Node-Sensitive GCN-LSTM for Geo-Distributed Forecasting

Massimiliano Altieri[1][0000−0002−7226−8036], Michelangelo Ceci[1,2][0000−0002−6690−7583], and Roberto Corizzo[3][0000−0001−8366−6059]

[1] University of Bari Aldo Moro, 70125 Bari, Italy
{massimiliano.altieri,michelangelo.ceci}@uniba.it
[2] Jožef Stefan Institute, 1000 Ljubljana, Slovenia
[3] American University, Washington, DC 20016, USA
rcorizzo@american.edu

Abstract. Forecasting models represent a paramount opportunity to support decision-making in modern real-world applications, especially in fields characterized by high variability due to geophysical phenomena. In this context, major challenges are the effective combination of temporal and spatial information from multiple geo-distributed nodes, and supporting scalability for larger and growing sensor networks. Current deep learning methods are generally unable to properly model node relationships throughout the time sequences, as they either treat all nodes independently, or exclusively focus on capturing high-level global network information. Additionally, their complexity usually grows rapidly as more nodes are added, making their applicability in large sensor networks costly. In this paper, we propose a novel geo-distributed forecasting method that simultaneously deals with these two challenges. In particular, we adopt a neural architecture combining recurrent and graph neural networks to jointly analyze sensor network time series data at two levels of granularity: while the GCN sub-network analyzes global network information, the LSTM sub-network is specific to a single node under consideration and extracts temporal autocorrelation from its time series. The model is designed so that multiple sub-models can be trained independently, one for each node of the sensor network, enabling high parallelization capabilities. Quantitative experiments with real-world energy datasets show that our method is highly competitive with respect to state-of-the-art forecasting methods.

Keywords: Time Series Forecasting · Graph Neural Networks · Sensor Networks.

1 Introduction

Forecasting models constitute a major source of decisional support for many modern real-world problems, such as renewable energy forecasting, where the observed phenomenon depends on several underlying variables. Crucial open challenges are the effective exploitation of spatio-temporal dependencies between multiple geo-distributed nodes, as well as the capability to process large-scale sensor networks.

Geo-distributed data collected from several physical locations defies the conventional assumption of independent and identically distributed samples, introducing spatio-temporal autocorrelation patterns that need to be exploited. Traditional autoregressive models (e.g. ARIMA, Prophet) can account for temporal autocorrelation, but they frequently fail to evaluate multivariate data and fully take advantage of spatial patterns [11]. Through the use of coefficients learned exclusively for a target feature of interest, Vector Autoregression (VAR)-based techniques may describe spatial relationships, albeit in a fairly simplistic manner [16]. Additionally, they are unable to model non-linear relationships between various values and are limited to a single target feature.

Deep learning methods based on recurrent neural networks or attention mechanisms, allow to deal with spatio-temporal correlations and non-linear interactions among features [6,7]. The work in [6] models the spatio-temporal autocorrelation in the sensor network by means of statistical indicators of spatial association. Similarly, [7] leverages 3D-CNNs for traffic flow forecasting. However, such deep learning approaches, typically based on LSTM-based architectures (e.g. Bi-LSTM, Attention-LSTM, CNN-LSTM, SVD-LSTM) usually attribute the same importance to all nodes and do not consider varying graph-based interactions among nodes. To address this issue, more advanced methods adopt graph neural network architectures, and are able to fully exploit the graph structure of the data, and to extract additional spatial information during the temporal processing [4,2,15,8].

From the scalability perspective, such sophisticated forecasting methods based on graph convolutional neural networks [1,5,9] (GCN-LSTM) usually lead to a higher overhead and the growth of a single model when adding new nodes to the network, constraining these methods to the limitations of hardware resources of a single computational worker. Some forecasting methods explicitly support large-scale data and distributed processing and analysis [10,12,14], but their modeling capabilities are rather shallow in the context of geo-distributed sensor networks.

In this paper, we propose a novel geo-distributed forecasting method that simultaneously deals with these two challenges. To deal with dynamic geo-distributed data, we adopt a neural network architecture that is able to jointly analyze the sensor network time series at two levels of granularity. Specifically, we use a graph convolutional sub-network to analyze, learn, and aggregate global network information, and an LSTM sub-network that is specific to a single node under consideration, and extracts temporal autocorrelation from its time series. The combination of these two components is able to contextualize local node time series with the general network state. The global information is extracted by a sequence summarizing operator, that computes several statistics from the input sequence of each node at run-time. These statistics are used to create an abstract network representation, propagating node information based on the correlation of their production output. This design allows to model the sensor network behaviour solely in terms of node observations, without the need of external static information such as the geographical node locations.

To deal with scalability, our neural network model is designed so that multiple sub-models can be trained independently, one for each node of the sensor network. As a result, the model is designed to provide high parallelization capabilities (e.g. adopting distributed learning frameworks such as Horovod [3]), while also considering the spatial dependencies among nodes in the sensor network. This trade-off is achieved using the LSTM sub-network to consider local time series for a single node, and the GCN sub-network to consider all time series.

Extensive quantitative experiments with real-world energy datasets show that our method achieves a competitive performance with respect to state-of-the-art forecasting methods.

2 Method

This section is broken down into two subsections. We start by formally defining the issue that this study is trying to address. Following that, we outline our suggested method in depth, focusing on the contribution of each component.

2.1 Problem Statement

In this study, we consider the scenario of N geo-distributed energy production plants, producing observations at a regular time frequency. For each discrete time point t, we consider all nodes together emitting an observation $\mathbf{x}_t \in \mathbb{R}^{N \times F}$, for F features. In our work, the timeline is split into non-overlapping sequences of length T, so that the k-th sequence is defined as:

$$\mathbf{X}_k = [\mathbf{x}_{kT+1}, \mathbf{x}_{kT+2}, \dots, \mathbf{x}_{kT+T}] \in \mathbb{R}^{T \times N \times F} \tag{1}$$

Let us use $\hat{\mathbf{x}}_t \in \mathbb{R}^N$ to denote the target feature value for the observation at time point t. Given a sequence $\mathbf{X}_k$ and a forecasting horizon P, the forecasting task consists in approximating the target feature of interest for the next P observations $[\hat{\mathbf{x}}_{kT+T+1}, \hat{\mathbf{x}}_{kT+T+2}, \dots, \hat{\mathbf{x}}_{kT+T+P}]$.

In our task, we model the interaction between the energy plants as a weighted, fully connected graph $\mathcal{G} = \langle \mathcal{V}, \epsilon \rangle$, where $\mathcal{V} = \{1, \dots, N\}$ is the set of nodes (power plants, in this case) and $\epsilon : \mathcal{V} \times \mathcal{V} \to [0, 1]$ is a weighting function of the edges for any pair of nodes, representing the intensity of their relationship.

Based on this formulation, data is organized into two structures:

- a *sequence tensor* $\mathcal{X} = [\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_{S-1}] \in \mathbb{R}^{S \times T \times N \times F}$, containing S contiguous, chronologically ordered sequences of length T;
- a *graph adjacency matrix* $\mathcal{A} \in \mathbb{R}^{N \times N}$, where $\mathcal{A}_{v,v'} = \epsilon(v, v')$ is a measure of correlation between nodes v and v'.

2.2 Proposed Model

The proposed model is divided into two sub-networks: one acts at sequence level and the other at timestep level. An overview of the architecture is shown in 1. Learning representations at two different levels of granularity can offer benefits that wouldn't be captured at a single granularity level. The information extracted this way is fused together, and the resulting vector is passed through a feed-forward layer that learns to combine this heterogeneous information into a prediction for the next P prediction timesteps. In this section, we expand on these components and describe the model architecture in detail.

Given a node n and a timestep t, we denote with $\mathbf{x}_t^{(n)}$ the observation for node n at time t, and with $\hat{\mathbf{x}}_t^{(n)}$ we denote only the target feature value for that observation.

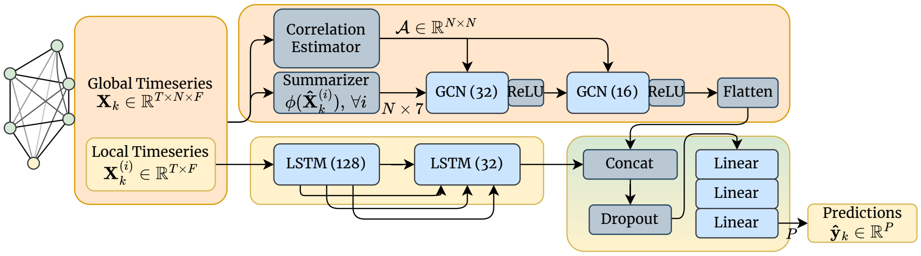
Global Timeseries
$\mathbf{X}_k \in \mathbb{R}^{T \times N \times F}$
Local Timeseries
$\mathbf{X}_k^{(i)} \in \mathbb{R}^{T \times F}$
Correlation Estimator
$\mathcal{A} \in \mathbb{R}^{N \times N}$
Summarizer
$\phi(\hat{\mathbf{X}}_k^{(i)}), \forall i$
$N \times 7$
GCN (32)
ReLU
GCN (16)
ReLU
Flatten
LSTM (128)
LSTM (32)
Concat
Dropout
Linear
Linear
Linear
P
Predictions
$\hat{\mathbf{y}}_k \in \mathbb{R}^P$

Fig. 1: Architecture of the proposed NS-GCN-LSTM model.

For every pair of nodes, the intensity of their relationship in the adjacency matrix is computed using the correlation of their respective observations for the target variable in the current batch of sequences.

Specifically, denoting with $\mathbf{X}_a \| \mathbf{X}_b = (\mathbf{x}_{a1}, \ldots, \mathbf{x}_{aQ}, \mathbf{x}_{b1}, \ldots, \mathbf{x}_{bR})$ the concatenation of sequences $\mathbf{X}_a = (\mathbf{x}_{a1}, \mathbf{x}_{a2}, \ldots, \mathbf{x}_{aQ})$ and $\mathbf{X}_b = (\mathbf{x}_{b1}, \mathbf{x}_{b2}, \ldots, \mathbf{x}_{bR})$, the adjacency matrix $\mathcal{A}_{[s,s']} \in \mathbb{R}^{N \times N}$ has elements

$$a_{ij} \triangleq \rho_{\hat{\mathbf{X}}^{(i)}_{[s,s']} \hat{\mathbf{X}}^{(j)}_{[s,s']}},$$

where $\hat{\mathbf{X}}^{(v)}_{[s,s']} = \hat{\mathbf{X}}^{(v)}_{s} \| \ldots \| \hat{\mathbf{X}}^{(v)}_{s'}$ is the iterative concatenation of sequences spanning from sequence s to sequence s', and $\rho_{\alpha\beta}$ denotes the Pearson correlation coefficient between variables α and β.

Using the correlation between the target feature allows us to model node relationships without the need of any external information about the network structure, such as the geographical location of nodes, thus computing the graph convolution solely based on the time series data. Two main advantages of this approach are that (i) it can be easily applied to other domains where it's not possible to clearly define a distance function between nodes, and (ii) it is especially useful in domains where the physical distance is not necessarily the best measure to use, as physically close sensors are not necessarily more strongly related (e.g. computer networks). In our work, the interval $[s, s']$ is set to be equal to the current batch of B sequences, where B is the batch size. This choice allows to have a dynamic adjacency matrix that reflects recent network conditions, and at the same time avoids recomputing $\mathcal{A}$ for each instance, maximizing a relevancy–computational efficiency trade-off.

Graph Convolutional Layer. To extract global network information, relevant historical data is summarized using statistical functions. Given a node i and the sequence of the target feature $\hat{\mathbf{X}}_k^{(i)} = \left(\hat{\mathbf{x}}_{kT+1}^{(i)}, \hat{\mathbf{x}}_{kT+2}^{(i)}, \ldots, \hat{\mathbf{x}}_{kT+T}^{(i)}\right)$ for i, and denoting with $\hat{\mathbf{X}}_k'^{(i)} = \left(\hat{\mathbf{x}}_{kT+\lfloor T/2 \rfloor}^{(i)}, \ldots, \hat{\mathbf{x}}_{kT+T}^{(i)}\right)$ the sub-sequence composed of the observations in the second half of $\hat{\mathbf{X}}_k^{(i)}$, we adopt a sequence summarizing function

$$\phi(\hat{\mathbf{X}}_k^{(i)}) = \left(\mu(\hat{\mathbf{X}}_k^{(i)}), \mu(\hat{\mathbf{X}}_k'^{(i)}), \sigma(\hat{\mathbf{X}}_k^{(i)}), \sigma(\hat{\mathbf{X}}_k'^{(i)}), \mathrm{Skew}(\hat{\mathbf{X}}_k^{(i)}), \mathrm{Kurt}(\hat{\mathbf{X}}_k^{(i)}), \mathrm{Slope}(\hat{\mathbf{X}}_k^{(i)})\right),$$

where μ denotes the mean, σ the standard deviation, Skew is the third standardized moment: $\mathrm{Skew}\left(\hat{\mathbf{X}}_k^{(i)}\right) = \mathbb{E}_{\hat{\mathbf{x}}^{(i)} \sim \hat{\mathbf{X}}_k^{(i)}}\left[\left(\frac{\hat{\mathbf{x}}_k^{(i)} - \mu(\hat{\mathbf{X}}_k^{(i)})}{\sigma(\hat{\mathbf{X}}_k^{(i)})}\right)^3\right]$, Kurtosis is the fourth standardized moment: $\mathrm{Kurt}\left(\hat{\mathbf{X}}_k^{(i)}\right) = \mathbb{E}_{\hat{\mathbf{x}}^{(i)} \sim \hat{\mathbf{X}}_k^{(i)}}\left[\left(\frac{\hat{\mathbf{x}}_k^{(i)} - \mu(\hat{\mathbf{X}}_k^{(i)})}{\sigma(\hat{\mathbf{X}}_k^{(i)})}\right)^4\right]$, and Slope is the angular coefficient of the best fitting[4] line for $\hat{\mathbf{X}}_k^{(i)}$. The summarization ϕ is applied locally to each node of the sequence $\hat{\mathbf{X}}_k$.

The output of the graph convolutional layer for a layer l and a sequence k is defined as:

$$H^{(l+1)} = \mathrm{ReLU}\left(\tilde{D}^{-1/2}\mathcal{A}_k\tilde{D}^{-1/2}\phi(\hat{\mathbf{X}}_k)W^{(l)}\right)$$ [5]

where $\mathcal{A}_k$ denotes the correlation adjacency matrix as described earlier in the section, $\tilde{D}$ denotes its degree matrix with $\tilde{D}_{ii} = \sum_j \mathcal{A}_{k;ij}$, and $\phi(\hat{\mathbf{X}}_k)$ is the summary matrix for sequence k.

Time series processing (LSTM). We employ an LSTM model alongside the graph neural network to analyze the sequence time series features. Unlike the GCN, which acts on the summarized values $\phi(\hat{\mathbf{X}}_k)$, the LSTM works on the raw data $\mathbf{X}_k$, and on all features, in a multivariate way.

In our model, we use two LSTM networks in series. The first one is used as a preprocessing step for the input sequence. It uses a cell state of size 128 and intermediate hidden states $h_1, \ldots, h_T$ are collected as output, so we have an hidden state h_t for each corresponding input time step t for the sequence, resulting in a matrix of shape $T \times 128$.

The second LSTM network is simpler and compresses this intermediate representation into a condensed vector of 32 values, corresponding to its last hidden state h'_T, without returning intermediate hidden states.

[4] According to least squares minimization.

[5] Self-loops are not required since $\mathcal{A}_k$ has diagonal equal to 1.

Fusion. As a last step, the outputs of the GCN and LSTM sub-networks are fused together via a concatenation layer, to obtain a unique representation combining local and global network information. Finally, a 50% dropout is applied to this vector representation[6], which is then fed to a series of three feed-forward neural networks that progressively compress it to the desired output size P, to extract a final prediction for the desired node and forecasting horizon.

3 Experiments

3.1 Experimental Setup

The datasets used to perform experiments are reported in Table 1. PV Italy contains hourly observations from 17 solar plants located in Italy, collected from 2:00 am to 8:00 pm. The time period spans from January 1st, 2012 to May 4th, 2014. Wind NREL was modeled using the Weather Research & Forecasting (WRF) model. Each plant consists of ten 3 MW turbines (for a total of 30 MW). Hourly aggregated observations range from January 1st, 2005 to December 31st, 2006. Wind NREL includes the following input features: temperature, pressure, wind speed, wind bearing, humidity, dew point, cloud cover. PV Italy includes all features in Wind NREL and also includes additional features for altitude, azimuth, irradiance and a weather summary feature. PV Italy observes a cutoff period between 9:00pm and 2:00am due to the absence of irradiance at that time.

Table 1: Datasets analyzed in our experiments.

Dataset	Domain	Time Frame	Horizon P	Nodes N	Features F
PV Italy	Energy	≈2.5 years	19	17	12
Wind NREL	Energy	2 years	24	5	8

3.2 Results

Table 2 presents the experimental results obtained for all methods and datasets considered in our study. It can be observed that our proposed method outperforms all competitors with all datasets according to the MAE and RMSE metric. Overall, it can be observed that Wind NREL presents the highest errors, which confirms that wind power forecasting is a more challenging than the solar counterpart, even though information from multiple nodes is taken into account. Among autoregressive approaches, Prophet significantly outperforms

[6] Dropout and batch normalization are recognized as beneficial regularization techniques. This rate was chosen subsequently to grid search using a validation set and values ≤ 0.5 following recognized heuristics [13].

ARIMA, possibly due to its improved exploitation of seasonal patterns. As expected, methods based on deep neural networks outperform the autoregressive approaches, confirming the positive contribution of exploiting the multi-variate nature of the data, the extraction of non-linear patterns, as well as the combined information from multiple nodes. One surprising result is that LSTM slightly outperforms Attention-LSTM on both datasets, and GRU outperforms all other deep learning methods, except for our proposed model, on Wind NREL. A similar pattern can be observed for SVD-LSTM obtaining slightly better results compared to GCN-LSTM. This result may depend on the fact that the global model learned by these methods is unable to uncover local spatio-temporal patterns and exploit them for the forecasting task at each single node. On the other hand, our method overcomes this issue, providing LSTM sub-models that are locally-optimized based on GCN features that globally consider the network structure and node relationships, but are learned separately for each node.

Table 2: Experimental results averaged over 10% evaluation sequences. The best performing method for each dataset and metric is marked in bold.

Model	PV Italy		Wind NREL	
	MAE	RMSE	MAE	RMSE
ARIMA	0.176	0.268	0.345	0.409
VAR	0.121	0.152	0.238	0.279
Prophet	0.077	0.119	0.261	0.302
LSTM	0.060	0.101	0.251	0.305
GRU	0.062	0.104	0.244	0.300
Bi-LSTM	0.061	0.105	0.274	0.324
Attention-LSTM	0.063	0.109	0.270	0.325
CNN-LSTM	0.060	0.101	0.303	0.369
SVD-LSTM	0.062	0.103	0.251	0.307
GCN-LSTM	0.061	0.103	0.257	0.309
NS-GCN-LSTM (Proposed)	**0.059**	**0.097**	**0.231**	**0.278**

4 Conclusion

In this paper, we proposed a novel time series forecasting method that leverages a neural architecture combining recurrent and graph neural networks to model geo-distributed sensor networks at two different levels of granularity (local and global). We showed that this characteristic allows us to learn relevant node-specific patterns while also extracting spatial autocorrelation from neighbouring nodes. This peculiar model architecture also allows multiple sub-models for the different nodes to be trained independently, providing high parallelization capabilities. Quantitative experiments with real-world energy datasets highlighted the competitiveness of our method against state-of-the-art forecasting methods. Future work will focus on assessing the scalability of our method in the presence of large sensor networks.

Acknowledgments

This work was partially supported by the project FAIR - Future AI Research (PE00000013), Spoke 6 - Symbiotic AI, under the NRRP MUR program funded by the NextGenerationEU.

References

1. Ali, A., Zhu, Y., Zakarya, M.: Exploiting dynamic spatio-temporal graph convolutional neural networks for citywide traffic flows prediction. Neural networks **145**, 233–247 (2022)
2. Ali, M.A., Venkatesan, S., Liang, V., Kruppa, H.: Test-gcn: Topologically enhanced spatial-temporal graph convolutional networks for traffic forecasting. In: 2021 IEEE International Conference on Data Mining (ICDM). pp. 982–987. IEEE (2021)
3. Altieri, M., Corizzo, R., Ceci, M.: Scalable forecasting in sensor networks with graph convolutional lstm models. In: 2022 IEEE International Conference on Big Data (Big Data). pp. 4595–4600. IEEE (2022)
4. Altieri, M., Corizzo, R., Ceci, M.: Gap-lstm: Graph-based autocorrelation preserving networks for geo-distributed forecasting. IEEE Transactions on Neural Networks and Learning Systems (2024)
5. Arastehfar, S., Matinkia, M., Jabbarpour, M.R.: Short-term residential load forecasting using graph convolutional recurrent neural networks. Engineering Applications of Artificial Intelligence **116**, 105358 (2022)
6. Ceci, M., Corizzo, R., Fumarola, F., Malerba, D., Rashkovska, A.: Predictive modeling of pv energy production: How to set up the learning task for a better prediction? IEEE Transactions on Industrial Informatics **13**(3), 956–966 (2016)
7. Chen, C., Li, K., Teo, S.G., Chen, G., Zou, X., Yang, X., Vijay, R.C., Feng, J., Zeng, Z.: Exploiting spatio-temporal correlations with multiple 3d convolutional neural networks for citywide vehicle flow prediction. In: 2018 IEEE international conference on data mining (ICDM). pp. 893–898. IEEE (2018)
8. Chen, P., Fu, X., Wang, X.: A graph convolutional stacked bidirectional unidirectional-lstm neural network for metro ridership prediction. IEEE Transactions on Intelligent Transportation Systems (2021)
9. Cirstea, R.G., Guo, C., Yang, B., Kieu, T., Dong, X., Pan, S.: Triformer: Triangular, variable-specific attentions for long sequence multivariate time series forecasting–full version. arXiv preprint arXiv:2204.13767 (2022)
10. Corizzo, R., Pio, G., Ceci, M., Malerba, D.: Dencast: distributed density-based clustering for multi-target regression. Journal of Big Data **6**, 1–27 (2019)
11. Makridakis, S., Hibon, M.: Arma models and the box–jenkins methodology. Journal of forecasting **16**(3), 147–163 (1997)
12. Pandya, A., Odunsi, O., Liu, C., Cuzzocrea, A., Wang, J.: Adaptive and efficient streaming time series forecasting with lambda architecture and spark. In: 2020 IEEE International Conference on Big Data (Big Data). pp. 5182–5190. IEEE (2020)
13. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research **15**(1), 1929–1958 (2014)
14. Syed, D., Refaat, S.S., Abu-Rub, H.: Performance evaluation of distributed machine learning for load forecasting in smart grids. In: 2020 Cybernetics & Informatics (K&I). pp. 1–6. IEEE (2020)

15. Wang, S., Zhang, M., Miao, H., Yu, P.S.: Mt-stnets: Multi-task spatial-temporal networks for multi-scale traffic prediction. In: Proceedings of the 2021 SIAM International Conference on Data Mining (SDM). pp. 504–512. SIAM (2021)
16. Zhao, Y., Ye, L., Pinson, P., Tang, Y., Lu, P.: Correlation-constrained and sparsity-controlled vector autoregressive model for spatio-temporal wind power forecasting. IEEE Transactions on Power Systems **33**(5), 5029–5040 (2018)