EAI Endorsed Transactions
on Energy Web
Research Article EAI.EU

Towards Multi-Model Big Data Road Traffic Forecast at Different Time Aggregations and Forecast Horizons

Riccardo Martoglia and Gabriele Savoia*

FIM - University of Modena and Reggio Emilia, Italy

Abstract

Due to its usefulness in various social contexts, from Intelligent Transportation Systems (ITSs) to the reduction of urban pollution, road traffic prediction represents an active research area in the scientific community, with strong potential impact on citizens' well-being. Already considered a non-trivial problem, in many real applications an additional level of complexity is given by the large amount of data requiring Big Data domain technologies. In this paper, we present the first steps of a novel approach integrating both classic and machine learning models in the Spark-based big data architecture of the H2020 CLASS project, and we perform preliminary tests to see how usually little-considered variables (different data aggregation levels, time horizons and traffic density levels) influence the error of the different models.

Received on 01 April 2022; accepted on 18 May 2022; published on 25 May 2022

Keywords: Big Data Analytics, Time Series, Traffic Forecast, Time Aggregation, ARIMA, Apache Spark, Machine Learning

doi:10.4108/ew.v9i39.1187

1. Introduction

Research in the urban logistics field, but more generally in a Smart City context, is experiencing a significant increase due to the numerous improvements it brings to public services. Specifically, the road traffic prediction task plays a fundamental role in terms of city mobility, and it is also useful as a decision support for defining traffic restrictions in order to reduce air pollution and improve public well-being. Since vehicles flows can be thought as time series, several statistical and machine learning traffic forecast models are exploited in this scenario. However, their accuracy depends on several factors which are often not sufficiently investigated, such as data granularity, forecast type, traffic conditions, etc.

The work presented in this paper starts from the H2020 CLASS project[1] and the real use-case given by the MASA[2] area; in the considered setting, an innovative big-data analytics framework [1] exploits cloud data management techniques based on Apache Spark offering efficient storage, real-time querying and updating of the high-frequency data incoming from the edge (pole-mounted cameras and smart/connected vehicles) at different granularity levels.

In this paper, we focus on the first steps and tests for supporting traffic forecasting in such a challenging scenario: (i) differently from many state-of-the-art proposals which concentrate either on "classic" forecast models (such as ARIMA) in non-big data settings [2, 3] or on machine learning models (such as Decision Trees, DT) when big data support is needed [4, 5], we present a novel approach integrating both worlds within the Apache Spark Big Data infrastructure by a joint exploitation of the Spark's MLlib (supporting DT) and Spark's Pandas Function API (for ARIMA); (ii) we perform preliminary tests on such algorithms in our real use-case; (iii) we analyze the accuracy trying go give useful first answers to a number of questions which are not usually contemplated (e.g., "How forecast accuracy varies in relation to the granularity of the data and to the traffic density?", "Are next-hour prediction more accurate than next-minute or next-15-minutes?"),

*Corresponding author. Email: riccardo.martoglia@unimore.it, gabrielesavoia98@gmail.com

[1] Edge and Cloud Computation: A Highly Distributed Software for Big Data Analytics (CLASS), https://class-project.eu/

[2] Modena Automotive Smart Area, https://www.automotivesmartarea.it

by considering the results of the different models at different data aggregation levels (1 minute, 15 minutes, 1 hour), time horizons (1 step, 1-3-6 hours), and traffic density levels. Finally, some execution performances will also be reported. The long term aim is that this initial research can eventually help in bringing us closer to better managed smart cities and services, improving citizens' well-being.

The rest of the paper is organized as follows: in Section 2 we briefly report on related work; Section 3 and 4 give an overview of the proposed approach and detail the specific data preprocessing steps, respectively; experimntal evaluation is discussed in Section 5. Finally, conclusions and future work are presented in Section 6.

2. Related works

Several recent research studies have demonstrated, at least conceptually, the possibility of utilizing and managing Big Data to improve and create new smart city services [6, 7]. While there are several works showing the benefits of big data information extraction / analysis, in many cases the focus is mainly application specific and on the analysis of the possible benefits rather than on presenting actual data management solutions / architectures [6]. Our past work [8], based on prior data management experiences in real-world smart city situations [9, 10], demonstrates a platform with data processing features for both real-time and historical data management; however, this is still based on a centralized relational architecture rather than on modern bigData/noSQL technologies.

Focusing on specific services in a Smart City context, scientific works concerning road traffic prediction are becoming increasingly common. As reported in a recent survey [2], while the most common approach is to use statistical forecasting models such as ARIMA, the use of machine learning (e.g., Decision Tree) and deep learning models like LSTM is becoming more and more popular. However, design patterns (such as type model selection and data management infrastructure) strongly depend on the specific application context. On one hand, there are approaches such as the one presented in [3], which tests ARIMA and compares it with a hybrid model also incorporating non-linearity, GARCH, concluding that ARIMA is better, or [2], which compares ARIMA with other "classic" forecast models. All these works consider a non-big data context, thus the exploitation of big data platforms such as Spark and the use of ML models are not considered. On the other hand, others propose and test machine learning models in Spark, including Decision Tree [4] and neural networks [5], but do not consider classic statistical forecast models. Instead, in this work we propose an approach based on Apache Spark supporting machine learning forecast models but also capable of integrating statistical forecasting models such as ARIMA through the Pandas Function API. Similarly to other works [11], we consider how the traffic volume affects the prediction. Moreover, unlike many researches [12–14] where the aim is rather to understand how external factors (atmospheric conditions or road indicators) affect the forecast, in this paper we consider how the concepts of data granularity and time horizon impact on the forecast accuracy.

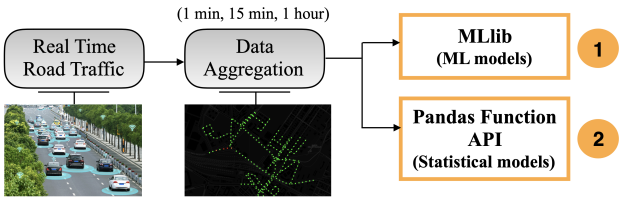

Figure 1. Overall architecture overview

3. Overview of the approach

The data management architecture we consider in this work is the one we devised in the CLASS project [1], which enables the management of real-time data (through Spark Structured Streaming) coming from the edge and their storing at different granularities (1 min, 15 min, 1 hour) by means of hierarchical aggregations (see Figure 1). In this context, the approach we propose extends this architecture and exploits two different data management paths to enable effective road traffic prediction:

- *MLlib* path: thanks to Spark's machine learning library we are able to efficiently execute ML models;
- *Pandas Function API* path: by means of this Spark's functionality, it is possible to integrate statistical forecasting models in the Spark ecosystem.

In this work we focus on the following two models, which are representative of each of the paths:

- *Decision Tree (DT)*: a supervised machine learning algorithm implemented in Spark MLlib, whose ability to solve regression problems makes it possible to forecast road traffic flow after an initial training step;
- *ARIMA*: one of the most common statistical models used for time series prediction (implemented with the support of Spark Pandas Function API). More precisely, we adopted ARIMAX, trough which consider time information (hour and/or weekend) as external regressors.

In addition to the different models available, the two paths also differ in their execution mode (Figure 2):

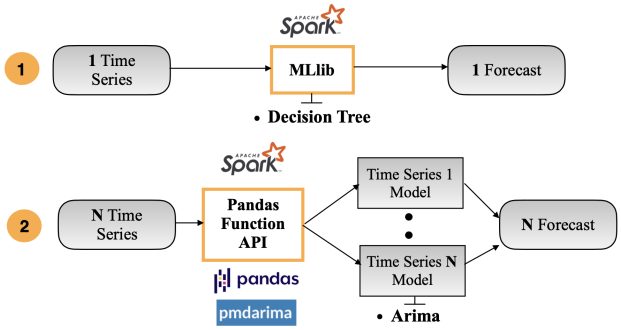

Figure 2. Execution mode of the two data management paths

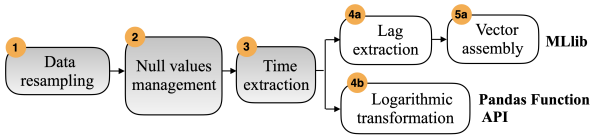

Figure 3. Data processing steps

whereas with MLlib ML jobs are submitted one at a time (but more jobs can be executed in parallel), with the Pandas Function API, *n* jobs can be directly run simultaneously. Moreover, while for the second path each single model can only manage time series whose dimension does not exceed the memory of the cluster executors, the first model enables the execution also on potentially very large time series.

4. Data processing

Depending on the different adopted methodologies, aggregated data needs specific (pre)processing steps. As reported in Figure 3, the first three steps are common to the two paths:

1. *Data resampling*: resampling is performed in order to get equally spaced observations in relation to the data granularity level;

2. *Null values management*: we associate zero values to null observations in order to handle no-vehicle-flow situations (fill forward or interpolation would lead to incorrect problem modeling);

3. *Time extraction*: time information like hour and weekend are extracted and used as additional features in forecasting models.

For ML models (MLlib path), two additional steps are required (and implemented through MLlib's Pipeline object): (4a) *Lag extraction* obtains lag values for each observation and (5a) *Vector assembly* creates a single vector containing all the extracted features. For statistical models (Pandas Function API path), instead, a *Logarithmic transformation* (4b) is needed, performed through the Pandas module.

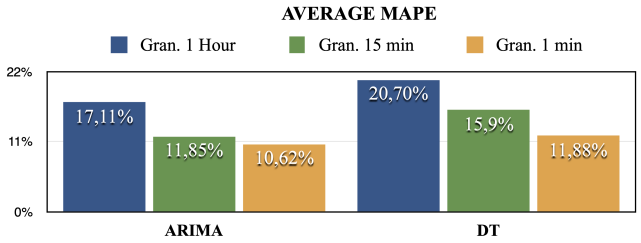

Figure 4. Average 1 Step MAPE for the two models at different data granularities

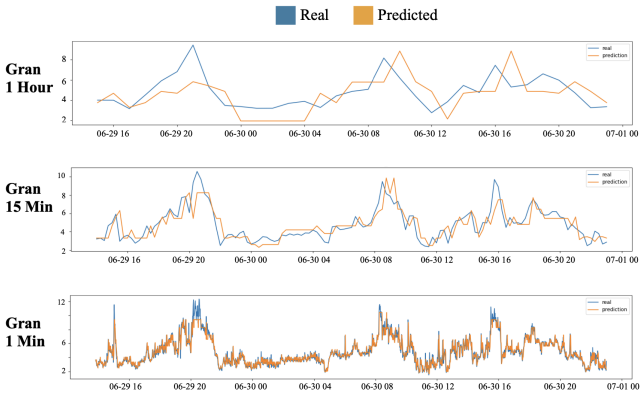

Figure 5. 1 Step forecast for a single time series in relation to different time granularities

5. Experimental Evaluation

We performed a series of preliminary tests on our real use-case in order to evaluate prediction accuracy and to give initial answers to how accuracy is influenced by data/prediction granularity, traffic density, and time horizon. Moreover, we also present preliminary efficiency figures. We considered the complete scenario of 500 different map points / time series (7 days duration) at the 3 granularities (1 hour, 15 min, 1 min), and from this we selected two groups of different significant road points, representative of high and low traffic densities. As to model configuration: for DT features like lag values, hour and weekend information are used, and for model parameters, *variance* is defined as the way to compute nodes impurity while *maxDepth* parameter is set to 5 with the aim to reduce the probability of overfitting; for ARIMA, we employed a grid-search methodology in which the best parameters are chosen in relation to Akaike's Information Criterion (AIC) value. The considered accuracy metric is the Mean Absolute Percentage Error (MAPE). Moreover, for the DT model, the training phase is done on the first 80 percent of total data and then tested on the remaining 20 percent; for ARIMA, a *cross validation on a rolling basis* is used so to respect temporal dependency between observations. Tests are executed on a server with 3.3 GHz Intel Core i7 CPU and 16 GB RAM.

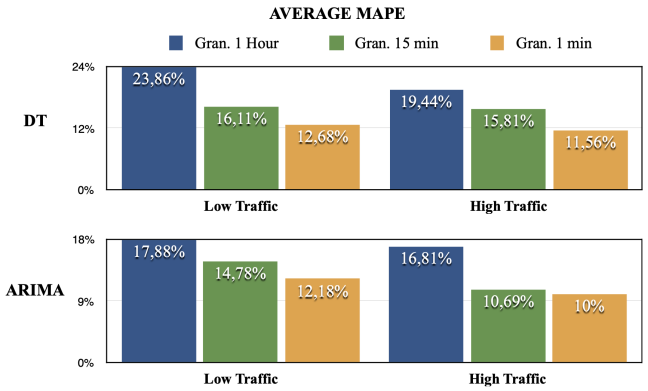

Figure 6. Average 1 Step MAPE details for the two traffic levels

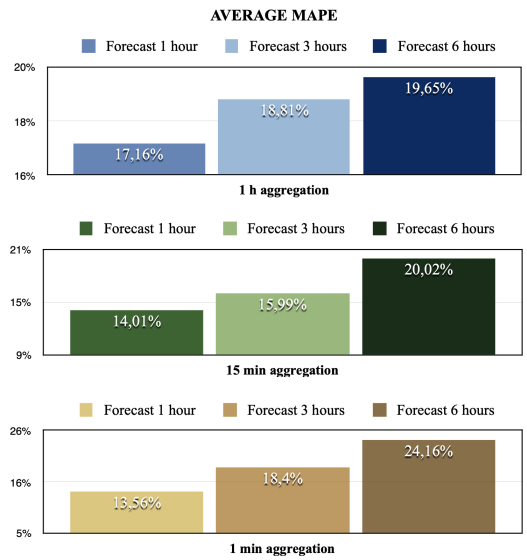

Figure 7. Average MAPE at different time horizons and time aggregations (ARIMA)

1 step prediction and time granularity impact. First of all we consider 1 step prediction accuracy with time series at the different granularities, to answer questions like: *are next-hour predictions more accurate than next-minute or next-15-minutes ones?* Figure 4 reports the obtained average errors. As expected, the use of a fine granularity and the consequent presence of a higher number of model information, leads to a decrease in the error (as also shown in Figure 5) On average, for all aggregation levels, ARIMA seems to get a higher accuracy value compared to DT.

Traffic density impact. Figure 6 reports average errors in relation to the two traffic levels for each considered model and at the different data granularities. We note that, relatively to all aggregation levels and for each model (ARIMA and DT), 1 step forecast produces, on average, a lower error when traffic density is *high*: when the average flow of vehicles is consistent (and the roads more congested), the data is possibly less subject to random fluctuations, thus making the forecast more accurate. Moreover, for the two traffic levels, the behaviour, w.r.t. the different data granularities, is in line with the one shown in the previous test.

Time horizon impact. In the previous tests we focused on 1 step prediction, in which the forecast was made with a short time horizon coinciding with the given data granularity. The aim of the following test is to make predictions over more distant time horizons (i.e., over the next 1 hour, 3 hours and 6 hours) in order to see how the forecast error changes, also on the basis of the different data aggregations. For example, if the target is to predict the next 3 hours average traffic density, we proceeded as follows: for 1 hour (15 minutes, 1 minute) data aggregation granularity, we made 3 (12, 180, respectively) steps forecast and then computed the average. In this preliminary evaluation, we focused on the ARIMA model and the results are reported in Figure 7.

From the obtained results it is possible to see that the increase of the time horizon leads to a consequent increase in the forecast error. In other words, this means that, for a given data aggregation granularity, we get a bigger error if we want to predict further into the future. As explained above, each data aggregation requires a specific n-steps forecast (where n is low for hourly data and increases for 15 minutes and 1 minute data). Due to this aspect, another interesting aspect to note is the different error rate growth between the different granularities. While with 1 hour aggregation we see an error increment of about 2 points, for 15 minutes and 1 minute it is about 6 and 11 points respectively. So, to answer questions like: *Given a specific forecast time horizon, which data granularity should we use to get the lowest error?* we could conclude that: (a) to predict the average next hour traffic density, the use of 1 minute data granularity leads to best accuracy levels compared to 15 minutes and 1 hour data; (b) to predict the average next 6 hours traffic density, the use of 1 minute data granularity (requiring a 60*6-step forecast) leads to a bigger error compared to 1 hour and 15 minutes aggregations which require 6-step and 4*6-step forecast respectively.

Preliminary efficiency evaluation. Even if in this first research phase we are not specifically focused on efficiency, we will nonetheless provide some early performance results derived from the execution of the two different data management paths, MLlib for DT and Pandas Fuction API for ARIMA, on our standard configuration (we plan to perform tests on dedicated parallel servers with cluster support in the future). In the test shown in Figure 8 (A), we compare the execution time between the Spark Pandas Fuction

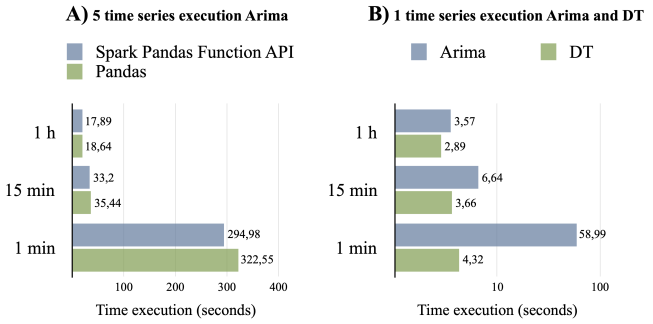

Figure 8. Execution time comparisons

API configuration we described in this paper and the standard Pandas execution: as it is possible to see, even if Spark is executed without cluster support, simultaneous execution of 5 different time series with ARIMA model results more efficient than in normal Pandas implementation, for each granularity level. This justifies this architectural choice not only for enabling ML models support but also from an efficiency point of view. Furthermore, in Figure 8 (B) we reported time execution for ARIMA and DT for a single time series computation and in relation to the different data aggregations. In this case, it is possible to note that ARIMA performances are good but require more time on very long time series in 1 minute granularity, since its time is affected by the complex automatic parameter optimization. On the other hand, DT is particularly efficient for all granularities also in this basic configuration setup; this makes us confident that future parallel optimized execution will enable a very high number of concurrent predictions to be made in real-time.

6. Conclusions and future work

In this paper we proposed a novel approach for traffic forecast integrating both classic and machine learning models in a Spark-based big data architecture. The preliminary tests allowed us to understand the impact of different variables which are often not considered together in state of the art (different data aggregation levels, time horizons and traffic density levels). Although the current work represents a good starting point, in the future we plan to continue the development and testing of our approach by considering further models to integrate and by improving the current ones through grid-search techniques for machine learning approaches, outlier detection mechanisms as well as the use of additional features like weather conditions. Moreover, building on recent data analytics experiences in different scenarios [15–18], we also plan to complement the approach with a complete dashboard for data analysis, possibly exploiting interpretable machine learning techniques. In the context of the MASA use-case and, more in general, in different smart city contexts, the techniques presented in this work could become the basis for supporting more complex tasks like public transportation logistic, road trip optimization and decision support to reduce air pollution, ultimately helping in improving citizens' well-being.

References

[1] Cavicchioli, R., Martoglia, R. and Verucchi, M. (2022) A novel real time edge-cloud big data management and analytics framework for smart cities. *Journal of Universal Computer Science* **28**(1). doi:10.3897/jucs.71645.

[2] Zhang, Y. (2020) Short-term traffic flow prediction methods: A survey. *Journal of Physics: Conference Series* **1486**: 052018. doi:10.1088/1742-6596/1486/5/052018.

[3] Chen, C., Hu, J., Meng, Q. and Zhang, Y. (2011) Short-time traffic flow prediction with arima-garch model. In *2011 IEEE Intelligent Vehicles Symposium (IV)*: 607–612. doi:10.1109/IVS.2011.5940418.

[4] Tsai, J., Chang, T.Y., Fang, Y.H. and Chang, E.S. (2018) A real-time traffic flow prediction system for national freeways based on the spark streaming technique. In *2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*: 1–2. doi:10.1109/ICCE-China.2018.8448998.

[5] Sundareswaran, A. and Sendhilvel, L. (2020) Real-time vehicle traffic prediction in apache spark using ensemble learning for deep neural networks. *International Journal of Intelligent Information Technologies* **16**. doi:10.4018/IJIIT.2020100102.

[6] Honarvar, A. and Sami, A. (2019) Towards sustainable smart city by particulate matter prediction using urban big data, excluding expensive air pollution infrastructures. *Big Data Research* **17**: 56–65.

[7] Kim, S., Choi, M., Lee, S., Park, H. and Park, S. (2020) Intelligent management system with energy data block in smart city. In *2020 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, January 4-6, 2020* (IEEE): 1–3.

[8] Carafoli, L., Mandreoli, F., Martoglia, R. and Penzo, W. (2016) A data management middleware for its services in smart cities. *Journal of Universal Computer Science* **22**(2): 228–246.

[9] Carafoli, L., Mandreoli, F., Martoglia, R. and Penzo, W. (2012) Evaluation of data reduction techniques for vehicle to infrastructure communication saving purposes. In *Proceedings of the 16th International Database Engineering and Applications Symposium, August 2012 (IDEAS 2012)* (Prague, Czech Republic): 61–70. URL http://www.isgroup.unimore.it/article/ideas12.pdf.

[10] Carafoli, L., Mandreoli, F., Martoglia, R. and Penzo, W. (2013) A framework for its data management in a smart city scenario. In *Proceedings of the 2nd International Conference on Smart Grids and Green IT Systems, May 2013 (SmartGreens 2013)* (Aachen, Germany).

[11] Mohammed, O. and Kianfar, J. (2018) A machine learning approach to short-term traffic flow prediction: A case study of interstate 64 in missouri. In

IEEE International Smart Cities Conference (ISC2): 1–7. doi:10.1109/ISC2.2018.8656924.

[12] Alajali, W., Zhou, W., Wen, S. and Wang, Y. (2018) Intersection traffic prediction using decision tree models. *Symmetry* **10**: 386. doi:10.3390/sym10090386.

[13] Essien, A., Petrounias, I., Sampaio, P. and Sampaio, S. (2019) Improving urban traffic speed prediction using data source fusion and deep learning. In *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*: 1–8. doi:10.1109/BIGCOMP.2019.8679231.

[14] Jia, Y., Wu, J. and Xu, M. (2017) Traffic flow prediction with rainfall impact using a deep learning method. *Journal of Advanced Transportation* URL https://doi.org/10.1155/2017/6575947.

[15] Ghidoni, G., Martoglia, R., Taccioli, C. and Vischioni, C. (2020) Instacircos: a web application for fast and interactive circular visualization of large genomic data. In *Proceedings of the 24 International Conference Information Visualisation (iV 2020)* (IEEE).

[16] Furini, M., Mandreoli, F., Martoglia, R. and Montangero, M. (2022) A predictive method to improve the effectiveness of twitter communication in a cultural heritage scenario. *ACM J. Comput. Cult. Herit.* .

[17] Martoglia, R. and Montangero, M. (2020) An intelligent dashboard for assisted tweet composition in the cultural heritage area (work-in-progress). In *Proc. of GOODTECHS20* (Association for Computing Machinery): 226–229.

[18] Vischioni, C., Bove, F., Mandreoli, F., Martoglia, R., Pisi, V. and Taccioli, C. (2022) Visual exploratory data analysis for copy number variation studies in biomedical research. *Big Data Research* **27**: 100298. doi:https://doi.org/10.1016/j.bdr.2021.100298, URL https://www.sciencedirect.com/science/article/pii/S2214579621001155.