Deployment and Management of Time Series Forecasts in Ocean Industry

Fearghal O'Donncha
AI 4 Digital Twins
IBM Research Europe
Dublin, Ireland
feardonn@ie.ibm.com

Albert Akhriev
AI & Quantum
IBM Research Europe
Dublin, Ireland
albert_akhriev@ie.ibm.com

Bradley Eck
AI 4 Digital Twins
IBM Research Europe
Dublin, Ireland
bradley.eck@ie.ibm.com

Meredith Burke
Department of Oceanography
Dalhousie University
Halifax, Canada
meredith.burke@dal.ca

Ramon Filgueira
Marine Affairs Program
Dalhousie University
Halifax, Canada
ramon.filgueira@dal.ca

Jon Grant
Department of Oceanography
Dalhousie University
Halifax, Canada
jon.grant@dal.ca

***Abstract*—Machine learning has not achieved the same degree of success in environmental applications as in other industries. Challenges around data sparsity, quality, and consistency have limited the impact of deep neural network approaches and restricted the focus to research applications. An alternative approach – that is more amenable to the characteristics of data coming from disparate IoT devices deployed at different times and locations in the ocean – is to develop many lightweight models that can be readily scaled up or down based on the number of devices available at any time. This paper presents a serverless framework that naturally marries a single IoT sensor device with a forecasting model. Aspects related to data ingestion, data processing, model training and deployment are described. The framework is applied to a fish farm site in Atlantic Canada.**

***Index Terms*—time series, imputation, ocean, environment, model management**

I. Introduction

Accurate forecasts of environmental variables are critical to many ocean industries such as aquaculture, shipping, and coastal management. Traditionally these forecasts rely on physics-based models that resolve ocean dynamics on a grid. These models are the linchpin of operational forecasting products but suffer from two shortcomings:

1) Running physics-based models at the high resolutions demanded by coastal forecasting applications can be prohibitively expensive; and,
2) the models require highly-skilled users to deploy them and maintain accurate forecasts.

Recent research at the interface of numerical modelling and High Performance Computing (HPC) examines the first point. Efforts include new programming environments to facilitate increased computational scaling [1] or innovative systems that combine extreme-scale computing with complex numerical models [2]. The former, however, are an emerging technology while the latter relies on large scientific collaboration and dissemination activities to improve understanding of ocean processes and simplify model tuning and deployment.

Instead of resolving the physics of the system with a set of partial differential equations (PDEs), machine learning offers an alternative that avoids many of these obstacles. Recent years has seen numerous applications of machine learning to forecasting ocean variables such as wave height [3], temperature [4], and primary productivity [5]. The concept of applying machine learning models to these datasets to capture the system dynamics and forecast future conditions is well established.

Two important considerations for operational forecasting products are the spatial resolution required to resolve system dynamics, and the forecast horizon that is necessary to enable decision support and response. Recent work suggests decision support systems will require an order of magnitude improvement in spatial resolution [6], and improvement in forecast confidence up to 14 days-ahead.

While advacnes in IoT, remote sensing, and numerical model products, have enriched data coming from the ocean, a key challenge remains related to developing machine learning solutions that fuses those data into a robust predictive forecasting system amenable to the different spatiotemporal scales necessary for industry operations. The most prominent bodies of work in that space involves combination of CNN and RNN for spatiotemporal forecasting (e.g. [7]), or hybrid approaches that combine physics models with machine learning (e.g. [8]). While these have demonstrated promising early results, the complexity associated with model setup, configuration, and interrogation makes flexible deployment challenging, and in particular is generally not easily transferred to other locations. Further since data from satellite or ocean model data are generally used to train the spatiotemporal models, the resolution is restricted to that of the original dataset.

An alternative approach is to consider relatively simple forecasting models and use many of them to resolve conditions across the region of interest. In many marine industries such as oil platforms, fish farms, and ports, there are 10s–1000s

of sensors sampling conditions on the marine environment. In a sense it reduces to a choice between a small number of complex models that aims to address many different spatial and temporal scales simultaneously, or a large number of simpler models that can respond to the huge spatial scales by instantiating different models for each location.

We target the latter situation with a system that considers the data ingestion, pre-processing and cleansing, model setup and training, and the management of these deployments. The framework is demonstrated on data from a real world fish farm. Contributions of the paper are as follows:

- We present a scalable framework to train, monitor, and deploy machine learning models for ocean datasets.
- We describe an automated data cleaning and pre-processing routine that adapts to the specific needs of ocean datasets.
- We assessed performance of the modelling framework on data from a chaotic, real-world environment where data is often sparse.
- Finally, we discuss the strengths and weaknesses of serverless computing for ocean industry applications.

The rest of the paper is structured as follows. In the next section, we discuss the current state of the art. Section III presents the system architecture, describes the data collected, and the experimental setup. This is followed by the results and analysis, while Section V presents conclusions from this research and future directions.

II. RELATED ART

Environmental forecasting applications have traditionally relied on physics-based approaches and an extensive literature exists related to hydro-environmental modeling and applications to numerous case studies. Complex software codes exist (generally written in FORTRAN), that resolve physical and biogeochemical ocean processes across both structured and unstructured grids. Prominent examples in this space include ROMS [9], DELFT3D [10], and EFDC [11].

Due to the heavy computational overhead of physical models, there is an increasing trend to apply data-driven DL or ML methods to model physical phenomena [12], [13]. There is an extensive body of literature related to ML based forecasting methods in environmental applications. These include a variety of shallow (e.g., decision trees, random forests, support-vector machines) and deep neural network (DNN) approaches. [4] compares the performance of different statistical, ML, and DNN methods to forecast sea surface temperature (SST). Results indicate that depending on availability of sufficient training data volumes, simpler methods achieve comparable accuracy to DNNs. Many studies have proposed frameworks to represent the spatiotemporal properties of geophsysical systems. The most widely used framework combines convolutional neural networks (CNN) with LSTM to represent both the spatial (CNN) and temporal (LSTM) dependencies within the data. This approach has been applied to a variety of geoscientific tasks such as precipitation nowcasting from rainfall radar maps [14] and forecasting sea surface temperature from satellite-derived observations [7].

A number of studies have investigated data-driven approaches to provide computationally cheaper surrogate models, applied to wave forecasting [15], air pollution [16], viscoelastic earthquake simulation [17], and water-quality investigation [18]. There are few (if any) machine learning based forecasting methods used in operational products in oceanography. The challenges of model trust and robustness, data ingestion and curation, and model management and monitoring limits the role that machine learning currently has in operational forecasting applications. All large scale operational products rely on physics-based models (e.g., [19], [20]) and additional work is necessary to gain acceptance of ML methods by environmental stakeholders.

III. METHODS

This paper presents a time series data preprocessing, imputation, and forecasting approach applied to ocean datasets. The entire modelling approach is deployed on a flexible serverless computing architecture. We describe the data ingestion, processing, model training/scoring, and model scheduling and deployment. The framework is applied to forecast ocean temperature and dissolved oxygen on a fish farm site in Atlantic Canada. The data are characterised by varying degrees of noise and anomalies as well as periods of missing values. We present a pragmatic preprocessing framework that filters outliers and imputes missing values (where feasible). In the remainder of this section, we describe the study site and the data being collected, describe the system architecture, describe data preprocessing and cleansing implementation, and explain the machine learning models and deployments.

A. The data

Temperature and dissolved oxygen (DO) observations were collected within a farm of sea cages cultivating Atlantic salmon. The study site, located in Saddle Island, Nova Scotia (coordinates: $44^\circ\ 30.225'$ N $64^\circ\ 2.923'$ W), is a commercially operated Atlantic salmon farm containing six cages, each measuring $150\,\mathrm{m}$ circumference and containing about 60,000 fish. Each cage was equipped with at least two RealTime Aquaculture [21] probes deployed at different depths in the water column and sampling at three minute intervals. Figure 1 gives a schematic of a typical cage configuration with sensors deployed to monitor horizontal and vertical gradients. Nineteen sensors were deployed across the farm each measuring temperature and dissolved oxygen. Battery capacity allowed four month deployments after which sensor was removed, cleaned, and recalibrated prior to another deployment (generally at another location in the the farm). Hence, data for machine learning models covered a four month period.

Temperature and DO are important variables to guide farm management and early warning decisions. Water temperature has a major impact on fish metabolism and therefore growth rates [22], while DO has obvious implications for respiration, health, and mortalities [23]. Further, DO can fluctuate rapidly

under the influence of external drivers and internal cage dynamics.

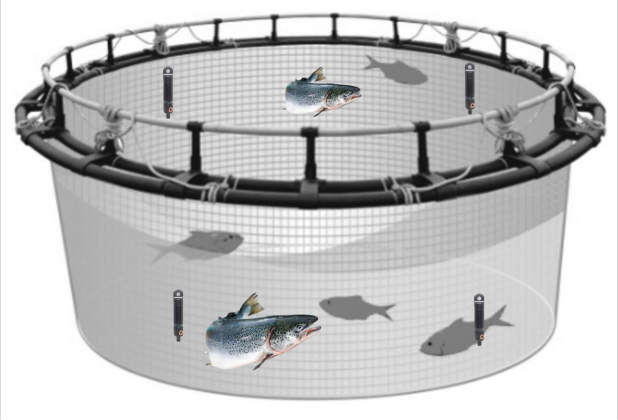

Fig. 1. Schematic of sensor configuration within a cage. Sensors are deployed at two levels in the vertical to measure vertical gradients and in specific cages four sensors were deployed to also measure horizontal gradients.

Sensor data were complemented with pertinent external data. Weather data were extracted from IBM Weather Operations Center available through their public API [24], while global ocean model data were downloaded from Copernicus Marine Service model repository [20]. Copernicus data is available at a $\approx 10\,\mathrm{km}$ horizontal resolution at hourly intervals and can be freely downloaded. Selected weather variables were air temperature, solar radiation, and wind speeds while the data extracted from the Copernicus marine service were surface temperature and dissolved oxygen as well as water elevation.

B. Castor

Castor is a cloud-native system for the management of time series data and models that makes extensive use of serverless technology, in particular IBM Cloud Functions ([25] [26]). Castor was previously applied to renewable energy forecasting and distribution grid optimization ([27], [28]). The present aquaculture application shares several characteristics that motivate the usage of Castor. Noisy and incomplete sensor data arrives from different sources without a fixed schedule. Production forecasts of a fixed horizon are made on a rolling basis, resulting in multiple forecasts for the same future time. Comparisons of several combinations of model algorithm and feature set are of interest. Thus, this work uses an instance of the Castor system to support several forecasting tasks:

- ingestion and storage of sensor data and weather forecasts,
- data pre-processing and cleaning,
- training and scoring of machine learning models, and,
- model management and performance evaluation.

Castor uses semantic concepts of *entity*, and *signal* to describe time series data and manage model deployments. In this application, entities corresponded to locations such as individual fish cages, and signals corresponded to water quality parameters such as dissolved oxygen or water temperature. Each time series has a unique identifier (tsid) which is then associated with an entity-signal pair. Thus, sensor data uploaded to the system may use either the tsid or semantic coordinates. A Castor instance is deployed for every farm and at this pilot site in Canada, it contains 1381 different time series, representing 146 entities and 23 signals.

Castor separates the code which implements a model from the configuration information needed to apply that model for a specific time series. Model configuration includes the semantic coordinates of the target time series, which automatically also provides the historical values and related time series. In this way, the semantics of entity and signal can be used for parametric and programmatic model deployment. As further described below, we implemented four different model classes:

1) Generalized additive models (GAM)
2) XGBoost
3) Random Forest (RF)
4) Multi-layer Perceptron (MLP)

Each class implemented Castor's model interface, allowing the system to manage model training and scoring and to evaluate performance.

C. Data Pre-processing and Cleansing

A key objective of our work is to enable forecasting with minimal human interaction. This requires the ability to request data based on a given context, automatically process and cleanse the data, and invoke an appropriate machine learning pipeline for model training and forecasting (or scoring in machine learning parlance). Data preparation is a core component of an applied data scientist's role with an oft-repeated trope that 80% of their time is spent cleaning data. A core part of data cleansing is handling missing data – this is especially true in time series forecasting where we desire complete coverage over the time period.

The data imputation approaches considered the different characteristics of data quality issues in aquaculture. The reality of operating in a chaotic environment result in multiple classes of data quality issues: failures in power and connectivity results in data gaps of hours to days; sensor fouling and damage impedes data quality and often demands bias and error corrections; while the multiple temporal and spatial scales of ocean processes often require robust post-processing and noise removal strategies. These issues are amplified by the fact that collecting sufficient data is often difficult and one is rarely able to replace significant portions of data due to data quality issues – one instead desires to repair the data.

While a fundamental and long-studied problem, it is not straightforward to devise a formal protocol and to categorise the "best" imputation system. The difficulty primarily stems from the fact that one requires a representative model of the original time series signal to enable reconstruction – of course this is generally not available in practise. Instead we wish to offer a variety of imputation approaches and allow the system to chose the best one based on changes or improvement in model skill. In practical terms, we are not interested in precisely replicating the true signal (this is not known and

impossible), but rather to impute in a manner that we replicate the statistical fidelity of the series.

We considered a variety of standard imputation approaches, as well as two algorithms we developed that are more amenable to time series data with distinct periodic signals. The standard imputers are well known and included: a simple imputer from Scikit-Learn Python package that substitutes a median value instead of a missing one, as well as linear, quadratic, cubic and polynomial interpolation. The two new imputation choices we offered were:

- A fast principal component projection imputer (FastPCPImputer) and a
- Low rank imputer (LRImputer)

Instead of interpolation at missing values, these imputers try to substitute the missing portion of a signal from other, uncorrupted parts of the same sequence. This is done via low-rank matrix approximation. Namely, we put the signal into a square matrix progressively row by row (possibly padding by NaNs at the end of last row). We call this matrix a data matrix. The rank of low-rank approximation is chosen as a square root of matrix size in either dimension.

The first imputer (FastPCPImputer) elaborates the idea described in the paper [29]. The fundamental rationale of the approach considers recovering a low-rank matrix (the principal components) from a high-dimensional data matrix despite the presence of sparse errors [30]. This has natural applicability for time series data where one may expect repeated patterns at different frequencies such as day (solar radiation), week (traffic volumes) or year (annual or seasonal cycles). We adopt a brute-force grid search approach to select value of regularization parameter λ to achieve best match at uncorrupted entries of the time series.

The second imputer (LRImputer) extends our paper [31], where low-rank approximation was achieved by decomposition of the data matrix into a product of two low-rank ones $L \cdot R$ (hence the name LR). We use robust loss function with a regularizer that promotes smoothness of the imputation result.

Both imputers are relatively fast. The Python implementation usually takes less than 10 seconds (often 2-3 seconds) on single-core CPU on time series with 10,000+ observations. This approach allows us to standardise data processing for machine learning models (since missing values are handled in an equivalent manner for every sensor), while different choices can be readily explored.

D. Model Setup and Training

While any model can conceptually be added to the model store, there are practical limitation in terms of model size, hence, we wish to limit to shallower choices (the data appetite of DNNs is also an issue in a challenging data collection environment such as oceans). For our applications we considered four different algorithms: Generalised Additive Models (GAM), Random Forest (RF) XGBoost, and Multi-Layer Perceptron (MLP). Extensive details on these algorithms are provided in many statistical and machine learning textbooks (e.g., [32], [33]) and considerations for their application to ocean datasets are described in [4].

For each model implementation, pertinent data was requested from the server, resampled to hourly intervals, preprocessed as described above to remove outliers and impute missing values, converted into appropriate time-aligned matrices for the model implementation, and used to train the machine learning model (or update a previously trained model). The train test split is an important consideration for time series data. Randomly shuffling the data can artificially inflate model performance, while a simple linear split (first 80% training and the remainder test) can lead to model drift. Our approach relies on a recurrent forecasting method where the model is trained on all available data up to a point and makes a prediction for the subsequent time point. The process then steps forward one time step and repeats the training and forecasting.

Hyperparameter optimisation was done using a greedy grid search approach that searched over a user-defined range of hyperparameters with an embarrassingly parallel approach. Pywren [34] managed the search, which then returned the parameters that minimised error for the training dataset, using a MapReduce operation. More details on hyperparameter optimisation and testing is provided in [4].

The trained model was then stored in a model management database and deployed in forecasting mode against a verification dataset to evaluate performance metrics and goodness-of-fit. Trained models are tagged and stored allowing flexibility to select different model iterations (e.g., roll back to a model that we know gives a certain level of performance while also testing new model implementations). At all stages, a variety of preprocessing and forecasting algorithms are available to the system to enhance forecasting skill and allow us to address many different variables and conditions. This gives us the added benefit that we can readily train and store different models with different combinations of features to allow flexible interrogation and demonstration of the implications of different forcing parameters.

IV. RESULTS AND DISCUSSION

The key objective of our work was to simplify the task of configuring, training, and deploying machine learning models, with a focus on environmental applications. The situation at hand consisted of multiple IoT sensors returning observations from different locations within the cage. Effective farm management required 5–10 day-ahead forecasts, and a robust model monitoring framework to identify deviations from ambient condition. Meeting these needs required a framework for data ingestion, data processing, model training, model deployment, and monitoring.

As described in Section III-B, sensor data is stored and managed via a context layer that provided the user with a logical access point to the data based on entity (generally sensors identified based on cage number and whether they are near surface or near bottom), and signal (water temperature or dissolved oxygen in this case).

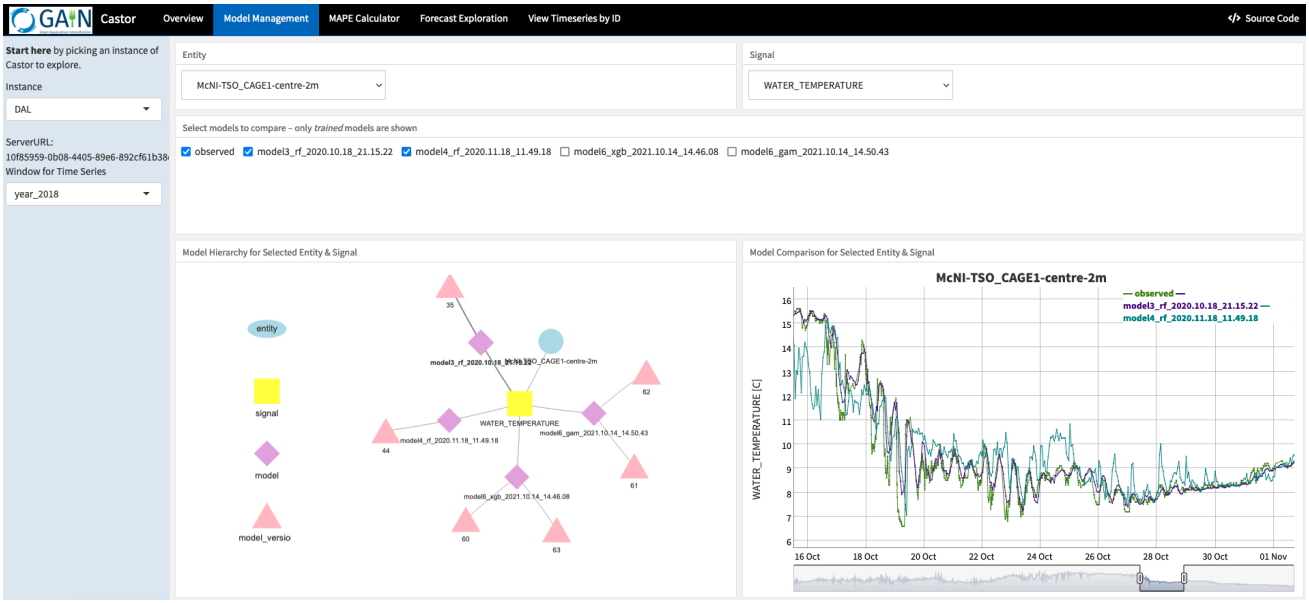

Fig. 2. Example dashboard showing Castor's model hierarchy and comparing performance of two random forest models for predicting water temperature in a cage at a depth of 2 m. The legend on the bottom left denotes an entity such as sensor in cage 1 at 2 m depth (blue ellipse), signal such as water temperature or DO (yellow square), model such as GAM, XGBoost, or Random Forest (purple diamond) and a tagged version or id for the model denoting different deployments (pink triangle)

A dashboard for the Castor system allows interactive navigation of data and forecasts. The instance/site of interest (DAL indicating Canadian fish farm) and the time window are selected from the left panel. The entity (a temperature, salinity, and oxygen (TSO) sensor from center of cage 1 at 2m depth) and signal (water temperature) are selected in the top row. The network diagram at the bottom left shows Castor's model hierarchy: how models and model versions relate to the target entity and signal. Perhaps of most interest is the time series plot on the bottom right where the forecast values from two different forest models are compared with observations.

Deployed model classes were labelled by model number (1–6) and algorithm (GAM, RF, XGBoost, and MLP). The model number refers to different combinations of features that draw from [4]: 1) autoregressive features (i.e. past values of the response variable), 2) atmospheric variables (air temperature, pressure, and wind speeds), and 3) Copernicus ocean model variables (physical model forcing at global scale ($\approx$10 km horizontal resolution). In this case, Figure 2 compares model results from *model3_rf* against *model4_rf*: *model3* is forced by autoregressive features only, while *model4* is informed by features from Copernicus ocean model only (in effect, *model4* attempts to downscale from the global scale (O(10 km) to the cage level (O(0.05 km). In our case *model3* reports better model skill, but *model4* avoids some of the limitations of autoregressive models and allows us make forecasts during periods we don't have sensor data. The eventual choice of models is driven by the specifics of a given farm. Naturally different combinations of these features can be readily combined (specified in configuration file) and the model store allows easy comparison and selection of desired choice. The scalability of our serverless approach places little practical limitations on the number of models that can be explored.

V. CONCLUSION

We present a scalable data ingestion, processing, and forecasting framework that is amenable to the requirements of forecasting in marine industry application. Challenges around data sparsity are addressed by fusing data from external sources, while shorter gaps in data are handled by a choice of imputation algorithms. The objective is a pragmatic approach that simplifies the repetitive tasks and allows the data scientist or domain expert to readily explore different model choice and make their selection based on given site characteristics and requirements.

The semi-automated approach greatly increases the number of models that can be easily deployed and managed by the user (in practical terms, the limitation is not computational but rather restricted by the user bandwidth). Metrics on model performance (RMSE and MAPE) are computed and provided to the user as a base sanity check on model performance. Future research is focused on developing more robust monitoring frameworks that are sensitive to data fluctuations or sensor drift that are not readily detected by more naive approaches like RMSE. In addition, increased penetration of machine learning in the industry is contingent on model trust and robustness. The fact that all models are stored provides a valuable resource for explainability solutions that allows the user to readily explore how different set of features contribute towards model forecast.

ACKNOWLEDGMENT

This project has received funding from the European Unions Horizon 2020 research and innovation programme as part of the RIA GAIN project under grant agreement No. 773330.

REFERENCES

[1] F. O'Donncha, R. Iakymchuk, A. Akhriev, P. Gschwandtner, P. Thoman, T. Heller, X. Aguilar, K. Dichev, C. Gillan, S. Markidis *et al.*, "AllScale toolchain pilot applications: PDE based solvers using a parallel development environment," *Computer Physics Communications*, vol. 251, p. 107089, 2020.

[2] P. Voosen, "Europe is building a 'digital twin' of Earth to revolutionize climate forecasts," 2020. [Online]. Available: https://www.sciencemag.org/news/2020/10/europe-building-digital-twin-earth-revolutionize-climate-forecasts

[3] F. O'Donncha, Y. Zhang, B. Chen, and S. C. James, "Ensemble model aggregation using a computationally lightweight machine-learning model to forecast ocean waves," *Journal of Marine Systems*, vol. 199, p. 103206, nov 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924796319300752

[4] S. Wolff, F. O'Donncha, and B. Chen, "Statistical and machine learning ensemble modelling to forecast sea surface temperature," *Journal of Marine Systems*, vol. 208, p. 103347, 2020.

[5] W. Tian, Z. Liao, and J. Zhang, "An optimization of artificial neural network model for predicting chlorophyll dynamics," *Ecological Modelling*, vol. 364, pp. 42–52, 2017.

[6] Mercator Ocean International, "Copernicus marine service workshop and training for the aquaculture sector," 2019. [Online]. Available: https://marine.copernicus.eu/wp-content/uploads/2019/11/Workshop-CMEMS4AQUACULTURE-Athens-SEP2019.pdf

[7] Y. Yang, J. Dong, X. Sun, E. Lima, Q. Mu, and X. Wang, "A cfcc-lstm model for sea surface temperature prediction," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 2, pp. 207–211, 2017.

[8] A. Wikner, J. Pathak, B. Hunt, M. Girvan, T. Arcomano, I. Szunyogh, A. Pomerance, and E. Ott, "Combining machine learning with knowledge-based modeling for scalable forecasting and subgrid-scale closure of large, complex, spatiotemporal systems," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 30, no. 5, p. 053111, 2020.

[9] A. F. Shchepetkin and J. C. McWilliams, "The regional oceanic modeling system (roms): a split-explicit, free-surface, topography-following-coordinate oceanic model," *Ocean modelling*, vol. 9, no. 4, pp. 347–404, 2005.

[10] J. Roelvink and G. Van Banning, "Design and development of delft3d and application to coastal morphodynamics," *Oceanographic Literature Review*, vol. 11, no. 42, p. 925, 1995.

[11] Z.-G. Ji, *Hydrodynamics and water quality: modeling rivers, lakes, and estuaries*. John Wiley & Sons, 2017.

[12] E. de Bezenac, A. Pajot, and P. Gallinari, "Deep Learning for Physical Processes: Incorporating Prior Scientific Knowledge," *arxiv preprint arXiv:1711.07970*, nov 2017. [Online]. Available: http://arxiv.org/abs/1711.07970

[13] S. Wiewel, M. Becher, and N. Thuerey, "Latent-space Physics: Towards Learning the Temporal Evolution of Fluid Flow," *arxiv preprint arXiv:1802.10123*, feb 2018. [Online]. Available: http://arxiv.org/abs/1802.10123

[14] S. Xingjian, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, 2015, pp. 802–810. [Online]. Available: http://papers.nips.cc/paper/5955-convolutional-lstm-network-a-machine-learning-approach-for-precipitation-nowcasting

[15] S. James, Y. Zhang, and F. O'Donncha, "A machine learning framework to forecast wave conditions," *Coastal Engineering*, vol. 137, 2018.

[16] P. Hähnel, J. Mareček, J. Monteil, and F. O'Donncha, "Using deep learning to extend the range of air pollution monitoring and forecasting," *Journal of Computational Physics*, vol. 408, p. 109278, 2020.

[17] P. M. R. DeVries, T. B. Thompson, and B. J. Meade, "Enabling large-scale viscoelastic calculations via neural network acceleration," *Geophysical Research Letters*, vol. 44, no. 6, pp. 2662–2669, mar 2017. [Online]. Available: http://doi.wiley.com/10.1002/2017GL072716

[18] E. Arandia, F. O'Donncha, S. McKenna, S. Tirupathi, and E. Ragnoli, "Surrogate modeling and risk-based analysis for solute transport simulations," *Stochastic Environmental Research and Risk Assessment*, pp. 1–15, may 2018. [Online]. Available: http://link.springer.com/10.1007/s00477-018-1549-6

[19] Y. Chao, J. D. Farrara, H. Zhang, K. J. Armenta, L. Centurioni, F. Chavez, J. B. Girton, D. Rudnick, and R. K. Walter, "Development, implementation, and validation of a california coastal ocean modeling, data assimilation, and forecasting system," *Deep Sea Research Part II: Topical Studies in Oceanography*, vol. 151, pp. 49–63, 2018.

[20] EU Copernicus, "Copernicus - Marine environment monitoring service," 2020. [Online]. Available: https://marine.copernicus.eu/

[21] Real Time Aquaculture. (2020) Real time aquaculture. [Online]. Available: https://www.innovasea.com/aquaculture-intelligence/environmental-monitoring/wireless-sensors/

[22] N. A. Shettigar, B. Bhattacharya, L. Mészáros, A. Spinosa, and G. El Serafy, "3d ensemble simulation of seawater temperature–an application for aquaculture operations," *Frontiers in Marine Science*, vol. 7, p. 1020, 2020.

[23] M. Burke, J. Grant, R. Filgueira, and T. Stone, "Oceanographic processes control dissolved oxygen variability at a commercial atlantic salmon farm: Application of a real-time sensor network," *Aquaculture*, p. 736143, 2020.

[24] IBM, "Weather Company Data API," The Weather Company, Tech. Rep., 2018. [Online]. Available: https://www.ibm.com/us-en/marketplace/weather-company-data-packages/details

[25] B. Chen, B. Eck, F. Fusco, R. Gormally, M. Purcell, M. Sinn, and S. Tirupathi, "Castor: Contextual iot time series data and model management at scale," in *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2018, pp. 1487–1492.

[26] B. Eck, F. Fusco, R. Gormally, M. Purcell, and S. Tirupathi, "Scalable deployment of ai time-series models for iot," in *AI for Internet of Things (AI4IoT) Workshop at 28th International Joint Conference on Artificial Intelligence*, 2019.

[27] ——, "Ai modelling and time-series forecasting systems for trading energy flexibility in distribution grids," in *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, ser. e-Energy '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 381–382. [Online]. Available: https://doi.org/10.1145/3307772.3330158

[28] F. Fusco, B. Eck, R. Gormally, M. Purcell, and S. Tirupathi, "Knowledge- and data-driven services for energy systems using graph neural networks," in *2020 IEEE International Conference on Big Data (Big Data)*, 2020, pp. 1301–1308.

[29] P. Rodriguez and B. Wohlberg, "Fast principal component pursuit via alternating minimization," in *2013 IEEE International Conference on Image Processing*, 2013, pp. 69–73.

[30] Z. Zhou, X. Li, J. Wright, E. Candes, and Y. Ma, "Stable principal component pursuit," in *2010 IEEE international symposium on information theory*. IEEE, 2010, pp. 1518–1522.

[31] A. Akhriev, J. Marecek, and A. Simonetto, "Pursuit of low-rank models of time-varying matrices robust to sparse and measurement noise," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 3171–3178, Apr. 2020. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/5714

[32] J. H. Friedman, *The elements of statistical learning: Data mining, inference, and prediction*. springer open, 2017.

[33] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[34] E. Jonas, Q. Pu, S. Venkataraman, I. Stoica, and B. Recht, "Occupy the cloud: Distributed computing for the 99%," in *Proceedings of the 2017 Symposium on Cloud Computing*, 2017, pp. 445–451.