Global ocean wind speed estimation with CyGNSSnet

Caroline Arnold
German Climate Computing Center, DKRZ
Bundesstrasse 45a, Hamburg, 20146, Germany
arnold@dkrz.de

Milad Asgarimehr
German Research Centre for Geosciences, GFZ
Telegrafenberg, 14473 Potsdam, Germany
milad.asgarimehr@gfz-potsdam.de

Abstract

The CyGNSS (Cyclone Global Navigation Satellite System) satellite system measures GNSS signals reflected off the Earth's surface. A global ocean wind speed dataset is derived, which fills a gap in Earth observation data, will improve cyclone forecasting, and could be used to mitigate effects of climate change. We propose CyGNSSnet, a deep learning model for predicting wind speed from CyGNSS observables, and evaluate its potential for operational use. With CyGNSSnet, performance improves by 29% over the current operational model. We further introduce a hierarchical model, that combines an extreme value classifier and a specialized CyGNSSnet and slightly improves predictions for high winds.

1 Introduction

The NASA CyGNSS (Cyclone Global Navigation Satellite System) is a constellation of eight microsatellites with the aim of improving hurricane intensity forecasts [1, 2]. CyGNSS picks up the signals from global navigation system satellites such as GPS and BeiDou, scattered off the Earth's surface. These reflected signals encode the ocean surface roughness, and thus the wind speed [3]. GNSS signals are only insignificantly affected by clouds and precipitation, and are thus suitable for remote sensing in adverse meteorological conditions [4]. CyGNSS covers tropical and subtropical regions ($\pm 35^\circ$ latitude) with an average revisit time of seven hours.

The CyGNSS wind speed dataset will be useful to mitigate the effects of climate change. Like other extreme weather events, cyclones are expected to increase in frequency and intensity in a warming climate, requiring improved forecasts [5]. Offshore wind turbines are a major renewable energy source with a projected installation of 205 GW capacity in the coming 10 years [6]. Global observations can help to better understand the relation between climate change, atmospheric conditions, and wind speed [7]. A wind atlas can be useful for identifying future offshore park locations [8]. Knowledge of extreme winds is beneficial for turbine safety engineering [9].

The main measurement in GNSS reflectometry is the Delay Doppler map (DDM), a 2D data array mapping the cross-correlation power of the original and the reflected GNS signal across bins of time delay and Doppler frequency shift. Wind speed retrieval algorithms have been successfully developed for CyGNSS data [10]. However, the algorithms are still being evaluated for field conditions and may contain undiscovered biases. Both for a previous mission with the TechDemoSat-1 satellite [11], as well as for CyGNSS [12, 13], it has been demonstrated that a feed forward neural network can estimate wind speed better than the conventional retrieval approach [11]. Convolutional networks

Tackling Climate Change with Machine Learning: workshop at NeurIPS 2021.

have been used to extract features from DDMs [14]. The evaluation on a large CyGNSS test set showed potential for operational use, with challenges at predicting high wind speeds [15].

We propose CyGNSSnet, a deep learning framework to predict wind speed from CyGNSS observational products using supervised learning. In this paper, we present our methodology and critically assess the performance of CyGNSSnet. Our focus is on extreme value prediction and potential operational use.

2 Methods

2.1 CyGNSS dataset

We use version 2.1 of the CyGNSS data set [16], covering 1 January 2018 – 20 February 2019. To exclude low-quality samples, we filter samples with a set of conditions, for details, see Appendix A.1. We use the first 215 days (7.2×10^6 samples) for training, the following 75 days (4.7×10^6 samples) for validation, and the remaining 127 days (8.8×10^6 samples) as a blind test set. The wind speed distribution is comparable across the three datasets, and the good quality samples are clustered in time. For details see Appendix A.2.

Each sample contains the bistatic radar cross section (BRCS) DDM [10, 17], a 17 x 11 pixel 2D data array that is treated like an image. Besides ancillary parameters are provided that are related to the measurement geometry, satellite status, and features processed from DDMs. For CyGNSSnet, we select 10 ancillary parameters in a data-driven way, for details, see Appendix A.3 and Table S2.

The global ocean wind speed labels for supervised learning are obtained from ERA5 reanalysis data [18] and interpolated to match the CyGNSS specular point coordinates. If samples are labeled with the same wind speed due to limited spatiotemporal resolution, we randomly select one of them for training. Note that this approach implicitly assumes that the wind speed is uniform over the entire spatial domain covered by the DDM.

As a baseline for the evaluation of CyGNSSnet, we include wind speed predictions obtained by a conventional method, the Minimum Variance Estimator for fully developed seas [10, 19]. These wind speeds are available as part of the Level 2 CyGNSS data product [20].

2.2 CyGNSSnet

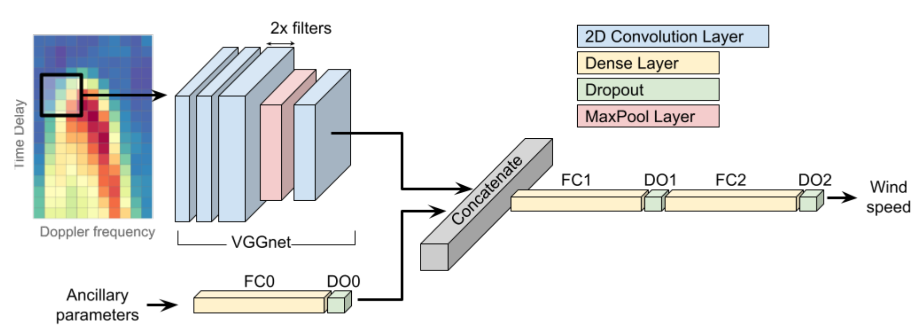


Figure 1: CyGNSSnet processes two lines of input. BRCS DDMs (17 x 11 2D arrays) are processed through a convolutional neural network (CNN) based on the VGGnet architecture [21]. Ancillary parameters are processed in a second input line with one dense layer. Both lines are concatenated and processed through two dense layers. Wind speed is predicted as a continuous variable.

The CyGNSSnet architecture is shown in Figure 1. To make use of the local features in the DDMs, we use two input lines: First, the DDMs are processed by a convolutional network based on the VGGnet architecture [21]. In a second input line, the ancillary parameters are processed through a dense layer. Then, both input lines are concatenated and processed through two dense layers. Dropout layers are added to improve regularization [22]. The model hyperparameters are optimized using the Tree-Parzen Estimator with the NNI package [23]. We average predictions across an ensemble of three models. For the full hyperparameter search space and the model configurations, see Appendix A.4. We use the Adam optimizer and the mean squared error loss function.

CyGNSSnet is implemented in Pytorch [24]. Training is conducted on single NVIDIA K80 GPUs and takes less than 12 hours per model to complete.

2.3 Extreme value classifier

Less than 5 % of the samples are labeled with a wind speed exceeding $12\,\mathrm{m/s}$. In order to improve the performance on these samples, we train a separate CyGNSSnet-X only on extreme values exceeding $10\,\mathrm{m/s}$, where 8.3×10^5 samples remain. For hyperparameters see Table S5.

Whether a given instance constitutes an extreme sample is decided by a separate classifier. We train an XGBoost classifier to state whether a sample exceeds $12\,\mathrm{m/s}$, allowing for some overlap with the CyGNSSnet-X training dataset. The classifier hyperparameters (Appendix A.4.2) are tuned on the validation set, such that the F_β score, $\beta = 0.5$, is maximized. This emphasizes precision over recall, since the model trained on extreme values will perform poorly on average samples.

Predictions are then made with a hierarchical model, where C refers to the classifier, $\mathcal{M}_S$ to the model trained on all available samples (CyGNSSnet), and $\mathcal{M}_X$ to the model trained only on extreme value samples (CyGNSSnet-X):

$$\hat{v}_i = \begin{cases} \mathcal{M}_X(x_i), & \text{if } C(x_i) = 1, \\ \mathcal{M}_S(x_i), & \text{otherwise.} \end{cases} \tag{1}$$

3 Results

3.1 General evaluation and model comparison

We evaluate CyGNSSnet on the hold-out test set covering Oct 17, 2018 – Feb 20, 2019. Table 1 shows the root mean square error (RMSE), with true values v and predicted values $\hat{v}$,

$$\mathrm{RMSE}(v, \hat{v}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{v}_i - v_i)^2},$$

for different deep learning algorithms: CyGNSSnet was trained on all samples, CyGNSSnet-X only on samples exceeding $10\,\mathrm{m/s}$. CyGNSSnet-C includes the classifier, cf. Eq. (1). MVE is the current operational retrieval algorithm.

Table 1: RMSE obtained on the test set for different architectures and wind speeds. CyGNSSnet-X was not trained at low winds, indicated by the round brackets. Best value highlighted in **bold**.

Architecture	All samples RMSE (m/s)	$v \leq 12\,\mathrm{m/s}$ RMSE (m/s)	$12\,\mathrm{m/s} < v \leq 16\,\mathrm{m/s}$ RMSE (m/s)	$v > 16\,\mathrm{m/s}$ RMSE (m/s)
CyGNSSnet	**1.36**	**1.31**	2.38	4.99
CyGNSSnet-X	(5.26)	(5.32)	1.48	4.40
CyGNSSnet-C	1.38	1.34	**2.26**	4.79
MVE	1.90	1.88	2.29	**3.39**

Across all samples and for winds below $12\,\mathrm{m/s}$, CyGNSSnet outperforms all other models. Compared to the MVE, the RMSE is reduced by $29\,\%$. Note that CyGNSSnet-X was not trained in this range.

For high winds, $v > 12\,\mathrm{m/s}$, MVE outperforms CyGNSSnet. CyGNSSnet-X, specifically trained for this region, reaches a lower RMSE than the MVE in the region $12\,\mathrm{m/s} < v \leq 16\,\mathrm{m/s}$. At very high wind speeds exceeding $v > 16\,\mathrm{m/s}$, it performs worse than MVE. Note that these are extreme values, even to the data seen by CyGNSSnet-X.

To evaluate the performance of CyGNSSnet-C, first the classifier accuracy is determined on the test set to $F_\beta = 0.35, (\beta = 0.5)$ (for details, see Appendix A.5). Even though many samples are incorrectly classified, extreme value predictions are improved by the combined model CyGNSSnet-C. For winds with $12\,\mathrm{m/s} < v \leq 16\,\mathrm{m/s}$, CyGNSSnet-C outperforms MVE slightly. Beyond $16\,\mathrm{m/s}$, CyGNSSnet-C misses the MVE baseline, but improves on the standard CyGNSSnet.

Figure 2 shows the log-scale density plot of the predicted and the true wind speed values for CyGNSSnet, CyGNSSnet-C, and MVE. The CyGNSSnet wind speeds are considerably closer to the 1:1 line than the MVE wind speeds. A slight overestimation at average wind speeds, as well as an underestimation for high wind speeds, is observed for all models. Comparing CyGNSSnet-C to standard CyGNSSnet, the bias at high wind speeds is slightly reduced.

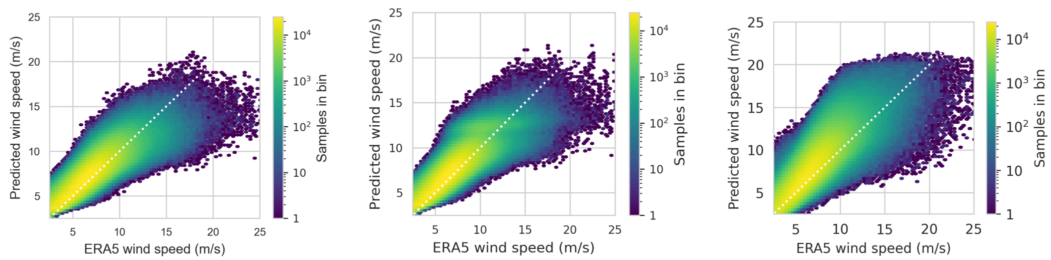
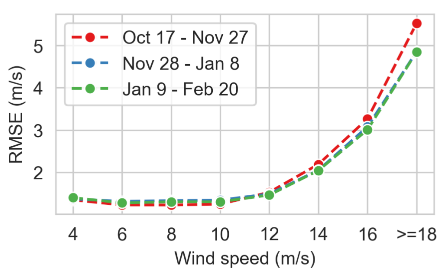


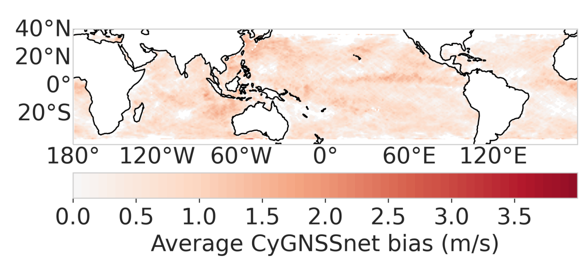
Figure 2: Log-scale density plot of predicted wind speed vs. true (ERA5) wind speed. *Left:* CyGNSSnet *Center:* CyGNSSnet-C *Right:* MVE

3.2 Evaluation in time and space

For an operational application, the deep learning algorithm must perform stable in time and space. Figure 3(a) shows the RMSE for different values of the ERA5 wind speed in three phases of the test set, spanning about six weeks each. The error is comparable throughout the phases and only slightly affected by the presence of stronger winds in the first phase. Figure 3(b) shows the CyGNSSnet bias, $\hat{v}_i - v_i$, averaged on a latitude-longitude grid with 1° resolution. The spatial patterns of wind speed overestimation are similar to the ones seen for MVE (see Fig. S8). Thus, they are likely resulting from satellite measurement, rather than from a shortcoming of the deep learning algorithm.



(a) CyGNSSnet applied in three different time spans within the test set. RMSE is given for different values of the wind speed target.



(b) CyGNSSnet bias, averaged on a latitude-longitude grid with 1° resolution.

Figure 3: CyGNSSnet evaluated in time and space.

4 Discussion

We introduced CyGNSSnet, a deep learning algorithm to predict global ocean wind speed from DDMs. The overall performance improves by 29% compared to the currently employed operational algorithm (MVE). At high wind speeds exceeding $12\,\mathrm{m/s}$, CyGNSSnet performs worse than the MVE. We demonstrate that a hierarchical model, including an extreme value classifier and a separate CyGNSSnet-X trained only on extreme values, can slightly improve performance in this range. The classifier could potentially be further improved to increase the benefit of this approach.

Note that all methods suffer from underestimation of high wind speeds, which can be linked to the sensitivity saturation of DDM observables in this regime [20, 25]. Besides, the high variability of strong wind speeds can introduce errors in the wind speed labels. With the current setup, CyGNSSnet is limited to reproduce essentially the ERA5 reanalysis wind speed dataset. Future work should explore other sources for wind speed labels, and potentially use self-supervised learning.

CyGNSSnet provides stable performance in time over the available test set. The performance evaluated on a global grid is comparable to existing bias patterns. In following work, we will aim to further reduce the bias by incorporating relevant parameters, such as precipitation [26, 27]. Already now, CyGNSSnet is a match for the operational wind speed retrieval algorithm and demonstrates the huge potential for deep learning in GNSS remote sensing.

Acknowledgments and Disclosure of Funding

This study is conducted with financial supports from the German Research Centre for Geosciences at GFZ, Potsdam, Germany. CA was funded by Helmholtz Association's Initiative and Networking Fund through Helmholtz AI [grant number: ZT-I-PF-5-01]. This work used resources of the Deutsches Klimarechenzentrum (DKRZ) 325 granted by its Scientific Steering Committee (WLA) under project ID AIM. The datasets used in this study are available free of charge and we thank scientific teams associated with the CyGNSS mission at NASA and the University of Michigan and ERA5 reanalysis estimates at the European Centre for Medium-Range Weather Forecasts (ECMWF). We thank Varvara Vetrova for helpful discussions.

References

[1] Christopher S. Ruf, Robert Atlas, Paul S. Chang, Maria Paola Clarizia, James L. Garrison, Scott Gleason, Stephen J. Katzberg, Zorana Jelenak, Joel T. Johnson, Sharanya J. Majumdar, Andrew O'brien, Derek J. Posselt, Aaron J. Ridley, Randall J. Rose, and Valery U. Zavorotny. New Ocean Winds Satellite Mission to Probe Hurricanes and Tropical Convection. *Bulletin of the American Meteorological Society*, 97(3): 385–395, 2016. doi: 10.1175/BAMS-D-14-00218.1.

[2] Christopher S Ruf, Clara Chew, Timothy Lang, Mary G Morris, Kyle Nave, Aaron Ridley, and Rajeswari Balasubramaniam. A new paradigm in earth environmental monitoring with the CYGNSS small satellite constellation. *Scientific reports*, 8(1):1–13, 2018.

[3] S. Gleason, S. Hodgart, Yiping Sun, C. Gommenginger, S. Mackin, M. Adjrad, and M. Unwin. Detection and processing of bistatically reflected gps signals from low earth orbit for the purpose of ocean remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 43(6):1229–1241, 2005. doi: 10.1109/TGRS.2005.845643.

[4] Milad Asgarimehr, Jens Wickert, and Sebastian Reich. Evaluating impact of rain attenuation on space-borne gnss reflectometry wind speeds. *Remote Sensing*, 11(9), 2019. doi: 10.3390/rs11091048.

[5] Thomas R. Knutson, John L. McBride, Johnny Chan, Kerry Emanuel, Greg Holland, Chris Landsea, Isaac Held, James P. Kossin, A. K. Srivastava, and Masato Sugi. Tropical cyclones and climate change. *Nature Geoscience*, 3(3):157–163, 2010. doi: 10.1038/ngeo779.

[6] Global Wind Report 2019, 2019. URL `https://gwec.net/global-wind-report-2019/`.

[7] Zhenzhong Zeng, Alan D. Ziegler, Timothy Searchinger, Long Yang, Anping Chen, Kunlu Ju, Shilong Piao, Laurent Z. X. Li, Philippe Ciais, Deliang Chen, Junguo Liu, Cesar Azorin-Molina, Adrian Chappell, David Medvigy, and Eric F. Wood. A reversal in global terrestrial stilling and its implications for wind energy production. *Nature Climate Change*, 9(12):979–985, 2019. ISSN 1758-6798. doi: 10.1038/s41558-019-0622-6.

[8] Paul Veers, Katherine Dykes, Eric Lantz, Stephan Barth, Carlo L. Bottasso, Ola Carlson, Andrew Clifton, Johney Green, Peter Green, Hannele Holttinen, Daniel Laird, Ville Lehtomäki, Julie K. Lundquist, James Manwell, Melinda Marquis, Charles Meneveau, Patrick Moriarty, Xabier Munduate, Michael Muskulus, Jonathan Naughton, Lucy Pao, Joshua Paquette, Joachim Peinke, Amy Robertson, Javier Sanz Rodrigo, Anna Maria Sempreviva, J. Charles Smith, Aidan Tuohy, and Ryan Wiser. Grand challenges in the science of wind energy. *Science*, 366(6464):eaau2027, 2019. doi: 10.1126/science.aau2027.

[9] Sara C. Pryor and Rebecca J. Barthelmie. A global assessment of extreme wind speeds for wind energy applications. *Nature Energy*, 6(3):268–276, 2021. doi: 10.1038/s41560-020-00773-7.

[10] M. P. Clarizia, C. S. Ruf, P. Jales, and C. Gommenginger. Spaceborne GNSS-R Minimum Variance Wind Speed Estimator. *IEEE Transactions on Geoscience and Remote Sensing*, 52(11):6829–6843, 2014. doi: 10.1109/TGRS.2014.2303831.

[11] M. Asgarimehr, I. Zhelavskaya, G. Foti, S. Reich, and J. Wickert. A GNSS-R Geophysical Model Function: Machine Learning for Wind Speed Retrievals. *IEEE Geoscience and Remote Sensing Letters*, 17(8): 1333–1337, 2020. doi: 10.1109/LGRS.2019.2948566.

[12] J. Reynolds, M. P. Clarizia, and E. Santi. Wind Speed Estimation From CYGNSS Using Artificial Neural Networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13: 708–716, 2020. doi: 10.1109/JSTARS.2020.2968156.

[13] Y. Liu, I. Collett, and Y. J. Morton. Application of Neural Network to GNSS-R Wind Speed Retrieval. *IEEE Transactions on Geoscience and Remote Sensing*, 57(12):9756–9766, 2019. doi: 10.1109/TGRS.2019.2929002.

[14] X. Chu, J. He, H. Song, Y. Qi, Y. Sun, W. Bai, W. Li, and Q. Wu. Multimodal Deep Learning for Heterogeneous GNSS-R Data Fusion and Ocean Wind Speed Retrieval. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:5971–5981, 2020. doi: 10.1109/JSTARS.2020.3010879.

[15] Milad Asgarimehr, Caroline Arnold, Tobias Weigel, Chris Ruf, and Jens Wickert. GNSS Reflectometry Global Ocean Wind Speed using Deep Learning: Development and Assessment of CyGNSSnet, Remote Sensing of Environment (accepted), 2021.

[16] CYGNSS. CYGNSS Level 1 Science Data Record Version 2.1, 2017. URL `https://podaac.jpl.nasa.gov/dataset/CYGNSS_L1_V2.1`.

[17] V. U. Zavorotny and A. G. Voronovich. Scattering of GPS signals from the ocean with wind remote sensing application. *IEEE Transactions on Geoscience and Remote Sensing*, 38(2):951–964, 2000. doi: 10.1109/36.841977.

[18] Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, et al. The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020.

[19] Christopher S Ruf and Rajeswari Balasubramaniam. Development of the CYGNSS geophysical model function for wind speed. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(1):66–77, 2018.

[20] Christopher S Ruf, Scott Gleason, and Darren S McKague. Assessment of CYGNSS wind speed retrieval uncertainty. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(1): 87–97, 2018.

[21] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]*, 2015.

[22] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15: 1929–1958, 2014.

[23] Microsoft Corporation. NNI (Neural Network Intelligence) 1.8, 2020. URL `https://github.com/microsoft/nni`.

[24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[25] Lixin Zeng and Robert A Brown. Scatterometer observations at high wind speeds. *Journal of Applied Meteorology*, 37(11):1412–1420, 1998.

[26] Milad Asgarimehr, Valery Zavorotny, Jens Wickert, and Sebastian Reich. Can GNSS reflectometry detect precipitation over oceans? *Geophysical Research Letters*, 45(22):12–585, 2018.

[27] Rajeswari Balasubramaniam and Christopher Ruf. Characterization of rain impact on L-Band GNSS-R ocean surface measurements. *Remote Sensing of Environment*, 239:111607, 2020.

[28] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[29] Jorge Querol, Alberto Alonso-Arroyo, Raul Onrubia, Daniel Pascual, Hyuk Park, and Adriano Camps. SNR degradation in GNSS-R measurements under the effects of radio-frequency interference. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(10):4865–4878, 2016.

A Appendix

A.1 Quality control in the dataset

To exclude low-quality samples, we filter for samples that meet a set of conditions [12]:

1. The BRCS DDM uncertainty is below 1 (ddm_brcs_uncert < 1)
2. The spacecraft roll is between $1°$ and $30°$, the pitch is between $1°$ and $10°$, or the yaw is between $1°$ and $5°$ (quality_flag = 4)
3. Nano star tracker attitude status is OK (nst_att_status = 1)
4. The receive antenna gain in the direction of the specular point is larger than 0 dBi (sp_rx_gain)
5. The range corrected gain figure of merit of the DDM is larger than 0 (prn_fig_of_merit)
6. The leading edge slope (ddm_les) is larger than 0
7. The zenith signal to noise ration is larger than 0 dB (direct_signal_snr)

To remove potentially mislabeled samples, we determine the 95% confidence interval of the wind-speed-dependent value of the normalized bistatic radar cross section (ddm_nbrcs) on the train dataset, see Fig. S4. By fitting an exponential function, we obtain

$$\mathrm{nbrcs}(v) = 27.53\, e^{-0.16v} + 7.99, \quad \mathrm{nbrcs}(v) = 285.0\, e^{-0.40v} + 18.96$$

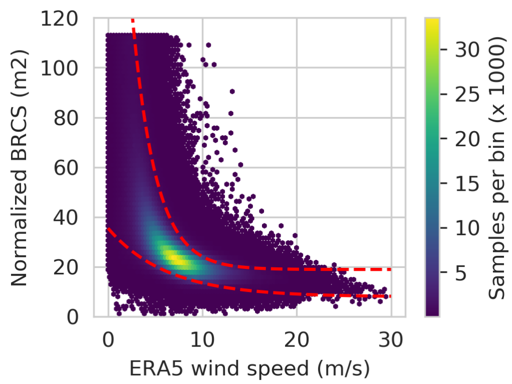


Figure S4: Density plot of the normalized bistatic radar cross section ddm_nbrcs and the target wind speed. The 95% confidence interval is indicated by dashed lines.

Samples labeled with wind speeds below $2.5\,\mathrm{m/s}$ are excluded. The DDM observables, particularly the NBRCS, are insensitive to winds below this threshold, as seen in simulations and empirically [26, 11].

A.2 Dataset statistics

The good-quality samples are clustered in time, see Fig. S5. Note that due to the randomized selection of DDMs with the same wind speed label, the sample count per day is lower in the training data range compared to the validation and test data ranges, where this selection is not applied. The wind speed distribution is comparable across the three datasets (Fig. S6).

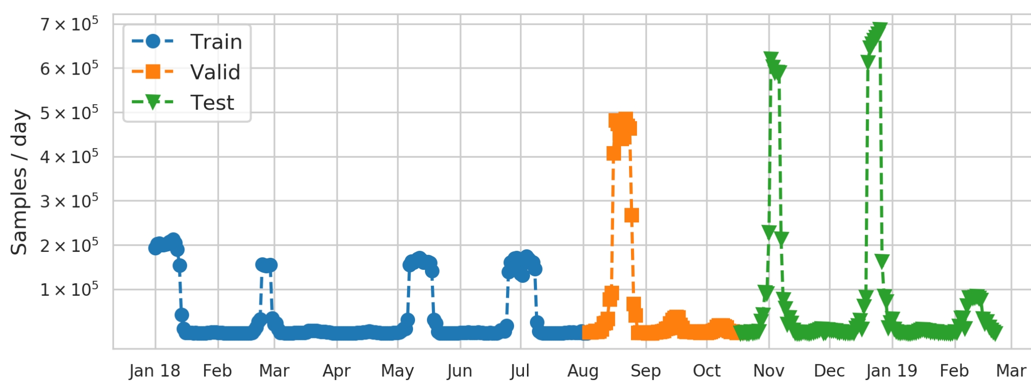


Figure S5: Good-quality samples per day for the train, validation, and test dataset.

A.3 Input features

We use a data driven strategy for determining the input parameters to CyGNSSnet. For each potential input parameter, we train 5 instances of CyGNSSnet and judge whether including the input parameter improves the loss on the validation set. If performance is improved in at least 3 out of 5 trained models, the input parameter is included. Thus, we form CyGNSSnet with 10 ancillary parameters, where the input parameters are given in Table S2.

A.4 Model parameters

A.4.1 Neural network hyperparameters

We tune the model hyperparameters using the NNI package [23]. Table S3 summarizes the hyperparameter search space.

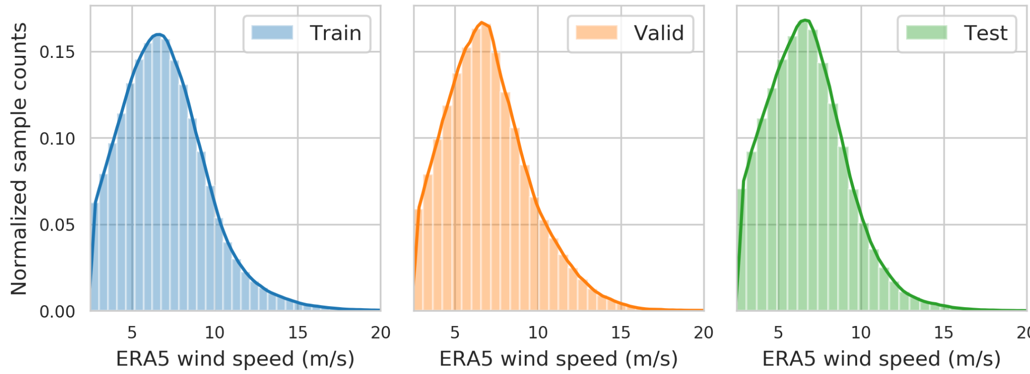


Figure S6: Wind speed distribution across the train, validation, and test dataset.

Table S2: Input parameters for CyGNSSnet. For the variables descriptions, see CyGNSS L1 V2.1 users's guide and data dictionary, `https://podaac-tools.jpl.nasa.gov/drive/files/allData/cygnss/L1/docs/148-0346-6_L1_v2.1_netCDF_Data_Dictionary.xlsx`.

Architecture	Maps	Map-related	Receiver-related	Geometry-related
CyGNSSnet	brcs	ddm_nbrcs, ddm_les, $\log_{10}$(les_scatter_area), $\log_{10}$(nbrcs_scatter_area), ddm_snr	gps_eirp, $\log_{10}$(rx_to_sp_range)	sp_inc_angle, sp_alt, sp_theta_orbit

For each architecture, the three best models are taken from the hyperparameter tuning run. They form an ensemble, their predictions on the test set are averaged. The hyperparameters are summarized in Table S4 (CyGNSSnet) and Table S5 (CyGNSSnet-X).

Table S3: Hyperparameter search space. Note that there are three fully connected layers and three dropout layers, which are optimized separately. Thus, the total number of tunable hyperparameters is 12.

Parameter	Search space
Learning rate	$5 \times 10^{-5} \ldots 1 \times 10^{-3}$
Batch size	$32 \ldots 2048$
Number of convolutional layers	$1 \ldots 8$
Filters in first convolutional layer	$8 \ldots 64$
Number of layers after which filters are doubled	$2 \ldots 8$
Number of layers after which pooling is applied	$1 \ldots 8$
Units in dense layers	$4 \ldots 256$
Dropout after dense layers	$0.0 \ldots 0.3$

Table S4: CyGNSSnet

Parameter	E1	E2	E3
Learning rate	1.4×10^{-4}	7.3×10^{-4}	2.8×10^{-4}
Batch size	64	1216	128
Number of conv. layers	3	4	5
Filters in first conv. layer	32	56	56
Filters doubled after layer	4	–	–
Pooling after layer	2	2	4
Units in FC0	188	72	244
Dropout after FC0	0.04	0.26	0.08
Units in FC1	20	216	96
Dropout after FC1	0.02	0.03	0.03
Units in FC2	12	176	24
Dropout after FC2	0.02	0.16	0.27

Table S5: CyGNSSnet-X (Model trained on extreme values exceeding $10\,\mathrm{m/s}$)

Parameter	E1	E2	E3
Learning rate	4.9×10^{-4}	4.1×10^{-4}	9.4×10^{-4}
Batch size	32	32	64
Number of conv. layers	2	7	7
Filters in first conv. layer	16	56	40
Filters doubled after layer	–	6	6
Pooling after layer	–	6	4
Units in FC0	68	164	20
Dropout after FC0	0.27	0.20	0.10
Units in FC1	72	36	124
Dropout after FC1	0.11	0.15	0.28
Units in FC2	156	48	136
Dropout after FC2	0.22	0.19	0.08

A.4.2 XGBoost hyperparameters

An XGBoostClassifier is trained to recognize extreme samples with wind speed larger than $12\,\mathrm{m/s}$. Since the model that is trained on the extreme values performs significantly worse at average wind speeds, we emphasize precision over recall and use the F_β score with $\beta = 0.5$ as an evaluation metric. We only use the ancillary variables (see Table S2) as inputs.

The hyperparameters are optimized on the validation set. We use the Tree Parzen Estimator (TPE) algorithm in its implementation in the `optuna` package [28] and optimize the hyperparameters in 80 trials. Note that the class imbalance is taken into account by the hyperparameter *scale positive weight*. The resulting hyperparameters are given in Table S6. For a full description of XGBoost hyperparameters, see `https://xgboost.readthedocs.io/en/latest/parameter.html#learning-task-parameters`

Table S6: XGBoost Classifier hyperparameters

Parameter	Search Space	Value
Maximum depth	$3 \ldots 15$	9
Learning rate	$0.01 \ldots 1$	0.54
Scale positive weight	$0 \ldots 100$	1.98
Min. child weight	$0 \ldots 1$	0.089
Gamma	$0 \ldots 100$	74.0
Subsample fraction	$0.1 \ldots 1$	0.20
Colsample by tree	$0.1 \ldots 1$	0.82

A.5 Classifier evaluation

The performance of the XGBoost classifier is evaluated on the test set. The confusion matrix is

$$C = \begin{pmatrix} TN & FP \\ FN & TP \end{pmatrix} = \begin{pmatrix} 8379728 & 216940 \\ 120245 & 115460 \end{pmatrix}$$

Overall, we reach an $F_\beta = 0.37$ score, where $\beta = 0.5$. Precision and recall are determined as

$$P = \frac{TP}{TP + FP} = 0.35, \quad R = \frac{TP}{TP + FN} = 0.49.$$

Thus, many samples are incorrectly classified as extreme values. This can be seen in the histogram plot in Fig. S7.

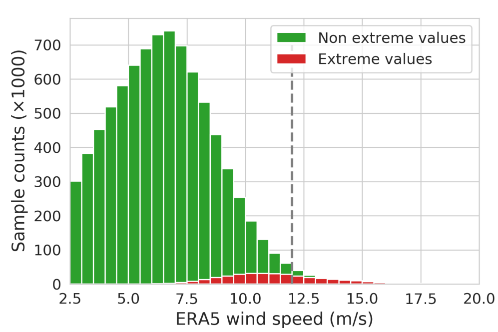


Figure S7: Binary class "extreme value", evaluated on the test set. The boundary at $v = 12\,\mathrm{m/s}$ is indicated by the dashed line.

The observed accuracy p_o and the expected accuracy p_e are

$$p_o = \frac{TP + TN}{n}, \quad p_e = \frac{(TP + FN) \times (TP + FP) + (TN + FP) \times (TN + FN)}{n^2}$$

with the total number of samples n. From there, we calculate Cohen's κ:

$$\kappa = 0.024,$$

which indicates slight agreement with a random classifier.

A.6 MVE global evaluation

The MVE algorithm is evaluated on the test set, and bias, $\hat{v}_i - v_i$, is averaged on a latitude-longitude grid with $1°$ resolution. The resulting spatial pattern is shown in Fig. S8. Especially in the Asia-Pacific regions, at longitudes between $50°$ W and $0°$, the bias is comparatively large. In this region, the Quasi-Zenith Satellite System is known to cause radio-frequency interference, which degrades the signal-to-noise ratio of GNSS-R measurements [29].

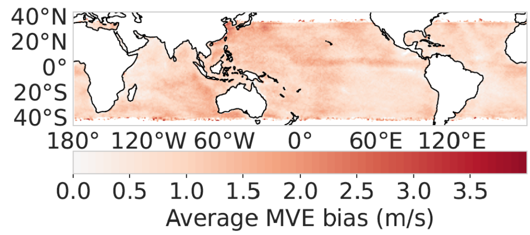


Figure S8: Bias on a latitude-longitude grid with $1°$ resolution for the MVE (current operational baseline algorithm).