Modeling Oceanic Variables with Dynamic Graph Neural Networks

Caio F. D. Netto^{*1} , **Marcel R. de Barros**^{*1} , **Jefferson F. Coelho**^{1} , **Lucas P. de Freitas**^{1} ,
Felipe M. Moreno^{1,3} , **Marlon S. Mathias**^{3,4} , **Marcelo Dottori**^{2} ,
Fábio G. Cozman^{1,3} , **Anna H. R. Costa**^{1,3} , **Edson S. Gomi**^{1,3} , **Eduardo A. Tannuri**^{1,3}
^{1}Escola Politécnica – University of Sao Paulo, Brazil
^{2}Instituto Oceanográfico – University of Sao Paulo, Brazil
^{3}Center for Artificial Intelligence (C4AI) – University of Sao Paulo, Brazil
^{4}Instituto de Estudos Avançados (IEA) – University of Sao Paulo, Brazil
{caio.netto, marcel.barros, jfialho, lfreitasp2001, felipe.marino.moreno, marlon.mathias, mdottori, fgcozman, anna.reali, gomi, eduat}@usp.br

Abstract

Researchers typically resort to numerical methods to understand and predict ocean dynamics, a key task in mastering environmental phenomena. Such methods may not be suitable in scenarios where the topographic map is complex, knowledge about the underlying processes is incomplete, or the application is time critical. On the other hand, if ocean dynamics are observed, they can be exploited by recent machine learning methods. In this paper we describe a data-driven method to predict environmental variables such as current velocity and sea surface height in the region of Santos-Sao Vicente-Bertioga Estuarine System in the southeastern coast of Brazil. Our model exploits both temporal and spatial inductive biases by joining state-of-the-art sequence models (LSTM and Transformers) and relational models (Graph Neural Networks) in an end-to-end framework that learns both the temporal features and the spatial relationship shared among observation sites. We compare our results with the *Santos Operational Forecasting System* (SOFS). Experiments show that better results are attained by our model, while maintaining flexibility and little domain knowledge dependency.

1 Introduction

Machine Learning (ML) has shown promising results in many fields, not only as an applied tool but also as a main driver of scientific discovery [Raccuglia *et al.*, 2016; Cranmer *et al.*, 2020; Jumper *et al.*, 2021]. If in the past environmental models were governed by first-principle models based on established science, it is now clear that this is not the ideal approach in a handful of situations. There are domains in which the underlying phenomena and the governing equations are not well understood or the complexity of the problem is enormous, making it unfeasible in practice to solve them through first-principle models.

One such domain is the oceanic one. Being able to understand and predict how the ocean dynamics works is a major concern for coastal countries both economically and socially. Techniques have been proposed so as to model ocean dynamics numerically [Blumberg and Mellor, 1987; Ribeiro *et al.*, 2019; Costa *et al.*, 2020]. The goal is often to anticipate extreme phenomena, for instance storm surges, which can cause transportation delays and accidents, when forecasting current velocity and sea surface height. However, as stated before, physical models are costly to design, implement, and maintain, given that they require accurate measurements of the region of interest topology for spatial representations and for boundaries conditions.

Conversely, ML models have been successfully applied to a variety of physical problems [Xu and Valocchi, 2015; Sanchez-Gonzalez *et al.*, 2020; Lira *et al.*, 2022] and in particular in the oceanic domain [Ibarra-Berastegi *et al.*, 2015; Xiao *et al.*, 2019; Netto *et al.*, 2020]. These models circumvent the computational cost of physical models while attaining excellent results. As measurement tools have improved, both public and private interest to record oceanic data has risen in recent years [PIANC MarCom, 2012]. One can explore ML models with such historical data, aiming to implement and deploy better ocean dynamics prediction systems. However, as sensors and measurement tools are "in the wild", the observation of hydrodynamic and meteorological variables are highly affected, for instance, by extreme or unexpected environmental conditions or technical glitches. So, key variables for ocean dynamics, such as current and wind velocity and sea surface height, tend to display noisy and missing data.

In order to build a data-driven model that deals with those constraints, specially missing data and diverse kinds of time series, a variety of ML models have been proposed. Sequence models like LSTM and Transformers [Hochreiter and Schmidhuber, 1997; Vaswani *et al.*, 2017] and Graph Neural Networks (GNNs) [Scarselli *et al.*, 2009] are suitable for this task as they incorporate both temporal and spatial inductive biases into their architecture.

*Equal contribution by first and second authors. All other authors made significant suggestions and the last five authors oversaw the whole project and secured funding.

In this work we aim to address a time series forecasting problem in the context of ocean dynamics, proposing a spatio-temporal GNN architecture to predict current velocity and sea surface height, and using multivariate time series data collected at the Santos-Sao Vicente-Bertioga Estuarine System. Forecasting oceanic variables is a major concern, in general, for both public actors and private port authorities, and it is even more in that area, which is home to the largest port in Latin America, the Port of Santos.

We summarize the main contributions of this work as follows:

- We propose a general model capable of dealing with problems that have both temporal and spatial dimensions and significant levels of missing data.
- We address a real problem in the context of physical sciences, using a data-driven method rather than physical model-based simulations.
- We demonstrate experimentally that the proposed model surpasses the physical model SOFS, our baseline, by 27% on water current speed prediction and 14% on water current direction prediction, while maintaining excellent performance in forecasting sea surface height, considering the Willmott index [Willmott, 1981], also known as the Index of Agreement (IoA).

Regarding the structure of the paper, we introduce in Section 2 previous works related to machine learning models in the context of relational learning, multivariate time series prediction and ocean domain. After, in Section 3 we describe the basic concepts and the problem we address, while in Section 4 we use that theory to model our problem. In Section 5 we detail our architecture. Sections 6 and 7 close the paper reporting our experimental setup, the obtained results and discussions, conclusions and planned future work.

2 Related Work

In an effort to address all dimensions in which our problem is embedded, we quickly highlight the most recent research that develops new methods for forecasting multivariate time series in the oceanic domain, where temporal and spatial biases matter and data-driven methods produce better results. A number of statistical methods have been proposed to address these kinds of problems. Wei (2019) presents in an intuitive and comprehensive way a handful of those methods, like autoregressive integrated moving-average for the case of multivariate time series data, applying them, for example, to real public health problems. Nonetheless, these statistical models are known to face difficulties in capturing long-term relations and seasonal components.

Due to their flexibility, ML models have shown excellent results in multivariate time series forecasting. For example, simple models like Quantile Regression Forests (QRF) were used in [Moreno *et al.*, 2022] to mitigate the error of a physics-based numerical model, built to forecast the surface current of an ocean region. The authors proposed a different and flexible framework that models the problem from the "backdoor": they exploit the existence of a physical model and build one that models the residual error between the former and the surface current true value.

Recent works have been focused on applying deep learning models or implementing their own, due to both the impressive performance in a variety of tasks and the capability of incorporating domain specific knowledge into those models' architecture. For instance, in [Ziat *et al.*, 2017] the authors proposed a spatio-temporal Recurrent Neural Network (RNN) model for time series forecasting, combining a sequential model with the dependence between time series of locations spatially separated. In the field of application of this work, others have even proposed the combination of state-of-the-art models like Convolutional Neural Networks (CNN) and RNN, building spatio-temporal time series models that benefit from both inductive biases [Xiao *et al.*, 2019].

In order to better model relationships between entities of a problem, Graph Neural Networks (GNN) showed promising results and well-established theory to do so. Lira *et al.* (2022) address the time series problem of frost forecasting, using GNNs with attention to model an experimental site spatially and temporally. In [Cao *et al.*, 2020], the authors compose both convolution and sequential learning in a relational architecture (GNN), extracting richer features in the frequency domain through Fourier Transform, aiming to model multivariate time series forecasting problems. Specifically to the ocean domain, [Netto *et al.*, 2020] propose an application of GNNs to model ocean variables in the form of time series, in a real problem of an economically important coastal region, similarly to [Lira *et al.*, 2022].

3 Background

In this section we outline the main ideas behind modeling time series as a graph and GNNs (Subsection 3.1), and the problem we address in this work (Subsection 3.2).

3.1 Dynamic Graph Neural Networks

Our work adopts terminology by [Kazemi *et al.*, 2020] in which a *continuous-time dynamic graph* (CTDG) is a pair $(\mathcal{G}, \mathcal{O})$ where $\mathcal{G}$ is a (static) initial graph and $\mathcal{O}$ is a set of observations/events of the form $(event\ type, event, timestamp)$ that can alter graph structure, node attributes, and edge attributes.

Each static graph is defined by a pair of sets $(\mathcal{V}, \mathcal{E})$. $\mathcal{V}$ is a set of nodes and $\mathcal{E}$ is a set of edges. Following the structure proposed by [Satorras *et al.*, 2022], we work with a fully connected graph. A *discrete-time dynamic graph* (DTDG) can be defined as a set of snapshots $\{\mathcal{G}^1, \mathcal{G}^2, \dots, \mathcal{G}^T\}$ sampled from an underlying CTDG.

Finally, a *Graph Neural Network* [Scarselli *et al.*, 2009] can be described as message passing mechanism which iteractively updates nodes' hidden representations. This mechanism can be summarized as follows:

$$\mathbf{x}_i^{(k)} = \gamma^{(k)}\left(\mathbf{x}_i^{(k-1)}, \square_{j \in \mathcal{N}(i)}\, \phi^{(k)}\left(\mathbf{x}_i^{(k-1)}, \mathbf{x}_j^{(k-1)}\right)\right), \quad (1)$$

where $\mathbf{x}_i^{(k)}$ is the hidden representation (or node features) of a node i at iteration k, $\phi^{(k)}$ is a differentiable *message function* that constructs a message to be propagated based on both sender ($\mathbf{x}_j$) and receiver ($\mathbf{x}_i$) nodes, $\square$ denotes a differentiable, permutation invariant *aggregation function* that aggregates all messages from neighbour nodes and $\gamma^{(k)}$ is a differentiable *update function*, that updates the node's hidden representation.

Both ϕ and γ are parametrized by a set of learnable parameters that are iteratively updated during the training process. Examples of such functions would be Multi Layer Perceptrons (MLPs).

3.2 The SSVBES Forecasting Problem

The Santos-Sao Vicente-Bertioga Estuarine System (SSVBES) is located on the southeast coast of Brazil, as part of the South Brazil Bight. This estuarine system, like others over the world, has its hydrodynamics driven mainly by three forcing processes, which are the astronomical tide, meteorological tide and river discharge. The meteorological tide is dependent on the synoptic winds that blow over the adjacent continental shelf, associated with simple Ekman dynamics and, therefore, is a process that occurs off the Santos Bay and enters it as a gravity wave. When winds blow from North-Northeast, sea surface height (SSH) decreases and, when winds from South-Southwest are dominant, SSH increases. We present that region in Figure 1 as well as its three main channels, Sao Vicente Channel, Santos Port Channel, and Bertioga Channel, that constitute the estuarine system.

An effort to understand the SSVBES behavior under the action of these forcings has been coded into the *Santos Operational Forecasting System* (SOFS) [Costa *et al.*, 2020], which provides daily forecasts for the region. SOFS adopts a finite difference model that uses the *Navier-Stokes* equation with sigma vertical coordinates, is forced by the winds, tides, density gradients and river discharge, with good results for both SSH and currents.

However, data availability of river discharge limits the accuracy outputs of this model because this is not a variable easily obtained in a near-real time frame. This limitation is mainly observed in current results, as river discharge in estuaries causes changes in flow, directly, and in vertical density stratification, which is also capable of changing currents.

4 Modeling SSVBES with DTDGs

We model SSVBES as a DTDG comprised of snapshots sampled from an underlying CTDG. Each node in this CTDG is a pair $(type, location)$ denoting a node type and a location. Possible node types are oceanic variables such as SSH, water current, and wind. Each different $type$ has k_{type} features of which l_{type} are labels. Possible locations are measuring stations in the SSVBES system from which data is collected. For this region, we have access to observations in six different sites located as indicated in Figure 2.

Each data point collected is an event associated with the node $(type, location)$. For example, a measurement from a water current velocity sensor in Praticagem station – located in the observation site 5 – is considered an event associated with the node $(current, praticagem)$.

4.1 Sampling the CTDG

To build a snapshot of the CTDG associated with an instant t we use the following procedure:

1. Define a past window time interval (*past_len*) and filter events having $timestamp \in [t - past\_len, t[$.
2. Define a future/prediction window time interval (*future_len*) and filter events having $timestamp \in [t, t + future\_len[$.
3. A node x_i associated with the pair $(type, location)$ is part of the snapshot if the past window has at least one event associated with that pair.
4. Each node x_i associated with the pair $(type, location)$ is built with three time-sorted sequences of events:
   - $x_i^{(past)} \in \mathbb{R}^{ps_i \times k_{type}}$: sequence with ps_i past events;
   - $x_i^{(future)} \in \mathbb{R}^{fs_i \times (k_{type} - l_{type})}$: sequence with fs_i future events without label columns;
   - $y_i \in \mathbb{R}^{fs_i \times l_{type}}$: sequence with fs_i future events with only label columns.

The future features sequence $x_i^{(future)}$ stores only features that are associated to future events but whose values are available beforehand such as astronomical tide and timestamp encoding columns.

Note that ps_i and fs_i may vary for each node x_i depending on its data collection periodicity and eventual missing data points (e.g. sensor outage).

To build the complete DTDG graph $\mathcal{G} = \{\mathcal{G}^1, \mathcal{G}^2, \ldots, \mathcal{G}^N\}$ we sample $\boldsymbol{N}$ uniformly spaced snapshots from the underlying CTDG.

4.2 Encoder-Decoder modeling

Following the definition from [Kazemi *et al.*, 2020], an encoder receives a dynamic graph as input and outputs an embedding function, while a decoder receives an embedding function as input and performs a forecasting task.

Our forecast model is comprised of two sets: a set of encoders $Enc = \{Enc^{type_1}, Enc^{type_2}, \ldots, Enc^{type_E}\}$, and a set of decoders $Dec = \{Dec^{type_1}, Dec^{type_2}, \ldots, Dec^{type_E}\}$, where E represents the number of oceanic variables being processed.

Each encoder Enc_{type} is comprised of two modules, namely, a *temporal encoder* and a *GNN* for information propagation between nodes. While the temporal encoder is different for each $type$, the *GNN* module is shared amongst all nodes.

The temporal encoder receives $x_i^{(past)}$ and $x_i^{(future)}$ as input and outputs a single fixed size hidden representation h_i.

The GNN updates all h_i based on graph neighbourhood as detailed by (1), enabling information from different nodes to be shared.

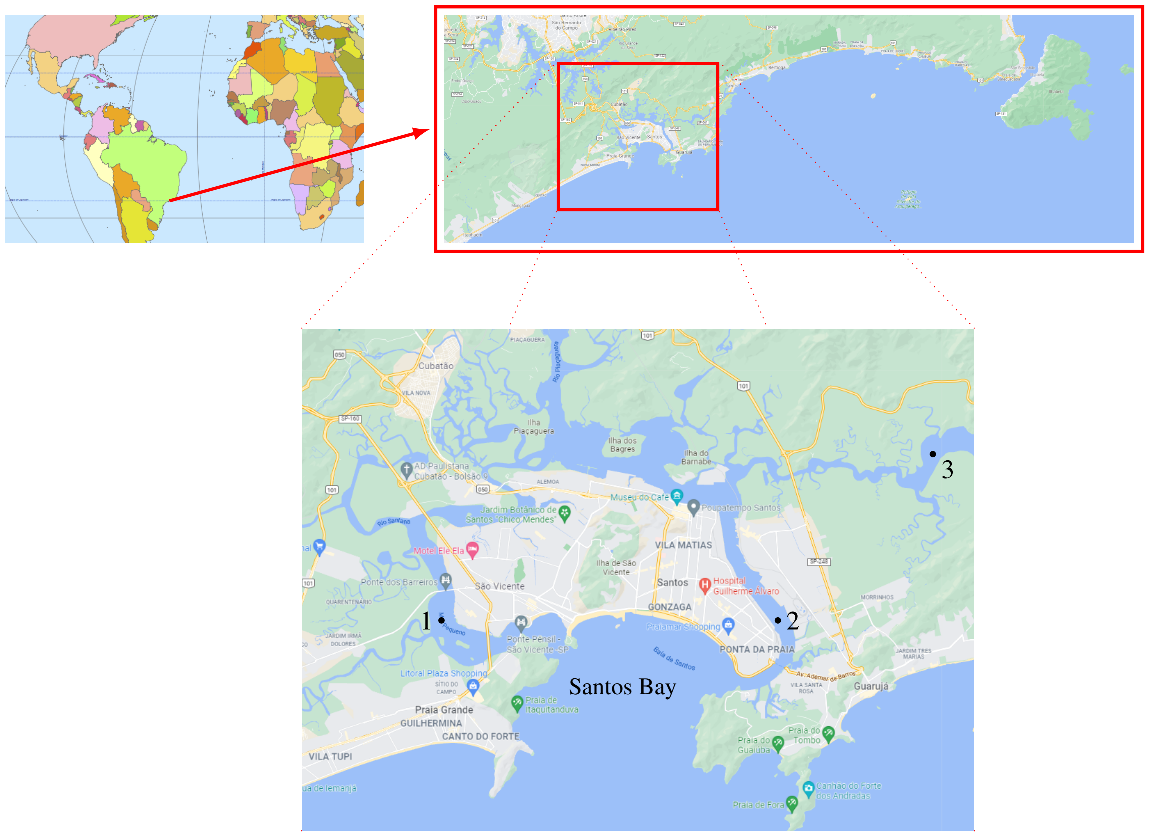

Figure 1: The Santos-Sao Vicente-Bertioga Estuarine System. In the bottom figure, the main locations: 1 - Sao Vicente Channel; 2 - Santos Port Channel; 3 - Bertioga Channel.

The decoder receives h_i as input and outputs a forecast $\hat{y}_i$ for a subset of $\mathcal{V}$.

This setup enables, for example, the sea surface height and water current nodes to have different *temporal encoders* and *decoders* that may have different architectures. This flexible approach can be adapted to a vastly different set of input variables, as any time series can be encoded into a node representation and have its information propagated through the graph structure.

5 Proposed Architecture

We propose a modularized architecture, shown in Figure 3, that fits in the encoder-decoder category, thus enabling the use of different sequence processing architectures for encoding and decoding of each different oceanic variable. Also, we separate the concept of a *GNN Block* from its inner *GNN Convolution* object allowing it to be replaced by newer graph convolution architectures while still maintaining normalization layers and skip connections. In this section we describe each one of these components.

5.1 Encoders

Each Enc_{type} encoder receives two sequences of variable length and outputs a single hidden representation h_i of fixed length $embed\_size$.

Temporal Encoding

We implement the *temporal encoder* described in Section 4.2 with two different sequence models, namely a Transformer and an LSTM, and compare results. Both models can be described as:

$$\xi\colon \mathbb{R}^{ps_i \times k_{type}} \times \mathbb{R}^{fs_i \times (k_{type} - l_{type})} \rightarrow \mathbb{R}^{embed\_size}.$$

In both cases, we use two instances of the same architecture to encode both $x_i^{(past)}$ and $x_i^{(future)}$. To form the final fixed size embedding, we concatenate the results into a single vector $h_i = [h_i^{(past)}, h_i^{(future)}]$.

Spatial Encoding

Given the fixed sized embedding h_i we use a GNN block to update the embedding with information from incoming edges. The GNN block is described as:

$$\mathbb{G}\colon \mathbb{R}^{embed\_size} \rightarrow \mathbb{R}^{embed\_size}.$$

Inspired by [He *et al.*, 2016] we establish a block with a skip connection and normalization layers as depicted in Figure 3.

Each GNN block contains graph convolutions that are responsible for aggregating neighbourhood information. Our implementation uses a Graph Attention Convolution, first proposed by [Veličković *et al.*, 2018] and then further improved by [Brody *et al.*, 2021].

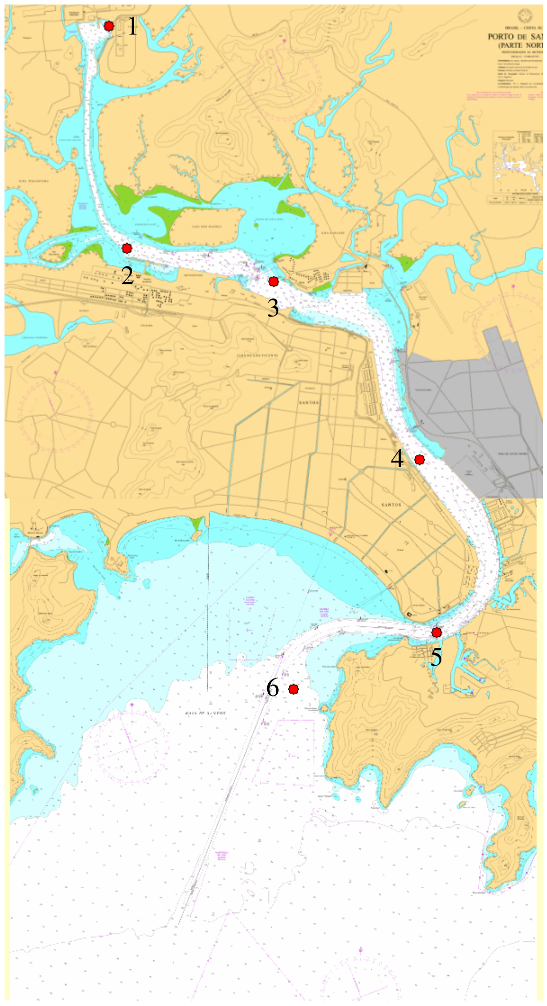

Figure 2: Santos-Sao Vicente-Bertioga Estuarine System, and the location of all six observation sites used in the experiments: 1) TIPLAM, 2) Alemoa, 3) Ilha Barnabé, 4) CPSP, 5) Praticagem, 6) Palmas.

5.2 Decoders

Each Dec_{type} decoder receives a single hidden representation h_i of fixed length $embed\_size$ as input and outputs a sequences of length fs_i. Our implementation uses two different decoder architectures. A *Fixed Output Size MLP* is employed to decode water current velocity data of the form:

$$\mathbb{D}_{fix} : \mathbb{R}^{embed\_size} \rightarrow \mathbb{R}^{max(fs_i) \times l_{type}}.$$

We call this architecture a fixed output size one, because $max(fs_i)$ is the maximum length of the output sequence and is defined beforehand based on training data. To decode sea surface height related data, we employ a *Dynamic Output Size MLP* that can be viewed as:

$$\mathbb{D}_{dyn} : \mathbb{R}^{embed\_size+embed\_size/2} \rightarrow \mathbb{R}^{l_{type}}.$$

In this architecture, the decoder also receives $h_i^{(future)}$ as input and concatenates it to the embedding h_i to form the final input to the MLP. This format resulted in much better results for SSH forecast.

We hypothesize that this is due to the fact that $h_i^{(future)}$ for $type = SSH$ contains astronomical tide information which is the major factor defining SSH behaviour. The model seems to benefit from receiving this information directly in the decoder. We leave a more detailed study of this behaviour to future work.

6 Experimental Setup

Here we describe our datasets and its characteristics (Subsection 6.1), the model configurations and the stack of tools used to implement and run our experiments (Subsection 6.3), and our experimental design (Subsections 6.2 and 6.4)

6.1 Datasets

As input to the model, we used environmental sensing data collected from the 6 regions of the SSVBES between 1/1/2019 and 1/9/2021 (totaling 974 days).

As the sensors responsible for data acquisition are directly affected by climate and environmental conditions, the input features presents the following percentage of missing data: Current: 24.3%, SSH: 42.1% and Wind: 84.1%, totaling approximately 50.1% missing data from the sensing date range. The data are also unbalanced as to the percentage of data obtained in the different locations, as seen in Figure 4.

The monitoring data was aggregated using a 30-minute step between windows, which means there are 3 graphs $\mathcal{G}$ in the flow data for each hour. In addition, data inputs are normalized by Z-score.

To deal with the characteristics of the available data, the proposed graph model does not need strict input shape or contiguous data. Each node of the model structure receives an independent data window as input after available features data encoding.

6.2 Scenarios

To assess our model performance, we designed 4 experimental scenarios.

Water Current and SSH Forecasting In the first two experiments we evaluated the performance of the proposed model in predicting the current velocity and sea surface height, separately. We compared these results to SOFS. We used the same set of hyperparameters to train both models.

Fully Connected *vs* Same-variable *vs* Disconnected Graph Next, we experimented and compared three graph topologies: a fully connected graph, a graph with connections only between nodes of the same *type* and a fully disconnected graph. Analogously to the previous experiment, all hyperparameters were fixed and we only modified our adjacency matrix.

LSTM *vs* Transformers as Temporal Encoders Lastly, given our framework flexibility, we compared different state-of-the-art sequence models as temporal encoders. For that experiment, we contrasted the results with an LSTM against Transformers as encoders, keeping the same set of hyperparameters for both cases.

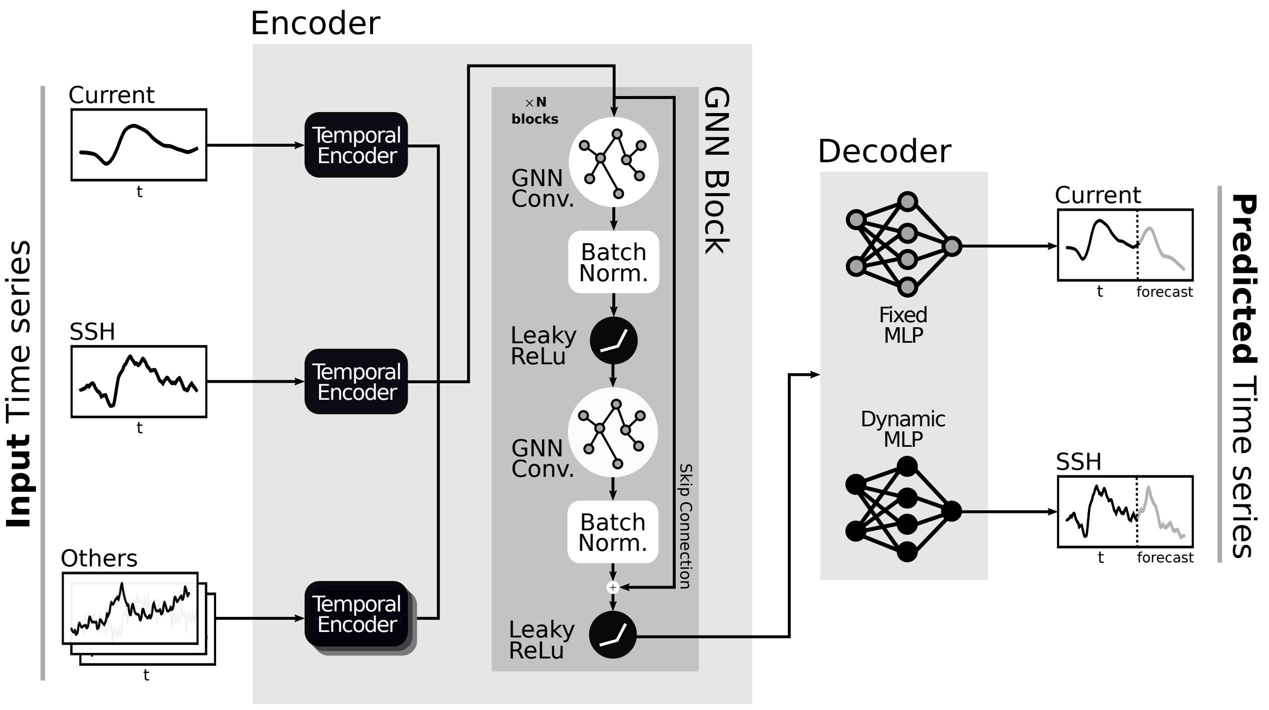

Figure 3: Overall model architecture indicating each *type* associated to a *temporal encoder* and a *decoder* module, while all *types* share the same GNN structure. Internally, the GNN is a sequence of GNN blocks with normalization layers, graph convolutions and nonlinearities.

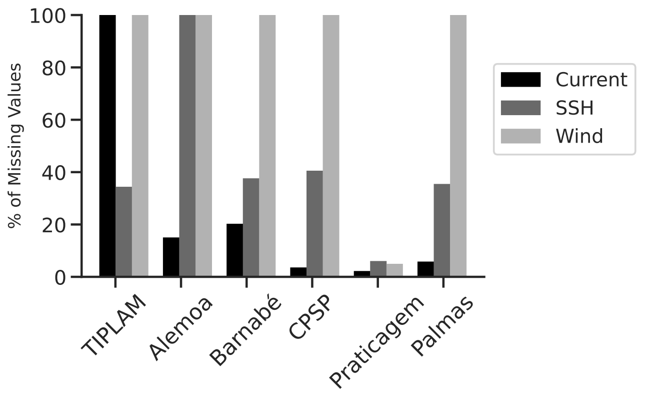

Figure 4: Distribution of missing values for different event types from the SSVBES observation sites.

We tested for each scenario a set of combinations of target variables, which are current velocity and SSH. In the first and second experiments (respectively, predicting current velocity and sea surface height), we fed as input to our model (i) only the corresponding target variable and (ii) both, to predict the corresponding target variable. In the last two experiments we used the 4 possible combinations of input variables. Table 1 summarizes this combination for input and target variables.

For all experiments we optimized our models with respect to the IoA [Willmott, 1981]. Hence, our loss function is defined as:

Scenarios	**Inputs**	**Targets**
1st Experiment	[Current; Current+SSH; Current+SSH+Wind]	[Current]
2nd Experiment	[SSH; Current+SSH]	[SSH]
3rd Experiment	[Current+SSH]	[Current]
4th Experiment	[Current; SSH; Current+SSH]	[Current; SSH]

Table 1: Inputs and target variables for each experimental scenario we designed.

$$\mathcal{L}(\hat{y_i}, y_i) = 1 - IoA = 1 - \frac{\sum_{n=1}^{N}(y_i - \hat{y_i})}{\sum_{n=1}^{N}(|\hat{y_i} - \bar{y_i}| + |y_i - \bar{y_i}|)^2} \quad (2)$$

We also present results for Root-Mean-Squared Error (RMSE) in meters per second, and degrees.

6.3 Model configurations

For experiments, we used well-known frameworks like *Pytorch* and *Pytorch Geometric* [Paszke *et al.*, 2019; Fey and Lenssen, 2019] to implement the model, and *Weights&Biases* [Biewald, 2020] to track all experiments.

We used an LSTM as temporal encoder with 1 layer and a hidden dimension ($embed\_size$) of size 20, and 2 GNN blocks with GATv2 [Brody *et al.*, 2021] as GNN convolution. The only exception is the Transformers experiments, where we used 3 layers of Transformer Encoder with 5 attention heads, the same hidden dimension size and numbers of GNN blocks, but with a less complex message passage GNN convolution in contrast to GATv2, due to overfitting issues during training.

We ran each experiment with 10 different seeds. We here report the average performance of these 10 runs. Again, the exception is the Transformers experiments, which we ran for the best 5 seeds used with the LSTM as temporal encoder and report the average performance for these 5 runs.

The results for the first and the second experiments are reported in Table 2 and Table 3. The third experiment is reported in Table 4, while the last experiment is reported in Table 5 and Table 6.

6.4 Discussion

Table 2 shows that our model surpasses SOFS in all scenarios by more than 17% and 8% for water current speed and direction respectively. This represents a considerable leap in quality of prediction that can be better visualized in Figure 5. While results for all scenarios consistently showed that adding graph connectivity can benefit the model's performance, most improvement is still observed with a single variable as input. This result indicate that the *temporal encoder* is the major component in this gain. Further indication of this can be found in Table 5 in which utilizing a Transformer architecture instead of an LSTM performed even better, surpassing SOFS by more than 27% and 14% for water current speed and direction, respectively.

Nevertheless, both Tables 2 and 5 show relevant gains to water velocity predictions when SSH nodes are added to $\mathcal{G}$. This shows that our model is able to aggregate information from SSH nodes into water current nodes' representation. SSH results in Table 3 and 6 did not benefit from the addition of current velocity data, but it is important to consider that SSH modeling in SOFS is already highly accurate and may present fewer opportunities of improvement.

Another important effect to note is that adding nodes can decrease the model's performance. A small indication of this effect can be seen by a slight decrease in IoA when wind data is added. More directly, by analysing Table 4 it is possible to note that a fully disconnected architecture performs slightly better than a graph with only nodes of the same *type* connected in the case of water current velocity modeling. This indicates that aggregating current velocity data from other observation sites did not benefit our model's accuracy.

This effect may be related to both *over-squashing* [Alon and Yahav, 2021; Topping *et al.*, 2021] of node embeddings and to the fact that our model uses a homogeneous GNN model in which functions ϕ and γ from expression (1) are the same for all *types*. Heterogeneous GNN models [Zhang *et al.*, 2019] have been demonstrated to perform well in scenarios with multiple node and edge types and constitute a future research direction for our work.

7 Conclusion

We presented a real problem whose structure benefits from a Graph Neural Network modeling strategy that encodes spatio-temporal features and their relations. Our experiments show the feasibility of a data-driven model capable of handling real datasets with missing data and able to incorporate and share multivariate time series information between problem entities that are spatially separated. Furthermore, our model has better performance when compared with physics-based models for the task of forecasting oceanic variables.

	Speed (m/s)		**Direction (degrees)**	
Scenarios	IoA ↑	RMSE ↓	IoA ↑	RMSE ↓
SOFS	0.599	0.178	0.755	85.18
Current	0.706	0.165	0.818	70.25
Current+SSH	**0.726**	0.158	0.842	65.68
Current+SSH+Wind	0.718	0.160	**0.843**	65.29

Table 2: Average results for 10 random seeds using data from all measuring stations to forecast Praticagem station water current speed and direction. In bold the best ones.

	SSH (m)	
Scenarios	IoA ↑	RMSE ↓
SOFS	0.935	0.133
SSH	**0.940**	0.124
Current+SSH	0.939	**0.123**

Table 3: Average results for 10 random seeds using data from all measuring stations to forecast Praticagem SSH. In bold the best ones.

In future work, we intend to look into the following directions:

Latent graph inference: When learning on graphs, one must assume the structure (topology) of the graph *a priori*. However, any such hypothesis may be wrong; in large graph scenarios we end up with an enormous combinatorial problem. So, schemes such as advanced by [Kazi *et al.*, 2022] offer a promising direction to address larger graphs that may include data from other meteorological stations near SSVBES, increasing the number of nodes to hundreds or even thousands, while we learn the nodes connectivity on training.

Heterogeneous GNNs: Our experiments suggest that distinct oceanic variables have different levels of influence in forecasting the underlying target variables. Thus, we must search for interesting ways to handle heterogeneous graphs, where the nodes may be of different types, such as in [Schlichtkrull *et al.*, 2018]. The heterogeneous setup may allow for different *message functions* to be learned for different types of relations.

Using SOFS as input: SOFS considers other types of information such as local topology; using that as a targetless node

	Speed (m/s)		**Direction (degrees)**	
Scenarios	IoA ↑	RMSE ↓	IoA ↑	RMSE ↓
Fully disconnected	0.719	0.161	0.830	68.89
Same *type* connections	0.706	0.165	0.818	70.25
Fully connected	**0.726**	0.158	**0.842**	65.68

Table 4: Average results for 10 random seeds comparing fully disconnected, same-variable connected and fully connected graphs in the [Current;SSH] predicting [Current] scenario. In bold the best ones.

	Speed (m/s)		**Direction (degrees)**	
	IoA		IoA	
Scenarios	Transformer	LSTM	Transformer	LSTM
Current	**0.734**	0.706	**0.841**	0.818
Current+SSH	**0.762**	0.726	**0.860**	0.842

Table 5: Comparison of results for different time series encoders. In these experiments we used a Transformer, a state-of-the-art sequence model, as our first encoders, and compare its results with the previous ones, where we used an LSTM instead. Given the computational cost of Transformers, we used, for each input experiment, the 5 random seeds that gave the best results using LSTM as encoder, and averaged them. We used the same data as the previous experiments to forecast the speed and direction of the water current at Praticagem. In bold the best ones.

	SSH (m)	
	IoA	
Scenarios	Transformer	LSTM
SSH	0.932	0.940
Current+SSH	0.939	0.939

Table 6: Comparison of results for different time series encoders. Same experimental design as showed in Table 5, except that we used SSH as solo input in one experiment, and we forecast the SSH at Praticagem.

in the DTDG model may improve the performance even further. Recent proposals point in that direction within *Physics-Informed Machine Learning* (PIML) [Willard *et al.*, 2020].

Temporal Graph Networks: The proposals by [Rossi *et al.*, 2020] may also be investigated to encode temporal relationships between nodes, thus eliminating the need for the sampling process to form a DTDG.

Acknowledgements

This work was carried out at the Center for Artificial Intelligence (C4AI-USP), with support by the Sao Paulo Research Foundation (FAPESP) under grant number 2019/07665-4 and by the IBM Corporation. This work is also supported in part by FAPESP under grant number 2020/16746-5, the Brazilian National Council for Scientific and Technological Development (CNPq) under grant numbers 312180/2018-7, 310127/2020-3, 310085/2020-9, the Coordination for the Improvement of Higher Education Personnel (CAPES Finance Code 001), Brazil, and *Itaú Unibanco S.A.* through the *Programa de Bolsas Itaú* (PBI) program of the *Centro de Ciência de Dados* (C^{2}D) of *Escola Politécnica* of USP. The authors also thank the Santos Marine Pilots for providing crucial data for this research.

References

[Alon and Yahav, 2021] Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. In *International Conference on Learning Representations*, 2021. (cit. on p. 7)

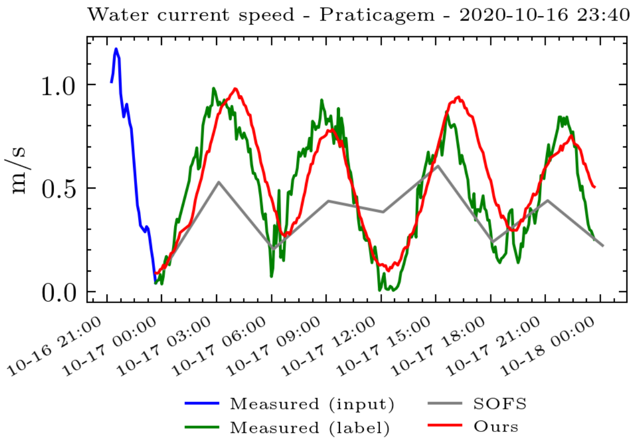

Figure 5: Comparison between our model and SOFS for current speed in meters per second for a sample t_0 in the test dataset. The full input window is 7 days long and is partially omitted for better visualization.

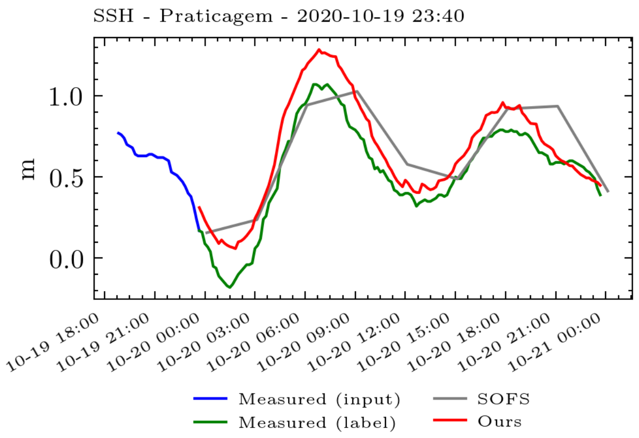

Figure 6: Comparison between our model and SOFS for SSH in meters for a sample t_0 in the test dataset. Values can be negative due to local SSH measurement offset.

[Biewald, 2020] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com. (cit. on p. 6)

[Blumberg and Mellor, 1987] Alan Blumberg and George Mellor. A description of a three-dimensional coastal ocean circulation model, three-dimensional coastal ocean models. *Coastal Estuarine Sci.*, 4, 01 1987. (cit. on p. 1)

[Brody *et al.*, 2021] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021. (cit. on p. 4, 6)

[Cao *et al.*, 2020] Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Congrui Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, et al. Spectral temporal graph neural network for multivariate time-series forecasting. *Advances in Neural Information Processing Systems*, 33:17766–17778, 2020. (cit. on p. 2)

[Costa *et al.*, 2020] Carine GR Costa, José Roberto B Leite, Belmiro M Castro, Alan F Blumberg, Nickitas Georgas, Marcelo Dottori, and Antoni Jordi. An operational forecasting system for physical processes in the santos-sao vicente-bertioga estuarine system, southeast brazil. *Ocean Dynamics*, 70(2):257–271, 2020. (cit. on p. 1, 3)

[Cranmer *et al.*, 2020] Miles Cranmer, Alvaro Sanchez Gonzalez, Peter Battaglia, Rui Xu, Kyle Cranmer, David Spergel, and Shirley Ho. Discovering symbolic models from deep learning with inductive biases. *Advances in Neural Information Processing Systems*, 33:17429–17442, 2020. (cit. on p. 1)

[Fey and Lenssen, 2019] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019. (cit. on p. 6)

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, NV, USA, June 2016. IEEE. (cit. on p. 4)

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. (cit. on p. 1)

[Ibarra-Berastegi *et al.*, 2015] Gabriel Ibarra-Berastegi, Jon Saénz, Ganix Esnaola, Agustin Ezcurra, and Alain Ulazia. Short-term forecasting of the wave energy flux: Analogues, random forests, and physics-based models. *Ocean Engineering*, 104:530–539, 2015. (cit. on p. 1)

[Jumper *et al.*, 2021] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, Aug 2021. (cit. on p. 1)

[Kazemi *et al.*, 2020] Seyed Mehran Kazemi, Rishab Goel, Kshitij Jain, Ivan Kobyzev, Akshay Sethi, Peter Forsyth, and Pascal Poupart. Representation Learning for Dynamic Graphs: A Survey. *Journal of Machine Learning Research*, 21(70):1–73, 2020. (cit. on p. 2, 3)

[Kazi *et al.*, 2022] Anees Kazi, Luca Cosmo, Seyed-Ahmad Ahmadi, Nassir Navab, and Michael Bronstein. Differentiable graph module (dgm) for graph convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. (cit. on p. 7)

[Lira *et al.*, 2022] Hernan Lira, Luis Martí, and Nayat Sanchez-Pi. A graph neural network with spatio-temporal attention for multi-sources time series data: An application to frost forecast. *Sensors*, 22(4), 2022. (cit. on p. 1, 2)

[Moreno *et al.*, 2022] Felipe M. Moreno, Luiz A. Schiaveto Neto, Fabio G. Cozman, Marcelo Dottori, and Eduardo A. Tannuri. Enhancing the forecast of ocean physical variables through physics informed machine learning in the santos estuary, brazil. In *OCEANS 2022 - Chennai*, pages 1–7, 2022. (cit. on p. 2)

[Netto *et al.*, 2020] Caio Netto, Eduardo Tannuri, Denis Mauá, and Fábio Cozman. Prediction of environmental conditions for maritime navigation using a network of sensors: A practical application of graph neural networks. In *Anais do VIII Symposium on Knowledge Discovery, Mining and Learning*, pages 233–240, Porto Alegre, RS, Brasil, 2020. SBC. (cit. on p. 1, 2)

[Paszke *et al.*, 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. (cit. on p. 6)

[PIANC MarCom, 2012] PIANC MarCom. Use of hydro/meteo information for port access and operations. Technical report, WG 117 Report 117, 2012. (cit. on p. 1)

[Raccuglia *et al.*, 2016] Paul Raccuglia, Katherine C. Elbert, Philip D. F. Adler, Casey Falk, Malia B. Wenny, Aurelio Mollo, Matthias Zeller, Sorelle A. Friedler, Joshua Schrier, and Alexander J. Norquist. Machine-learning-assisted materials discovery using failed experiments. *Nature*, 533(7601):73–76, May 2016. (cit. on p. 1)

[Ribeiro *et al.*, 2019] Renan Braga Ribeiro, Alexandra Franciscatto Penteado Sampaio, Matheus Souza Ruiz, José Chambel Leitão, and Paulo Chambel Leitão. *First Approach of a Storm Surge Early Warning System for Santos Region*, pages 135–157. Springer International Publishing, Cham, 2019. (cit. on p. 1)

[Rossi *et al.*, 2020] Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal Graph Networks for Deep Learning on Dynamic Graphs. *arXiv:2006.10637 [cs, stat]*, October 2020. (cit. on p. 8)

[Sanchez-Gonzalez *et al.*, 2020] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International Conference on Machine Learning*, pages 8459–8468. PMLR, 2020. (cit. on p. 1)

[Satorras *et al.*, 2022] Victor Garcia Satorras, Syama Sundar Rangapuram, and Tim Januschowski. Multivariate Time Series Forecasting with Latent Graph Inference, March 2022. (cit. on p. 2)

[Scarselli *et al.*, 2009] F. Scarselli, M. Gori, Ah Chung Tsoi, M. Hagenbuchner, and G. Monfardini. The Graph Neural

Network Model. *IEEE Transactions on Neural Networks*, 20(1):61–80, January 2009. (cit. on p. 1, 2)

[Schlichtkrull *et al.*, 2018] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling Relational Data with Graph Convolutional Networks. In Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna Tordai, and Mehwish Alam, editors, *The Semantic Web*, Lecture Notes in Computer Science, pages 593–607, Cham, 2018. Springer International Publishing. (cit. on p. 7)

[Topping *et al.*, 2021] Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M. Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations*, September 2021. (cit. on p. 7)

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. (cit. on p. 1)

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *arXiv:1710.10903 [cs, stat]*, February 2018. (cit. on p. 4)

[Wei, 2019] William W S Wei. Multivariate Time Series Analysis and Applications. *Wiley*, page 528, 2019. (cit. on p. 2)

[Willard *et al.*, 2020] Jared Willard, Xiaowei Jia, Shaoming Xu, Michael Steinbach, and Vipin Kumar. Integrating scientific knowledge with machine learning for engineering and environmental systems. *arXiv preprint arXiv:2003.04919*, 2020. (cit. on p. 8)

[Willmott, 1981] Cort J. Willmott. On the validation of models. *Physical Geography*, 2(2):184–194, 1981. (cit. on p. 2, 6)

[Xiao *et al.*, 2019] Changjiang Xiao, Nengcheng Chen, Chuli Hu, Ke Wang, Zewei Xu, Yaping Cai, Lei Xu, Zeqiang Chen, and Jianya Gong. A spatiotemporal deep learning model for sea surface temperature field prediction using time-series satellite data. *Environmental Modelling & Software*, 120:104502, October 2019. (cit. on p. 1, 2)

[Xu and Valocchi, 2015] Tianfang Xu and Albert J. Valocchi. Data-driven methods to improve baseflow prediction of a regional groundwater model. *Computers & Geosciences*, 85:124–136, 2015. Statistical learning in geoscience modelling: Novel algorithms and challenging case studies. (cit. on p. 1)

[Zhang *et al.*, 2019] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. Heterogeneous Graph Neural Network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 793–803, New York, NY, USA, July 2019. Association for Computing Machinery. (cit. on p. 7)

[Ziat *et al.*, 2017] Ali Ziat, Edouard Delasalles, Ludovic Denoyer, and Patrick Gallinari. Spatio-Temporal Neural Networks for Space-Time Series Forecasting and Relations Discovery. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 705–714, November 2017. (cit. on p. 2)