Towards LSTM-Driven Forecasts for Next-Gen Water Level Monitoring

LUCAS A. RAICU*, University of Chicago, USA
DANIEL GRZENDA, University of Chicago, USA
KYLE CHARD, University of Chicago, USA

Accurate forecasting of water levels is essential for tracking climate change and flood mitigation. Traditionally, predictions have been based on harmonic analysis and sensor networks maintained by the National Oceanographic and Atmospheric Administration. However, these methods struggle with increasingly erratic sea-level dynamics. To more accurately capture complex temporal dependencies, TidalMark leverages deep learning models, specifically Long-Short-Term Memory networks. These models, due to their ability to use gates to selectively retain or forget information over time, are able to learn long-range patterns in sequential time-series data, making them perfect to use for water-level predictions. Through extensive hyperparameter sweeps and comparisons across model variants, we have evaluated tradeoffs in accuracy, generalization (for time and architecture), and scalability. Our results show that properly tuned machine learning models consistently outperform the scientific-standard harmonic approaches between 2.1X and 4.7X (between 7D to 1D predictions) with the goal towards achieving adaptive, scalable, and accurate forecasting of coastal water levels. Our project moves toward achieving adaptive, scalable, and accurate forecasting of coastal water levels.

1 Introduction

Coastal communities face increasing threats from flooding, sea-level rise, and extreme weather. [7] Since the 1800s, harmonic analysis has been used to predict water levels by decomposing tides into cyclical components [1, 5, 8]. The National Oceanographic and Atmospheric Administration (NOAA) [4] maintains hundreds of sensors measuring coastal water levels, and has adopted harmonic analysis to predict future water levels. While effective under stable conditions, it assumes linearity and stationarity (long-term sea-level rise), limiting accuracy under rapid environmental change. Figure 3 shows a major weather system over the course of several days that produced flooding water levels (red line) while both NOAA predictions and Forecast Guidance estimated water levels to be well in normal ranges.

Harmonic analysis decomposes the water-level signal into a fixed set of sinusoidal constituents, each with constant amplitude and phase (see Figure 2b). In reality, coastal systems exhibit non-stationary behavior (e.g., seasonal shifts, long-term sea-level rise) and nonlinear interactions among constituents (e.g., overtides, compound tides), which harmonic analysis cannot adapt to dynamically. By design, harmonic analysis captures only astronomic (gravitational) drivers of tides (see Figure 2a). It cannot account for wind, atmospheric pressure changes, or storms. As climate change alters baseline sea levels and potentially the resonance characteristics of estuaries or bays, the fixed-parameter nature of harmonic analysis means constituents estimated from historical data may become increasingly unsuitable for future predictions.

Many modern approaches such as Long Short-Term Memory (LSTM) based models learn from both periodic and aperiodic patterns, adapt to changing dynamics, and fuse spatial context in ways that harmonic analysis alone cannot [2]. LSTMs are designed to retain context over long sequences by using memory cells and gating mechanisms (input, forget, and output gates). While LSTM also features repetitive modules like traditional RNNs, LSTM incorporates four interacting layers that

Authors' Contact Information: Lucas A. Raicu, University of Chicago, Chicago, IL, USA, lucas.raicu@gmail.com; Daniel Grzenda, University of Chicago, Chicago, IL, USA, grzenda@uchicago.edu; Kyle Chard, University of Chicago, Chicago, IL, USA, chard@uchicago.edu.

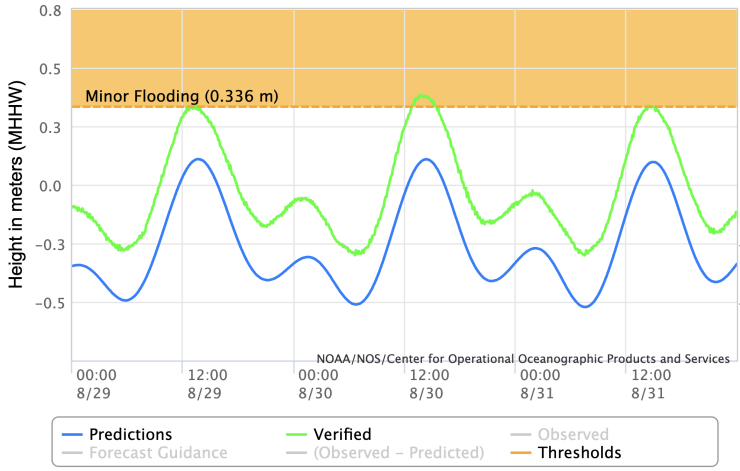
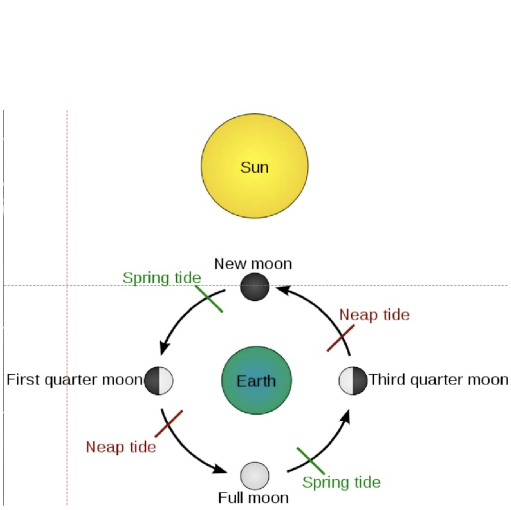
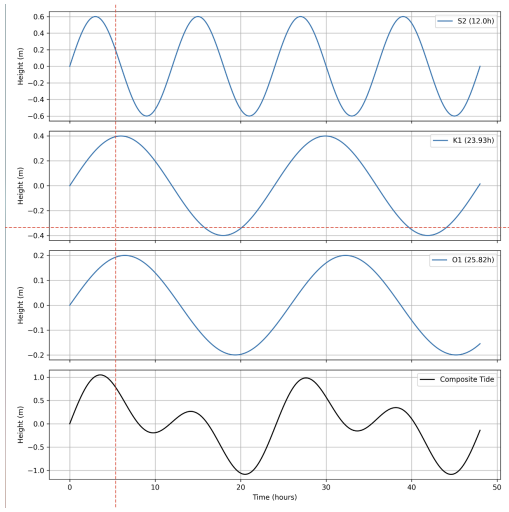


Fig. 1. NOAA's predictions during non-periodic events



(a) Harmonic analysis decomposes tidal forces into fixed-frequency constituents



(b) Tide modeling via additive synthesis of harmonics

Fig. 2. Gravitational Pull and Harmonic Analysis

communicate internally. This multi-layered design allows the model to retain complex long-term dependencies.

We propose TidalMark, a system that applies LSTM models to improve forecast lead times and adaptability in coastal water-level prediction. This research work's main contribution is that it identified Machine Learning models (LSTM) and its configuration parameters that can outperform prediction accuracy of traditional harmonic analysis between 2.1X and 4.7X (between 7D to 1D predictions). Figure 3 shows the histogram of all predictions using the LSTM compared to NOAA's predictions, showing the error size in meters.

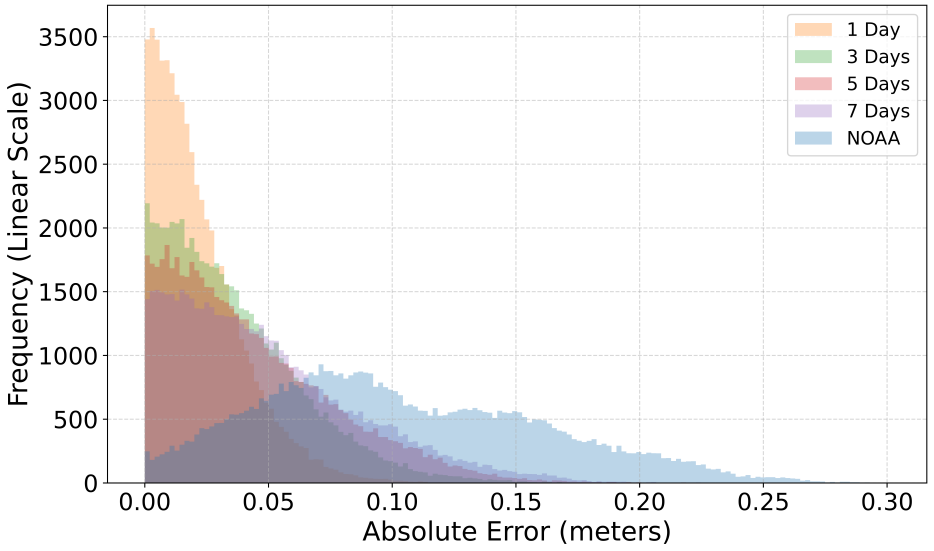


Fig. 3. LSTM vs NOAA predictions at the Nawiliwili station, HI

2 Proposed Work

We implemented our LSTM models using PyTorch using the `nn.LSTM` module. We used sequential models to capture long-term-dependencies in water-levels. We performed a single-shot walk-forward validation. We split up our training data, validation data, and testing data in traditional 80:10:10 split. We trained models with MSE loss and early stopping based on validation loss.

The model's behavior is strongly influenced by its hyperparameters:

- **Sequence length:** controls the input time window. Longer sequences (e.g., 14) allow the model to learn long-term dependencies but require more memory and may introduce noise.
- **Batch size:** impacts generalization and training time. Smaller batches (32–64) often lead to better generalization but increase training time. Larger batches (256–512) can converge faster but may overfit.
- **Learning rate:** governs how quickly the model adapts during training. We found 10^{-3} offered the best tradeoff between convergence speed and stability, aligning with prior hydrological studies [3].
- **Hidden size:** affects representational capacity. Larger hidden states (64) increase expressiveness but may cause overfitting if not regularized.
- **Number of layers:** increases abstraction depth. Our experiments showed that 2 layers offered marginal gains over 1, consistent with diminishing returns seen in prior literature [3].

We used a number of visualization techniques to evaluate the trained models, such as absolute error histograms, train vs validation loss, correlation heatmap, and hyper-parameter comparisons through Box, Scatter, as well as Violin Plots.

2.1 Data & Preprocessing

Our primary dataset comes from NOAA's National Water level Observation Network (NWLON) system, spanning 217 stations (see Figure 4b) and over 127 million measurements, each taken

at six-minute intervals. The full dataset totals over 82 GB. Our work focuses on the station in Nawiliwili, HI (Station ID: 1611400, see Figure 4a) due to its completeness and coastal variability. Before feeding the data into our neural networks, we undertake a rigorous preprocessing routine:

(a) NOAA's Nawiliwili station (b) U.S. Map of Stations

Fig. 4. Target station and dataset coverage

first, we filter the raw TSV files to retain only validated records for the target station. Next, we sort all entries chronologically to preserve the temporal sequence. Finally, we partition the cleaned series by station. Figure 5 shows the first four entries in the final dataset.

time	value	sigma	quality	inferred		flat		roc	threshold	station	datum
2018-01-01 00:00:00	0.273	0.002	v	0	0	0	0	1611400	mllw		
2018-01-01 00:06:00	0.278	0.003	v	0	0	0	0	1611400	mllw		
2018-01-01 00:12:00	0.277	0.003	v	0	0	0	0	1611400	mllw		
2018-01-01 00:18:00	0.276	0.006	v	0	0	0	0	1611400	mllw		

Fig. 5. Sample data

2.2 Modeling Framework

We developed a modular pipeline in PyTorch for training and evaluating water-level forecasting models. Each model receives a fixed-length window of prior water levels as input and predicts future water levels at multiple time horizons (1, 3, 5, and 7 days). Input features are normalized per sequence using standard scaling.

To understand how architectural design and input dimensionality influence performance, we explored three major experiments.

(1) Univariate Hyperparameter Sweep

We evaluated 120 configurations of a standard LSTM model, varying five key hyperparameters. Table 1 summarizes the grid search space. Each model was trained with mean squared error (MSE) loss and early stopping based on validation RMSE.

(2) Model Architecture Comparison

To assess the effect of architecture choice, we fixed the best LSTM configuration and compared it against other recurrent models across 180 trials. Table 2 lists the architectures and configuration ranges. This experiment evaluated whether added architectural complexity led to meaningful gains in forecast accuracy or generalization.

(3) Multivariate Input Extension

Table 1. Hyperparameter grid for univariate LSTM sweep.

Parameter	Values Tested
Sequence Length	7, 14
Batch Size	32, 64, 128, 256, 512
Learning Rate	$1x10^{-3}$, $1x10^{-4}$, $1x10^{-5}$
Hidden Size	32, 64
Number of Layers	1, 2

Table 2. Recurrent architectures tested in model comparison.

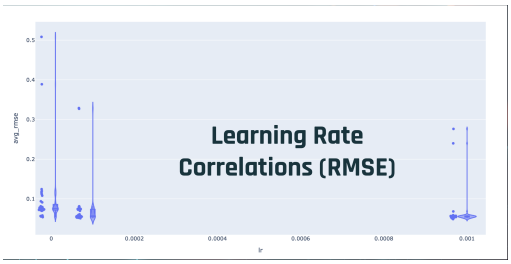
Model Type	Variants Tested	Parameter Ranges
LSTM	Vanilla, BiLSTM	Hidden sizes: 32, 64
Conv-LSTM	1D convolution	Learning rates: $1x10^{-3}$, $5x10^{-4}$, $1x10^{-4}$
GRU	Single, stacked	Layers: 1, 2
Attention-LSTM	Scaled dot-product	Batch sizes: 32, 64, 128

We extended the LSTM input space to include additional sources of temporal information. This aimed to determine whether richer inputs improve model performance or simply increase the risk of overfitting.

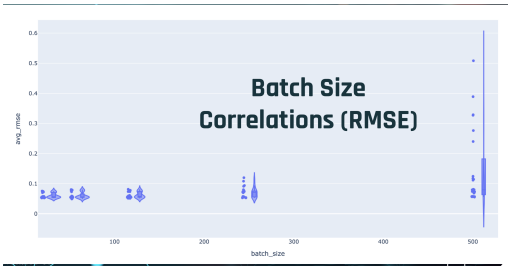
- Neighboring station water levels
- NOAA tidal predictions
- All 37 harmonic constituents from Nawiliwili

3 Performance Evaluation

Learning rate was the dominant factor. Models trained with 1×10^{-3} converged fastest and achieved the highest R^2, outperforming lower values by a wide margin (see Figure 6a). Batch size and number of layers had modest effects. Smaller batches (32–64) improved stability and generalization. One or two layers offered comparable results (see Figure 6b).



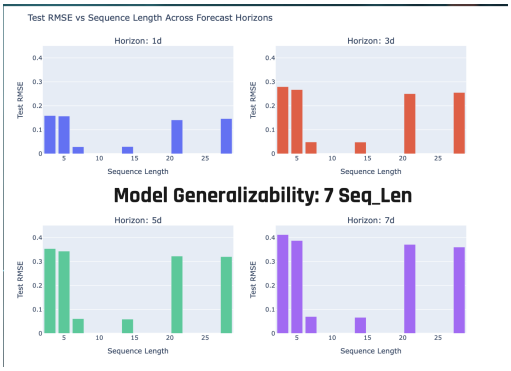
(a) Learning rate vs RMSE correlation



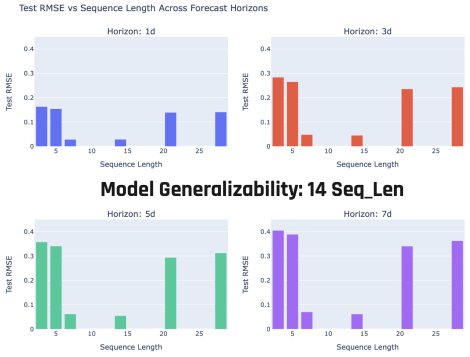
(b) Batch size vs RMSE correlation

Fig. 6. Learning rate and batch size

Longer sequences (14 vs. 7) slightly improved accuracy but at a significant training time increase. Models tuned for one seq length also did not generalize well for other sequence lengths (see Figure 7a and Figure 7b).



(a) Sequence Length of 7 Days

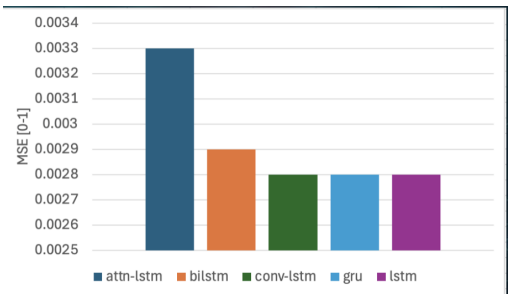


(b) Sequence Length of 14 Days

Fig. 7. Model Generalizability

3.1 Architecture Comparisons

Despite their theoretical advantages, BiLSTM, GRU, and Attention-based models did not outperform the tuned vanilla LSTM.



(a) MSE distribution across architectures

	BEST	Model	R^2	MSE	MAE	MaxAE	seq	batch	lr	hidden	layers
Horizon 1d	R^2	conv-lstm	0.980	0.001	0.022	0.133	7	64	0.0005	32	1
	MAE	conv-lstm	0.980	0.001	0.022	0.133	7	64	0.0005	32	1
	MaxAE	lstm	0.978	0.001	0.023	0.116	7	32	0.001	32	2
Horizon 3d	R^2	lstm	0.942	0.002	0.038	0.173	7	64	0.001	32	1
	MAE	conv-lstm	0.942	0.002	0.037	0.206	7	64	0.0005	32	1
	MaxAE	bilstm	0.940	0.002	0.039	0.169	7	32	0.0005	32	1
Horizon 5d	R^2	gru	0.910	0.004	0.047	0.251	7	64	0.0001	64	2
	MAE	lstm	0.910	0.004	0.047	0.227	7	128	0.0005	32	1
	MaxAE	gru	0.903	0.004	0.049	0.219	7	32	0.001	32	1
Horizon 7d	R^2	conv-lstm	0.891	0.004	0.053	0.251	7	64	0.001	32	1
	MAE	conv-lstm	0.891	0.004	0.053	0.251	7	64	0.001	32	1
	MaxAE	gru	0.880	0.005	0.056	0.222	7	32	0.001	32	1

(b) Forecast performance per horizon across architectures

Fig. 8. Architecture comparison

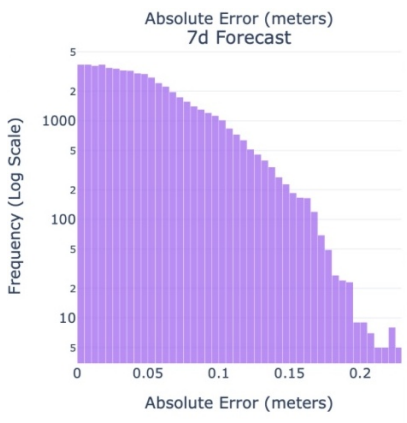
As shown in the figures above, variance was high, but performance distributions largely overlapped. This suggests that careful tuning matters more than exotic architecture selection.

3.2 Multivariate Modeling

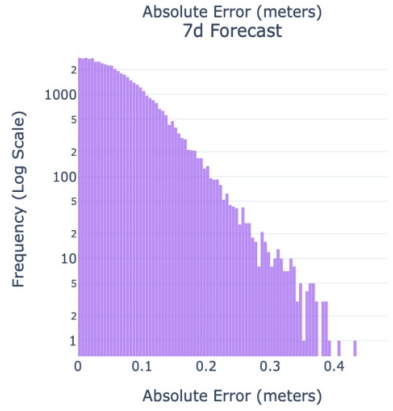
Contrary to expectations, adding auxiliary inputs hurt performance in most configurations. Multivariate models tended to overfit, especially when trained without retuning hyperparameters. Figure shows how absolute error (maximum and distribution) increases for neighbor, NOAA, and constituent-based models compared to univariate LSTM.

3.3 Testbed Performance

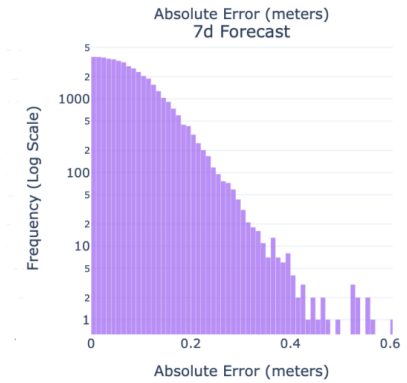
To gauge real-world viability, we benchmarked training speed on two platforms. We trained models on both a local MacBook Pro (6-core Central Processing Unit, 16 GB RAM) and the Chameleon Cloud platform (24-core CPU, 192 GB RAM, NVIDIA RTX 6000 Graphics Processing Unit). On CPU,



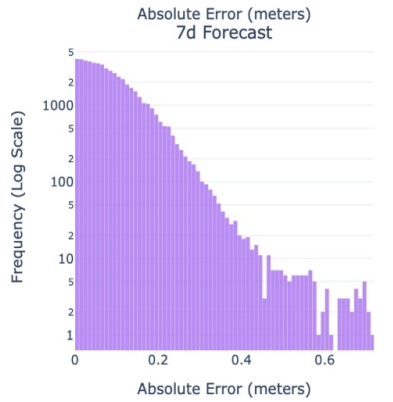
(a) Univariate LSTM (best configuration) - MaxAE: 0.241



(b) Multivariate with Neighbor Station - MaxAE: 0.445



(c) Multivariate with NOAA Predictions - MaxAE: 0.603



(d) Multivariate with 37 Constituents - MaxAE: 0.687

Fig. 9. Absolute error histograms (log scale) across 7d horizons for various input types

2 epochs on 500,000 samples took 2 hours. On GPU, we trained 200 epochs in the same time—a 100× speedup.

This gap highlights that GPU acceleration is essential for large-scale hyperparameter sweeps and production deployment.

4 Related Work

Harmonic Analysis remains the industry standard for long-term sea-level forecasting. However, LSTM-based forecasting has seen success in hydrological studies [3] and Earth system science more broadly [6]. Dynamic Graph Neural Networks (DGNNs) have also been recently been proposed for spatiotemporal learning [9].

5 Conclusion and Future Work

TidalMark demonstrates that well-tuned LSTM architectures can outperform traditional harmonic-based forecasts in dynamic environments. Our results show that properly tuned machine learning models consistently outperform the scientific-standard harmonic approaches between 2.1X and 4.7X (between 7D to 1D predictions) with the goal towards achieving adaptive, scalable, and accurate forecasting of coastal water levels. Figure 3 shows the histogram of all predictions using the LSTM compared to NOAA's predictions, showing the error size in meters. These findings align with broader trends in Earth system modeling, where deep learning methods are increasingly surpassing classical statistical and physical models in accuracy and adaptability [6].

With next steps in spatial modeling and hyperparameter automation, TidalMark moves us closer to more adaptive and scalable water-level forecasting systems. Building on these findings, we plan to extend TidalMark into a full spatiotemporal Graph Neural Network framework, where each station is a node and edges represent geophysical relationships. This will allow the model to reason across space, not just time. We also aim to apply it to inland river systems. Additionally, we aim to explore automated hyperparameter optimization using Bayesian search. Finally, we plan to streamline the framework for production use, including edge deployment for early flood warnings.

Acknowledgments

This research was inspired by work at the University of Chicago's Globus Labs. Thank you to Daniel Grzenda, Dr. Kyle Chard, and Dr. Foster for the technical help and profound insight. Thank you also to Chameleon Cloud team for their support and computational resources.

References

[1] Sergio Consoli, Diego Reforgiato Recupero, and Vanni Zavarella. 2014. A survey on tidal analysis and forecasting methods for Tsunami detection. *arXiv preprint arXiv:1403.0135* (2014). Traditional tidal forecasting methods based on harmonic analysis require long-term data and fail under non-astronomical influences such as weather.

[2] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[3] Frederik Kratzert, Daniel Klotz, Mathew Herrnegger, Andrew K Sampson, Sepp Hochreiter, and Grey Nearing. 2019. Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. *Hydrology and Earth System Sciences* 23, 12 (2019), 5089–5110.

[4] National Oceanic and Atmospheric Administration (NOAA). 2025. National Oceanic and Atmospheric Administration. https://www.noaa.gov/. Accessed: 2025-08-08.

[5] David Pugh. 2004. *Tides, Surges and Mean Sea-Level* (2nd ed.). Wiley.

[6] Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, Nuno Carvalhais, and Prabhat. 2019. Deep learning and process understanding for data-driven Earth system science. *Nature* 566, 7743 (2019), 195–204.

[7] NOAA Ocean Service. 2024. What threats do coastal communities face? Online facts page. Lists threats including extreme natural events, sea level rise and coastal erosion..

[8] Wikipedia contributors. 2025. Tide — Harmonic analysis historical introduction by William Thomson. Wikipedia, accessed 2025-08-09. Describes development of harmonic analysis in the 1860s for tidal prediction..

[9] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. 2020. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems* 32, 1 (2020), 4–24.