Prediction of Environmental Conditions for Maritime Navigation using a Network of Sensors: A Practical Application of Graph Neural Networks

C. F. Deberaldini Netto[1], E. A. Tannuri[1], D. D. Mauá[2], F. G. Cozman[1]

[1] Escola Politécnica - Universidade de São Paulo, Brazil
caio.netto@usp.br, eduat@usp.br, fgcozman@usp.br
[2] Institute of Mathematics and Statistics - Universidade de São Paulo, Brazil
denis.maua@usp.br

Abstract. This paper describes a real application of graphical neural networks (GNNs) in the dynamic estimation of spatially distributed buoys that are of central importance in maritime navigation. We describe the techniques we used to process both data and background knowledge about the domain, indicating why GNNs are particularly well suited for this sort of task. We report our empirical results, demonstrating that GNNs profitably use the avaible relational structure.

CCS Concepts: • **Applied computing**;

Keywords: forecasting, graph neural networks, relational learning, time series

1. INTRODUCTION

Many practical situations ask for predictions that must be based both on collected data and on structural patterns. If attributes are related through predicates, so that data come in the form of graphs, one may exploit relational learning to build a prediction model through a Graph Neural Network [Gori et al. 2005; Sperduti and Starita 1997]. A different scenario that seems to also ask for some graph-theoretical structure ensues if we have background knowledge about relations amongst attributes that display repetitive patterns, for example because they are related through spatial and/or temporal phenomena. Graph Neural Networks seem to offer the ideal vehicle to exploit underlying time/space structures.

In this paper we examine a real problem of technological and economic importance, where one must deal with a substantial flow of temporal data and also with a known spatial structure. The problem is to determine the speed of water current at a particular location within a major port area in South America, using measurements collected by a network of nearby sea buoys. Sensors in buoys periodically collect data, but a significant number of points are missing due to technical glitches. Prediction of water conditions in the port area is a major concern for port authorities that must coordinate a large number of heavy ships transporting for instance oil or ore. Current physical models

The first author has been supported by a scholarship from Itau-Unibanco. The second author has been partially supported by Conselho Nacional de Desenvolvimento Cientifico e Tecnologico (CNPq) grant 304784/2017-6. The third author has received financial support by the Conselho Nacional de Desenvolvimento Cientifico e Tecnologico (CNPq), grants 304012/2019-0 (PQ). The fourth author has been partially supported by the Conselho Nacional de Desenvolvimento Cientifico e Tecnológico (CNPq), grant 312180/2018- 7. The work was also supported by the Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), grants 2016/18841-0 and 2019/07665-4, and also by the Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior (CAPES) - finance code 001.

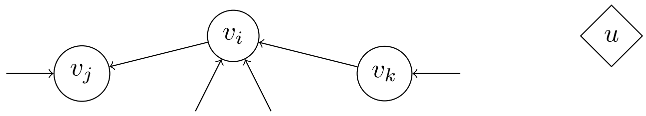


Fig. 1. Graph: nodes, edges, and global attribute.

are costly to develop and maintain, as they require high-quality measurements of the environment, boundary conditions and a 3D accurate spatial representation of the location.

Abstractly, this is a problem where sequential data are collected through a sensor network with a fixed spatial structure where sensors are often faulty. Thus our specific application is representative of a large class of significant prediction challenges faced by several industries and that could greatly exploit the properties of GNNs.

We have developed a GNN-based model whose structure captures spatial relations in the domain and whose parameters are learned from historical and real data collected at the Sepetiba/Ilha Grande Bay in Brazil. In this paper we describe the design we have adopted for our GNN and examine a number of nontrivial structural assumptions that can be imposed in our model, comparing their impact. We show that the GNN attains excellent performance and, more importantly, that it benefits from the spatial structure imposed on it.

In Section 2 we describe the essential ingredients of graph neural networks and the problem we address. In Section 3 we introduce our modeling decisions. Results are described and discussed in Section 4. Section 5 closes the paper with a summary of our conclusions and planned future work.

2. BACKGROUND

In this section we summarize relevant notions about GNNs (Section 2.1) and then we describe the prediction problem we have tackled in this work (Section 2.2).

2.1 Graph Neural Networks

Deep neural networks now offer excellent, and often surprising, performance in a variety of settings where patterns must be identified and acted upon. However, fully connected neural networks face difficulties when capturing repetitive relationships between entities [Battaglia et al. 2018]. In some cases neural networks adopt weights that are shared across many units, for instance in Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNN); however, even in those cases the connections are mostly local within the model (spatially or temporally).

Graph Neural Networks (GNNs) have been developed so as to take into account the structure of a domain, expressed through relations amongst entities [Scarselli et al. 2009; Sperduti and Starita 1997]. The goal is to work in the intersection between neural techniques and symbolic modeling, where the underlying relations and connections capture the symbolic aspects of the domain. To codify the domain, a GNN uses nodes (that belong to one or more classes), edges (that correspond to binary predicates), and global attributes. Figure 1 depicts a fragment of a graph with such objects.

In order to better illustrate the components of a GNN, consider a simple setting that is not related to the application we latter describe. Take the physical problem of predicting the mass position in a mass-spring system. One can model the physical system using a graph with nodes representing masses of the system and edges representing relations between nodes — that is, interactions between masses due to the springs. The masses attributes v_i are their positions, velocities and mass values. The edges attributes e_k are the stiffness and the natural length of the spring that connects each mass (node). Lastly, the global attribute u, shared with all entities, is the gravity force. These objects define the underlying graph. To predict the features of the model, a prediction function is to be applied to

Fig. 2. Sepetiba/Ilha Grande Bay, and the location of the main port terminals: 1 - Angra Terminal (Oil) TEBIG; 2 - Guaiba Island Terminal (Ore) TIG; 3 - Port of Sepetiba; 4 - CSN Terminal.

entities; for instance, one might want to predict one mass (node) position after 5 time steps in the hypothetical mass-spring system problem.

To determine the behavior of a GNN, a number of functions must be specified; in the most general scheme [Battaglia et al. 2018], we have functions associated with nodes, edges, and global attributes:

$$\begin{aligned} e'_k &= \phi^e(e_k, v_{r_k}, v_{s_k}, u), & \qquad \bar{e}'_i &= \rho^{e\to v}(E'_i), \\ v'_i &= \phi^v(\bar{e}'_i, v_i, u), & \bar{e}'_i &= \rho^{e\to u}(E'), \\ u' &= \phi^u(\bar{e}', \bar{v}', u), & \bar{v}'_i &= \rho^{v\to u}(V'), \end{aligned}$$

where updated attributes are primed, and edges, nodes and global attributes, respectively e_k, v_i and u, are given by ϕ functions. Besides, edges attributes are aggregate by ρ functions, which can be the mean, median, sum (any aggregation function) so as to be produced by nodes that are pointed by the edges or by the global attributes. The same is true for aggregating node attributes for the purpose of updating global attributes. For a GNN, each ϕ function is encoded by a neural network; all of these networks must have their structure designed and their weights learned — as we discuss latter for our particular application.

2.2 The Sepetiba/Ilha Grande Bay Prediction Problem

The Sepetiba/Ilha Grande Bay is located in the State of Rio de Janeiro, southeast region of Brazil. It is a sheltered area close to Rio de Janeiro city, with different port facilities and heavy traffic of ships. Figure 2 shows the region of interest within South America (left) and the four main port terminals, TEBIG, TIG, Port of Sepetiba, and CSN (right).

The state of the sea and the weather conditions have a direct impact on the operations of commercial sea-going vessels, largely determining the safety of the navigation. In fact, during extreme events, such as low visibility or strong currents, the traffic of ships must be interrupted. The short-term prediction of environmental parameters (within 24 to 48 hours) is an essential duty of port authorities; in the case of shared channels with multiple ship sizes (as in the present case), prediction becomes even more critical. We are interested in predictions within a time frame of 24 hours, as this is the most typical prediction window in this context.

The prediction of environmental parameters is generally made with an array of physical models of atmospheric and hydrodynamic circulation. The inputs of the method are the boundary conditions such as tidal variation, low-resolution satellite or global model information and local wind measurements [PIANC MarCom 2012]. They also depend on an accurate 3D grid model of the area (coastline and bathymetry). Recent years have witnessed a growing number of measured data, due to new sensing, data transmission and storage technologies, and a consequent call for more sophisticated data-driven techniques for time-series prediction [Wu et al. 2019]. The data-based model is independent of the physical models and external inputs, being automatically updated as soon as a new set of

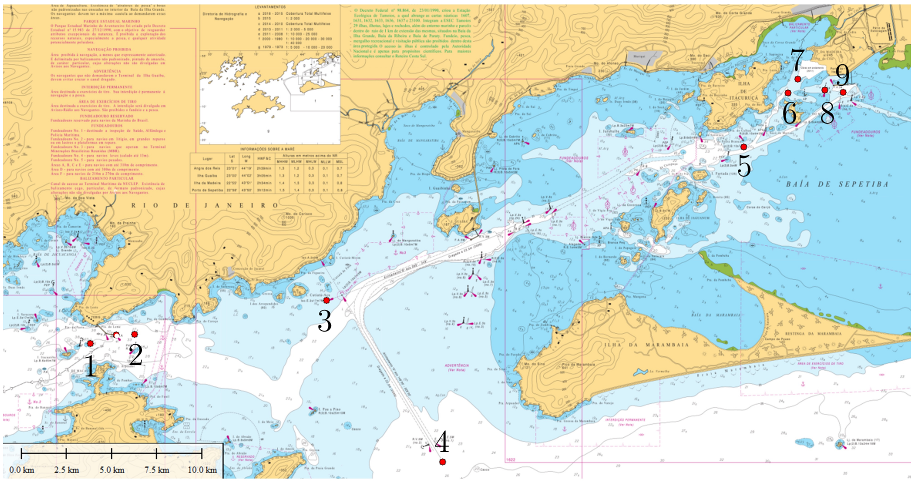


Fig. 3. Location of buoys measuring current, tide, wind and visibility. Names of buoys: 1) B18 2) BEV 3) TIG 4) TIG 1 5) Bifurcation 6) Evolut 7) Pier 8) BPA 9) B22

measurements is available. It is within this trend towards data-driven modeling that we operate.

Here we adopt a novel modeling approach that uses a spatially distributed network of sensors along the Sepetiba/Ilha Grande Bay so as to predict the speed of water current at a single buoy referred to as Bifurcation. The network consists of nine navigational buoys located as indicated in Figure 3 (note that Bifurcation appears there over number 6). Each buoy collects measurements of variables related to tide elevation, current and wind speed, and visibility. The water current speed in a bay strongly depends on the tide elevation; there is also weaker dependency on wind speed (superficial drag) and on meteorological effects (indirectly indicated by the visibility). All dependencies, as well as their temporal evolution, must be captured by a predictor.

Each buoy collects a round of measurements every ten minutes. Due to the relatively slow dynamics of the system, we decided to sub-sample and look at twenty minute intervals between measurements. Notably, data collection is rather faulty: several features are missing in many measurement rounds, and in some cases buoys do not report on one or more features for months. This calls for the benefit of a GNN approach that exploits the spatial information.

3. A GNN FOR PREDICTION

Following Battaglia et al. [Battaglia et al. 2018], we take a (directed) attributed graph to be a tuple $G = (V, E)$, where $V = \{\mathbf{v}_1, \ldots, \mathbf{v}_N\}$ is a set of node attributes (real-valued vectors), and $E = \{(\mathbf{e}_1, r_1, s_1), \ldots, (\mathbf{e}_M, r_M, s_M)\}$ is a set of triples containing an edge attribute $\mathbf{e}_k$ (a real-value vector) over the edge (r_k, s_k).[1]

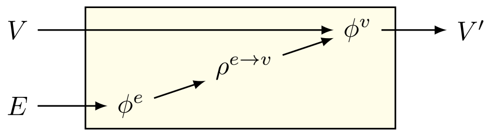
Our underlying graph is specified as follows. Each node represents a buoy whose attributes are current velocity component in x-axis, current velocity component in y-axis, wind velocity component in x-axis, wind velocity component in y-axis, local sea level, local temperature and local visibility. As for the edges, an expert provided domain knowledge by selecting which attributes of nodes should affect other node attributes. The edge attributes are a projection of the adjacent node attributes current velocity component in x-axis, current velocity component in y-axis and local sea level.

[1] [Battaglia et al. 2018] also defines a global attribute $\mathbf{u}$ which we do not use in this work.

We consider two types of graph topology: a fully connected version (which we call “non-local” model) and a fully disconnected version (which we call “local” model). The first version captures effects between buoys (as they all share a geographic location and are spatially related); the second version does *not* use the power of Graph Neural Networks and was built only for purposes of comparison.

That is, we have two schemes:

• A non-local neural network updates node attributes taking all relational information into account [Wang et al. 2018]. We can represent this architecture as:



The function $\mathbf{e}'_k = \phi^e(\mathbf{e}_k)$ updates each edge attribute $\mathbf{e}_k$. These updated attributes are aggregated via a simple sum as $\bar{\mathbf{e}}_i = \rho^{e\to v}(\{\mathbf{e}_k\}_{k:r_k=i}) = \sum_{k:r_k=i} \mathbf{e}'_k$. Finally, the function $\mathbf{v}'_i = \phi^v(\bar{\mathbf{e}}_i, \mathbf{v}_i)$ updates each node attribute $\mathbf{v}_i$ based on the aggregate attribute $\bar{\mathbf{e}}_i$. As stated in Section 2.1, in a GNN approach, the functions applied to graph's entities are neural networks. In this sense, the update functions described above as ϕ^e and ϕ^v are neural networks (NN) fed by entity's attributes. In other words, given the buoys' attributes on a time step, that are in graph's nodes and edges, their values in the next step are predicted by updating edges' attributes, $\mathbf{e}'_k = NN^e(\mathbf{e}_k)$, which is an independent variable for updating nodes' attributes: $\mathbf{v}'_i = NN^v(\bar{\mathbf{e}}_i, \mathbf{v}_i)$.

• A local neural network that only looks at attributes of the node; that is, we have only $\mathbf{v}'_i = \phi^v(\mathbf{v}_i)$.

Note that the second scheme (the local one) benefits from the graph structure as the function ϕ^v is shared among all nodes, although information between nodes are not shared between them.

To enable the GNNs to capture the temporal evolution of signals, we concatenate, for each node and edge, measurements from the past 48h, collected every 20 minutes. This effectively generates node attributes $\mathbf{v}_i \in \mathbb{R}^{145\times 7}$ and the edge attributes $\mathbf{e}_k \in \mathbb{R}^{145\times 3}$ (i.e., 144 observed data points plus the next-step point to be predicted).

4. EXPERIMENTS

We took a large dataset consisting of data from 2018-01-01 00h00 to 2019-12-31 23h50 of the four locations (TEBIG, TIG, CSN and Port of Sepetiba), which gives us, approximately, 105k data points containing measurements from nine buoys. About 3.2 million of cells were missing in this dataset which corresponds to roughly 43% of the cells.

We completed the dataset using two distinct methods. First we implemented an average-based imputation method as follows. Suppose we consider a buoy and a feature is missing there at some time step, then, if a feature was missing for some buoys and that feature was present in the majority of other buoys, we imputed their average when their variance was small, and we imputed the median otherwise; if instead the majority of buoys also lacked that feature, then we took an average in time around the time step of interest just for measurements of the buoy. Second, we used MICE imputation, a popular MAP-based scheme that fits local statistical models to impute values [Buuren and Groothuis-Oudshoorn 2011]. Our experiments indicated that the average-based imputation led to slightly better performance, so later we report results with an average-completed dataset.

We implemented our GNN using DeepMind's Graph Nets library[2], adapting it as needed to our model. Given the modeling assumptions described in the previous section, we have 2.415.367 pa-

[2] `https://github.com/deepmind/graph_nets`.

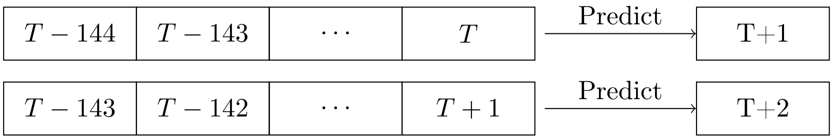


Fig. 4. Forecasting procedure example.

rameters to fit in the non-local scheme, and 2.020.359 parameters in the local scheme. Models were trained in a 5k datapoint batch with sliding window of 6 datapoints for training and 3 datapoints for validation (a 67%/33% split of training/validation set). The Adam optimizer was used with a learning rate of 1e-4. The models were trained for 5 iterations. The neural networks' architectures used for entities had 5 layers and 256 units/layer. All these hyperparameters — learning rate, number of layers and number of units per layer — were tuned using grid search.

We then used the model to predict the water current at Bifurcation for the next 24h in a sequential fashion. That is, we use data from the past 48h to predict the next measurement. We then shift our time window to incorporate this prediction as if it were a measurement, and predict measurements for the second 20 min interval of the day. We continue this way until all $[24h] \times [3 \text{ measurements}/h] = 72$ measurements have been predicted. A 2-step prediction example is shown in Figure 4. As stated before, given that the objective is predicting the water current to the following 24h, 72-steps similar to these showed in Figure 4 are needed.

Empirical tests indicated that average-imputation led to the best results, so we kept it. Testing also indicated that for both local and non-local neural networks schemes (presented in Section 3), learned models are successful in tracking the dynamic behavior of signals (i.e., both models are good at following the signal behavior). In particular, predictions were able to fit the number of peaks in the prediction period, a nontrivial behavior as the number of peaks is not constant, as shown in Figures 5 and 6. However, both models had difficulty adjusting to the value of peaks in signals. Overall mean squared error is 0.02 $knots^2$ for the non-local model and 0.10 $knots^2$ for the local model (squared error is measure by subtracting observed and predicted values across time).

Figure 5 depicts a prediction run, comparing our model with baseline models extensively used in time series prediction — specifically, ARIMA [Brockwell and Davis 1987] and LSTM [Hochreiter and Schmidhuber 1997]. Both baselines were trained in a 20k datapoint and 50 iterations. Our model performs significantly better than ARIMA, and performs slightly better than LSTM, but with notable data efficiency — less data points and iterations to train (about a fourth of data points). Comparing the two graphical models, non-local and local, best results are attained with the first one, which indicates that our hypothesis that a model could benefit from sharing and combining information among its entities is true. Given that LSTM baseline and GNN non-local were the best models, we also present in Figure 5 the squared error of both models along the time window.

Note that at the Bifurcation buoy, where we predicted current speed, the main current component is the x-axis; positive current speed means, essentially, that the speed vector points to the east; a negative current speed means that the speed vector points to the west.

5. CONCLUSION

We have described a real application where the structuring resources of Graphical Neural Networks are advantageous: they simplify modeling, facilitate understanding of the model, and lead to accuracy gains as related to purely local modeling. Besides, our application should be a paradigmatic example for prediction scenarios that depend on sensor networks, as such networks usually operate with a spatial structure. We thus believe that results reported here can be of great interest to practitioners and researchers.

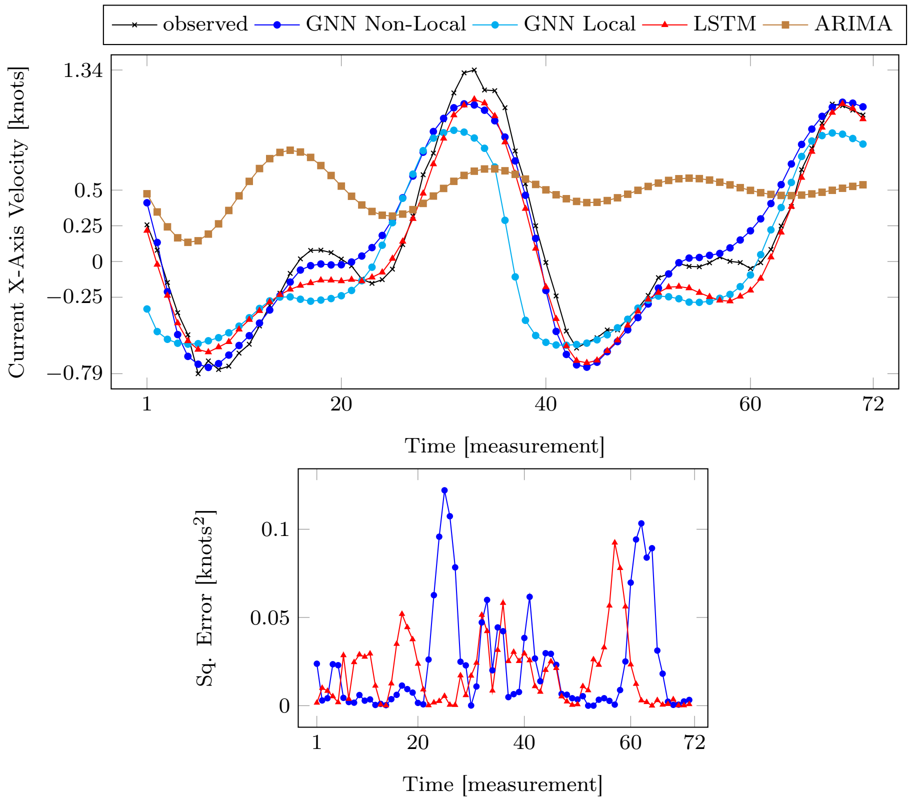


Fig. 5. Results for a selected time window, comparing our models and baselines.

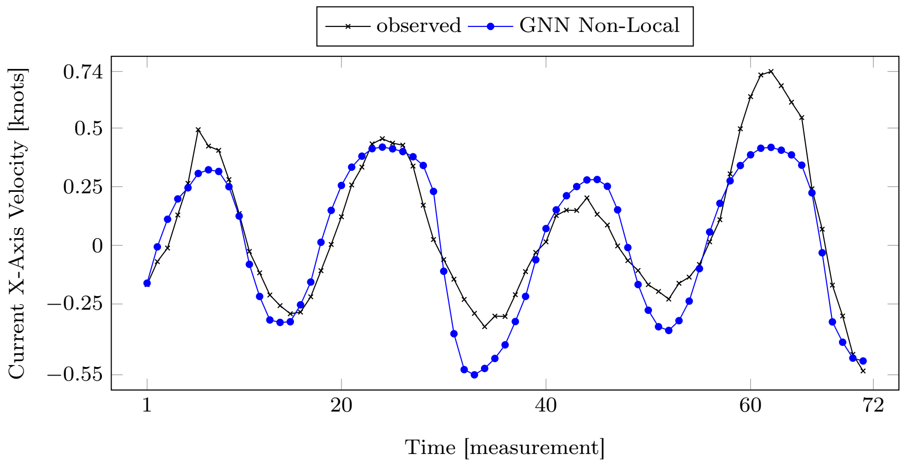


Fig. 6. Results for a different time window.

Much can be done to refine our initial efforts. We feel there is room for improvement concerning hyperparameters — explore hyperparameters' space with local search heuristics — and some critical design decisions, such as the use of 20 minute intervals; a more comprehensive study will require quite some data manipulation and computation time but it may be worth the investment. Also, we should explore other imputation techniques, as imputation is critical given that 43% of the input dataset is missing; while our current solution is acceptable, we hope to run an EM-like iterative scheme. Finally, we should compare our data-driven solution with existing hydrodynamic modeling tools; usually such tools are not as flexible as one would like as they require extensive fiddling when conditions change, but they may provide valuable additional inputs, and at least they should offer an additional comparison.

REFERENCES

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gulcehre, C., Song, F., Ballard, A., Gilmer, J., Dahl, G., Vaswani, A., Allen, K., Nash, C., Langston, V., Dyer, C., Heess, N., Wierstra, D., Kohli, P., Botvinick, M., Vinyals, O., Li, Y., and Pascanu, R. Relational inductive biases, deep learning, and graph networks. Tech. rep., arXiv:1806.01261, 2018.

Brockwell, P. J. and Davis, R. A. *Time Series: Theory and Methods.* Springer, 1987.

Buuren, S. and Groothuis-Oudshoorn, K. MICE: Multivariate imputation by chained equations in R. *Journal of statistical Software* 45 (3): 1–67, 2011.

Gori, M., Monfardini, G., and Scarselli, F. A new model for learning in graph domains. In *IEEE International Joint Conference on Neural Networks.* Vol. 2. pp. 729–734, 2005.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation* vol. 9, pp. 1735–80, 1997.

PIANC MarCom. Use of hydro/meteo information for port access and operations. Tech. rep., WG 117 Report 117, 2012.

Scarselli, F., Gori, M., Tsoi, A., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE Transactions on Neural Networks* 20 (1): 61–80, 2009.

Sperduti, A. and Starita, A. Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks* 8 (3): 714–735, 1997.

Wang, X., Girshick, R., Gupta, A., and He, K. Non-local neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition.* pp. 7794–7803, 2018.

Wu, M., Stefanakos, C., Gao, Z., and Haver, S. Prediction of short-term wind and wave conditions for marine operations using a multi-step-ahead decomposition-ANFIS model and quantification of its uncertainty. *Ocean Engineering* vol. 188, pp. 106300, 2019.