

A Deep Learning Model for Improved Wind and Consequent Wave Forecasts[✉]

YUVAL YEVNIN^a AND YARON TOLEDO^a

^a *School of Mechanical Engineering, Tel-Aviv University, Tel-Aviv, Israel*

(Manuscript received 24 November 2021, in final form 6 July 2022)

ABSTRACT: The paper presents a combined numerical–deep learning (DL) approach for improving wind and wave forecasting. First, a DL model is trained to improve wind velocity forecasts by using past reanalysis data. The improved wind forecasts are used as forcing in a numerical wave forecasting model. This novel approach, used to combine physics-based and data-driven models, was tested over the Mediterranean. The correction to the wind forecast resulted in ~10% RMSE improvement in both wind velocity and wave height over reanalysis data. This significant improvement is even more substantial at the Aegean Sea when Etesian winds are dominant, improving wave height forecasts by over 35%. The additional computational costs of the DL model are negligible compared to the costs of either the atmospheric or wave numerical model by itself. This work has the potential to greatly improve the wind and wave forecasting models used nowadays by tailoring models to localized seasonal conditions, at negligible additional computational costs.

SIGNIFICANCE STATEMENT: Wind and wave forecasting models solve a set of complicated physical equations. Improving forecasting accuracy is usually achieved by using a higher-resolution, empirical coefficients calibration or better physical formulations. However, measurements are rarely used directly to achieve better forecasts, as their assimilation can prove difficult. The presented work bridges this gap by using a data-driven deep learning model to improve wind forecasting accuracy, and the resulting wave forecasting. Testing over the Mediterranean Sea resulted in ~10% RMSE improvement. Inspecting the Aegean Sea when the Etesian wind is dominant shows an outstanding 35% improvement. This approach has the potential to improve the operational atmospheric and wave forecasting models used nowadays by tailoring models to localized seasonal conditions, at negligible computational costs.

KEYWORDS: Forecasting; Forecasting techniques; Hindcasts; Numerical weather prediction/forecasting; Operational forecasting; Statistical forecasting; Neural networks; Numerical analysis/modeling; Ocean models; Optimization; Postprocessing; Reanalysis data; Artificial intelligence; Deep learning; Machine learning; Regression

1. Introduction

Wind velocity accuracy has been established as one of the most significant factors in achieving an accurate ocean waves forecast (Bidlot et al. 2002). For this reason, operational wave forecasting models aim to use the most accurate wind fields available, with a high resolution in both space and time. The models producing these wind fields are highly computationally expensive, simulating many layers in the atmosphere. These atmospheric models are assimilated with data acquired by measurement instruments to create reanalysis results. The reanalysis data are used to assess, study, and improve the wind forecast ability (Hersbach et al. 2020).

Wave forecasting models, such as WAM (Hasselmann et al. 1988) or WAVEWATCH III (Tolman 1991), use wind forecast as an input. Although the driving force for wave generation is surface wind, the parameter used by most models is wind velocity at 10 m above the sea surface (U10), as this property is easier to measure and predict. This means only a

single property at a single level of the atmospheric model actually affects the wave model. A semiempirical source term is used by wave models to convert U10 to wave action forcing. Optimizing atmospheric models is highly complex, both in terms of computational costs and in terms of improved physical equations accounting for multiple flow parameters. Thus, a model which can optimize U10 independently, decoupled from the physics-based model and with low computational costs, is very desirable.

In the last few years, deep learning (DL) models have been used in multiple fields to solve complex, highly nonlinear problems (Wang et al. 2019; Brunton et al. 2020). These DL models are data-driven, meaning they generally do not possess any prior physical knowledge, but are instead trained to predict a given “ground truth” data. After the model is trained using a training dataset to achieve good performance, it is verified over an independent test dataset. The training process usually requires significant computational resources, though it is still relatively small compared to numerical models. Afterward, the resulting model can be used to produce accurate predictions at very minimal computational cost for specific problems.

DL methods are highly relevant for geophysical problems (Reichstein et al. 2019) and can be used for various functions. First, DL is used for making forecasts of various atmospheric variables directly, which are data-driven and independent of

[✉] Supplemental information related to this paper is available at the Journals Online website: <https://doi.org/10.1175/JPO-D-21-0280.s1>.

Corresponding authors: Yuval Yevnin, yuval.yevnin@gmail.com; Yaron Toledo, toledo@tau.ac.il

physical equations and numerical models (Weyn et al. 2020; Rasp and Thuerey 2020; Arcomano et al. 2020). Second, these methods are used in hybrid numerical–DL models, where the DL model usually replaces some functions or parameterization of the numerical model in order to increase computational efficiency (Schneider et al. 2017; Gentine et al. 2018; Brenowitz and Bretherton 2019; Wikner et al. 2020). Finally, machine learning (ML) and DL methods are used for post-processing and measurement assimilation (Vannitsem et al. 2020; Haupt et al. 2021). These usually use an ensemble of atmospheric models with different initial conditions as an input to an ML model based on random forest or a fully connected neural network (NN) (Zjavka 2015; Rasp and Lerch 2018), while recently some work has been done using convolutional NN (Grönquist et al. 2021; Veldkamp et al. 2020).

The presented paper uses a DL model with U10 wind velocity forecasts as the input and predicts the U10 reanalysis data, considered as ground truth. This is a form of postprocessing and is intended to improve wind prediction used as an input to a numerical wave model. Unlike previous works, the current model focuses on using advanced DL architecture to improve forecasts using only the predicted variable as input (e.g., only wind velocity). This allows the DL model to be used in wave forecasting as a wind preprocess source term as the wind input, which has the strongest effect on wave modeling. Improving the wind input is a common way of improving the quality of wave forecasting. This is done either by improving the equation of the physical interaction source term or by adjusting the wind values directly to better fit measurements (Ardhuin et al. 2007). To the best of our knowledge, this is the first attempt to create such an integrated numerical–deep learning process to improve wind forecasting that can be used to improve operational wave forecasting directly.

2. Model database: ECMWF wind velocity

The datasets used in this paper are ECMWF ERA5 reanalysis (REAN) and the forecasts (FC) that were used as initial model for the reanalysis (Hersbach et al. 2020). ERA5 was chosen as it was found to be a very accurate reanalysis for surface winds (Ramon et al. 2019). The parameters of wind velocity in the zonal and meridional directions at 10-m height (u_{10} , v_{10}) were used, where FC data were used as the deep learning models (DLM) input and the REAN as the ground truth.

The FC is initiated from a wind analysis every 12 h at 0600 and 1800 UTC, and consists of 18 hourly steps. This means there is an overlap between consecutive forecasts. In this work the time steps 7–18 were used in the training and evaluation, as these were furthest from the initial analysis and had the largest errors. The REAN data are an hourly high-resolution model incorporated with measurements.

The spatial grid chosen was of the Mediterranean region, with latitude between 30.2° and 45.7°N and steps of 0.5°, and longitude between 2.1°W and 36.0°E and steps of 0.3°. This results in a power of two grid dimensions 32 × 128, making it efficient for processing with a DLM.

3. Recurrent-convolutional model

In Roitenberg and Wolf (2019) a general DLM architecture for spatiotemporal forecasting problems was introduced and tested for public transportation demand. This model was used as a base for a new DLM by removing the encoder and making several adjustments to the decoder part (Fig. 1). The new DLM begins with an input sequence of FC instances. Next is an encoder comprised of convolutional layers with gradually increasing width and dilation. Increasing the width allows each layer to capture more information, while larger dilation allows a wider receptive field, taking into account the effects of further spatial information. Using dilation instead of more traditional approaches of strided convolution or pooling layers keeps the original input dimensions and thus prevents spatial information loss (Yu and Koltun 2015).

Following the encoder, convolutional gated recurrent unit (CGRU) (Ballas et al. 2015) layers were used. These layers combine the ability of the GRU layer (Chung et al. 2014) to learn temporal connections with the convolutional layer capability of spatial modeling. This is done by replacing the matrix multiplication of a GRU with a convolution, and the parameter matrices and vectors with smaller kernels. The CGRU is governed by a set of equations:

$$\begin{aligned} \mathbf{z}_t &= \sigma(\mathbf{W}_z \otimes \mathbf{X}_t + \mathbf{U}_z \otimes \mathbf{h}_{t-1}) \\ \mathbf{r}_t &= \sigma(\mathbf{W}_r \otimes \mathbf{X}_t + \mathbf{U}_r \otimes \mathbf{h}_{t-1}) \\ \bar{\mathbf{h}}_t &= \tanh[\mathbf{W} \otimes \mathbf{X}_t + \mathbf{U} \otimes (\mathbf{r}_t \odot \mathbf{h}_{t-1})] \\ \mathbf{h}_t &= (1 - \mathbf{z}_t)\mathbf{h}_{t-1} + \mathbf{z}_t\bar{\mathbf{h}}_t \end{aligned} \quad (1)$$

where \otimes is the convolution operation; \odot is the Hadamard product; σ is the sigmoid nonlinearity function; and \mathbf{W} , \mathbf{W}_z , \mathbf{W}_r , \mathbf{U} , \mathbf{U}_z , \mathbf{U}_r are the convolutional kernels that can be trained. Parameter \mathbf{h}_t is called the hidden state, a tensor that is both the cell's output at time t and is used as an input to the next time the cell is used, in a recurrent operation. Parameter \mathbf{r}_t is called the reset gate, which controls what information is “remembered” from the previous hidden state \mathbf{h}_{t-1} to the current cell's candidate hidden state $\bar{\mathbf{h}}_t$. Parameter \mathbf{z}_t is called the update gate, and it sets a relative importance between the previous hidden state and current candidate hidden state.

Each instance of the input sequence is introduced separately to the encoder and to the following CGRU, and the last output of the CGRU is concatenated with the last input instance into it. This forms a skip connection over the CGRU, allowing both to bypass it where needed and to improve it by adding a residual. Using residual connections was shown to be extremely effective in improving the learning ability of the neural networks compared to modeling absolute values (Littwin and Wolf 2016).

Finally, the new decoder consists of convolution layers mirroring the structure of the encoder in width and dilation. The output of the decoder was summed with the last input instance to the model, forming another residual connection. This last residual connection helps the model by providing it with a prior model (the original forecast). It means the model

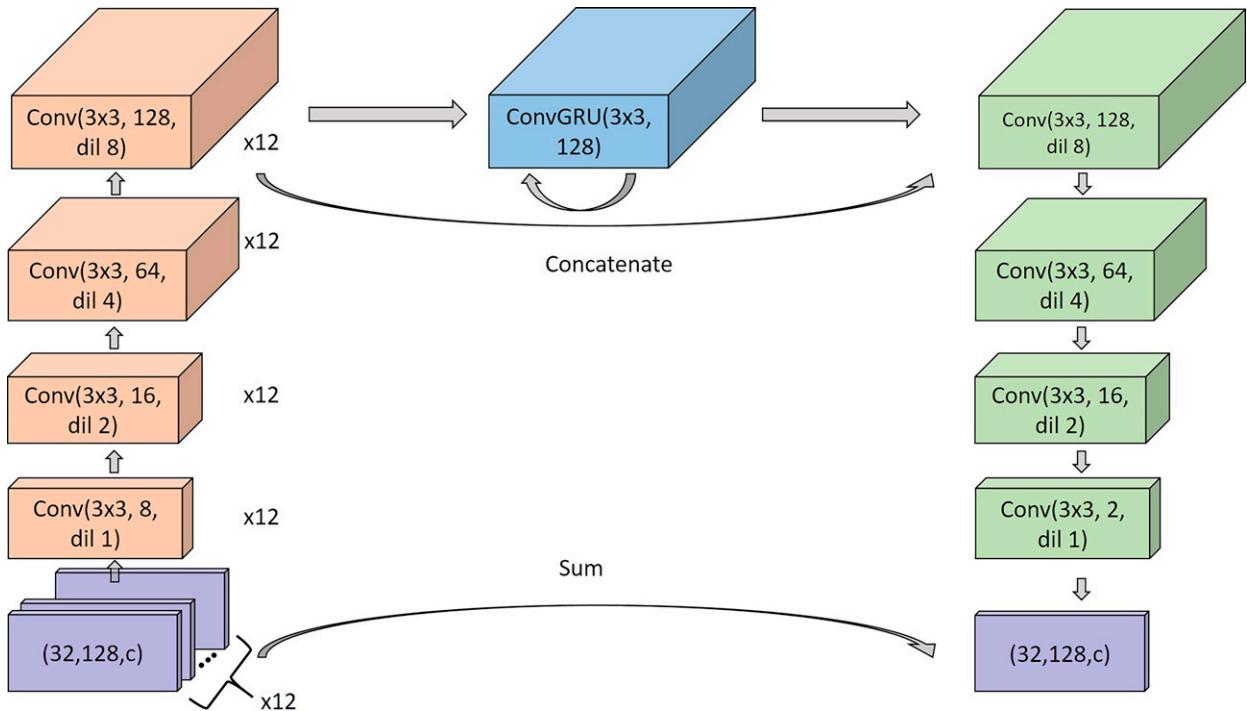


FIG. 1. Model architecture from bottom left: input (purple) in the form of a sequence of FC instances with c channels (variables) is passed one at a time to the encoder (orange), comprised of convolutional layers with increasing filters and dilation. The output of the encoder is fed to a CGRU (blue). The last output of the resulting sequence is concatenated with the last input into it, and introduced to the decoder (green), comprised of convolutional layers mirroring the encoder. The final result is summed with the last instance of the input sequence to form a residual connection (purple).

is actually learning to correct the original forecast input, instead of learning to make an independent prediction.

4. Deep learning wind prediction experiments

Four types of wind input to the DLM were tested for effectiveness in producing a more accurate wind input for wave forecasting. The input data for all experiments consisted of 12 consecutive hourly time steps from the FC dataset. The target was the REAN at the time of the last input. This effectively means improving the wind field at a given time t by using time steps $(t - 11, t)$. The network hyperparameters were initially set to those of Roitenberg and Wolf (2019). A short training period of the years 2010/11 and validation period of the year 2012 was used to test changes to the architecture. The chosen architecture (shown in Fig. 1) consisted of a four convolutional layers encoder with (8, 16, 64, 128) filters and a dilation of (1, 2, 4, 8), followed by a single CGRU layer with input and output dimensions of 128. The decoder consisted of four convolutional layers with (128, 32, 16, 2) filters and (8, 3, 2, 1) dilations. The datasets were split into a training set between the years 2001 and 2016, a validation set of the year 2000, and a test set of the year 2017. The validation set was used for hyperparameter tuning and internal model verification. It was separated from the test set to prevent similarities between the two. The presented results refer only to the test set. The DLM was trained and evaluated using an NVidia

GeForce GTX 2080 Ti GPU with a 12 GB memory. The FastAI API (Howard and Guggler 2020) was used with Pytorch API as a base. The model was optimized using Adam (Kingma and Ba 2015), an adaptive moment estimation algorithm using per-parameter learning rates based on second moments for minimizing the loss function. Weight decay was set to 10^{-3} , and the minibatch size was 16. A changing learning rate with the one-cycle approach of Smith (2018) was used, and each model was trained for eight cycles of two epochs. The max learning rate started at 10^{-3} , and was divided by the cycle number as learning progressed. After training, the validation set was used to identify the cycle with best performance. The weights of this cycle defined the new DLM, and its performance was evaluated on the test set. The resulting RMSEs (see supplemental material) in space and time of all wind input types are shown in Table 1 and compared to the original FC data. Additional statistics and figures are available in the supplemental material.

a. Input type 1: Wind velocity magnitude

The first experiment optimized prediction of wind velocity magnitude (UMag), defined as $U = \sqrt{u_{10}^2 + v_{10}^2}$. The magnitude was chosen as it seemed easier to predict, being always positive, nondirectional, and independent property in space. This resulted with input and output tensors with dimensions of (time = 12, $c = 1$, lat = 32, lon = 128). The resulting U was also transformed back to the form of u_{10} and v_{10} using the

TABLE 1. Wind velocity RMSE in space and time by wind input types.

Model	Property	DLM RMSE	FC RMSE	RMSE improved
UMag, inp. type 1	U (m s ⁻¹)	0.5999	0.6673	10.1%
	u_{10} (m s ⁻¹)	0.7075	0.7291	2.97%
	v_{10} (m s ⁻¹)	0.7065	0.7278	2.88%
UVec, inp. type 2	U (m s ⁻¹)	0.615	0.6673	7.8%
	u_{10} (m s ⁻¹)	0.6616	0.7291	9.26%
	v_{10} (m s ⁻¹)	0.6594	0.7278	9.39%
UDir, inp. type 3	$\cos\theta$	0.2307	0.2469	6.55%
	$\sin\theta$	0.229	0.2463	7.04%
	u_{10} (m s ⁻¹)	0.6906	0.7291	5.28%
	v_{10} (m s ⁻¹)	0.69	0.7278	5.19%
UFrc, inp. type 4	U (m s ⁻¹)	0.6162	0.6673	7.65%
	u_{10} (m s ⁻¹)	0.663	0.7291	9.06%
	v_{10} (m s ⁻¹)	0.6613	0.7278	9.14%

original FC direction. As expected, U improved significantly, as it is the main objective of the UMag DLM. It is interesting that the resulting u_{10} and v_{10} are improved by a much smaller percentage.

b. Input type 2: Wind velocity vector

The second experiment was performed to test the DLM's ability to improve the wind velocity vector (UVec) directly. The input was set as the FC u_{10} and v_{10} , and the output as the matching prediction, resulting with (12, 2, 32, 128) tensors. Although the improvement in the main objective of each DLM is smaller, the resulting wind vector improvement is almost 3 times as much as that of the UMag model.

c. Input type 3: Wind direction vector

The third experiment was predicting the direction of the wind velocity vector (UDir). The normalized directional unit vector (unit vector) was defined as

$$\begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} = \begin{pmatrix} u_{10}/U \\ v_{10}/U \end{pmatrix}, \quad (2)$$

and was set as both the input and output of the DLM. The test set output was multiplied by U to produce a wind velocity vector. Examining the results of this DLM found it similar to the UVec model with smaller improvement.

d. Input type 4: Wind friction velocity vector

Finally, an experiment was carried out to try and make a connection between a physical wave forecasting model and the DLM for wind prediction. The wave model uses the wind input through a source term (ST) which converts it to wave energy. Such a ST combines analytical and empirical derivations, with a

varying degree of complexity. The relatively simple wind friction velocity vector (UFrc) of WAM 3 (Hasselmann et al. 1988)

$$\mathbf{u}_* = \begin{pmatrix} u_{10}\sqrt{0.8 + 0.065u_{10}} \\ v_{10}\sqrt{0.8 + 0.065v_{10}} \end{pmatrix}, \quad (3)$$

was used in the DLM cost function, which should make it better fitting as an input to the ST. This still lacks the local wave action spectrum used in the source term, but as they are the result of an independent model with high computational cost, such a coupled model was not tested. This DLM's results were almost identical to the UVec model.

5. Wave forecasting with deep learning wind prediction

The effects of the new DLM output (the wind velocity prediction) on ocean waves forecasting was examined by using it as a forcing of the WAVEWATCH III v6.07 (WW3) model. WW3 ran with an unstructured grid of the eastern (Levant) area of the Mediterranean Sea, using 36 directions, 36 frequencies in the range 0.04–0.427 Hz and a time step of $dt_{\text{global}} = 10$ min. The wind forcing source term of Ardhuin et al. (2010) was used, alongside a linear wind interpolation. Six input configurations were tested: ECMWFs FC and REAN, and the four DLM outputs. WW3 ran separately with each forcing for the year 2017. The resulting wave forecast mean field parameters of significant wave height (H_s), mean wave direction (dir) and mean wave period ($T_{m0,-1}$) are shown in Table 2. All DLM outperformed the FC, as expected. Surprisingly, UMag had the best performance for both wave height and period, while UVec results with a better mean direction. UDir was outperformed by the other models and UFrc was almost identical to UVec with slightly worse results. Thus, only UMag and UVec are shown in the following analysis.

TABLE 2. Model wave mean parameters RMSE in space and time by wind input types. Bold values indicate the best result.

Property	FC	UMag (%improved)	UVec (%)	UDir (%)	UFrc (%)
H_s (m)	0.0765	0.0676 (11.6%)	0.0698 (8.7%)	0.0762 (0.4%)	0.0705 (7.8%)
Dir (°)	44.4	42.8 (3.4%)	42.2 (4.9%)	43.8 (1.3%)	42.5 (4.3%)
$T_{m0,-1}$ (s)	0.309	0.283 (8.4%)	0.286 (7.4%)	0.307 (0.05%)	0.287 (7.1%)

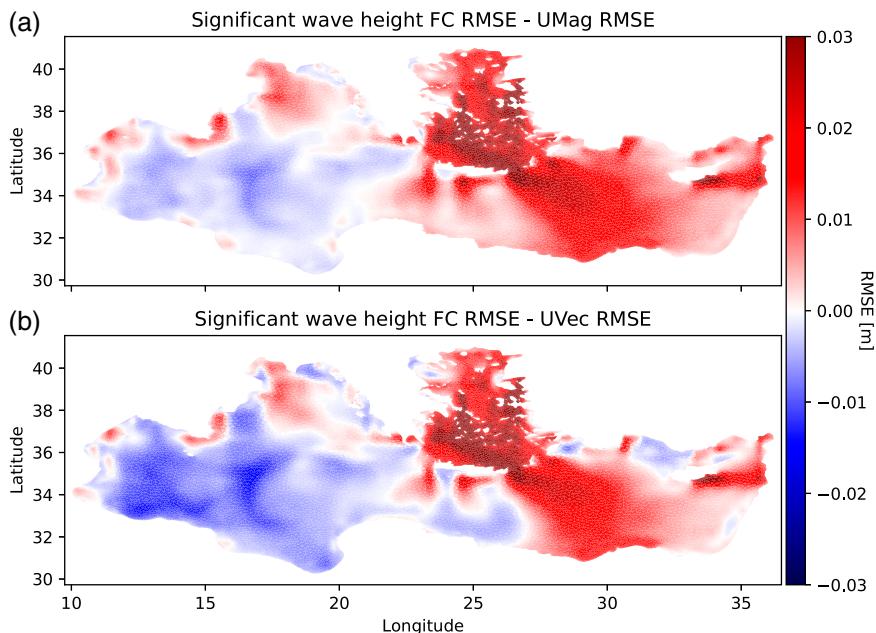


FIG. 2. Time-mean RMSE difference map of significant wave height H_s for (a) FC RMSE – UMag RMSE and (b) FC RMSE – UVec RMSE. FC with larger error is in red, and DLM is in blue.

A spatial map of H_s time-mean RMSE differences can be seen in Fig. 2. The RMSE difference was taken as $\text{RMSE}_{\text{FC}} - \text{RMSE}_{\text{DLM}}$, meaning the new DLM has better performance where positive and vice versa. It is immediately apparent that both DLMs outperform the original FC in the eastern part of the basin, especially in the Aegean Sea where the local improvement is $\sim 20\%$. The FC slightly outperforms the DLM at the southwestern part. This spatial difference is correlated to a much higher RMSE in the original FC data at the eastern half, specifically in the Aegean Sea (see Fig. S8 in the online supplemental material). The large RMSE results in larger gradients while training the DLM, and thus greater improvement. The improved performance of UMag can be attributed to more accurate results over the western half, including improved performance along the coastal area.

A temporal comparison of spatial-mean RMSE of the DLM and FC is given in Fig. 3. This shows that the main improvement of both DLM was during the spring to autumn period, most prominently during the summer months, implying correction of the Etesian wind. Examining the Aegean Sea during the Etesian results in a staggering 35% RMSE improvement. This examination was done by looking at the RMSE of the Aegean Sea from mid-May to mid-September. The current model can be used as is, or as a seasonal model, alongside a separate seasonal model trained specifically for the winter season or even for stormy conditions. Such models can work as an ensemble to produce better results by using localized seasonal models.

6. Summary and discussion

In this work a novel deep learning model for wind velocity postprocessing was presented. The model allows to improve

wind and waves numerical, physics-based models' accuracy by using a deep learning, data-driven model (DLM). The DLM's input were the forecasts (FC) which were used in ECMWF ERA5 reanalysis (REAN), and the ground truth was the REAN data themselves. Several input types were tested, and the best performance was achieved by using the wind magnitude (UMag), disregarding the wind direction. This model consisted of a convolutional encoder and decoder, with a convolutional gated recurrent unit in between. The DLM's output was used as a forcing for a wave forecasting model (WAVEWATCH III), and the resulting significant wave height, mean wave direction and mean wave period were examined. The new model showed significant improvement in all wind and wave parameters.

The model can be deployed to operational atmospheric models as a postprocess of the wind forecast, or to a wave model as a preprocess of the wind input source term. In both cases the DLM has to be trained to match the grid of the desired model and use recent wind forecast and reanalysis to fine-tune and improve the results. This fine-tuning process might also have to be repeated as data or model drifts might occur, for example, due to climate change. This periodic fine-tuning process is a common practice in operational deep learning, and it helps to "future-proof" the model by allowing it to learn from the most recent and relevant data available.

The presented DLM was used to improve wind velocity but could easily be trained to improve any other parameter of the atmospheric model, such as geopotential height or temperature. It could also be trained over different locations, or as a global model. Furthermore, another very interesting usage is training toward seasonal localized models. These could be optimized over specific time periods and locations where

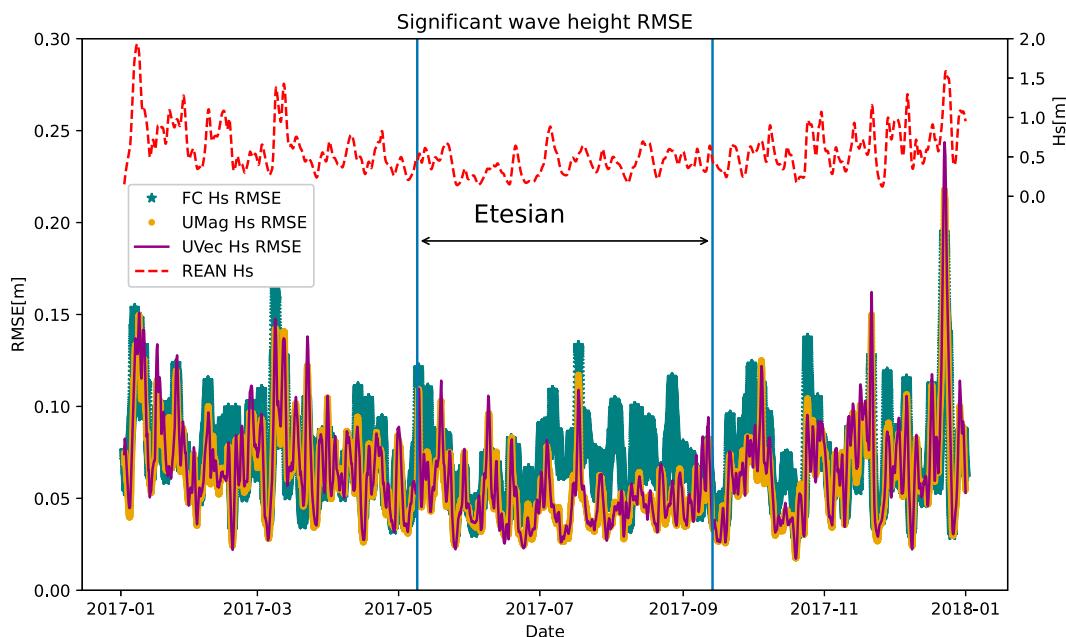


FIG. 3. The 24-h moving average of the spatial-mean (east Mediterranean) RMSE of the significant wave height H_s of FC (thick teal), UMag (medium orange), and UVec (thin purple). The right axis is the REAN H_s (dashed red line), for reference. Improvement is apparent between May and September when Etesian wind is dominant over the Aegean Sea.

weather conditions are hard to predict and result in significant improvement. One such example is shown in this work at the Aegean Sea, where the Etesian wind is dominant from mid-May to mid-September. Even without training specifically for this task, the presented model improves the significant wave height forecast over the Aegean Sea at this period by $\sim 35\%$.

The new model has very minimal computational costs, which are negligible when compared to either the numerical wind or wave forecasting models. Running the entire Mediterranean Sea over an entire year as was done in this work took the numerical model the equivalent of over 4000 CPU hours, while the DLM postprocessing took less than 1 CPU hour. Furthermore, it could easily be implemented, as it does not require any adjustment to any of the currently used operational models, while providing significant improvement in forecasting results.

Acknowledgments. This research was supported by the ISRAEL SCIENCE FOUNDATION (Grant 1601/20) and the Gordon Center for Energy Studies. The authors thank Prof. Lior Wolf for his insights and helpful discussion. Author contributions: Y.Y. conceived the research, created the deep learning model, and performed the experiments. Y.T. supervised the work. Both authors wrote the manuscript.

Data availability statement. The code to recreate the work is available at https://github.com/yuvalyevnin/Wind_Processing/. The ERA5 wind data are available from ECMWF and the Copernicus Climate Data Store database at <https://cds.climate.copernicus.eu/>. The WAVEWATCH III wave model is available at <https://polar.ncep.noaa.gov/waves/>.

REFERENCES

- Arcomano, T., I. Szunyogh, J. Pathak, A. Wikner, B. R. Hunt, and E. Ott, 2020: A machine learning-based global atmospheric forecast model. *Geophys. Res. Lett.*, **47**, e2020GL087776, <https://doi.org/10.1029/2020GL087776>.
- Ardhuin, F., L. Bertotti, J.-R. Bidlot, L. Cavaleri, V. Filipetto, J.-M. Lefevre, and P. Wittmann, 2007: Comparison of wind and wave measurements and models in the western Mediterranean sea. *Ocean Eng.*, **34**, 526–541, <https://doi.org/10.1016/j.oceaneng.2006.02.008>.
- , and Coauthors, 2010: Semiempirical dissipation source functions for ocean waves. Part I: Definition, calibration, and validation. *J. Phys. Oceanogr.*, **40**, 1917–1941, <https://doi.org/10.1175/2010JPO4324.1>.
- Ballas, N., L. Yao, C. Pal, and A. Courville, 2015: Delving deeper into convolutional networks for learning video representations. *4th Int. Conf. on Learning Representations*, San Diego, CA, ICLR, 11 pp., <https://doi.org/10.48550/arXiv.1511.06432>.
- Bidlot, J.-R., D. J. Holmes, P. A. Wittmann, R. Lalbeharry, and H. S. Chen, 2002: Intercomparison of the performance of operational ocean wave forecasting systems with buoy data. *Wea. Forecasting*, **17**, 287–310, [https://doi.org/10.1175/1520-0434\(2002\)017<0287:IOTPOO>2.0.CO;2](https://doi.org/10.1175/1520-0434(2002)017<0287:IOTPOO>2.0.CO;2).
- Brenowitz, N. D., and C. S. Bretherton, 2019: Spatially extended tests of a neural network parametrization trained by coarse-graining. *J. Adv. Model. Earth Syst.*, **11**, 2728–2744, <https://doi.org/10.1029/2019MS001711>.
- Brunton, S. L., B. R. Noack, and P. Koumoutsakos, 2020: Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.*, **52**, 477–508, <https://doi.org/10.1146/annurev-fluid-010719-060214>.
- Chung, J., C. Gulcehre, K. Cho, and Y. Bengio, 2014: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv, 1412.3555, <https://doi.org/10.48550/arXiv.1412.3555>.

- Gentine, P., M. Pritchard, S. Rasp, G. Reinaudi, and G. Yacalis, 2018: Could machine learning break the convection parameterization deadlock? *Geophys. Res. Lett.*, **45**, 5742–5751, <https://doi.org/10.1029/2018GL078202>.
- Grönquist, P., C. Yao, T. Ben-Nun, N. Dryden, P. Dueben, S. Li, and T. Hoefler, 2021: Deep learning for post-processing ensemble weather forecasts. *Philos. Trans. Roy. Soc.*, **379**, 2194, <https://doi.org/10.1098/rsta.2020.0092>.
- Hasselmann, K., and Coauthors, 1988: The WAM model—A third generation ocean wave prediction model. *J. Phys. Oceanogr.*, **18**, 1775–1810, [https://doi.org/10.1175/1520-0485\(1988\)018<1775:TWMTOG>2.0.CO;2](https://doi.org/10.1175/1520-0485(1988)018<1775:TWMTOG>2.0.CO;2).
- Haupt, S. E., W. Chapman, S. V. Adams, C. Kirkwood, J. S. Hosking, N. H. Robinson, S. Lerch, and A. C. Subramanian, 2021: Towards implementing artificial intelligence post-processing in weather and climate: Proposed actions from the Oxford 2019 workshop. *Philos. Trans. Roy. Soc.*, **A379**, 20200091, <https://doi.org/10.1098/rsta.2020.0091>.
- Hersbach, H., and Coauthors, 2020: The ERA5 global reanalysis. *Quart. J. Roy. Meteor. Soc.*, **146**, 1999–2049, <https://doi.org/10.1002/qj.3803>.
- Howard, J., and S. Gugger, 2020: Fastai: A layered API for deep learning. *Information*, **11**, 108, <https://doi.org/10.3390/info11020108>.
- Kingma, D. P., and J. L. Ba, 2015: Adam: A method for stochastic optimization. *3rd Int. Conf. on Learning Representations*, San Diego, CA, ICLR, <https://doi.org/10.48550/arXiv.1412.6980>.
- Littwin, E., and L. Wolf, 2016: The loss surface of residual networks: Ensembles and the role of batch normalization. arXiv, 1611.02525, <https://doi.org/10.48550/arXiv.1611.02525>.
- Ramon, J., L. Lledó, V. Torralba, A. Soret, and F. J. Doblas-Reyes, 2019: What global reanalysis best represents near-surface winds? *Quart. J. Roy. Meteor. Soc.*, **145**, 3236–3251, <https://doi.org/10.1002/qj.3616>.
- Rasp, S., and S. Lerch, 2018: Neural networks for postprocessing ensemble weather forecasts. *Mon. Wea. Rev.*, **146**, 3885–3900, <https://doi.org/10.1175/MWR-D-18-0187.1>.
- , and N. Thuerey, 2020: Data-driven medium-range weather prediction with a Resnet pretrained on climate simulations: A new model for Weather Bench. *J. Adv. Model. Earth Syst.*, **13**, e2020MS002405, <https://doi.org/10.1029/2020MS002405>.
- Reichstein, M., G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais and Prabhat, 2019: Deep learning and process understanding for data-driven earth system science. *Nature*, **566**, 195–204, <https://doi.org/10.1038/s41586-019-0912-1>.
- Roitenberg, A., and L. Wolf, 2019: Forecasting traffic with a convolutional GRU decoder conditioned on adapted historical data. *ICML 2019 Time Series Workshop*, Long Beach, CA, ICML, 6 pp., https://roseyu.com/time-series-workshop/submissions/2019/timeseries-ICML19_paper_29.pdf.
- Schneider, T., S. Lan, A. Stuart, and J. Teixeira, 2017: Earth system modeling 2.0: A blueprint for models that learn from observations and targeted high-resolution simulations. *Geophys. Res. Lett.*, **44**, 12 396–12 417, <https://doi.org/10.1002/2017GL076101>.
- Smith, L. N., 2018: A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay. US Naval Research Laboratory Tech. Rep. 5510-026, 21 pp., <https://doi.org/10.48550/arXiv.1803.09820>.
- Tolman, H. L., 1991: A third-generation model for wind waves on slowly varying, unsteady, and inhomogeneous depths and currents. *J. Phys. Oceanogr.*, **21**, 782–797, [https://doi.org/10.1175/1520-0485\(1991\)021<0782:ATGMFW>2.0.CO;2](https://doi.org/10.1175/1520-0485(1991)021<0782:ATGMFW>2.0.CO;2).
- Vannitsem, S., and Coauthors., 2020: Statistical postprocessing for weather forecasts: Review, challenges and avenues in a big data world. *Bull. Amer. Meteor. Soc.*, **102**, E681–E699, <https://doi.org/10.1175/BAMS-D-19-0308.1>.
- Veldkamp, S., K. Whan, S. Dirksen, and M. Schmeits, 2020: Statistical post-processing of wind speed forecasts using convolutional neural networks. *Mon. Wea. Rev.*, **149**, 1141–1152, <https://doi.org/10.1175/MWR-D-20-0219.1>.
- Wang, S., J. Cao, and P. S. Yu, 2019: Deep learning for spatio-temporal data mining: A survey. *IEEE. Trans. Data Eng.*, **34**, 3681–3700, <https://doi.org/10.1109/TKDE.2020.3025580>.
- Weyn, J. A., D. R. Durran, and R. Caruana, 2020: Improving data-driven global weather prediction using deep convolutional neural networks on a cubed sphere. *J. Adv. Model. Earth Syst.*, **12**, e2020MS002109, <https://doi.org/10.1029/2020MS002109>.
- Wikner, A., J. Pathak, B. Hunt, M. Girvan, T. Arcomano, I. Szunyogh, A. Pomerance, and E. Ott, 2020: Combining machine learning with knowledge-based modeling for scalable forecasting and subgrid-scale closure of large, complex, spatio-temporal systems. *Chaos*, **30**, 053111, <https://doi.org/10.1063/5.0005541>.
- Yu, F., and V. Koltun, 2015: Multi-Scale Context Aggregation by Dilated Convolutions. *Fourth Int. Conf. on Learning Representations*, San Juan, Puerto Rico, ICLR, 13 pp., <https://doi.org/10.48550/arXiv.1511.07122>.
- Zjavka, L., 2015: Wind speed forecast correction models using polynomial neural networks. *Renew. Energy*, **83**, 998–1006, <https://doi.org/10.1016/j.renene.2015.04.054>.