Ishak, S., Al-Deek, H.: Performance evaluation of short-term time-series traffic prediction model. J. Transp. Eng. **128**(6), 490–498 (2002)
10. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
11. Kumar, K., Parida, M., Katiyar, V.: Short term traffic flow prediction for a non urban highway using artificial neural network. Procedia - Soc. Behav. Sci. **104**, 755–764 (2013)
12. Loce, R.P., Bala, R., Trivedi, M.: Computer Vision and Imaging in Intelligent Transportation Systems. Wiley, Hoboken (2017)
13. Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F.Y.: Traffic flow prediction with big data: a deep learning approach. IEEE Trans. Intell. Transp. Syst. **16**(2), 865–873 (2015)
14. Ma, J., Theiler, J., Perkins, S.: Accurate on-line support vector regression. Neural Comput. **15**(11), 2683–2703 (2003)
15. Ma, X., Tao, Z., Wang, Y., Yu, H., Wang, Y.: Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transp. Res. Part C: Emerg. Technol. **54**, 187–197 (2015)
16. Nati, M., Gluhak, A., Abangar, H., Headley, W.: SmartCampus: a user-centric testbed for Internet of Things experimentation. In: 2013 16th International Symposium on Wireless Personal Multimedia Communications (WPMC), pp. 1–6, June 2013
17. Qolomany, B., Maabreh, M., Al-Fuqaha, A., Gupta, A., Benhaddou, D.: Parameters optimization of deep learning models using particle swarm optimization. In: 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 1285–1290. IEEE (2017)
18. Shuai, M., Xie, K., Pu, W., Song, G., Ma, X.: An online approach based on locally weighted learning for short-term traffic flow prediction. In: Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, p. 45. ACM (2008)
19. Tian, Y., Pan, L.: Predicting short-term traffic flow by long short-term memory recurrent neural network. In: 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), pp. 153–158. IEEE (2015)
20. Van Der Voort, M., Dougherty, M., Watson, S.: Combining Kohonen maps with arima time series models to forecast traffic flow. Transp. Res. Part C: Emerg. Technol. **4**(5), 307–318 (1996)
21. Werbos, P.J.: Backpropagation through time: what it does and how to do it. Proc. IEEE **78**(10), 1550–1560 (1990)