Energy-efficient Learning for Traffic Prediction

Assunta De Caro[1], Angelo Furno[2], Lorenzo Goglia[1], and Eugenio Zimeo[1]

[1]*Dept. of Engineering*, Università degli Studi del Sannio, Benevento, Italy
[2]*LICIT-ECO7 UMR-T9401 ENTPE*, Univ. of Gustave Eiffel, Lyon, France

***Abstract*—Green AI is an emerging research area that focuses on AI environmental sustainability, *i.e.*, minimize energy consumption of AI-based models while preserving their accuracy. In this extended abstract, we compare different training approaches (static and dynamic) for optimizing the accuracy-energy trade-off in traffic forecasting through *Graph Neural Networks* and spatial temporal transformers.**

***Index Terms*—Green AI, Continual Learning, Traffic Forecasting, Graph Neural Networks**

I. Introduction

Green AI proposes making *efficiency* an evaluation criterion alongside accuracy and related metrics [1]. The design of AI-based models is usually and strongly driven by the sole objective of improving accuracy. The consequence of this approach is the proliferation of increasingly complex models, with increasingly sophisticated modules, which achieve even small gains at high computational costs. *Continual Learning* (CL) represents a viable solution for reducing the energy footprint of such models while keeping accuracy metrics at an acceptable level. This preliminary study explores the use of different learning techniques, including CL-based, to increase the performance of state-of-the-art models in terms of energy consumption and evaluates their impact on Smart Mobility applications.

In traffic forecasting, *Graph Neural Networks* (GNN) and transformer-based networks dominate the scene due to their ability to capture the intricate spatial and temporal relationships hidden in the data. In particular, we consider the so-called *Dynamic Multiview Spatio-Temporal Network* (DMSTG) [2], which is the most recent proposal in this domain. It consists of one input layer, a certain number of stacked blocks, and one output layer. Each block is mainly composed of four modules: *i)* a *dynamic spatial-temporal graph convolutional network* (DSTGCN); *ii)* a *dynamic spatial-temporal transformer* (DSTFormer); *iii)* a *long-term spatial-temporal graph convolutional network* (LSTGCN); and *iv)* a spatio-temporal fusion module. The first two are responsible for extracting short-term and local spatio-temporal information. The third captures the long-term and global spatio-temporal information. Finally, the fourth fuses all the extracted features from the different views through a hierarchical knowledge fusion process. While capable of mining the most complex patterns, all these components inevitably add more and more processing steps. Our experiments reveal the optimal approach for balancing prediction error and resource consumption across various time horizons.

This work has been partially supported by the Borgo 4.0 P-Mobility project funded by Regione Campania.

II. Evaluation of learning approaches

CL involves learning from dynamic data distributions, *i.e.*, models are trained incrementally, using data samples only once as they arrive. Despite its ability to provide a model with an adaptation mechanism that makes it capable of responding to external changes, this paradigm introduces a side effect, known as *catastrophic forgetting*, which represents a critical challenge: learning from a new distribution generally results in forgetting what has been learned from the old ones. Dealing with this dilemma means finding a trade-off between *learning plasticity*, *i.e.*, the ability of a model to integrate new knowledge, and *memory stability*, *i.e.*, the ability of a model to remember past knowledge, while achieving resource efficiency, *i.e.*, the overhead of model updates should be close to those that would be obtained if learning were performed only on new samples. The latter aspect makes the naive approach of retraining from both new and all old samples infeasible. For this reason, several alternative approaches have been proposed in the literature. Among them, based on the availability of source code, we tested and compared the *Experience Replay* (ER) and the *Averaged Gradient Episodic Memory* (A-GEM) models, along with the classic *pre-training* (PT) one. ER stores a certain amount of old training samples within a memory buffer. Those samples are then used during the retraining phases to enrich the batches containing new data. To efficiently use the limited memory available, ER uses reservoir sampling: buffered elements are randomly selected for replacement with new samples. A-GEM exploits an episodic memory to compute loss bounds to be used to constrain the loss score calculated in the current retraining phase. Finally, PT requires one single network that is continuously trained with new samples. Specifically, the starting network to be trained with the new data is the network already trained with the old data. The above approaches were compared with two naive strategies, namely *naive* (NV) and *static* (ST), together with the *cumulative* (CM) learning process in which the train set is incrementally built by accumulating data samples as they arrive and the training/test phases are always run on all the available data. In particular, NV does not involve with training since it uses the actual values from the previous time window as predictions of the current one. In ST, instead, training is performed only on a subset of data capturing all the possible patterns (*e.g.*, data from a restricted time interval).

	MAE [#vehicle per 5m]				Energy consumption [Wh]				Execution Time [s]			
	T=1d	*T=7d*	*T=10d*	*T=14d*	*T=1d*	*T=7d*	*T=10d*	*T=14d*	*T=1d*	*T=7d*	*T=10d*	*T=14d*
A-GEM	24.21	22.73	23.78	23.56	10.64	91.89	195.14	291.80	190.01	1641.29	3524.66	5555.72
ER	24.29	23.03	14.62	14.54	10.49	87.06	177.78	296.19	263.73	2508.53	5170.34	7417.95
PT	27.73	19.69	14.90	14.82	5.39	48.49	101.36	142.00	93.00	845.86	1777.29	2477.87
ST	15.62	15.05	15.31	15.22	2.38	3.27	3.25	141.72	5.45	25.84	32.65	2473.54
NV	38.48	34.41	49.63	34.28	0	0	0	0	1.74	14.67	19.32	42.63
CM	14.68	14.87	14.81	14.81	532.432	359.87	447.71	434.75	6566.92	6408.73	7782.16	7493.58

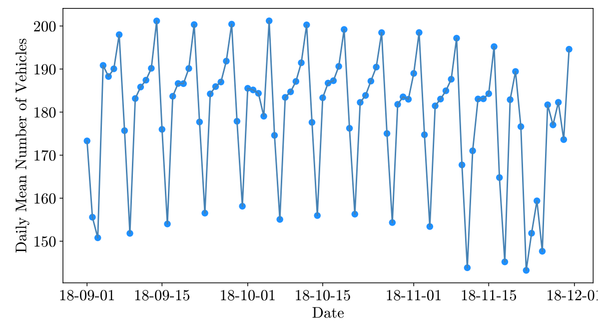
Note: NV has zero energy consumption because it does not use any models.

TABLE I: Summary of the results - Median MAE, energy consumption, and execution time.

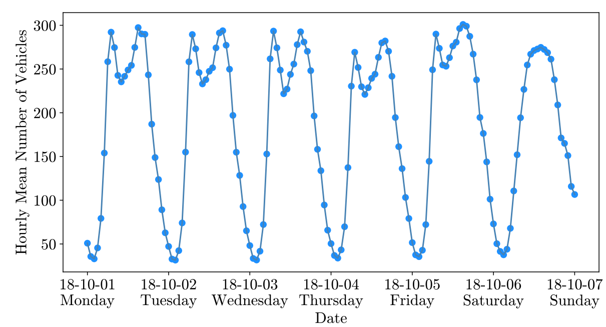
III. Experiments and Results

The experimental evaluation has been conducted considering the *PeMS03* dataset, which contains traffic flow data collected in Sacramento, California from September 1st to November 30th, 2018. The data is aggregated in 5-minute intervals and covers 91 days. The dataset includes 358 nodes and 26,208 time steps, with 288 time steps per day. The DMSTG network used as the base model is optimized by training it on the whole dataset, as in a classic learning process, before applying the different approaches.

The results obtained are summarized in Tab. I. In particular, we report the median values of *mean absolute error* (MAE), execution time (including training and inference contributions), and energy consumption, assuming different sizes of the time window T: one day (*1d*), one week (*7d*), ten days (*10d*), and two weeks (*14d*). In general, T represents the time horizon according to which *i)* data is used for training and testing, and *ii)* the whole dataset is continuously consumed. For example, T=*1d* means that one day of data is used for the current training phase and one day of data (the next one) is used for the current testing phase; the latter is then used for the subsequent training phase, while a new day will be considered for the corresponding new testing phase. As it can be seen, with T=*10d* and T=*14d*, ER and PT achieved MAE values similar to those achieved with CM, but they did it while consuming significantly less energy. Note that energy consumption increases as T grows due to longer execution times, *i.e.*, the time needed for both training and inference. As for NV, it represents the best choice from the energy consumption perspective, since it does not involve training or querying a network, but it generates the predictions simply by replicating past data. However, the MAE values are the highest. The fact that PT exhibits the same performance as ER from the point of view of accuracy depends on the periodicity of the data (see Fig. 1). For this reason, having a memory buffer storing past samples does not introduce benefits: the values assumed by the observed quantity, *i.e.*, traffic flow, are repeated throughout the entire time span (*e.g.*, daily, weekly, monthly). Another direct consequence deriving from such a particular trend of data is that it is sufficient to train a model only for a restricted time interval, and then use it to generate predictions for every subsequent moment. In our experiments, for ST we considered only the first month of data. Results confirm what has just been said: ST is able to achieve accuracy levels close to those of ER and PT, which are the best CL approaches, and of CM. Furthermore, it notably reduces energy consumption. The outputs of our experiments lead us to conclude that the trigger of a new training phase should not be based on time, but rather on events that substantially alter the existing relationships between the data, regardless of the specific learning approach employed.



(a) Daily aggregate data.



(b) Hourly aggregate data (for one week).

Fig. 1: PeMS03 - Average Traffic Flow.

Future experiments will involve more granular error measurements and their temporal and spatial impact.

IV. Acknowledgments

We want to thank Costanzo Dell'Ostia and Pasquale Martorano for their support in carrying out the experimentation.

References

[1] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green ai," 2019. [Online]. Available: https://arxiv.org/abs/1907.10597

[2] Z. Diao, X. Wang, D. Zhang, G. Xie, J. Chen, C. Pei, X. Meng, K. Xie, and G. Zhang, "Dmstg: Dynamic multiview spatio-temporal networks for traffic forecasting," *IEEE Transactions on Mobile Computing*, vol. 23, no. 6, p. 6865–6880, Jun. 2024. [Online]. Available: http://dx.doi.org/10.1109/TMC.2023.3328038