Charlotte Wertz

CPTS 570

11715300

Generating Probabilistic Storm-Surge Scenarios for Tropical Cyclones

1) Abstract

This report investigates data-driven methods to generate a storm surge for a given hurricane scenario based on its parameters. In order to determine vulnerability in electric power systems from tropical cyclones, it is highly important to get a realistic, yet fast representation of coastal flooding. Two data-driven methods are investigated: the multi-layer perceptron, and the random forest algorithm. Ultimately, the random forest algorithm produces a more accurate result that closely resembles real flooding data.

2) Introduction

Electrical power systems of the future come with many challenges: a changing mix of energy resources from traditional fossil fuels to renewables, a changing load with the increase in electric vehicles, and, crucially, a changing climate. With climate change, extreme weather events such as hurricanes will become both larger and more frequent, and our power systems need to be built to resist these extreme events.

However, predicting where and how these extreme weather events will happen is a challenging problem. Power system damage due to hurricanes is not only caused by the extreme winds, but also the massive storm surges and flooding that happen on coastlines. Predicting a storm surge based on a given hurricane is not an easy task, as it often involves multivariate calculus and physics based on hurricane parameters and the coastline shape (see fig. 1 for some of these variables). Since power systems solutions are often needed quickly, yet still accurately, these complex equations will not suffice.

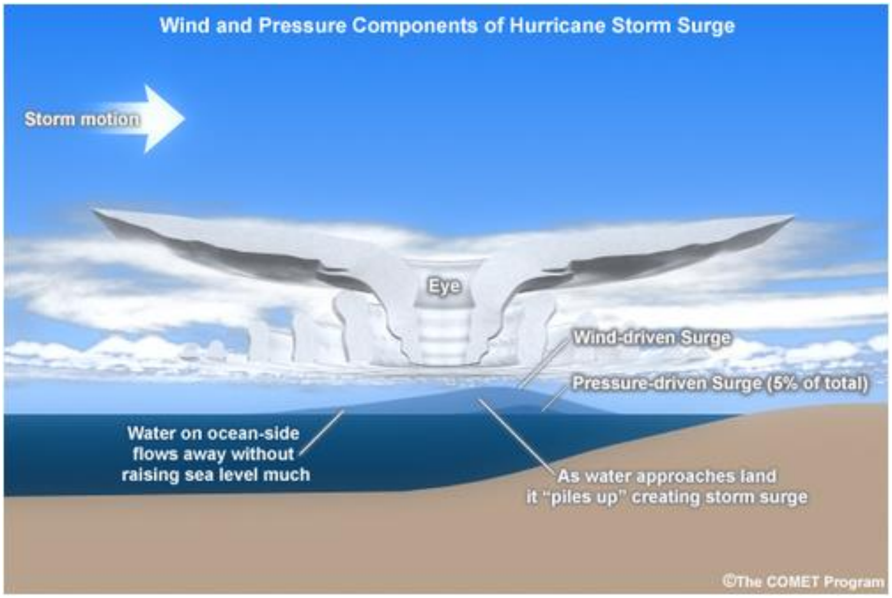

Figure 1: Wind and Pressure Components of Hurricane Storm Surge [1]

This is where data-driven methods have many benefits. They are not only faster than traditional methods of computing potential flooding, but they are also easier to understand for someone who does not have an extensive knowledge of climate modeling and geography. Because of this, my research group aims to generate coastal flooding scenarios for given probabilistic hurricane tracks.

3) Problem Setup

Currently, we are studying the power system impact on the Texas basin. This is due to a multitude of reasons; one being that Texas is particularly vulnerable to tropical cyclone events, and another being that their power grid is isolated from the rest of the county, meaning it is harder to supply emergency generation during an outage. Finally, Texas is currently one of the largest renewable energy producing states in the U.S, meaning that it provides a good outlook for what future power systems will look like [2]. Because of this, it makes a good case study for this project.

Texas lies on the North Atlantic basin, and much of its flooding vulnerability is in its respective sub-basin, Corpus Christi. For now, I will focus on this region for analysis, but future work will extrapolate the same methods for all five sub-basins that lie within the state of Texas. The solution to tackle is, for any given hurricane track, we want to predict a realistic and accurate flooding model based on its parameters.

A notable limitation of this model is it will not predict flooding caused by rainfall. Rainfall is a separate independent event that causes hurricane flooding and is related to the same variables as storm surge. Thus, it will be analyzed separately in future works.

4) Solution Approach

The framework can be illustrated in figure 2. To summarize, each input is a matrix representing the time-step values of each point in the hurricane track. After putting it through a black box data-driven algorithm, we hope to achieve a grid of storm surge values.

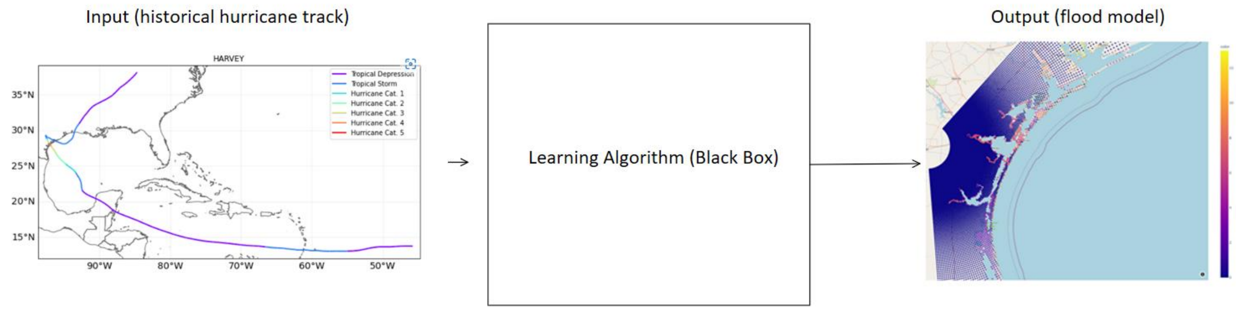

Figure 2: Solution Framework

However, there are several problems with this approach; for one, every input hurricane has a different number of hurricane track points, depending on when the hurricane was first detected and how long it was categorized as a tropical cyclone. We cannot put inputs of varying feature sizes into a machine learning algorithm. Because of this, we must analyze the most impactful time-step values of these tropical cyclones. For our analysis, we choose the time step values that are approximately 300 km from the Texas coastline. This number is chosen due to literature review showing this is where a tropical cyclone begins to impact the storm surge and can be somewhat forecasted [3].

Secondly, the inputs and outputs are not 1-1; not every input has a single appropriate label. Instead, each input will have many labels. This requires a multi-output regression analysis. In order to achieve this, we must use a neural network algorithm that has multiple output nodes. To simplify the process, I have chosen a Multi-Layer Perceptron to start with but have not excluded other methods that can handle this level of depth and data, such as Random Forest and Convolutional Neural Networks that have been used in other literature [4]

This model will be validated in two ways; one will be finding R^2 value, and additionally, simple visual inspection on flood predictions using plotly express.

5) Experiments and Results

For input data, I will use the International Best Track Archive for Climate Stewardship (IBTRACS) [5]. IBTRACS is regularly updated by the National Oceanic and Atmospheric Administration and provides many parameters on hurricane tracks in 3-hour time intervals. This data is easy to extrapolate and from a very reliable source.

The chosen features from this dataset are Latitude, Longitude, Wind Speed, Pressure, Storm Direction, Storm Speed, and Distance to Landfall. All of these play a crucial role in developing a storm surge [4]. For each of these time-series tracks, to reduce complexity and the scope of this project, the points that are approximately 300 km from the Texas coastline are used for this analysis.

There have been twenty-two tropical cyclones in the Corpus basin from 1980 to the present day. Of these twenty-two storms, fifteen of them have usable data for this surge analysis. Therefore, there will be 15 examples, each with 7 features.

The output data is a list of coordinates from the Corpus sub-basin and its corresponding depth caused by the given storm. This information is extracted from Sea, Lake, and Overhead Surges from Hurricanes (SLOSH) database [6], from the National Hurricane Center. The surge height listed is known as the Maximum Envelope of Water (MEOWs), which is the maximum height the given coordinate's water level reaches above its elevation. In other words, this dataset provides a "worst-case scenario" estimate. From this basin we filter out any points that lie beyond the coastline in the ocean. This leaves us with 15,341 output nodes.

The first 12 examples with their corresponding surge levels are used for training data, and the remaining 3 tropical cyclones are used for testing. The example in figure 4 is taken from Hurricane Delta.

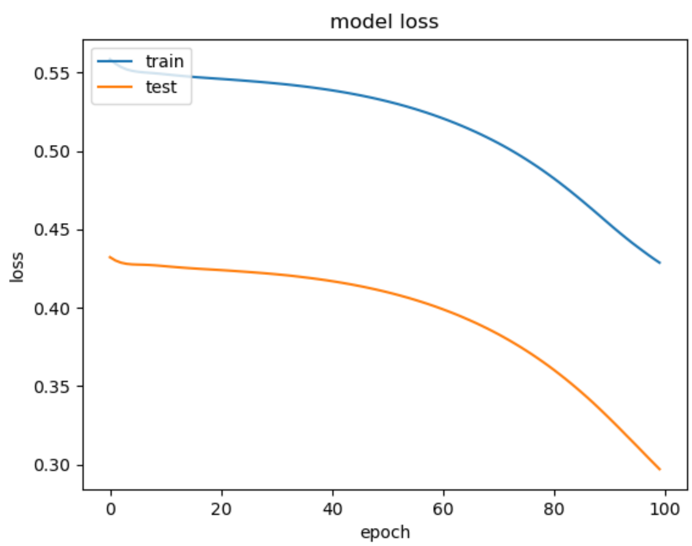

Figure 3: Model Loss on Multi-Layer Perceptron Algorithm

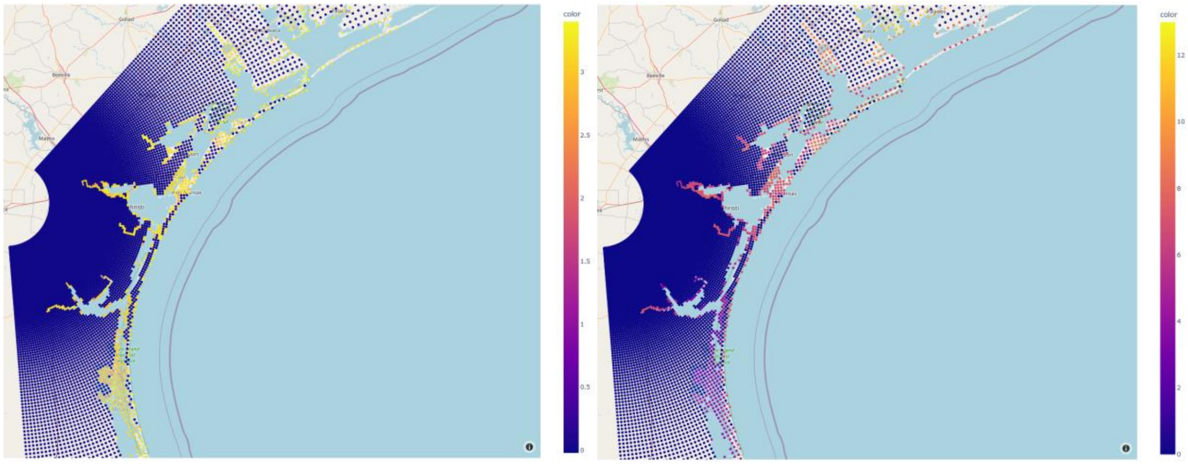

Figure 4: Predicted Surge Levels on Hurricane Delta (a), Compared to Actual Surge Levels (b), using Multi-Layer Perceptron

As can be seen, the loss decreases over time as desired. However, the visual results tell a different story; while the predictions for inland values that are unlikely to have coastal surge are predicted correctly at 0 feet, the specific surge values predicted seem to be several feet off.

The storm surge data is extremely dense; there are over 15,000 output nodes in this model. Because of this, it is inherently sparse; most coordinates will not witness a storm surge from a given hurricane. However, this becomes problematic. For example, if only 1% of coordinates are flooded, it is still crucial to track where this 1% of coordinates are, even if predicting no flood at all outputs a 99% accuracy. This is especially challenging to model with neural networks since they do not have high accuracy under sparse conditions. I believe this is what's lowering the predicted values from their observed flood levels.

Because of this, I also tried Random Forest, which is known to have notably better performance with sparse data (source). The training and testing data remains the same, the number of estimators is set to 1000, and the random state is set to 42. The results can be compared in figure 5 below.

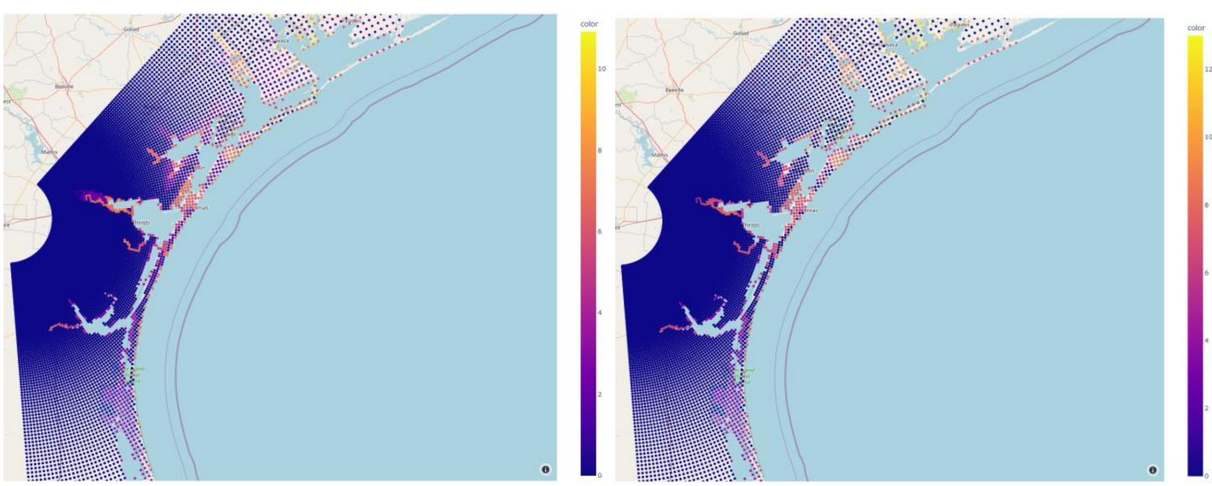

Figure 5: Predicted Surge Levels on Hurricane Delta (a), Compared to Actual Surge Levels (b), using Random Forest

It can easily be seen upon visual inspection that Random Forest performs significantly better than the multi-layer perceptron, and these results are quite precise. The flooding is still underestimated, but not by a significant margin as seen prior. However, for validation's sake, we will also compute the R^2 value for both the multi-layer perceptron and random forest.

Table 1: Accuracy of Multi-Layer Perceptron and Random Forest on Testing Data

	R^2 Value
MLP	-0.081846
RF	0.89537

Now, it is clear that not only is random forest better than MLP using the given parameters, but it is also quite accurate.

Another note of interest between these methods is the difference in output values, even for non-flooded regions. The output values for the neural network in non-flood regions are actually continuous, but very small values that are close to 0. However, Random Forest correctly identifies them as nodes without flooding. An example of this for a coordinate set is below.

Table 2: Example Output Values for given Coordinates: Random Forest vs Multi-Layer Perceptron

Coordinates	[29.31251, 96.4725]
Surge Height (MLP)	0.00478 ft
Surge Height (RF)	0.0 ft

This confirms the suspicion that the multi-layer perceptron algorithms has its faults with highly sparse output data. However, this could likely be mitigated in future with hyperparameter tuning and data filtering.

6) Conclusions and Future Work

To conclude, I learned that the ML algorithm chosen can have a significant impact on results depending on the type of data being analyzed. Also, I learned the importance of having abundant training data. Because only 15 historical storms fit the criteria of this project and had reliable data, our training pool is quite small. In future, this model can be expanded on using additional data, such as creating synthetic tracks based on historical data. Additionally, there may be post-processing methods that are adequate for this study. Both multi-layer perceptron and random forest models seem to do quite well at knowing where a flood could happen, but the specific depth height could be improved upon. It may be more practical to adjust this after the model has been formed. Finally, this model could benefit from further hyper-parameter and feature tuning. Overall, it has produced intriguing results and I will continue to improve this model.

7) Acknowledgments

I would like to thank my research team members, Abodh Poudyal and Sajjad Uddin-Mahmud, for their technical guidance and help with data pre-processing.

References

[1] Storm Surge Overview. National Oceanic and Atmospheric Administration, https://www.nhc.noaa.gov/surge/

[2] Nilson, Elsa. Wind and solar are propping up the Texas power grid amid a brutally long heat wave. CNN Business, July 2023. https://www.cnn.com/2023/07/28/business/texas-renewable-energy-heat-wave-climate/index.html

[3] Lockwood, Joseph W, Lin, Ning, Oppenheimer, Michael, Lai, Ching-Yao. Using Neural Networks to Predict Hurricane Storm Surge and to Assess the Sensitivity of Surge to Storm Characteristics. Journal of Geophysical Research: Atmospheres. 2022, https://doi.org/10.1029/2022JD037617

[4] Jun-Whan Lee, Jennifer L. Irish, Michelle T. Bensi, Douglas C. Marcy. Rapid prediction of peak storm surge from tropical cyclone track time series using machine learning. Coastal Engineering, Volume 170, 2021, 104024, ISSN 0378-3839. https://doi.org/10.1016/j.coastaleng.2021.104024.

[5] International Best Track Archive for Climate Stewardship (IBTrACS). National Oceanic and Atmospheric Administration, https://www.ncei.noaa.gov/products/international-best-track-archive

[6] Sea, Lake, and Overland Surges from Hurricanes (SLOSH). National Hurricane Center, https://www.nhc.noaa.gov/surge/slosh.php

[7] https://machinelearningmastery.com/deep-learning-models-for-multi-output-regression/